Computational Studies on Discrete Logarithm Problem

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

By

R. Padmavathy

Supervisor**: Prof. Chakravarthy Bhagvati**

Department of Computer and Information Sciences
School of Mathematics and Computer & Information Sciences, University of Hyderabad, Hyderabad.

2009

Computational Studies on Discrete Logarithm Problem

R. Padmavathy (Roll No. 03MCPC09)

under the guidance of

Prof. Chakravarthy Bhagvati

Dept.of Computer and Information Sciences,

University of Hyderabad,

Hyderabad 500046 (India)

CERTIFICATE

This is to certify that the thesis entitled Computational Studies on Discrete Logarithm Problem being submitted to the University of Hyderabad by R.Padmavathy (Reg. No. 03MCPC09), in partial fulfillment for the award of the degree of Doctor of Philosophy in Computer Science, is a record of bonafide work carried out by her under my supervision

The matter embodied in this report has not been submitted to any other University or Institution for the award of any degree or diploma.

Prof. Chakravarthy Bhagvati
Supervisor,
Department of CIS,
University of Hyderabad,
Hyderabad-500046.

Prof. Arun Agarwal,
Head,
Department of CIS,
University of Hyderabad,
Hyderabad-500046

Prof. T. Amarnath,
Dean,
School of MCIS,
University of Hyderabad,
Hyderabad-500046

DECLARATION

I, R.Padmavathy, hereby declare that the work presented in this thesis has been carried out by me under the supervision of Prof. Chakravarthy Bhagvati, Department of Computer and Information Sciences University of Hyderabad, Hyderabad, India, as per the PhD ordinances of the University. I declare, to the best of my knowledge, that no part of this thesis has been submitted for the award of a research degree of any other University

R.Padmavathy

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Chakravarthy Bhagvati for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee, Prof. Arun K. Pujari and Dr. Durga Bhavani, for their encouragement, insightful comments, and hard questions.

My sincere thanks also goes to Prof. Arun Agarwal, Head of the Computer and Information Sciences department, UOH for his encouragement to complete this work. I also thank Prof.T. Amarnath, Dean of School of Mathematics and Computer and Information Sciences, UOH for his support to pursue my doctoral work.

I am very grateful to Ms. Devi Prasanna, Animal Sciences and Ms.G. Uma, Computer and Information Sciences for their consistent moral support and help rendered to me when required. Finally I would like to thank my husband, in-laws and my daughter for their continuous support throughout the tenure. Last but not the least, I would like to thank my parents and my sister, for supporting me throughout my life.

Contents

1 Introduction **1**

1.1 Problem definition 2

1.2 Motivation 3

1.2.1 Index Calculus Method 5

1.2.2 Smooth numbers over Z_p^* 6

1.2.3 DLP for Ephemeral key 7

1.2.4 Pollard-Rho to solve ECDLP 7

1.2.5 The DLP $\alpha^a\beta^b$ 8

1.3 Contributions 8

1.4 Organization 9

2 Literature Survey **11**

2.1 Public key cryptography 11

2.2 DLP based public key cryptosystems 14

2.2.1 Diffie-Hellman 14

2.2.2 El-Gamal cryptosystem 17

2.2.3 XTR cryptosystem 19

2.2.4 Elliptic Curve Cryptosystem 22

2.3 Attacks on DLP based public key cryptography 24

2.3.1 Improving the computation of DLP using the number theoretic approach 24

2.3.2 Improving the DLP using the computational approach . . . 35

2.3.3 Improving the computation of DLP using the structure of group, exponents and order of other elements related to DLP 36

2.4 Summary 40

3 On the Computation of Index Calculus Method 41

3.1 Introduction 42

3.2 Linear sieve method 42

3.2.1 Linear sieve for generating the linear relations 43

3.2.2 Methods to solve the relations 44

3.2.3 Computational parameters 47

3.3 Analysis on linear sieve method 48

3.3.1 Empirical analysis on structured Gaussian elimination 49

3.3.2 Analysis on bound(B) and sieve length(C) for the ratio(R) . 51

3.4 Algorithm for pre-computation step of linear sieve method 54

3.4.1 Improved algorithm for linear sieve method 56

3.4.2 Performance study and numerical results 56

3.5 Comparative analysis 61

3.6 Conclusion 61

4 A New Method for Computing DLP Based on Extending Smooth Numbers to Finite Fields 63

4.1 Introduction 63

4.2 Smooth numbers and their distributions for certain primes 64

4.2.1 Identified patterns 66

4.2.2 New algorithms for DLP 74

4.3 Computing the discrete logarithm in a given range 74

4.3.1 Exponents in a specific range in a group of order $2q$ 75

4.3.2 Exponent in a specific range in a prime order subgroup of a safe prime of the form $2q+1$ or a random prime of $2\rho+1$. 78

4.3.3 Random prime with $p-1$ as factors 2ρ 81

4.4 Experimental results 83

4.4.1 Selected list of solved problems 86

4.5 Conclusion 88

5 Ephemeral Key Recovery using Index Calculus Method 91

5.1 Introduction . . . 91

5.2 Ephemeral key recovery using the properties of a special case of smooth numbers over Z_p^* . . . 93

5.2.1 Smoothness . . . 93

5.2.2 Distribution of smooth numbers over Z_p^* on different types of primes . . . 96

5.2.3 Results and discussion . . . 100

5.2.4 Algorithms for ICM . . . 101

5.3 Ephemeral key recovery using the properties of generators . . . 107

5.3.1 Our approach . . . 107

5.3.2 Motivation to develop the new methodology . . . 107

5.3.3 Algorithms for ICM . . . 109

5.4 Conclusion . . . 113

6 Cryptanalysis on the DLP using Random method, on the ECDLP using Pollard-Rho and the DLP $\alpha^a\beta^b$ 116

6.1 Performance analysis of Index Calculus Method . . . 116

6.1.1 Performance analysis on Random method . . . 117

6.1.2 Improved Random method using the properties of smooth number over Z_p^* . . . 128

6.1.3 Experimental results . . . 131

6.2 DLP on elliptic group using Pollard-Rho . . . 131

6.2.1 Experimental results . . . 135

6.3 Cryptanalysis on the DLP $\alpha^a\beta^b$ . . . 138

6.3.1 The encryption scheme based on the DLP $\alpha^a\beta^b$ . . . 138

6.3.2 Improved version of cryptosystem . . . 141

6.3.3 Security analysis of newly proposed system . . . 142

6.3.4 Limitations . . . 143

6.4 Conclusion . . . 145

7 Discussion and Conclusions **146**

7.1 Discussion 146

7.2 Extensions 149

7.3 Contributions 151

8 Publications **153**

List of Figures

3.1 Optimal ratio of problems from 17 digits to 30 digits 53
3.2 Size of reduced matrix for a problem of size 25 digits 53
3.3 The relationship between the Bound(B) and the Sieve length (C) for a problem of size 25 digits 54
3.4 Size of reduced matrix of different bound for different size of problems 55
3.5 Bound and the Sieve length for different size of problems 55

List of Tables

2.1 Comparable problem sizes solved using ICM 30

3.1 Optimal values for parameters 47
3.2 Results on improved linear sieve method with k=1 59
3.3 Comparison between the running time of improved linear sieve method and the empirical results in table 3.1 with k=1 60

4.1 Patterns generated from different types of primes 73
4.2 Running time of methods to solve DLP on safe primes 89
4.3 Running time of methods to solve DLP on prime order subgroups . 89
4.4 Running time of methods to solve DLP on random primes 90

5.1 Methods for Index Calculus Method 101
5.2 Running time of method-1 and method-2 105
5.3 Individual logarithm step 106
5.4 Running time of pre-computation and individual logarithm steps of ICM 114
5.5 Running time of individual logarithm step and Pohlig-Hellman . . . 115

6.1 Running time of Random method 124
6.2 Running time of partial linear sieve method 124
6.3 The difference in the number of smooth integers before and after the reduction 125
6.4 The running time with or without reduction in the range with respect to problem size in Random method 126

6.5 Difference in running time with or without reduction in the range with respect to problem size 127
6.6 Difference in the running time of traditional and improved Index Calculus Method 132
6.7 Difference in the running time of traditional and improved Index Calculus Method 133
6.8 Results on Pollard-Rho method 136
6.9 Problems solved using Pollard-Rho method 137

ABSTRACT

The broad objective of the present work is computational analysis on the mathematically hard Discrete Logarithm Problem(DLP). The DLP forms the basis for several popular public key cryptosystems. For a given prime number p, a generator $g \in Z_p^*$ and an element $y \in Z_p^*$, the problem of finding an x $(0 \leq x \leq p-2)$ such that $g^x \equiv y \bmod p$ is known as the DLP. The DLP is also defined over other groups. There are several methods to solve the DLP and the Index Calculus Method (ICM) is currently the best general-purpose algorithm.

A more specific objective of the thesis is the investigation of the ICM. The ICM has two stages: a pre-computation stage where logarithms of the elements of a subset of the group, known as a *factor base*, are computed, and a solution stage where the desired logarithm is computed with the help of pre-computed logarithms of the factor base. The pre-computation stage itself has two steps. The first is generating the necessary linear system of equations and the second is solving the linear system. Linear sieve is a popular method for ICM and is used for efficiently generating the equations relating the logarithms of the elements in the factor base. However, the method generates a large system that is normally reduced (by using structured Gaussian elimination or other techniques) to yield a faster solution.

There are several computational parameters such as the factor base and its size and the sieve length to be used in linear sieve that have been shown in literature to have an impact on the overall performance of the ICM. The first major contribution is the development of a new technique by combining the size of the factor base and the sieve length to achieve a performance improvement of 30% and more for the pre-computation step on problem sizes of over 120 bits. Another contribution is the extension of the concept of smooth numbers over factor bases and finite fields. This new concept is used for improving the solutions for certain instances of the DLP previously assumed to be hard. In particular, it is shown that exponents chosen near the middle of the groups over safe primes (i.e., primes of the form $2q+1$ where q is a prime) and prime order subgroups may be solved efficiently using the new technique. Results are shown on problems of sizes upto 1024 bits. The same concept is also used to improve the performance

of ICM for safe primes. A variant of the ICM, analogous to Pohlig-Hellman, when the factors of $p-1$ are small is developed on a special case of the newly defined smooth numbers. It is shown that this new method has certain advantages over Pohlig-Hellman for attacking ephemeral keys.

At more abstract level the Pollard-Rho to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the DLP $\alpha^a\beta^b$ are also investigated. The ECDLP is a DLP defined on elliptic curve group. Recently, Kashyap et al., proposed a cryptosystem based on the DLP, $\gamma = \alpha^a\beta^b$, where α and β are generators, a and b are distinct integers. The above cryptosystem and the corresponding cryptanalysis are further analyzed in the present study. Also an improved version of the encryption scheme based on the cryptosystem published by Kashyap et al., is proposed.

At a broader level, we introduce a new approach to cryptanalysis that is based on deriving computationally optimal choices. For example, the choices of factor bases and types of primes have to the best of our knowledge no direct theoretical basis. The computational approach taken is not related to finding efficient ways (either using hardware or special purpose libraries) of implementing known algorithms; nor is it based on a number-theoretic approach. Thus the approach followed in this thesis is a novel computer-science approach to cryptanalysis in contrast with number-theoretic methods.

Chapter 1

Introduction

The focus of this thesis is on investigating the Discrete Logarithm Problem (DLP) from a computational perspective. In particular Index Calculus Methods are studied, and new methods and improvements are proposed for cases that have been considered hard in literature. The DLP forms the basis for the popular El-Gamal public key cryptosystem [30], Diffie-Hellman key exchange [29] and several digital signature schemes. For a given prime number p, a generator $g \in Z_p^*$ and an element $y \in Z_p^*$, the problem of finding an x ($0 \leq x \leq p-2$) such that $g^x \equiv y \bmod p$ is known as the DLP. DLP is also defined over several other groups, such as the multiplicative group of F_{2^m} and the collection of points defined by an elliptic curve over a finite field.

Several methods of solving DLP have been proposed in literature. Shanks baby-step-giant-step [85] is a well-known deterministic algorithm with both time and space complexities of $O(\sqrt{n})$ where n is the order of the group. The probabilistic Pollard-Rho method [73] also has a $O(\sqrt{n})$ running time but avoids the large space requirements. Another variant of the Pollard-Rho algorithm, called the Pollard-Lambda, solves the DLP in a time of $O(\sqrt{w})$ if the exponent is known to lie in an interval of width w [73]. A special case algorithm is the Pohlig-Hellman method [71] which reduces the DLP to search over small subgroups and uses the Chinese Remainder Theorem to combine the results and solve the DLP. Pohlig-Hellman is a very effective method when the order of the group Z_p^*, i.e., $p-1$, has no large factors.

Index Calculus methods (ICM) are some of the best general-purpose methods for solving DLP if there is more structure in the group than merely the set of elements and a binary operation [25]. Specifically, if the group elements can be expressed as a product of a smaller subset of group elements (called the *factor base*), then index calculus methods solve the DLP in sub-exponential time.

Although the earlier attacks tried to solve the DLP in an efficient way, still there is a need for further analysis on the DLP to develop new and efficient algorithms and computational techniques. Hence, an attempt has been made to formulate such computational techniques. In this thesis, we propose some new methods and improved traditional methods to solve DLP for cases that have been considered hard in the literature. In particular, we show that our methods are effective on the so-called *safe primes* of the form $p = 2q + 1$, where q is a prime and for prime order subgroups.

1.1 Problem definition

The main aim of the present study is to improve the ICM and to formulate new techniques to solve DLP. The ICM has two stages: a pre-computation stage where logarithms of the elements of a subset of the group, known as a *factor base*, are computed, and a solution stage where the desired logarithm is computed with the help of pre-computed logarithms of the factor base. The pre-computation stage itself has two steps. The first is generating the necessary linear system of equations and the second is solving the linear system. Linear sieve is a popular method for ICM and is used for efficiently generating the equations relating the logarithms of the elements in the factor base. However, the method generates a large system that is normally reduced (by using structured Gaussian elimination or other techniques) to yield a faster solution.

There are several computational parameters such as the factor base and its size and the sieve length to be used in linear sieve that have been shown in literature to have an impact on the overall performance of the ICM. In this thesis, various studies on improving the performance by selecting suitable parameters, such as investigating the effect of factor base., etc are reported. Also, distributions of

numbers factorizable into the subset of elements defined in the factor base are specially exploited for patterns that lead to faster solutions of the DLP.

1.2 Motivation

Today, there is no polynomial time algorithm for solving the DLP. Development of secure and efficient public key systems is driven by cryptanalysis of their underlying hard mathematical problems such as Integer Factorization Problem (IFP) for RSA and the DLP for El-Gamal. RSA has been one of the most popular cryptosystems to date and is also one of the most extensively analyzed. It has been shown by researchers that DLP is as hard as RSA. The largest number factored by a general-purpose factoring algorithm for RSA is 663 bits long, using a state-of-the-art distributed implementation. The DLP on integer field of 120 digits and $GF(2^n)$ of 607 bits are solved.

The following paragraph summarizes the results of the detailed analysis on cryptographic key sizes as discussed by Lenstra and Verheul [52]. The discrete logarithm based schemes are broadly divided into two categories such as TDL (Traditional Discrete Logarithm) and SDL(Subgroup Discrete Logarithm) systems. TDL is the DLP, where the DLP is defined on the the group of order $p-1$ of prime field p and SDL is the DLP, where the DLP is defined on the subgroup of order q of prime field p. In the literature the computing power is often measured in Mips-Year, where a Mips-Year is defined as the amount of computation that can be performed in one year by a single DEC VAX 11/780. Suppose one is developing a commercial application in the year 2000 in which the confidentiality or integrity of the electronic information has to be guaranteed for 20 years, i.e., until the year 2020. The security computationally equivalent to that offered by the DES in 1982 from 2000 to 2020 is obtained by using the following key sizes.

- Symmetric keys of at least 86 bits, and hash functions of at least 172 bits;
- RSA moduli of at least 1881 bits;
- Subgroup discrete logarithm systems with subgroups of at least 151 bits with

finite fields of at least 1881 bits. Thus, for an SDL system such as XTR it follows that log2 (q) 151 and 6 log2 (p) 1881.

- Elliptic curve systems over prime fields of at least 161 bits if one is confident that no cryptanalytic progress will take place, and at least 188 bits if one prefers to be more careful.

The approach followed in [52] to estimate the parameters of different classical cryptosystems may fail, a single bright idea may prove that all currently popular cryptographic protocols are considerably less effective than expected. For example in 1980's the popular knapsack-based cryptosystems were suddenly wiped out by a new type of attack.

All the above analysis on the key sizes of various cryptosystems are based on the conventional attacks available in the literature. Cryptanalysis is the area of research that decides the security implication of various popular cryptosystems such as RSA, ECC, El-Gamal and XTR to name a few by developing classical attacks. Thus the problem is formulated to analyze computationally the DLP through which to launch the effective attacks on all popular DLP based cryptosystems.

In general, cryptanalysis is seen to be performed in one of the three major approaches:-

- Developing new methods through a number theoretic approach.
- Efficient implementation of traditional methods, developing new techniques and speeding up the traditional methods through a computational approach. For example, efficient implementation of linear sieve, cubic sieve, number field sieve, parallelized Pollard-Rho and Pollard-Lambda.
- Exploiting the group structure, structure of the exponents, weakness of the field, weakness of the protocol, order of the other elements related to the DLP and through the parameters involved in the traditional method.

The present work improves the computation of DLP through the third approach. Weaknesses in cryptosystems are almost always found and analyzed by using number-theoretic considerations to prove that certain instances have specially designed algorithms, e.g., Pohlig-Hellman, MOV and anomalous attacks on ECDLP

(A DLP defined over an elliptic group). In this thesis, we also explore cryptosystems for weaknesses from a purely computational perspective. Generally number theoretic results lead to the development of computationally efficient algorithms. It is also possible that efficient algorithms may be discovered by systematically studying the parameters associated with the cryptosystems and their effect on computation times. Such studies tend to be laborious and become possible only with the recent increases in computational power of even inexpensive computers. It is this approach that is taken in this thesis and the following list summarizes the approaches:

- An improved ICM is proposed by using a computational study on key parameters involved in the method (chapter-3)
- New methods of solving the DLP and a variant of ICM to solve the DLP for the ephemeral key are developed by exploiting the structure of $p-1$ i.e., the order of the group. (chapters 4 and 5).
- Three miscellaneous algorithms are summarized in chapter 6.

At more abstract level ECDLP and the DLP $\alpha^a\beta^b$ are also investigated. Specifically, the present thesis consists of the following studies on DLP and ICM.

1.2.1 Index Calculus Method

The ICM has two steps, namely, a pre-computation and individual logarithm computation . In the pre-computation step, the logarithms of elements of a subset of group, known as a factor base, is computed and in the individual logarithm step, the DLP is computed with the help of pre-computed logarithms of the factor base. In the present work the pre-computation step is studied. There are three steps that have a significant impact on the performance of the pre-computation step, namely, the generation of relations involving the logarithms of the primes in the factor base, the optional step of reducing their numbers for computation efficiency and solving the system for logarithms of elements in factor base. In the present study, the performance of ICM is improved through the efficient way of producing the smaller size matrix for the third step by using the reduction step

and the generation step collectively. An improved algorithm is developed for the pre-computation step. Through the improved algorithm the performance of ICM is enhanced by 30 to 40%

On a similar note, the performance of a primitive method of ICM, known as Random method is enhanced through partial linear sieve to reduce the search space. Partial linear sieve is defined from the well known linear sieve and Pollard-Rho method and analyzed. The partial linear sieve and Random method are compared based on their running time. It is observed that a range parameter is influencing the running time of the partial linear sieve method, which leads to outperform the partial linear sieve method than the Random method. This range parameter is introduced in the Random method, due to which the probability of numbers getting smoothing is improved. It is shown that the performance of Random method is enhanced by more than 50% for problems of size $\approx$ 20 digits .

1.2.2 Smooth numbers over Z_p^*

This thesis introduces a new concept of B-smooth numbers over Z_p^*. Their distribution for different types of primes is studied and it is shown that there are two major classes. These lead to the development of an efficient algorithm for solving DLP, when the exponent lies near the middle of a group, i,e., $\frac{p}{2}$ or the prime order subgroup, i.e., $\frac{q}{2}$ for primes of the form $p = 2q + 1$. Also with a simple extension the algorithm may be used for $p = 2\rho + 1$, where ρ is a product of primes. The examination on problems of size up to 1024 bits reveals that an exponent near the middle of the group can be retrieved easily in reduced cost.

The pre-computation step of Random method is based on the probability of finding field elements that are smooth i.e., the factors of field elements are less than a prescribed bound. In the present study, the Random method is improved through the smoothness concept over the prime field Z_p^* for certain instances of primes. The property based on the characteristics of smooth numbers over Z_p^* allows to map the field elements from one subset to another. Based on this concept, an improved ICM by increasing the probability of getting smooth numbers is devised. Through the experimental results it is observed that the performance of ICM is enhanced

by $\approx 50\%$.

1.2.3 DLP for Ephemeral key

A special case of ICM, analogous to Pohlig-Hellman, when the factors of $p-1$ are small is studied. In the literature, Pohlig-Hellman is the best known method to solve the DLP, when the factors of $p-1$ are small and known, while ICM is an efficient method for general DLP. Two algorithms are proposed to improve the efficiency of the pre-computation step of the ICM by using the equivalence classes formed from a special case of B-smooth numbers over Z_p^*. The two algorithms proposed here are useful in recovering ephemeral keys often used for session-based security. In ephemeral key security, the underlying field and generator of the cryptosystem are held static but each session uses different keys. In some systems, the ephemeral keys are changed periodically (e.g once every 5 minutes). For such systems, the use of ICM allows the pre-computation to be performed once with one search each for the individual logarithm of the ephemeral key. In addition to the above two algorithms, another approach to recover the ephemeral key based on the property of generators is also presented.

Traditionally the ICM is viewed as a general purpose algorithm with respect to solve the DLP. In the present study, the ICM is viewed as a special purpose method and investigated, when the factors of $p-1$ are small. Similarly, the traditional way of computing the logarithms of factor base is to obtain the logarithms of first-t primes. In this thesis, another way of choosing and computing the logarithms of factor base is proposed. Based on the new concept of choosing the factor base, a variant of ICM (pre-computation and individual logarithm step) is proposed. The newly proposed individual logarithm (computation) step outperforms the Pohlig-Hellman method on some special cases. This leads to recover the ephemeral keys used in the DLP based schemes in reduced time.

1.2.4 Pollard-Rho to solve ECDLP

Pollard-Rho method of solving the ECDLP is considered and its performance is analyzed. Since the Pollard-Rho method is known to solve the ECDLP on all

types of elliptic curves in an exponential time, independent of group structure. Teske [92] conducted experiments on the elliptic group based on prime field i.e., $E_{a,b}(F_q)$ and presented results based on the running time and number of steps. In the present study, Pollard-Rho method is considered to solve the ECDLP over the field $E_{a,b}(F(2^m))$, due to its wide cryptographic applications. In the present work the experiments are conducted, for the problems of different size of bits, based on the proposed simple algorithm. Through our experimental results, it is observed that the magnitude of a private key value influences the running time of the attack and the number of group operations. It is also shown that the above property holds good for a group order of any size. Further it is shown that the large values of the private key can be computed in reduced time bound.

1.2.5 The DLP $\alpha^a\beta^b$

Recently, Kashyap et al., [43] proposed a cryptosystem based on the DLP, $\gamma = \alpha^a\beta^b$, where α and β are generators, a and b are distinct integers. The above cryptosystem and the corresponding cryptanalysis are further analyzed in the present study. Sramka [87] claimed that for the cipher text (c_1, c_2, c_3) of the cryptosystem discussed in [43], the random integer, say k, can be obtained by the simple relation, such as $c_1c_2 = (\alpha\beta)^k$. Further, he reported an attack, in obtaining the plain text from a valid cipher text, by computing the single traditional DLP. In the present study, it is shown that k can be obtained only for some special cases and those cases are discussed. Also an improved version of the encryption scheme based on the cryptosystem published by Kashyap et al., [43] is proposed. It is observed that the proposed cryptosystem is invulnerable to the attack proposed by Sramka [87].

1.3 Contributions

- A new approach to improve the performance of ICM is developed. This leads to an improved algorithm for the generation step of ICM and new results are reported on the parameters used to generate and solve the relations. The

performance of ICM is improved by 30 - 40 % by using this new approach.

- A new idea, partial linear sieve, is introduced and used to improve the performance of one of the primitive method of ICM known as Random method. It is shown that the performance is enhanced by more than 50% for problems of size ≈ 20 digits.

- A new concept of B-smooth numbers over Z_p^* is defined in the present work and a detailed analysis on the distribution is presented. Through these properties, the DLP is solved in reduced cost, in particular on safe primes and prime order subgroups.

- The characteristics of smooth numbers over Z_p^* lead to an improved Random method for pre-computation step of ICM and aid in improving the probability of getting smooth numbers. We achieved $\approx 50\%$ of improvement in the performance of ICM.

- A special case of smooth numbers over Z_p^* is defined and the properties are analyzed on different types of primes. A variant of ICM is proposed to recover the ephemeral keys based on the properties of smooth numbers over Z_p^* and the generators of Z_p^*.

- At more preliminary level the ECDLP and DLP $\alpha^a\beta^b$ are analyzed.

1.4 Organization

The subsequent chapters of this thesis are organized as follows. Chapter 2 presents the literature survey on the popular cryptosystems and the cryptanalysis on these systems in detail. Specifically the cryptanalysis on the cryptosystems based on the DLP with respect to solve the DLP is reported. The research contributions on the development in devising, designing, implementing and enhancing the algorithms to solve the DLP are discussed.

Chapter 3 presents a new approach to improve the performance of pre-computation step of ICM. The ICM is viewed into two groups, such as the generation and reduction as one group and solving as another. Empirical results on the computational

parameters are reported based on the new insight of ICM. An improved algorithm is developed and reported based on the empirical results of computational parameters involved in ICM.

Chapter 4 introduces the smoothness concept over Z_p^* and discusses the analysis on the properties of new smoothness concept over Z_p^*. The analysis on the computation of DLP for some instances of prime fields, such as $p = 2\rho + 1$, where ρ is a prime or product of primes is reported. New methods are presented to solve the DLP in the above group using the characteristics of a new smoothness concept over Z_p^*.

Chapter 5 introduces a special case of smoothness concept over Z_p^*. The detailed analysis on the properties of above smoothness concept is presented. A new variant for ICM is proposed through the properties of smooth numbers over Z_p^* to recover the ephemeral key. Similarly, an alternate approach is discussed to recover the ephemeral keys by using the property of generators of Z_p^*.

Chapter 6 reports the study on the performance of Random method, on the Pollard-Rho method of solving the ECDLP and the cryptanalysis on the DLP $\alpha^a\beta^b$. The Random method is investigated from two perspective, one is by introducing a new parameter and another is through the properties of smooth numbers over Z_p^*.

Chapter 7 is the concluding remarks of this thesis with the future work. There is a discussion of the salient features, trade-offs and design decisions. Several possible future directions for extending and improving the current study along with a list of the major and other contributions are also presented in this chapter and chapter 8 is the publications.

Chapter 2

Literature Survey

This chapter describes the popular cryptosystems and the cryptanalysis on these systems in detail. Further the research contributions in devising, developing or enhancing the algorithms in solving the DLP for the last 20 years are discussed.

The security goals on basic communication model are confidentiality, data integrity, data authentication, entity authentication and non repudiation. The cryptographic systems are basically designed to achieve these goals. Cryptology is the combination of cryptography and cryptanalysis. Cryptography deals with the design and analysis of mathematical techniques that enable secure communication in the presence of malicious adversaries while cryptanalysis deals with the secure systems (cryptosystems) for vulnerabilities. The cryptosystems can be divided into two types, such as symmetric key and asymmetric key (public key cryptosystems).

2.1 Public key cryptography

Public-key refers to a cryptographic mechanism. It has been named public-key to differentiate it from the traditional and more intuitive cryptographic mechanism known as: symmetric-key, shared secret, secret-key and also called private-key. Symmetric-key cryptography is a mechanism by which the same key is used for both encrypting and decrypting; it is more intuitive because of its similarity with the same key. This characteristic requires sophisticated mechanisms to securely distribute the secret-key to both parties. Differential [21, 55, 11, 12, 13] and linear

analysis [57, 58, 59] are the best known analysis on symmetric algorithms. Public-key on the other hand, introduces another concept involving key pairs: one for encrypting, the other for decrypting. This concept, is very clever and attractive, and provides a great deal of advantages over symmetric-key, such as simplified key distribution, digital signature, long-term encryption. Public-key is commonly used to identify a cryptographic method that uses an asymmetric-key pair: a public-key and a private-key. Public-key encryption uses that key pair for encryption and decryption. The public-key is made public and is distributed widely and freely. The private-key is never distributed and must be kept secret. Given a key pair, data encrypted with the public-key can only be decrypted with its private- key; conversely, data encrypted with the private-key can only be decrypted with its public- key. All public key cryptosystems are based on NP-problems such as DLP (Discrete Logarithm Problem), IFP (Integer Factorization Problem) to name a few. For example, the popular RSA public key cryptography depends on the IFP, while the Diffie-Hellman key exchange, Elliptic curve cryptosystem, El-Gamal and XTR are based on the DLP.

The popular cryptosystem RSA was publicly described in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman at MIT; the letters RSA are the initials of their surnames, listed in the same order as on the paper [74]. The Diffie-Hellman key agreement was invented in 1976 during a collaboration between Whitfield Diffie and Martin Hellman and was the first practical method for establishing a shared secret over an unprotected communications channel. Another popular public key cryptosystem is El-Gamal. It was described by Taher El-Gamal in 1984 [30].

RSA is the first algorithm known to be suitable for signing as well as encryption, and one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations [63, 74, 26]. The security of the RSA cryptosystem is based on two mathematical problems: the problem of factoring large numbers (IFP) and the RSA problem. Full decryption of an RSA ciphertext is thought to be infeasible on the assumption that both of these problems are hard, i.e., no efficient algorithm exists for solving them. Providing security against partial decryption may require the addition of a secure padding

scheme. The RSA problem is defined as the task of taking e^{th} roots modulo a composite n: recovering a value m such that $c = m^e mod\ n$, where (n, e) is an RSA public key and c is an RSA ciphertext. Currently the most promising approach to solving the RSA problem is to factor the modulus n. With the ability to recover prime factors, an attacker can compute the secret exponent d from a public key (n, e), then decrypt c using the standard procedure. To accomplish this, an attacker factors n into p and q, and computes $(p-1)(q-1)$ which allows the determination of d from e. No polynomial-time method for factoring large integers on a classical computer has yet been found, but it has not been proven that none exists.

As of 2005, the largest number factored by a general-purpose factoring algorithm was 663 bits long, using a state-of-the-art distributed implementation. RSA keys are typically 1024 to 2048 bits long. Therefore, it is generally presumed that RSA is secure if n is sufficiently large. If n is 256 bits or shorter, it can be factored in a few hours on a personal computer, using software already freely available. Keys of 512 bits (or less) have been shown to be practically breakable in 1999 when RSA-155 was factored by using several hundred computers. A theoretical hardware device named TWIRL and described by Shamir and Tromer in 2003 called into question the security of 1024 bit keys. It is currently recommended that n be at least 2048 bits long

On the other hand, the Diffie-Hellman, El-Gamal and XTR are based on DLP. The key exchange protocol is one of the most elegant ways of establishing secure communication between pair of users by using a session key. The session key, which is exchanged between two users assures the secure communication for later sessions. The first practical key exchange protocol is proposed by Diffie-Hellman. Since the introduction of key exchange protocol by Diffie-Hellman, various versions and improvements in key exchange protocol have been developed. El-Gamal is the popular and first cryptosystem based on DLP for both encryption and digital signatures. The XTR is another public key cryptosystem based on sub group DLP. The DLP defined on a sub group and the efficient representation of sub group elements assure the security of this system. Similarly, the elliptic curve cryptosystem is a popular public key cryptosystem based on ECDLP. The ECDLP

is a DLP defined on an elliptic group.

As of now, the DLP on integer field of 120 digits and $GF(2^n)$ of 607 bits are solved. This leads to choose 1024 bits long for Diffie-Hellman and El-Gamal keys. Since the XTR uses an efficient representation of subgroup elements, the keys are 170 bits of subgroup on 1024 bits of field. The ECC keys are much shorter and comparable with XTR with 160 to 200 bits. The ECDLP on the field of size 108 bits is solved so far.

2.2 DLP based public key cryptosystems

As discussed in the previous section the DLP is the basis for many public key cryptosystems. The generalized DLP is defined as follows:- For a given prime number p, a generator $g \in Z_p^*$ and an element $y \in Z_p^*$, the problem of finding an x $(0 \leq x \leq p-2)$ such that $g^x \equiv y \bmod p$ is known as the discrete logarithm problem. DLP is also defined over other groups such as the multiplicative group of Galois field GFp^n and the collection of points defined by an elliptic curve over a finite field. For example the XTR cryptosystem is based on the DLP defined over a prime order sub group of $GF(p^6)$. The XTR uses the trace over $GF(p^2)$ to represents elements of subgroup of $GF(p^6)^*$ of the order $p^2 - p + 1$. The trace representation leads to achieve an efficient arithmetic on $GF(p^2)$. Even though the arithmetic are to be performed on $GF(p^2)$, the security is achieved on $GF(p^6)$ [53, 54]. Another example is the elliptic curve cryptosystems. They are based on the DLP defined over elliptic curve group. The above two methods uses the basic principle for generating the keys. Later they uses either Diffie-Hellman or El-Gamal for encryption.

2.2.1 Diffie-Hellman

Diffie-Hellman key exchange (D-H) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher. Synonyms

of Diffie-Hellman key exchange include:

- Diffie-Hellman key agreement
- Diffie-Hellman key establishment
- Diffie-Hellman key negotiation
- Exponential key exchange

The scheme was first published publicly by Whitfield Diffie and Martin Hellman in 1976, although it later emerged that it had been separately invented a few years earlier within GCHQ, the British signals intelligence agency, by Malcolm J. Williamson but was kept classified. In 2002, Hellman suggested the algorithm be called Diffie-Hellman-Merkle key exchange in recognition of Ralph Merkle's contribution to the invention of public-key cryptography (Hellman, 2002). Although Diffie-Hellman key agreement itself is an anonymous (non-authenticated) key-agreement protocol, it provides the basis for a variety of authenticated protocols, and is used to provide perfect forward secrecy in Transport Layer Security's ephemeral modes.

History

Diffie-Hellman key agreement was invented in 1976 during a collaboration between Whitfield Diffie and Martin Hellman and was the first practical method for establishing a shared secret over an unprotected communications channel. Ralph Merkle's work on public key distribution was an influence. John Gill suggested application of the discrete logarithm problem. It had been discovered by Malcolm Williamson of GCHQ in the UK some years previously, but GCHQ chose not to make it public until 1997, by which time it had no influence on research in academia. The method was followed shortly afterwards by RSA, another implementation of public key cryptography using asymmetric algorithms.

Diffie-Hellman key exchange

The key is assumed to be exchanged between two communicators, say Alice and Bob. Alice and Bob wishes to agree on a secret random element in the group

which could be used as a key for a higher speed symmetric algorithm like the DES. They wish to make this agreement over an insecure channel without having exchanged any information previously.

The public items are the group G, and an element $g \in G$ of large known order. Based on this assumption the following procedure is carried out.

Alice generates a random integer $x_A \in 1.....\#G - 1$ and sends to bob the element g^{x_A}

Bob generates a random integer $x_B \in 1.....\#G - 1$ and sends the element g^{x_B} to Alice

Then Alice can compute

$$g^{x_A x_B} = (g^{x_B})^{x_A}$$

and Bob can compute

$$g^{x_A x_B} = (g^{x_A})^{x_B}$$

Thus, the secret key exchanged is $g^{x_A x_B}$ [29].

Security

The protocol is considered secure against eavesdroppers if the group G and the generator g are chosen properly. This is currently considered difficult. An efficient algorithm to solve the discrete logarithm problem would make it easy to compute a or b and solve the Diffie-Hellman problem, making this and many other public key cryptosystems insecure. The order of G should be prime or have a large prime factor to prevent the attacks based on the order of group. For this reason, a Sophie Germain prime q is sometimes used to calculate $p = 2q + 1$, called a safe prime, since the order of G is then only divisible by 2 and q. g is then sometimes chosen to generate the order q subgroup of G, rather than G, so that the Legendre symbol of g^a never reveals the low order bit of a. If Alice and Bob use random number generators whose outputs are not completely random and can be predicted to some extent, then Eve's task is much easier. The secret integers a and b are discarded at the end of the session. Therefore, Diffie-Hellman key exchange by itself trivially achieves perfect forward secrecy because no long-term private keying

material exists to be disclosed [29, 37, 28, 86, 36]. Perfect Forward Secrecy (or PFS) is the property that ensures that a session key derived from a set of long-term public and private keys will not be compromised if one of the (long-term) private keys is compromised in the future. Forward secrecy has been used as a synonym for perfect forward secrecy.

2.2.2 El-Gamal cryptosystem

The El-Gamal encryption system is an asymmetric key encryption algorithm for public-key cryptography. It was described by Taher El-Gamal in 1984 [30]. El-Gamal encryption is used in the free GNU Privacy Guard software, recent versions of PGP, and other cryptosystems. The Digital Signature Algorithm is a variant of the El-Gamal signature scheme. El-Gamal encryption can be defined over any cyclic group G. Its security depends upon the difficulty of a certain problem in G related to computing discrete logarithms.

El-Gamal cryptosystem

Alice wishes to send a message, say m, to Bob. The message m is assumed to be encoded as an element in the group. Bob has a public key consists of g and $h = g^x$ where x is the private key of Bob.

Alice generates a random integer $k \in 1...\#G - 1$ and computes

$$a = g^k, b = h^k m$$

Alice sends the cipher text (a, b) to Bob.

Bob can receive the message from the equation as follows

$$ba^{-x} = h^k m g^{-kx} = g^{xk-xk} m = m$$

Security

Analysis based on the best available algorithms for both factoring and discrete logarithms show that the RSA system and the El-Gamal system have similar se-

curity for equivalent key lengths. The security of the El-Gamal scheme depends on the properties of the underlying group G as well as any padding scheme used on the messages. If the Computational Diffie-Hellman assumption holds the underlying cyclic group G, then the encryption function is one-way. If the Decisional Diffie-Hellman assumption (DDH) holds in G, then El-Gamal achieves semantic security. Semantic security is not implied by the Computational Diffie-Hellman assumption (CDH) alone. The CDH assumption states that, given (g, g^a, g^b) for a randomly-chosen generator g and random $a, b \in \{0, \ldots, q-1\}$, it is computationally intractable to compute the value g^{ab}. The DDH assumption states that, given g^a and g^b for randomly-chosen $a, b \in \mathbb{Z}_q$, the value g^{ab} "looks like" a random element in the group G. This intuitive notion is formally stated by saying that the following two probability distributions are computationally indistinguishable (in the security parameter q): (g^a, g^b, g^{ab}), where a and b are randomly and independently chosen from $\mathbb{Z}_q$. (g^a, g^b, g^c), where a, b, c are randomly and independently chosen from $\mathbb{Z}_q$.

El-Gamal encryption is unconditionally malleable, and therefore is not secure under chosen ciphertext attack. For example, given an encryption (c_1, c_2) of some (possibly unknown) message m, one can easily construct a valid encryption $(c_1, 2c_2)$ of the message $2m$. To achieve chosen-ciphertext security [27, 6, 7], the scheme must be further modified, or an appropriate padding scheme must be used. Depending on the modification, the DDH assumption may or may not be necessary [15].

Other schemes related to El-Gamal which achieve security against chosen ciphertext attacks have also been proposed. The Cramer-Shoup system is secure under chosen ciphertext attack assuming DDH holds for G. Its proof does not use the random oracle model. Another proposed scheme is DHAES [1], whose proof requires an assumption that is weaker than the DDH assumption.

Digital signature schemes

Digital signature schemes can be devised for data authentication, data integrity and to facilitate the provision of non-repudiation services. An entity A would use

the signature generation algorithm, say $SIGN$, of a digital signature scheme and his private key d_A to compute the signature of a message as $S = SIGN_{d_A}(m)$. Upon receiving m and S an entity B who has an authentic copy of A's public key e_A uses a signature verification algorithm to confirm that S was indeed generated from m and d_A. All public key cryptosystems provide elegant solution to key distribution, key management and the provision of non-repudiation. The popular digital signature schemes are DSA and DSS. The following paragraph discusses the El-Gamal digital signature.

El-Gamal digital signature:-

Bob wants to sign a message m. He can use the public and private key pair h and x as discussed in the previous section. First he generates a random integer $k \in 1....\#G - 1$ and computes

$$a = g^k$$

Then he computes a solution b to the congruence

$$m \equiv xf(a) + bk(modG)$$

and sends the signature (a, b) and message m to Alice.

Alice verifies the signature by checking the following equation holds [30].

$$h^{f(a)}a^b = g^{xf(a)-kb} = g^m$$

2.2.3 XTR cryptosystem

XTR stands for ECSTR. This is an abbreviation for Efficient and Compact Subgroup Trace Representation. It is a novel method that makes use of traces to represent and calculate powers of elements of a subgroup of a finite field. The XTR uses the trace over $GF(p^2)$ to represents elements of the order $p^2 - p + 1$ subgroup of $GF(p^6)^*$, thereby achieving a 3 factor reduction. Also, the resulting calculations are appreciably faster than the standard representation.

Many cryptographic protocols used to be based on generator of a full multiplicative group of finite field. Schnorr introduced the idea of replace this generator by the generator of a relatively small subgroup of sufficiently large prime order q.

The same idea is used in XTR. XTR uses a subgroup of prime order q of the order $p^2 + p - 1$ subgroup of $GF(p^6)^*$. The latter group is referred as *XTR supergroup* and the subgroup of order q as *XTR subgroup*

Selection of p and q

The primes p and q have to be selected in such a way that q divides $p^2 - p + 1$ and such that the resulting fields and subgroups with stand known attacks. Furthermore in order to able to use the fast $GF(p^2)$ arithmetic, the p should be 2 mod 3.

XTR-Diffie-Hellman

Let p, q and $Tr(g)$ be shared XTR data, where $Tr(g)$ is a trace of generator g. If Alice and Bob want to agree upon a secrete key K they do the following:

- Alice selects a random integer $a \in [2, q-3]$ and compute the following equation with $n = a$ and $c = Tr(g)$

  $$S_a(Tr(g)) = (Tr(g^{a-1}), Tr(g^a), Tr(g^{a+1})$$

  and sends $Tr(g^a)$ to Bob.

- Bob selects a random integer $b \in [2, q-3]$ and compute the following equation with $n = b$ and $c = Tr(g)$

  $$S_b(Tr(g)) = (Tr(g^{b-1}), Tr(g^b), Tr(g^{b+1})$$

  and sends $Tr(g^b)$ to Alice.

- Alice compute the following with $n = a$ and $c = Tr(g^b)$

  $$S_a(Tr(g^b)) = (Tr(g^{(a-1)b}), Tr(g^{ab}), Tr(g^{(a+1)b})$$

  and determines the key K as $Tr(g^{ab})$.

- Bob compute the following with to $n = b$ and $c = Tr(g^a)$

  $$S_b(Tr(g^a)) = (Tr(g^{(a(b-1)}), Tr(g^{ab}), Tr(g^{a(b+1)})$$

and determines the key K as $Tr(g^{ab})$.

note: Refer [53, 54] for the computation of $S_n(c)$.
In the similar way XTR-El-Gamal also can be developed using the XTR parameters [53, 54].

Security

The best attacks are Pollards rho method in the order q subgroup, or the Discrete Logarithm variant of the Number Field Sieve, another variant of ICM in the full multiplicative group $GF(p^6)^*$. With primes p and q of about $1024/6 \approx 170$ bits the security of XTR is equivalent to traditional subgroup systems using 170-bit subgroups and 1024-bit finite fields. But with XTR, subgroup elements can be represented using only about $2 * 170$ bits, which is substantially less than the 1024-bits required for their traditional representation.

Full exponentiation in XTR is faster than full scalar multiplication in an Elliptic Curve Cryptosystem (ECC) over a 170-bit prime field, and thus substantially faster than full exponentiation in either RSA or traditional subgroup discrete logarithm systems of equivalent security. XTR keys are much smaller than RSA keys of comparable security. ECC keys allow a smaller representation than XTR keys, but in many circumstances (e.g. storage) ECC and XTR key sizes are comparable. Key selection for XTR is very fast compared to RSA, and orders of magnitude easier and faster than for ECC. As a result XTR may be regarded as the best of two worlds, RSA and ECC. It is an excellent alternative to either RSA or ECC in applications such as SSL/TLS (Secure Sockets Layer, Transport Layer Security), public key smartcards, WAP/WTLS (Wireless Application Protocol, Wireless Transport Layer Security), IPSEC/IKE (Internet Protocol Security, Internet Key Exchange), and SET (Secure Electronic Transaction).

The XTR key selection is very easy. This makes it easily feasible for all users of XTR to have public keys that are not shared with others, unlike ECC where a large part of the public key is often shared between all users of the system. Also, compared to ECC, the mathematics underlying XTR is straightforward, thus avoiding two common ECC-pitfalls: ascertaining that unfortunate parameter choices are

avoided that happen to render the system less secure, and keeping abreast of, and incorporating additional checks published in, newly obtained results. As a consequence the draft IKE protocol (part of IPSec) for ECC was revised.

XTR is the first method that uses $GF(p^2)$ arithmetic to achieve $GF(p^6)$ security, without requiring explicit construction of $GF(p^6)$. Let g be an element of order $q > 6$ dividing $p^2 - p + 1$. Because $p^2 - p + 1$ divides the order $p^6 - 1$ of $GF(p^6)^*$ this g generates an order q subgroup of $GF(p^6)^*$. Since q does not divide any $p^s - 1$ for $s = 1, 2, 3$, the subgroup generated by g cannot be embedded in the multiplicative group of any true subfield of $GF(p^6)$. However the arbitrary powers of g can be represented using a single element of the subfield $GF(p^2)$, and that such powers can be computed efficiently using arithmetic operations in $GF(p^2)$ while avoiding arithmetic in $GF(p^6)$.

2.2.4 Elliptic Curve Cryptosystem

In 1985 Neal Koblitz [45, 46] and Victor Miller [65] independently proposed ECC using the group of points on an elliptic curve defined over a finite field in discrete logarithm cryptographic systems. The primary advantage that elliptic curve systems over multiplicative group of finite field is the absence of a sub-exponential time algorithm that could find discrete logarithms in these groups. Consequently, one can use an elliptic curve group that is smaller in size while maintaining the same level of security. The necessary condition for the security of all elliptic curve cryptographic schemes is that the ECDLP is intractable. Similarly like DLP, no proof that the ECDLP is indeed a hard problem. evidence for its hardness has been gathered over the years. First, the problem has been extensively studied by researchers for the last 16 years and no general-purpose sub-exponential time algorithm has been discovered. An elliptic curve E over the field F is a smooth curve in the so called "long Weierstrass form"
$Y^2 = +a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$
We let $E(F)$ denote the set of points $(x, y) \in F^2$ that satisfy this equation along with a point at infinity denoted as O.
Two finite fields are of particular interest. The finite field F_p and the finite field

F_q^m with $q = p^r$ elements. Since setting $p = 2$ the arithmetic in this field will be well suitable for implementation in hardware.

Elliptic curve is a new emerging class of public key cryptosystems that may successfully compete in the future with the current monopoly of the RSA cryptosystem. If the elliptic curve is chosen correctly, the best known algorithm for finding the discrete logarithm is of exponential difficulty. The cryptographic keys may be significantly shorter for elliptic curve cryptosystems compared to RSA, which results in faster and more compact implementations. The other advantage of using elliptic curves for constructing cryptosystems is that each user may (but does not need to) choose his/her own curve. If this is the case, reconstructing the private key of one user, does not provide any information that could be used to reconstruct the private key of another user. According to the current knowledge, choosing an appropriate elliptic curve is equivalent to calculating the number of points on the elliptic curve, and checking whether this number has a large prime divisor. Several methods have been devised for choosing an appropriate elliptic curve. The anex to the IEEE 1363 standard lists three of them: counting number of points on a randomly chosen curve using Schoof's algorithm, constructing the curve using the Weil theorem, and constructing the curve using the method of complex multiplication.

Selection of parameters

The system parameters, such as the elliptic curve group of order n with p as the largest prime order factor, the fixed point P of order p to be chosen as follows:-

- Two constants a and b are selected randomly, such that $4a^3 + 27b^2 \neq 0$
- The order of the elliptic curve (n) to be computed by using the Schoof's algorithm or Weil's theorem based on the type of the curve
- The fixed point P of order p is chosen

ECC-Diffie-Hellman

The Elliptic curve cryptosystem is implemented by using the Diffie-Hellman protocol for exchanging the session key. Later, the symmetric key cryptosystem such as DES, IDEA to name a few is used for encryption by using the session key. The procedure is discussed below.

1. Alice selects a random integer m and computes $Q = mP$ and sends this Q to Bob.
2. Bob selects a random integer n and computes $R = nP$ and sends this R to Alice.
3. Alice computes $X = mR = mnP$, thus the session key is X.
4. Bob computes $X = nQ = nmP$, thus the session key is X.

The session key X is used as a secrete key for encryption using one of the symmetric key cryptosystems. Another option for encryption is by using the El-Gamal encryption scheme.

2.3 Attacks on DLP based public key cryptography

From the above discussion, it is observed that the DLP is the basis for many popular cryptosystems. Thus the present study deals with the cryptanalysis of the DLP. In this section the methods to improve the computation of DLP are discussed.

2.3.1 Improving the computation of DLP using the number theoretic approach

As discussed in the introduction chapter the computation of DLP can be improved through three approaches. The following section briefs the attacks developed to solve the DLP by using the first approach such as through the number theoretic properties.

Generic attacks on any group structure

The following attacks can work on any group structure. These types of attacks are popularly known as square root attacks, as they need $O(\sqrt{n})$ group operations, where n is the order of the generator.

Shanks Baby Step and Giant Step Algorithm The algorithm is originally developed by D.Shanks on 1971 [85, 44, 88, 90, 91]. This algorithm requires the construction of two arrays of group elements as follows:-

Giant step is defined by

$$S = \{(i, g^{i\lceil\sqrt{n}\rceil} | i = 0.....\lceil\sqrt{n}\rceil\}$$

Baby step is defined by

$$T = \{(j, y \times g^{j}) | j = 0.....\lceil\sqrt{n}\rceil\}$$

To compute the discrete logarithm, find group element that appear in both the list. Thus the logarithm is

$$log_g^y \equiv i\lceil\sqrt{n}\rceil - j \text{mod n}$$

Pollard-Rho Method This is the popular algorithm proposed by J.Pollard on 1978. The Pollard-Rho works by first defining a pseudo - random sequence of elements from a group and then looking for a cycle to appear in the sequence [73]. The sequence can be defined by

$$\begin{cases} y \times Y_i & : \quad Y_i \in S_1 \\ Y_i^2 & : \quad Y_i \in S_2 \\ g \times Y_i & : \quad Y_i \in S_3 \end{cases}$$

Where S_1, S_2, S_3 are an arbitrary partition of the group into roughly equal sized sets

The procedure is as follows

1. Find a_i, b_i at random and compute $Y = g^{a_i} y^{b_i}$

2. Find the next sequence by using the equation given above

3. At one point $Z_{i-1} = Z_i$

4 Find x such that $x = (a_i - k)(b_i - l)^{-1} \bmod n$ (i.e) $g^{a_i} y^{b_i} = g^k y^l (\bmod n)$.

Sub-exponential time algorithms

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 [60] is that if

$$\prod_{i-1}^{m} x_i = \prod_{j-1}^{n} y_i \tag{2.1}$$

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

$$\sum_{i-1}^{m} \log_g x_i \equiv \sum_{j-1}^{n} \log_g y_j (\bmod q - 1) \tag{2.2}$$

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. This is similar to the situation in integer factorization, discussed greater detail in [51], in which one needs to find a linear dependency among a system of linear equation modulo 2. For more details on index calculus methods for discrete logarithms refer [78]. Progress in index calculus algorithms has come from better ways of producing relations that lead to equations such as 2.2. However even with the primitive method known as Random Method one can obtain running time bounds of the form

$$\exp((c + O(1))(\log p)^{\frac{1}{2}}(\log \log p)^{\frac{1}{2}}) \tag{2.3}$$

for some constant c.

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 $\log_g y$.[78, 89].

The pre-computation step is computationally the more expensive step, and it 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 is the primitive method of ICM and it is described below[18].

General algorithm

INPUT a generator g of a cyclic group G of order n and an element y
OUTPUT $\log_g y$.

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

$$g^k = \prod_{i=1}^{t} p_i^{c_i}, c_i > 0, \tag{2.4}$$

    for any k. Then, $k \equiv \sum_{i=1}^{t} c_i \log_g p_i$
  * Repeat the above steps to get the value of $t+c$ equations.
- Solve this linear system to obtain $\log_g p_i$
- Compute $\log_g y$
  * Select a random integer, k, $(0 \leq k \leq n-1)$ and compute yg^k
  * Try to write yg^k as a product of elements in FB

$$yg^k = \prod_{i=1}^{t} p_i^{d_i}, \tag{2.5}$$

for any k. Then, $\log_g y = (\sum_{i=1}^{t} d_i \log_g p_i - k) \mod n$

The very first analysis of the asymptotic running time of Index calculus algorithms appeared in the 1970s and were of the form 2.3. All the progress in the 1970s and 1980s was in obtaining better values of c, and for a long time $c = 1$ was the record, both for discrete logs modulo primes and for integer factorization. Coppersmith and Odlyzko [25] presented three versions of index calculus method in 1986. Later LaMacchia and Odlyzko [49] reported the implementation of two of these three versions namely, linear sieve and Gaussian integer methods in 1991. An implementation of cubic sieve method is reported by Abhijit Das and Veni Madhavan in 2005 [2]. For fields $GF(q)$ with $q = p^n$ for small p, Coppersmith's algorithm [22] offered running time of the form

$$\exp((C - o(1)))(\log q)^{\frac{1}{3}}(\log \log q)^{\frac{2}{3}}) \tag{2.6}$$

for a positive constant C [68]. For some fields $GF(q)$ with $q = p^n$ in which both p and n grow even bounds of the first form were not available. This lack of progress led to fairly wide speculation that running times for Integer factorization and for discrete logs in prime fields could not be improved beyond 2.3 with $c = 1$.

However in 1988 Pollard found a new approach for factoring integers. This method was developed into the Special Number Field Sieve (SNFS) by Hendrik Lenstra, and later into the General Number Field Sieve (GNFS) through a collaboration of several researchers. Initially there was wide skepticism as to whether this method would be practical, but those doubts have been dispelled. The first version of the GNFS for discrete logs was developed by [34]. Gordon's algorithm was improved by [76, 77], later Weber efficiently implemented the algorithm [99, 100]. Adleman [3] has invented the function field sieve, which can be regarded as a generalization and often an improvement of the Coppersmith algorithm for fields of small characteristic. As a result, we now posses a variety of discrete log algorithms with running time of the form 2.6. For fields GF(q) with $q = p^n$, where n is large, the running time bound of 2.6 holds with $C = 1.5262.....$ For n small,

in general we know only that 2.6 holds with $C = 1.9229....$ for special primes p, which initially were just the primes of the Cunningham form with $p = r^n + a$, where r and a are small runs in times of the form 2.6 with $C = 1.5262$ or even less using NFS [82, 83, 84].

Sub-exponential index calculus algorithms have been developed for a variety of discrete log problems. All algorithms for discrete logs that are claimed to run in time of the form 2.6 for some constants C are heuristic, in that there is no proof they will run that fast. If one willing to settle for running times of the form 2.3, then it is possible to obtain rigorous probabilistic algorithms.

Smoothness The main concept associated with ICM is smoothness property of integers. The smoothness property of integers is the base for sub-exponential time algorithms to solve the DLP. An element is called as y smooth, if it has no prime divisor larger than some bound y. The distribution of smooth integers is studied extensively [40, 39, 41]. Berstein presented a tight bounds on the distribution of smooth integers [8], a linear time algorithm to list y-smooth integer up to x and smooth part of integers [9, 10] and several algorithms for number of integers free of large prime factors [10]. The results on smoothness of integers are available [40], smoothness on algebraic integers in [17] and smoothness of polynomials are in [70]. Smoothness estimates of this type are also crucial for the few rigorous proofs of running times of probabilistic algorithms.

Linear systems over finite fields Index calculus algorithms require solutions of large sets of linear equations over finite fields. For a long time in the 1970s and early 1980s this step was regarded as a major bottleneck, affecting the asymptotic running time estimates of algorithms such as the continued fraction method and the quadratic sieve. Fortunately the linear systems of equations produced by all index calculus algorithms are sparse. This makes possible development of algorithms that take advantage of this sparsity and operate faster than general ones. The introduction of Structured Gaussian elimination [48, 67] and of the finite field versions of the Lanczos and conjugate gradient algorithms [25, 67, 48], and the subsequent discovery of the Wiedemann [102] algorithm led to a reduction in the

estimate of the difficulty of the equation solving phase. However, practice lagged behind the theory for a long time. The main advances in linear algebra for index calculus algorithms in the 1990s came from the parallelization of the Lanczos and Wiedemann algorithms by Coppersmith [23, 24]. Currently the most widely used parallel method is Montgomery's [66] version of the Lanczos algorithm, where it is used after structured Gaussian elimination reduces the matrix to manageable size. These parallelization methods essentially speed up the basic algorithms over the field of two elements by factors of 32 or 64 and are very effective. There are concerns that linear equations might can be done on a network of distributed machines, each with modest memory requirements and minor communication needs, linear equation solutions require a closely coupled system of processors with a large memory. Still, those requirements are not too onerous [69].

In discrete logarithm, the linear algebra is more serious problem, since solutions have to be carried out not modulo 2, but modulo large primes. Hence the parallelizations of Coppersmith and Montgomery do not provide any relief, and the original structured Gaussian elimination, Lanczos, and conjugate gradient methods as implemented in [48] are still close to best possible. Also, Lambert reported some careful analysis and improvements [50]. The table 2.1 presents the problems solved in ICM using various method. The algorithms reported in the table used

Table 2.1: Comparable problem sizes solved using ICM

Method	First phase	Second phase	problem size	Year
Linear sieve	$\surd$	$\surd$	192bits	1992
Gaussian integer	$\surd$	$\surd$	200bits	2000
Cubic sieve	$\surd$	—	150bits	2005
Number Field Sieve (special primes)	$\surd$	$\surd$	129digits	
Number Field Sieve	$\surd$	$\surd$	120 digits	2002
Number Field Sieve on $GF(2^n)$	$\surd$	$\surd$	2^{607}	2002

one of the methods such as linear sieve, Gaussian integer, cubic sieve and Number Field Sieve in the first phase and one of the linear algebra techniques such as Lanczos, conjugate gradient and Wiedemann methods in the second phase.

From the above discussions, it is observed that the popular methods for generating linear relations are linear sieve, cubic sieve, Gaussian integer and Number Field Sieve. The linear sieve method is considered in the present study. Generally, the generation of relation is highly dependent on the smoothness of elements in the group. The smoothness in turn depends on the size of the factor base. Consequently the number of relations to be generated depends on the size of factor base and the sieve length i.e., the range of numbers to be checked for smoothness. These two characteristics, namely size of factor base and the sieve length are studied and reported in the literature. Next, reducing the number of relations also studied by LaMacchia and Odlyzko through the experimental work on structured Gaussian elimination for a problem of size around 192 bits, in which it is mentioned that the linear system ought to be sparse and there should be considerable number of relations than unknowns to get better reduction [48]. Recently Roberto Avanzi proposed a new filtering technique called as harvesting method to reduce the size of the linear system, which removes duplicate equations and singletons from the system. He also claimed that it can improve the performance of ICM by more than 30% [5] . Finally the popular linear algebra techniques for solving the reduced linear systems are Lanczos, conjugate gradient and Wiedemann.

Some special algorithms

In this section we discuss briefly some algorithms that apparently do not work very well. In a field $GF(p)$, Well's [101] has shown that for any u, $1 \leq u \leq p-1$, if g is a primitive root modulo p, then one can write

$$\log_g u \equiv \sum_{f=1}^{p-2} (1-g^f)^{-1} u^f (mod p) \tag{2.7}$$

This was not designed as an algorithm at all. The Herlestam-Johannesson [38] method was designed to work over the field $GF(2^n)$, and was reported by those authors to work efficiently for fields as large as $GF(2^{31})$. However, the heuristics

used by those authors in arguing that the method ought to work efficiently in larger fields as well as to be very questionable. As usual, $GF(2^n)$ is represented as polynomials over $GF(2)$ modulo some fixed irreducible polynomial $f(x)$ of degree n over $GF(2)$. In order to compute the logarithm of $h(x)$ to base x. Herlestam and Johannesson proposed to apply a combination of the transformations.

$$h(x) \leftarrow h(x)^{2^r}, h(x) \leftarrow x^{-2^s} h(x) \tag{2.8}$$

so as to minimize the Hamming weight of the resulting polynomial, and apply this procedure iteratively until an element of low weight, for which the logarithm was known, was reached. There is no reason to expect such a strategy to work, and considerable numerical evidence has been collected which shows that this method is not efficient [14], and is not much better than a Pollard-random walk through the field. However some unusual phenomena related to the algorithm have been found whose significance is not yet understood. Another approach to computing discrete logarithms in fields $GF(2^n)$ was taken by Arazi [68]. He noted that if one can determine the parity of the discrete logarithm of u, then one can quickly determine the discrete logarithm itself. Arazi showed that one can determine the parity of discrete logarithms to base g fast if g satisfies some rather complicated conditions. Since being able to compute discrete logarithms to one base enables one to compute them to any other base about equally fast. However, so far no algorithm has been found for finding such primitive elements g in large fields $GF(2^n)$, nor even a proof that any such elements exists [68].

Attacks on ECC

Generic attacks All generic attacks such as Shanks baby step and giant step, Pollard-Rho method and Pohlig-Hellman methods are applicable to solve ECDLP.

Attacks using group structure of elliptic curve This section presents the attacks developed based on the structure of the elliptic curve group. The popular attacks of this kind are MOV, Weil-Descent and anomalous attack to name a few. The MOV attack depends on Weil-pairing or Tate-pairing.

Weil-Pairing

Weil-pairing is defined as follows

$e_n : E[n] \times E[n] \rightarrow \mu_n$ where $E[n]$ is set of n-torsion points such that $\{P \in E(k)|[n]P = O\}E[n] \cong Z/nZ \oplus Z/nZ, \mu_n = \{\alpha|\alpha^n = 1, \alpha \in \overline{k}\}$ If P and Q are n-torsion points then $e_n(P,Q)$ is an n^{th} root of unity in $\overline{k}$

$$e_n(P,Q) = \begin{cases} 1 & : \quad P = Q \\ \frac{f_P(Q)}{f_Q(P)} & : \quad \text{otherwise} \end{cases}$$

The properties of Weil pairing are interesting

(1)Bilinear: For $P_1, P_2, Q_1, Q_2 \in E[n]$

$$e_n(P1 \oplus P2, Q1) = e_n(P1, Q1).e_n(P2, Q2)$$

$$e_n(P1, Q1 \oplus Q2) = e_n(P1, Q1).e_n(P2, Q2)$$

(2) Non degenerate: If $e_n(P,Q) = 1 \quad \forall Q$ then $P = O$.

Similarly, the tate-pairing is a map of $e_n : \ E(F_q)[n] \times E(F_{q^k})[n] \rightarrow F^*_{q^k}$

MOV Attack

Consider E/F_p and $|E(F_p)| = l$ prime and therefore $E(F_p) \leq E[l]$ as subgroup. $E[l] = Z/lZ \oplus Z/lZ$ and $|E[l]| = l^2$;$E[l] = E(F_p) \oplus Z/lZ.Q$ for some $Q \in E[l]$

The Elliptic Curve Discrete Logarithm Problem is, given $T \in E(F_p)$ to compute m such that $T = mS$. The attack is

1.Compute $e_l(S,Q)$

2.Compute $e_l(T,Q) = e_l(mS,Q) = e_l(S,Q)^m$

by bilinearity property this become

$a = e_l(T,Q) = e_l(mS,Q) = e_l(S,Q)^m$

Suppose $e_l(S,Q) = b$ then $a = b^m$;$a = \mu_l^m$

m can be calculated if discrete logarithm can be calculated in $\mu_l \leq \overline{F_p}$. Actually an extension of F_p that contain μ_l will do. So we find k such that F_{p^k} contain μ_l i.e we need l to divide $|F^*_{p^k}| = p^k - 1$ [62].

Anomalous Attack

This attack is possible on the anomalous curve (i.e) $|\#E(F_p)| = p$. Here the attack on super anomalous curve is presented. Super anomalous curve is an

extension of anomalous curve. Satoh-Araki-Smart algorithm used to solve DLP over super anomalous elliptic curve is explained below [75]

Super anomalous elliptic curve over a ring $Z/nZ(n = \prod_{i=1}^{k} p_i^{e_i})$ is defined by extending anomalous elliptic curve over a prime field F_p. They have n points over a ring Z/nZ and p_i points over F_{p_i} $\forall$ p_i

The Elliptic curve $\tilde{\mathrm{E}}(F_p)$ is the set (x, y) on the curve $\tilde{\mathrm{E}}$= $y^2 = x^3 + a_p x + b_p (\bmod p)$ including a point at infinity Q_p.

If $\tilde{\mathrm{E}}(F_p) = p$ (anomalous) then $\tilde{\mathrm{E}}^*(F_p) \in \tilde{\mathrm{E}}(F_p)$ - $\{Q_p\}$

The lifting curve $E = y^2 \equiv x^3 + ax + b$ is an elliptic curve satisfying $a \equiv a_p (\bmod p)$ and $b \equiv b_p (\bmod p)$ and $a_p, b_p \in Fp$ and $a_p, b_p \in ZorZ/p^2Z$.

A point $(x, y) \in E(F_p)$ or $E(Z/p^2Z)$ satisfying $x \equiv x_p (\bmod p)$ and $y \equiv y_p (\bmod p)$

Elliptic curve over Z/nZ

Let the composite $n = \prod_{i=1}^{k} p_i^{e_i}$, k is the number of distinct prime factors of n and E be an elliptic curve $E = y^2 \equiv x^3 + ax + b$. A group $E(Z/nZ)$ is defined as the direct sum of k groups (i.e) $E(Z/nZ) = \oplus_{i=1}^{k} E(Z/p_i^{e_i}Z)$

The other popular attacks are Weil-descent attack [42] and faster attack on ECC using Pollard-Rho [92, 93, 94].

Experimental results on ECC attacks

Certicom [19] initiated an ECDLP challenge in november 1997 in order to encourage and simulate research on ECDLP. Their challenges consist of instances of ECDLP on a selection of elliptic curves. The challenge curves are divided into three categories listed below. In the following, $ECC_p - k$ denotes a randomly selected elliptic curve over a field F_p. $ECC2 - k$ denotes a randomly selected elliptic curve over a field F_{2^m}, and $ECC2K - k$ denotes a Koblitz curve over F_{2^m}. In all cases, the bit size of the order of the underlying finite field is equal or slightly greater than k.

- Randomly generated curves over F_p, where p is prime: $ECC_p - 79, ECC_p - 89, ECC_p - 97, ECC_p - 109, ECC_p - 131, ECC_p - 163, ECC_p - 191, ECC_p - 239, ECC_p - 359$

  .

- Randomly generated curves over F_{2^m}, where m is prime:$ECC2-79, ECC2-89, ECC2-97, ECC2-109, ECC2-131, ECC2-163, ECC2-191, ECC2-238$ and $ECC2-353$.

- Koblitz curves over F_{2^m}, where m is prime: $ECC2K-95, ECC2K-108, ECC2K-130, ECC2K-163, ECC2K-238$ and $ECC2K-358$.

Escott et al. [31] reported on their 1998 implementation of parallelized Pollard's rho algorithm which incorporates some improvements of Teske [92]. The hardest instance of the ECDLP they solved was the Certicom ECC_p-97 challenge. For this task they utilized over 1200 machines from at least 16 countries, and found the answer in 53 days. The following challenges are solved

- $ECC_p-79, ECC_p-89, ECC_p-97$
- $ECC2-79, ECC2-89, ECC2-97$
- $ECC2K-95, ECC2K-108$

Recently Menezes et al. [64] solved for any instance of the ECDLP over any elliptic curve on the week fields. They showed that the ECDLP can be solved significantly in less time using their method in comparison with the Pollard-Rho method to solve the hardest instances. Solving the ECDLP using the ICM is an open problem, but recently the DLP in hyper elliptic curves are solved using this method [5].

2.3.2 Improving the DLP using the computational approach

The computation of DLP can be improved by using efficient implementation of the traditional methods. The above algorithms solves the DLP in exponential and sub-exponential time. The running time of the Shanks and Pollard algorithms have not been improved to any substantial extent. Only improvements by constant factors have been obtained [72, 92, 98]. There has been progress, on the other hand, in obtaining fast parallel versions in which the elapsed time for the computation shrinks by a factor that is linear in the number of processors used [35, 72, 98].

However, the basic processing for any of these algorithms still requires a total of about $p^{1/2}$ steps, where p is the largest prime dividing the order of g. The lack of progress in several decades is very important. Many modern public key cryptosystems based on discrete logarithms, such as the U.S Digital Signature Algorithm (DSA) [63, 79], rely on the Schnorr method [80], which reduces the computational burden normally imposed by having to work in a large finite field by working within a large multiplicative subgroup Q of prime order q. The assumption is that the discrete log problem in Q cannot be solved much faster than $q^{1/2}$ steps. For q of order 2^{160}, as in DSA, this is about 10^{24} group operations. A mips-year is equivalent to about 3.10^{13} instructions, so breaking DSA, say, with the Pollard or Shanks algorithms would require over 10^{12} mips-year, which appears to be adequate for a while at least.

From the ICM point of view, the implementation of linear sieve and Gaussian integer method proposed by LaMacchia and Odlyzko is an example of improving the DLP though the computational approach. In the same line the cubic sieve method is implemented by Abhijit Das and Veni Madhavan. Further the Number Field Sieve method is implemented and improved though computationally [76, 77]. Through, a new filtering technique the ICM is improved by 30% [5]. The technique is used to reduce the large linear system generated in the first step of ICM to smaller system for the solving step.

From ECDLP perspective, the parallel collision search on binary anomalous curves is one way of improving the computation of DLP. The equivalence classes derived from the anomalous curve aid in improving the Pollard-Lambda method [33]. This allows to perform the parallel collision search to solve the ECDLP.

2.3.3 Improving the computation of DLP using the structure of group, exponents and order of other elements related to DLP

The algorithms developed to solve the DLP with additional information other than g and y such as the order of the group, the exponents and other details regarding the DLP are reported. The efficient algorithm to solve the DLP, when the factors

of $p-1$ are known and small is Pohlig-Hellman method.

Pohlig-Hellman Method

This is a popular algorithm introduced by Pohlig-Hellman on 1978. If the order of the group is known along with the complete factorization and the factors are relatively small then this attack is possible [71].
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

$$g^x = y \quad mod \quad p \Rightarrow \text{ can be reduced into}$$

$$\alpha^x = \beta \text{ mod p where} \alpha \text{ is } g^{(\frac{p-1}{p_1^{e_1}})}, g^{(\frac{p-1}{p_2^{e_2}})}, \cdots g^{(\frac{p-1}{p_k^{e_k}})}$$

and

$$\beta \text{ is } y^{(\frac{p-1}{p_1^{e_1}})}, y^{(\frac{p-1}{p_2^{e_2}})} .....y^{(\frac{p-1}{p_k^{e_k}})}$$

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 $x \bmod p-1$.

Other attacks

Apart from the above algorithms to solve the DLP, it is worth noticing the other popular attacks on this hard problem such as timing attacks, attacks on short exponents, attacks on prime order subgroup, malleability attack, man in the middle attack, man in the meet attack, insider's attacks, outsider's attack to name a few. Some of the above attacks are designed on the vulnerabilities of the selection of parameters for the protocols based on this hard problem, such as the generation of p , q and g. Minding p's and q's by R. Anderson and S.Vaudenay addresses the selection of these parameters and the vulnerabilities [4, 96].

The exponent selection is also a crucial point in the implementation of protocols. This may leads to give a way to powerful attacks on the systems. One of the attacks of this kind is the attack on short exponents by Van Oorschot's and Wiener [97]. In the current technology, it is considered as infeasible to compute

the DLP in a group of order ≈ 1024 bits. Van Oorschot and Wiener examined the difficulty of computing the DLP for an exponent of size ≈ 160 combined with the random prime p of size ≈ 1024 bits. They suggested the use of prime order subgroup along with the short exponents or the use of safe primes. Lim and Lee [56] have shown a specific interest in investigating the computation of the DLP in a prime order subgroup of order ≈ 160 bits of p of size ≈ 1024 bits by extracting $x \bmod O(\beta)$, where β is the product of elements of smooth order and the prime p is assumed to be of random and the $p-1$ has many small factors apart from the large one with 160 bits. In both the cases random primes are used for the computations and assumed to have many small factors apart from the large one for $p-1$. The difference with respect to the structure of $p-1$ is that, the former solved the problems of generators of order $|p-1|$ and the later solved the problems of generators of order $|q|$ and computations are restricted to prime order subgroup q.

Further Dan Boneh, Antoine Joux, Q. Phong and Nguyen reported an attack on Textbook El-Gamal and RSA encryption on the encryption of messages of smaller size [16]. Since the encryption procedure followed in the above two cryptosystems are simple operations such as exponentiation in RSA and multiplications or XOR in El-Gamal, the above attack breaks these systems, when the smaller size messages are used in the encryption, for example a session key to be encrypted using the above algorithms.

The chosen-ciphertext attack is a popular attack on DLP based schemes to recover the plaintext. The additional information used in this type of attack is the chosen ciphertext.

Chosen-ciphertext attack A chosen-ciphertext attack (CCA) is an attack model for cryptanalysis in which the cryptanalyst gathers information, at least in part, by choosing a ciphertext and obtaining its decryption under an unknown key. A number of otherwise secure schemes can be defeated under chosen-ciphertext attack. For example, the El-Gamal cryptosystem is semantically secure under chosen-plaintext attack, but this semantic security can be trivially defeated under a chosen-ciphertext attack. When a cryptosystem is vulnerable to chosen-ciphertext attack, implementers must be careful to avoid situations in which an

adversary might be able to decrypt chosen-ciphertexts (i.e., avoid providing a decryption oracle). For example, given an encryption (c_1, c_2) of some (possibly unknown) message m, one can easily construct a valid encryption $(c_1, 2c_2)$ of the message $2m$.To achieve chosen-ciphertext security, the scheme must be further modified, or an appropriate padding scheme must be used. The other scheme related to El-Gamal i.e., Cramer Shoup resists the chosen cipher text attacks [27, 6, 7].

Timing attacks are another class of attacks, based on the computational time required by the prominent operations of the DLP based schemes.

Timing attacks

Timing attacks attempt to exploit the variations in computational time for private key operations to guess the private key. This type of attack is primitive in the sense that no specialized equipment is needed. An attacker can break a smart card key by simply measuring the computational time required by the card to respond to user inputs and recording those user inputs. The viability of this attack is important to any smart card implementation using vulnerable cryptosystems. An attacker with prolonged passive eavesdropping ability may be able to break the private key and gain access to the information stored on the card. This will give the attacker access to sensitive information.

Timing Attack on ECC

The timing attack requires to find a partition of an input message set into groups that require different amounts of computational time depending on the key bits. For a timing attack to work, there must be some predictable variation in computational time dependent upon the input messages. The variations in computing times for multiplications and inverses allows to predict the key bits in an implementation of El-Gamal cipher in ECC. These two steps are the most time consuming steps in the process of adding two elliptic curve points or doubling an elliptic curve point. The implementation of these operations in ECC may leads a way to timing attacks.

2.4 Summary

In this chapter, we reviewed the contributions of researchers, specifically in terms of solving the DLP. This helps in identifying the importance of cryptanalysis on the DLP. Solving DLP is viewed by the researchers from various perspectives. The present study improves the computation of DLP using the third approach as discussed earlier. The approach exploits the additional information regarding the DLP other than g and y. In general, the methods developed by using the above approach help the cryptanalyst to create a trap door to break the system and for the cryptographer to built a secure system from the vulnerabilities caused by the trap doors. In particular, the computation of DLP is improved through the traditional ICM, specialized ICM and new methods. The ICM is the most effective attack on solving the DLP. The key parameters of ICM such as the size of the factor base and sieve length are studied from generation of relations perspective. The reduction in the linear system from larger to smaller system, which makes the second phase efficient is also a key factor for the overall performance of algorithm. Thus in the present study, the reduction, size of the factor base and sieve length are studied all together as a group from the perspective of both the generation and solving linear systems phases of ICM.

The ICM is studied so far as a general method for prime fields and for special type of primes using number field sieve. In the present study, the basic ICM is modified using the combination of Pohlig-Hellman and traditional ICM for some instances of primes to recover the ephemeral keys. Ephemeral keys are used in each session between the communicators in the popular DLP based cryptosystems. A new smoothness concept named as smooth numbers over Z_p^* is defined in the present study and the DLP is solved efficiently in reduced time using various techniques based on the above new smooth concept. Furthermore, the Random method of ICM and the third step (individual logarithm step) of ICM are improved through the characteristics of smooth numbers over Z_p^*. From the ECDLP point of view, Pollard family of algorithms are the most effective algorithms. In the present work the experiments are conducted on the finite field of $E_{a,b}(F_{2^m})$.

Chapter 3

On the Computation of Index Calculus Method

This chapter presents a new methodology for the pre-computation step of Index Calculus Method (ICM). The ICM has two steps, namely, a pre-computation and individual logarithm computation . In the pre-computation step, the logarithm of elements of a subset of group, known as a factor base, is computed and in the individual logarithm step the DLP is computed with the help of pre-computed logarithms of the factor base. In the present work, the pre-computation step is studied. As discussed in chapter 2, from the general algorithm of ICM, it is known that there are three steps that have a significant impact on the performance of the pre-computation step, namely, the generation of relations involving the logarithms of the primes in the factor base, the optional step of reducing their numbers for computation efficiency and solving the system for logarithms of elements in factor base. In the present study, the performance of ICM is improved through the efficient way of producing the smaller size matrix for the third step by using the reduction step and the generation step collectively. An improved algorithm is developed for the pre-computation step. Through the improved algorithm the performance of ICM is enhanced by 30 to 40%

3.1 Introduction

Several methods are proposed in the literature for ICM, such as Random method, linear sieve method to name a few. The linear sieve method is considered in the present study. This method consists of linear sieve to generate the linear system, structured Gaussian method to reduce the size of the linear system and one of the linear algebra techniques such as Lanczos, conjugate gradient and Wiedemann to solve the linear system. The Lanczos is considered in the present study due to its simplicity and suitability for sparse linear system.

Apart from the methods used in the linear sieve method, the computational parameters involved in the method are the bound (B) based on which the factor base is defined and the sieve length (C) i.e., a range of numbers chosen for the smooth test. The smooth test is used to form the linear relation. Traditionally, the steps in ICM are viewed into two groups such as generation as one group and reduction and solving as another. In the present study the steps are viewed as generation and reduction as one group and solving as another. This leads to introduce a parameter, namely, ratio (R) between the number of relations to number of unknowns and the performance of pre-computation step of linear sieve method is improved by optimizing the tuple (B, C, R), where B is the bound, C is the sieve length and R is the ratio.

3.2 Linear sieve method

The linear sieve method is an efficient method to solve the Discrete Logarithm Problem (DLP) on the prime field (Z_p^*). The steps involved in the pre-computation step of linear sieve method are as follows:

- The first phase is to find the linear relations relating the logarithms of the primes in the factor base by using linear sieve.
- The second phase is to reduce the system by structured Gaussian method and solve the system by using techniques from linear algebra.

The following sections review the methods involved in the linear sieve method.

3.2.1 Linear sieve for generating the linear relations

The popular sieving method, namely, the polynomial sieve is used in this method. The detailed steps involved in linear sieve are as follows:

1. Define a factor base based on the bound (B).

2. Form the linear system by using the polynomial sieve. The polynomial used in this sieving is $(H + c_1)(H + c_2)$, where $H = \sqrt{p}$, c_1 and c_2 lies in the range $0 < c_1 \leq c_2 \leq C$ and C is the sieve length i.e., range of numbers to be checked for smoothness.

The polynomial sieve is to compute the least non-negative residue of $(H+c_1)(H+c_2)$ and check for smoothness as given below:-

For some arbitrary value J, If $H^2 = p + J$ then,

$$\begin{aligned}(H + c_1)(H + c_2) &= H^2 + (c_1 + c_2).H + c_1.c_2 \\ &= p + J + (c_1 + c_2).H + c_1.c_2 \\ &\equiv J + (c_1 + c_2).H + c_1.c_2 \end{aligned} \tag{3.1}$$

For a given $c_1$andc_2, if the residue is smooth then it can be factored as

$$(H + c_1)(H + c_2) \equiv p1_1^{e_1}.....p_k^{e_k} \mod \text{p} \tag{3.2}$$

and the linear relation is given by

$$\log_g^{(H+c_1)} + \log_g^{(H+c_2)} = e_1.\log_g^{p_1} + ...e_k.\log_g^{p_k} \mod \text{p-1} \tag{3.3}$$

For fixed value of c_1 and by varying c_2, the value of c_2, for which the residue is smooth can be found using the following formula.

$$\begin{aligned}(H + c_1)(H + c_2) \mod \text{p} &= J + (c_1 + c_2)H + c_1c_2 \\ &= J + c_1H + c_2H + c_1c_2 \\ &= J + c_1H + c_2(H + c_1)\end{aligned}$$

$$= J + c_1 H + c_2(H + c_1)$$

$$c_2 = -(J + c_1 H)(H + c_1)^{-1} \text{ mod p}$$

Thus, c_2 is calculated for the powers of each prime in the factor base based on bound B. It is well known that for any polynomial $f(x)$, If the polynomial is divisible by some prime say p then $f(x + p)$ is also divisible by the same p. Since we consider the polynomial as $(H + c_1)(H + c_2)$, we add the logarithms of the prime in a sieve array indexed by c_2 i.e the logarithm of the prime that divides the polynomial for a specific c_2 is added to the sieve array.

For a fixed value of c_1 and by varying c_2, the residue $(H + c_1)(H + c_2)$ mod p is calculated. The residue is considered as smooth, if the logarithm of the residue is approximately near to the logarithm of the sieve array of index c_2. From the literature, the exact match can be obtained for .1 difference between the logarithm of the residue and the logarithm of the sieve array of index c_2. However, the difference parameter, say α, can be chosen such that $\alpha \leq 2.5$ [2]. The parameter α is chosen as 1 in the present study. If the residue is smooth it will be factored and the linear relation will be formed. The above step is repeated till the number of relations equals the number of unknowns in the factor base.

3.2.2 Methods to solve the relations

In general the linear system formed in the first phase of pre-computation step is larger in size. The linear system needs to be solved does not only have the logarithms of the primes in the factor base as unknowns. The logarithms of $H + c_1$ and $H + c_2$ factors are also unknowns in the factor base. Thus, the number of relations needed to form the system in the generation step increases predominantly. This is the reason for obtaining a large matrix from the generation step. The structured Gaussian method is an efficient method to reduce the large matrix in to a matrix of required size. The following section addresses the structured Gaussian elimination method and the subsequent section presents the Lanczos method.

Structured Gaussian elimination Method

This method is an adaptation and systematization of some of the standard techniques used in numerical analysis to minimize fill-in during Gaussian elimination. Additionally some more steps are designed to take the advantage of the special structure present in matrices arising from integer factorization and discrete logarithm algorithms. The structured Gaussian elimination method acts as a bridge between the sieving and solving phases of ICM. This method reduces a larger system into a smaller system and works efficiently when the matrix has more number of relations than the unknowns [48, 50]. LaMacchia and Odlyzko [48] described the following important steps involved in the structured Gaussian elimination method:-

- Label each column either light or heavy, depends on the number non-zero coefficients as less or many, respectively.
- Delete all the light columns that have one or fewer non-zero coefficients in comparison with the other columns and the rows, in which those columns have nonzero-coefficient.
- Delete some of the excess rows, which have the largest number of non-zero coefficients in the light columns.
- For any row, which has only a single non-zero coefficient that is equal to ± 1 in a light column, subtract the appropriate multiples of that row from all other rows that have non-zero coefficients on that column so as to make those coefficients as zero.

The above steps are repeated until the matrix has been reduced to the desired amount. The identification of light or heavy columns in the above procedure is purely dependent on the sparsity of the matrix generated in the generation step.

Lanczos Method

The Lanczos method is originally devised for solving the systems over the real numbers. The method for solving the systems over finite field is exactly the same

as the algorithm that is used over the real numbers [89]. We first describe the Lanczos algorithm. Suppose that we have to solve the system

$$|A|x = w \tag{3.4}$$

for a column n vector x, where A is symmetric $n \times n$ matrix and w is given n-column vector.

Let

$$w_0 = w \tag{3.5}$$

$$v = |A|w_0 \tag{3.6}$$

$$w_1 = v_1 - \frac{(v_1, v_1)}{(w_0, v_1)} w_0 \tag{3.7}$$

$$\tag{3.8}$$

and then, for $i \geq 1$, define,

$$v_{i+1} = |A|w_i \tag{3.9}$$

$$w_{i+1} = v_{i+1} - \frac{(v_{i+1}, v_{i+1})}{(w_i, v_{i+1})} w_i - \frac{(v_{i+1}, v_i)}{(w_{i-1}, v_i)} w_{i-1} \tag{3.10}$$

The algorithm stops, when it finds a w_j that is conjugate to itself, i.e. such that (w_j, Aw_j)=0. This happens for some $j \leq n$. If $w_j = 0$, then,

$$x = \sum_{i=0}^{j-1} b_i w_i \tag{3.11}$$

is a solution to equation 3.4 , where

$$b_i = \frac{(w_i, w)}{(w_i v_{i+1})} \tag{3.12}$$

If($w_j \neq 0$), the algorithm fails.

The Lanczos algorithm was invented to solve the systems with real coefficients. To solve systems over finite fields, simply the equations to be transformed into finite field. This causes possible problems, such as, over a finite field, a non zero vector may conjugate to itself. However, this difficulty can be overcome in practice. In addition to self conjugacy, the systems need to be solve are in general not symmetric but rather are of the form

$$|B|x = u \tag{3.13}$$

where B is $m \times n$, x is an unknown column n- vector and u is a given m- vector.

To solve the above equation a diagonal matrix of size $m \times m$ is chosen such that the diagonal elements are selected from the field F_p and the following equations are calculated.

$$|A| = |B|^T |D|^2 |B| \tag{3.14}$$

$$w = |B|^T |D|^2 u \tag{3.15}$$

.

The solution of equation 3.13 is then a solution to equation 3.4, and with high probability a solution to equation 3.4 will be a solution to equation 3.13. The random choice of D ensures that the rank of A will be the rank of B, this will not run into self-conjugate w_j in the Lanczos algorithm.

3.2.3 Computational parameters

Important computational parameters involved in ICM are the size of the factor base and the number of linear relations generated for computing the logarithms. Generally the factor base consists of all primes up to a certain bound B and the number of relations is governed by a parameter called the sieve length C. The time complexity of the pre-computation step, optimal Bound (B) and optimal sieve length (C) are derived in the literature in terms of $L_p[s; c]$ function. These two parameters are also analyzed experimentally by considering half the time for generation and half the time for solving in the paper [89] and the results are given below:-

Table 3.1: Optimal values for parameters

Parameter	Theoretical	Experimental
B	$O(L_p(\frac{1}{2}; \frac{1}{2} + O(1))$	$3.33L_p(0.5, 0.476)$
C	$O(L_p(\frac{1}{2}; \frac{1}{2} + O(1))$	$2.78L_p(0.5, 0.417)$

where $L_p[s; c]$ is a function of $\exp^{c(\log p)^s (\log\log p)^s}$. Since ICM is sub-exponential and the time complexity is always derived in terms of $L_p[s; c]$ and s indicates ICM

is sub-exponential, the parameters B and C are also derived in terms of same function.

3.3 Analysis on linear sieve method

The third step in ICM is the most significant step. The performance of this step can be improved by reducing the size of the linear system to be solved. The reduced system can be obtained from the reduction step. Thus, in the present study the performance of ICM is improved through the efficient way of producing the smaller size matrix for the third step by using the reduction step and the generation step collectively. The experimental results show that gains of nearly 30% may be achieved.

Traditionally, the steps in linear sieve method are viewed into two groups, such as generation as one group and reduction and solving as another. In the present study, the steps are viewed into two groups: generation and reduction of relations as one group and solving as another. The linear system is generated and reduced such that a smaller matrix is produced for the third step. Odlyzko [48] proposed that the more number of relations than unknowns, the more is the reduction using structured Gaussian method. This allows us to introduce a new parameter R (the ratio between the number of rows and number of columns) in the current study.

As the first step in linear sieve method is viewed collectively as generation and reduction of linear system, the parameters B, C and R become interdependent. The parameters involved in the method are viewed as a tuple (B, C, R). Thus, the problem is formulated as to investigate the optimal tuple (B, C, R) and develop an algorithm to obtain a smaller matrix for the second step (solving).

In the initial phase of our work the characteristics, namely, the size of the factor base, sieve length and the ratio are studied in detail all together as a group to improve the overall performance. The following methodology is used to obtain an optimal tuple (B, C, R) empirically.

- Analyze the structured Gaussian method to obtain R that produces as high a reduction is possible;

- Obtain a suitable pair (B, C), which will produce a linear system with the required R.

3.3.1 Empirical analysis on structured Gaussian elimination

In the first phase of our experiments, the performance of structure Gaussian method is analyzed. This method reduces a large system into smaller system and works efficiently when the matrix has more number of relations than unknowns [48, 50]. The performance of structured Gaussian elimination is analyzed using an appropriate database of safe primes to obtain an optimal ratio between the rows and columns of linear system. Therefore, obtaining an improved reduction will leads a way to achieve a smaller matrix for the solving step of linear sieve method. Since safe primes are considered as hard for exponential time algorithms, such as Shanks, Pollard-Rho, Pollard-Lambda, [67, 68, 69] to name a few, they are considered and solved through linear sieve method.
Let us describe as follows:
First we produce a data file consists of a list of tuples (p, g, q, m, M, s, S, k) with the following properties :- p is a problem to be solved, which is a safe prime, g is a generator, q is the largest prime factor of $p-1$, m is the minimum bound, M is the maximum bound, s is the minimum sieve length, S is the maximum sieve length and k is the size of problem p. The tuples are computed as follows:

- Select k between 15 to 40.
- Select q, an integer of size k digits.
- Check $p = 2q + 1$ is a prime or not using probabilistic primality test algorithm.
- Calculate (m, M) and (s, S) approximately using the minimum and maximum bound and sieve length for linear sieve from the literature [89].

Having built up the file the following procedure is computed

Heuristic Algorithm-1

1.Read the tuple from the data file

1.1 For every B from $m \leq B \leq M$

1.1.1 For every C from $s \leq C \leq S$ repeat the following

1.1.1.1 Generate the Linear relation for p B and c. If the ratio between rows and column is less than 1 skip the following step.

1.1.1.2 Reduce the system using structured Gaussian elimination method and solve the linear system using Lanczos.

1.1.2 Plot the reduced size of matrix to the ratio between the rows and the columns and obtain the ratio that minimize the size of the reduced matrix for a bound B.

1.2. From the results of optimal ratio and the size of reduced matrix of each bound B, obtain the ratio that minimizes the size of reduced matrix for the given problem p.

2. Plot the problem size to the ratio, which is obtained in the previous step and obtain the ratio that can be considered as the ratio that achieves the maximum reduction.

The first set of experiments on the structured Gaussian method are designed to find the ratio R, which produces the best reduction for a set of different sized problems. The plot of the ratio versus the size of reduced matrix for a given problem shows that when the ratio increases the size of reduced matrix decreases and after reaching some point it starts increasing. The valley gives the best reduction for a given problem. Figure 3.2 shows the reduction, when ratio increases for a specific problem of size 25 digits. This figure shows that the size of reduced matrix decreases, when the ratio increases and starts increasing after reaching some point. In particular, figure 3.2 shows that the linear relations of a 25 digits problem is reduced into much smaller system, when the ratio is around 1.8, which is considered as the optimal ratio for this problem.

From the different sized problems, it is observed that the reduced size does not vary much for ratios between 1.7-2.4. Figure 3.1 shows the optimal ratio between the rows and the columns obtained by implementing the above heuristic algorithm-1 for problems of size between the range 17-30 digits considering the minimal set of bound and sieve length. The figure 3.1 shows that the optimal ratio is ranges from 1.8 to 3.0. We have chosen to fix the ratio of 2.0 for deciding the number of relations to be generated in the generation step of pre-computation step of ICM. In other words, we generate twice as many relations as there are unknowns.

Similarly, figure 3.4 shows the size of reduced matrix, when bound increases for different size of problems. This figure shows that the size of reduced matrix is less, when the bound is less. It starts increasing as the bound increased and saturates after reaching some point. From the figure 3.4, it can be said that the bound also influences the percentage of reduction. The reduction is high (the size of reduced matrix is low) for smaller bounds than the higher bounds.

3.3.2 Analysis on bound(B) and sieve length(C) for the ratio(R)

After fixing R as 2.0, the next set of experiments are conducted on obtaining the relationship between B and C, that could produce the linear relations in the required ratio 2.0. The following procedure is adapted for retrieving the above mentioned pair.

Heuristic Algorithm-2

1. Read the tuple from the data file.

1.1 For every B from $m \leq B \leq M$ repeat the following

1.2.1 Choose the sieve length (C) from the range $s \leq C \leq S$.

1.2.2 Generate the linear relation till the ratio is met.

otherwise increment or decrement sieve length.

1.2.3 Obtain the sieve length.

1.3 Plot bound vs sieve length.

2. Find the relationship between bound and sieve length for fixed ratio 2.0 that minimizes the size of the reduced matrix from the above results

The plots of bound versus sieve length for a set of different size problems show that the sieve length is exponentially high, when the bound is small. It starts decreasing as the bound increased, and finally saturates. The number of primes in the factor base of smaller bounds are less compared with larger bounds. This makes the search for smooth elements difficult for smaller bounds and needs more number of elements (sieve length) to be searched. This leads to increase the time for generation step. On the other hand for larger bounds, the search for smooth elements is easy and needs more number of smooth elements to form the linear system. This leads to increase the time for solving step.

The figure 3.3 shows that the result on sieve length against the bound for the fixed optimal ratio for the problems of size around 25 digits. The figure 3.3 reports that the sieve length is exponentially very high for some smaller bounds, this will drastically increase the running time of generation step. On the other hand for the larger bounds, even though the sieve length is less the number of primes in the factor base is high, this will increase the time for solving step. Figure 3.3 also shows that the optimal bound ranges from 2500 to 10000 and the sieve length ranges from 1000 to 6000. Similarly the figure 3.5 shows the relation between the bound and the sieve length for different size problems. By examining the results of different size of problems, we found that the lower bound where $C \leq 2B$ is an appropriate choice, beyond this will drastically increase the generation time. With the above results we developed an algorithm for pre-computation step of linear sieve method. In the proposed algorithm the linear system is generated iteratively by satisfying the ratio and the relationship between the bound and sieve length. The initial bound is the lower bound for the linear sieve method. The following section demonstrates the algorithm developed for the pre-computation step of linear sieve method along with the experimental results.

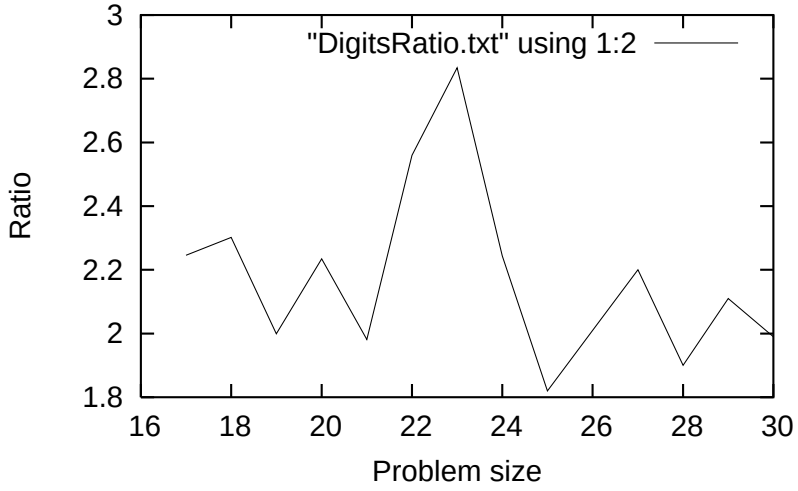

Figure 3.1: Optimal ratio of problems from 17 digits to 30 digits

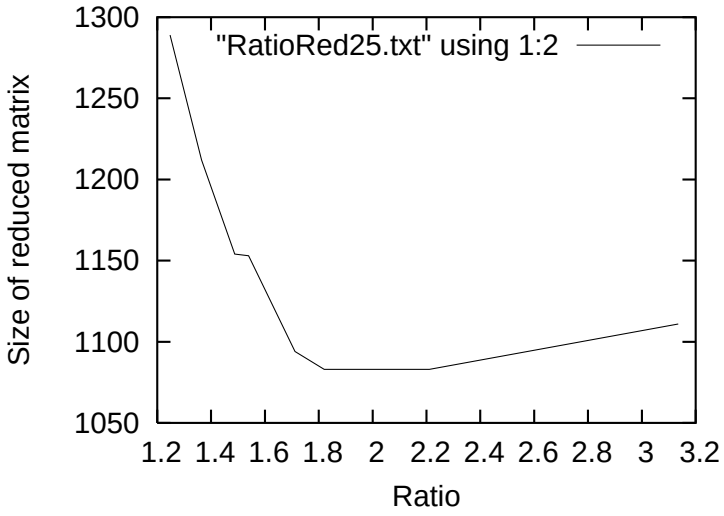

Figure 3.2: Size of reduced matrix for a problem of size 25 digits

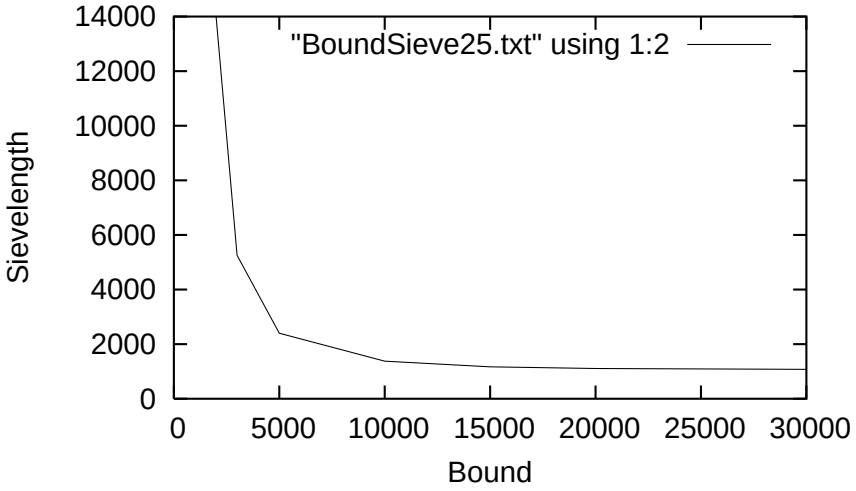

Figure 3.3: The relationship between the Bound(B) and the Sieve length (C) for a problem of size 25 digits

3.4 Algorithm for pre-computation step of linear sieve method

In the present study, the parameters bound, sieve length and the ratio are studied collectively as a group. An algorithm is developed based on the results we obtained. The performance of the algorithm is compared with the information given in the table 3.1 reported in section 3.2.3. The algorithm presented in the following section is based on two information, one is the performance of structured Gaussian elimination as it decides the optimal ratio between the rows and the column and the another one is the dependency between the bound and the sieve length, that corresponds to optimal ratio. These are considered as key issues as they are independent from implementation point of view and need to be known to develop an algorithm to improve the performance of linear sieve method in all aspects. Initial value for the bound is computed from the lower bound for the linear sieve reported in the literature [89]. From the conjecture of bound and the sieve length presented in the section 3.2.3 and the empirical values reported in the table 3.1 , the initial value for the sieve length is assigned as the initial value of the bound. The linear relation is generated iteratively till the optimal ratio and

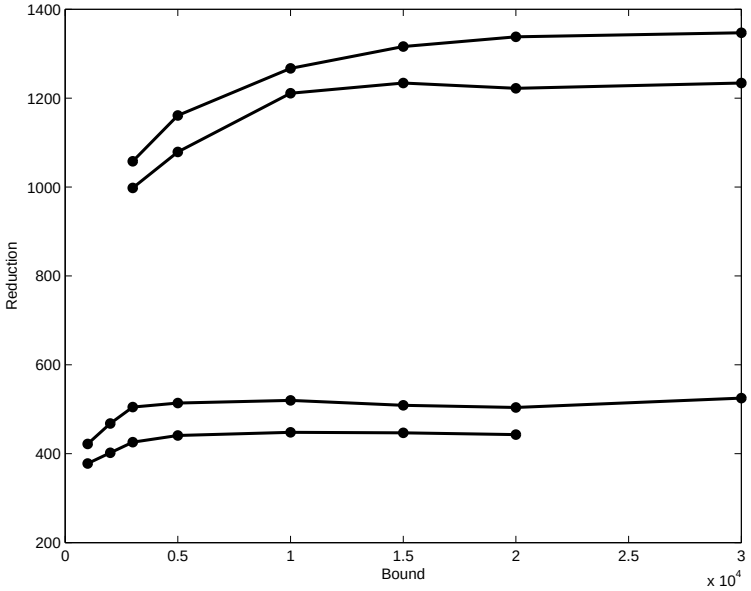

Figure 3.4: Size of reduced matrix of different bound for different size of problems

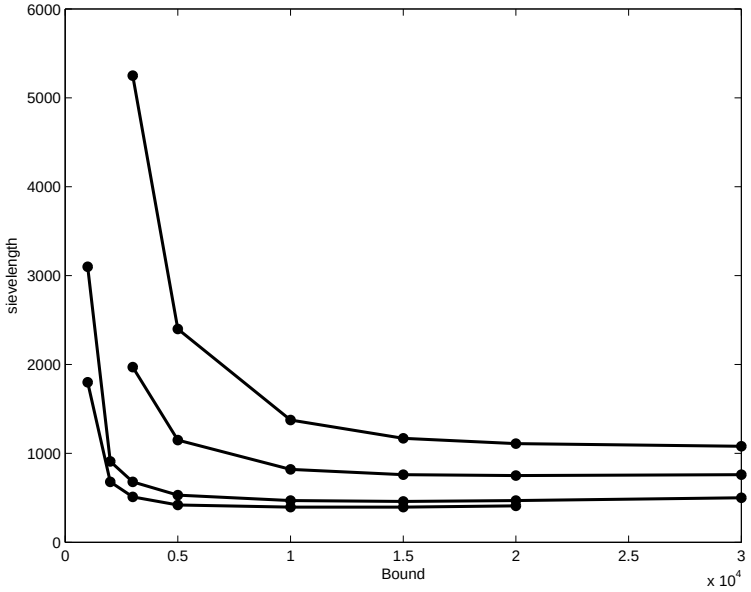

Figure 3.5: Bound and the Sieve length for different size of problems

the relationship between the bound and the sieve length are satisfied.

3.4.1 Improved algorithm for linear sieve method

In the Algorithm-1, step 8 is to generate the relation in required ratio, step 22 is to reduce the linear system into smaller one and step 23 is to solve the linear system. The terminating condition $C = k * B$ is chosen from the heuristic analysis given above, where k may be chosen arbitrarily ≤ 2.

3.4.2 Performance study and numerical results

We implemented the Algorithm-1 to ensure the efficiency of the algorithm and conducted experiments on an appropriate data base of safe primes i.e., primes of the form $2q + 1$, where q is also a prime. The use of safe prime is considered as safe for cryptographic applications. The advantage of using safe primes is that, all group elements of Z_p^* other than ± 1 are known to have order either q or $2q$. Consequently $g = 2$ known to be of generator for full group of order $2q$ elements if 2 is quadratic non residue *mod* p and generator for the group of order q when 2 is quadratic residue *mod* p. In modular exponentiation using $g = 2$, the multiples cost is decreases substantially and it almost disappears, only squaring cost remains in the exponentiation. The another usage of this prime is that it precludes the attacks, which are working on the small factors of $p - 1$. Apart from the large prime factor q, to retrieve the partial information about the exponent x, the partial Pohlig-Hellman decomposition is used for random primes. This way of obtaining the partial information about the x yields only one bit of information from safe primes. In the present study, the pre-computation step of linear sieve method is implemented using the algorithm discussed in the previous section to solve the DLP defined on the safe primes. The data base of safe primes is created based on the following algorithm.

Algorithm-2

- Select k between 15 to 50.

Algorithm 1 Algorithm for the pre-computation step of linear sieve method

INPUT : p (Problem to be solved)

OUTPUT : RESULT (The discrete logarithm for the primes in the factor base)

VARIABLES : $n.B, B, C, Ratio, tag = 0, flag = 0, p$

1: Read the input(p)
2: Set $n.B = \lceil e(\sqrt{(logp)}\rceil$; $C = n.B$; $B = n.B$.
3: Repeat the following till the required size of matrix is obtained for the bound and sieve length
4: BEGIN
5: $C = k * B * m$, where m is to be close to 0
6: Repeat the following steps 7 to 20
7: BEGIN
8: Generate the relations using the linear sieve method
9: Compute the ratio between the rows and columns
10: **if** $C \leq k * B$ and ratio is equal to 2.0 **then**
11: Go to step 22
12: **else**
13: **if** $C = k * B$ and ratio is $\neq$ 2.0 **then**
14: $B = B + l * n.B$, where l is close to zero when ratio is close to 2.0 or close to 1 when ratio is close to zero.
15: Go to step 5
16: **end if**
17: **else**
18: $C = k * B * m$, where m is to be varied based on the value of C and the ratio
19: **end if**
20: END
21: END
22: Reduce the linear system using the structured Gaussian method
23: Solve the linear system using the Lanczos method

- Select q, an integer of size k digits.
- Check $2q+1$ is a prime or not using probabilistic primality test algorithm.
- Find the generator g, such that it should be a prime and $g \in FB$, where FB is a factor base formed from the lower bound of linear sieve.
- Create a 3-tuple as (p, q, g), where p is a prime of the form $2q+1$, q is a prime and g is a generator.

Having built up the database the following algorithm is executed to test the Algorithm-1 discussed in the previous section.

Algorithm-3

- Read the 3-tuple (p, q, g).
- Generate the linear relation using the **Algorithm-1**.
- The linear relation is reduced using structured Gaussian elimination.
- The reduced linear system is of the form $Ax = 0$, where A is the coefficient matrix and x is a one dimensional unknown matrix. This linear system is transformed into the form $Ax = B$, where B is formed by using the logarithm of the generator. This linear system is solved by using the Lanczos method.
- The logarithms $mod\ p$ of factor base elements can be obtained by using Chinese Remainder Theorem for logarithms $mod\ p$ from logarithms $mod\ q$ and $mod\ 2$.
- keep the running time for each tuple.
- Repeat the above procedure for all the tuples.

The above algorithm is executed on a data base of safe primes and the reports are tabulated in table 3.2. Table 3.2 reports the average running time and the number of iteration for problems of size from 17 digits to 50 digits. The algorithm is tested for both sieving and solving phase up to 40 digits, beyond that

Table 3.2: Results on improved linear sieve method with k=1

Problem size in digits	Running time in sec	Bound	Sieve length	No of iteration
17	4	1042	795	2
18	7	1250	895	2
19	11	1492	1067	2
20	14	1770	1255	2
21	26	2094	2094	2
22	32	2466	2466	2
23	53	2894	2894	2
24	72	3384	3384	2
25	121	4000	4000	2
26	171	4586	4586	2
27	238	5314	5314	2
28	338	6140	6140	2
29	459	7200	7200	2
30	677	8200	8200	2
31	985	10897	10897	3
40	6hours	38000	38000	3
45	-	81000	81000	3
50	-	160000	160000	4

Table 3.3: Comparison between the running time of improved linear sieve method and the empirical results in table 3.1 with k=1

Problem size	Bound from table 1	Bound obtained from Algorithm	Running time using results from table 1 in sec	Running time of Algorithm in sec
17	998	1042	4	4
18	1233	1250	6	7
19	1515	1492	10	11
20	1900	1770	13	14
21	2260	2064	26	26
22	2740	2466	38	32
23	3310	2894	58	53
24	4000	3384	88	72
25	4800	4000	130	120
26	5730	4586	191	171
27	6840	5314	265	238
28	8140	6140	395	338
29	9660	7200	578	459
30	11440	8200	947	677

only linear relation is generated and the parameters bound and sieve length are obtained. Similarly the traditional linear sieve method is executed with the same data base of safe primes except the linear relation is created using the bound and sieve length computed from the table 3.1, discussed in section 3.2.3. The running time is compared with the running time of linear sieve method implemented using the proposed algorithm discussed in the previous section. Table 3.3 tabulates the difference in the running time between the traditional and Algorithm-1

3.5 Comparative analysis

The table 3.3 ensures the performance of improved pre-computation step of Linear sieve method. The investigation of the relationship among the parameters in the tuple (B, C, R) aids in improving the performance of linear sieve method. This section reports the comparative analysis between the parameters listed in the table 3.1 discussed in section 3.2.3 with tuple (B, C, R) investigated in the present study. The bound (B) and the sieve length (C) in the table 3.1 are functions of problem size (p). The bound and the sieve length in the tuple (B, C, R) are related such that $C \leq 2B$ and this is obtained through the experimental results for the fixed ratio R. The bound B chosen for the list of problems in the newly proposed pre-computation step are comparatively less with the empirical results given in the table 3.1. The sieve length presented in the table 3.1 is comparatively less with the sieve length chosen in the newly proposed algorithm.

For example a 20 digit problem is solved by using a bound $B \approx 1900$ and the sieve length of $C \approx 800$ obtained from the table 3.1, this shows that the B and C are related with independent of ratio as $C = .421B$. The bound and the sieve length chosen for the same problem are $B \approx 1770$ and $C \approx 1400$ with dependent on ratio as $R = 2.0$, when it is solved by using the newly proposed pre-computation step of linear sieve method. Through the experimental results we obtained $C \approx .5B$ to $.9B$ for smaller problems and $C = B$ for problems of size ≥ 21. In the present study, the problems of size up to ≈ 40 digits are solved by using both generation and solving steps of Algorithm-1 and the parameters in the tuple (B, C, R) are $R \approx 2$ and $C = B$. The problems of size up to ≈ 55 digits are solved by using generation step and the tuple (B, C, R) are $R \approx 2$ and $C = B$.

3.6 Conclusion

In this chapter we discussed a new approach to improve the performance of ICM. An improved algorithm is proposed for the pre-computation step of ICM and new results are reported on the parameters used to generate and solve the relations. The parameters are analyzed as a group on the performance of ICM. We showed

that including reduction alongside generation permits the selection of B and C such that the overall time needed for the pre-computation step of ICM is reduced. Also, we reported that, if the number of relations is twice the number of unknowns, we get improved performance in the structured Gaussian step resulting in a smaller set of relations for solving by Lanczos method. Finally we conclude that the performance of ICM is improved by 30 to 40 % by selecting the smallest bound B such that $C \leq 2B$.

Chapter 4

A New Method for Computing DLP Based on Extending Smooth Numbers to Finite Fields

This chapter introduces the concept of B-smooth numbers over Z_p^*. Their distribution for different types of primes is studied and it is shown that there are two major classes. These lead to the development of an efficient algorithm for solving DLP, when the exponent lies near the middle of a group, i,e., $\frac{p}{2}$ or the prime order subgroup, i.e., $\frac{q}{2}$ for primes of the form $p = 2q + 1$. Also with a simple extension the algorithm may be used for $p = 2\rho + 1$, where ρ is a product of primes. The examination on problems of size up to 1024 bits reveals that the exponent in the above interval can be retrieved easily in reduced cost.

4.1 Introduction

The popular DLP based public key cryptosystems are based on the hardness of solving the DLP. The DLP is a mathematical problem defined on a finite field Z_p^*. To ensure the security of the system the prime p to be chosen appropriately regarding the bit size and the structure. The bit size of 1024 and with a large prime factor of $p - 1$ are recommended and considered as safe.

The DLP can be solved through the trapdoors in reduced time bound, when

additional information is available apart from g and y. The subgroup confinement attack is one of the popular attacks of this kind. In this attack the element y is confined to a small subgroup and retrieved in reduced time using exponential time algorithm. Similarly, Van Oorschot and Weiner [98] presented a method to retrieve the short exponents (x) and suggested the use of prime order subgroup or the safe primes. Further, Lim and Lee [56] proposed a key recovery attack on prime order subgroup for certain DLP based schemes.

This chapter introduces a new smoothness concept over Z_p^*. This is an analogous definition of Y-smooth integers. An integer is Y-smooth if all its factors are less than or equal to Y. The distribution of smooth numbers over Z_p^* is studied and it is found to exhibit different patterns depending on the type of primes. The primes are classified based on the factors of $p-1$.

The patterns generated from the smooth number set over Z_p^* aid in developing some new methods to solve the DLP. The DLP is solved with the additional information regarding the structure of p and the range of the exponents. The structure of p, such as $p-1=2\rho$, where ρ is a prime or product of primes is chosen. The range of exponent x is assumed to lie near the middle of the group i.e., $\frac{\rho}{2}$. In general the primes with one of the factors of $p-1$ is large and in particular when $p=2\rho+1$ and one of the factors of ρ is large are considered as safe in the literature. In the present study the DLP in the above groups are solved .

4.2 Smooth numbers and their distributions for certain primes

An integer is called *Y-smooth* if it has no prime factors larger than Y. For example, the number 48 is 3-smooth because its prime factors are only 2 and 3. An analogous definition is proposed for smoothness over Z_p^* as follows:

Definition 1 *Let Z_p^* be a multiplicative group of a prime field p and let B be a factor base ($B \subset Z_p^*$). Then an element $s \in Z_p^*$ is B-smooth over Z_p^* iff*

$$s \equiv \prod_{b_i \in B} {b_i}^{c_i} \bmod p \tag{4.1}$$

In general, the factor base B consists of all primes less than an upper bound p_B and the above definition is equivalent to the conventional definition of smooth numbers (p_B smooth in this case). The difference is that the factor base may now contain *an arbitrary set of primes* and the new definition of smoothness is necessary to handle such a case. The importance of this difference becomes more evident in the rest of the chapter.

Let S_p^B denote the set of all B-smooth numbers over Z_p^*. It should be noted that if the factor base B contains a generator of Z_p^*, then S_p^B contains all the elements of the group. Otherwise, $S_p^B \subset Z_p^*$. It is the latter case that we explore in this chapter and exploit in our approach.

Let b be an element of Z_p^*. We define a smoothness function

$$F_s(b) = \begin{cases} 1 & \text{if } b \in S_p^B \\ 0 & \text{otherwise.} \end{cases} \tag{4.2}$$

Let $K = \{x : x \in Z_p^* \text{ and } 1 \leq x \leq \frac{p-1}{2}\}$ and $L = \{x : x \in Z_p^* \text{ and } \frac{p-1}{2} < x \leq p-1\}$, and let

$$X = \{F_s(b), b \in K\}, \text{ and} \tag{4.3}$$

$$Y = \{F_s(b), b \in L\} \tag{4.4}$$

Note that the cardinality of the sets X and Y is the same.

Let w_1 represent the string of 0's and 1's obtained by writing out in order the elements of the set X and w_2 represent a similar string obtained from the set Y.

Let a_i be the i^{th} element of w_1 and b_j be the j^{th} element of w_2.

Definition 2 (Reverse Pattern) *w_2 is defined as the reverse of w_1 and denoted as $w_2 = w_1^R$ if $a_i = b_{p-i}, \forall a_i$.*

Definition 3 (Complement Pattern) *w_2 is defined as the complement of w_1 and written as $w_2 = w_1^C$ if $a_i = b'_{p-i}, \forall a_i$ where $'$ denotes complement operation.*

Definition 4 (Palindrome Pattern) *Let w denote the string w_1w_2. Then w exhibits a palindrome pattern if w_2 is the reverse of w_1.*

Definition 5 (Inverse Palindrome Pattern) *Let w denote the string w_1w_2. Then w exhibits an inverse palindrome pattern if w_2 is the complement of w_1.*

4.2.1 Identified patterns

The properties of B- smooth numbers are investigated on five types of primes, classified based on the factors of $p-1$ as given 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 1 *Let p be a prime with $p-1 = 2q$, where q is a prime. Let the factor base B be chosen such that it contains elements of order $(p-1)/2$, i.e., of order q. In such a case, the distribution of B-smooth numbers over Z_p^* exhibits an inverse palindrome pattern.*

Proof.

1. Let $x \in Z_p^*$ be an element of $O(q)$. Then $x^q \equiv 1$ and $-x^q \equiv -1$ which means that $-x$ is a generator and non-residue. Therefore, if $x \in K$ and $F_s(x) = 1$ then $-x \in L$ and $F_s(-x) = 0$; and if $x \in L$ and $F_s(x) = 1$ then $-x \in K$ and $F_s(-x) = 0$.

2. Let x be an element of $O(2q)$. Then x is a generator and non-residue.

$$x^{\frac{p-1}{2}} \equiv -1; -x^{\frac{p-1}{2}} \equiv 1$$

That is, $-x$ is an element of $O(q)$. Therefore, if $x \in K$ and $F_s(x) = 0$ then $-x \in L$ and $F_s(-x) = 1$; and if $x \in L$ and $F_s(x) = 0$ then $-x \in K$ and $F_s(-x) = 1$.

3. Let $x \in K$ be an identity element; then, $-x$ is an element of order 2 and $x \in L$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 0$.

(1), (2) and (3) prove that $w_1 w_2$ exhibits an *inverse palindrome pattern.*

It may also be noted that if B, the factor base, contains only elements of $O(2)$, then $w_1 w_2$ represents a *palindrome pattern.*

Theorem 2 *Let p be a prime with $p - 1 = 2q_1q_2$ where q_1 and q_2 are primes. Let the factor base B be chosen such that it contains elements of $O(q_1q_2)$. Then the set S_p^B contains elements of $O(q_1), O(q_2)$ and $O(q_1q_2)$ and the distribution of $B-$smooth numbers over Z_p^* exhibits an inverse palindrome pattern.*

Proof.

1. Let $x \in Z_p^*$ be an element of $O(q_1q_2)$. Then,

$$x^{q_1q_2} \equiv 1; -x^{q_1q_2} \equiv -1$$

That is, $-x$ is a generator.

2. Let $x \in Z_p^*$ be an element of $O(q_1)$. Then,

$$x^{q_1} \equiv 1; -x^{q_1} \equiv -1$$

It shows that $-x$ is an element of $O(2q_1)$. Similarly, if x is an element of $O(q_2)$, then $-x$ is an element of $O(2q_2)$.

From (1) and (2), it may be seen that if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 0$; if $x \in L$ and $F_s(x) = 1$, then $-x \in K$ and $F_s(-x) = 0$.

3. Let $x \in Z_p^*$ be an element of $O(2q_1q_2)$. Then,

$$x^{q_1q_2} \equiv -1; -x^{q_1q_2} \equiv 1$$

As $q_1 \neq q_2$ and $O(2q_1)$ and $O(2q_2)$ are even, $-x$ is an element of $O(q_1q_2)$.

4. Let x be an element of $O(2q_1)$. Then,

$$x^{q_1} \equiv -1; -x^{q_1} \equiv 1$$

Thus, $-x$ is an element of $O(q_1)$. Similarly, it may be shown that $-x$ is an element of $O(q_2)$ if x is an element of $O(2q_2)$.

From (2) and (3), it may be seen that if $x \in K$ and $F_s(x) = 0$, then $-x \in L$ and $F_s(-x) = 1$; if $x \in L$ and $F_s(x) = 0$, then $-x \in K$ and $F_s(-x) = 1$.

5. Let $x \in K$ be an identity element, then $-x \in L$ and is an element of $O(2)$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 0$.

It is proved from the above cases that $w_1 w_2$ exhibits an *inverse palindrome pattern.*

Corollary 1 *Let B contain only elements of $O(2q_1)$. Then, S_p^B consists of elements of $O(2), O(q_1)$ and $O(2q_1)$ and the distribution of $B-$smooth numbers exhibits a palindrome pattern.*

Corollary 2 *Let B contain only elements of $O(2q_2)$. Then, S_p^B consists of elements of $O(2), O(q_2)$ and $O(2q_2)$ and the distribution of $B-$smooth numbers exhibits a palindrome pattern.*

Both the above corollaries are easily proven from (2) and (4) of theorem 2. Similarly, it may also be shown that if B contains only elements of $O(q_1)$ or $O(q_2)$, the distribution of $B-$smooth numbers show no discernible patterns.

Theorem 3 *Let p be a prime with $p-1 = 2q^n$, where q is a prime. Let the factor base B be chosen such that it contains elements of order $(p-1)/2$, i.e., of order q^n. In such a case, the distribution of B-smooth numbers over Z_p^* exhibits an inverse palindrome pattern.*

Proof.

1. Let $x \in Z_p^*$ be an element of $O(q^m)$, where m varies from 1 to n Then,

$$x^{q^m} \equiv 1; -x^{q^m} \equiv -1$$

That is, $-x$ is a generator for $O(2q^m)$.

2. Let $x \in Z_p^*$ be an element of $O(2q^m)$. Then,

$$x^{\frac{2q^m}{2}} \equiv -1; -x^{\frac{2q^m}{2}} \equiv 1$$

It shows that $-x$ is an element of $O(q^m)$.

From (1) and (2), it may be seen that if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 0$; if $x \in L$ and $F_s(x) = 1$, then $-x \in K$ and $F_s(-x) = 0$, since the B-smooth numbers set consists of elements of order q^m.

3. Let $x \in K$ be an identity element, then $-x \in L$ and is an element of $O(2)$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 0$.

It is proved from the above cases that w_1w_2 exhibits an *inverse palindrome pattern.*

Theorem 4 *Let p be a prime with $p-1 = 2q^n$. Let the factor base B be chosen such that it contains elements of order $2q^m$, where m varies from 1 to $n-1$. Then the set S_p^B consists of elements of $O(2), O(q^m), O(2q^m)$, where m varies from 1 to $n-1$ and the distribution of B-smooth numbers over Z_p^* exhibits a palindrome pattern.*

Proof.

1. Let $x \in Z_p^*$ be an element of $O(2q^m)$. Then,

$$x^{\frac{2q^m}{2}} \equiv -1; -x^{\frac{2q^m}{2}} \equiv 1$$

That is, $-x$ is an element of $O(q^m)$.

2. Let $x \in Z_p^*$ be an element of $O(q^m)$. Then,

$$x^{q^m} \equiv 1; -x^{q^m} \equiv -1$$

It shows that $-x$ is an element of $O(2q^m)$.

From (1) and (2), it may be seen that if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 1$; if $x \in L$ and $F_s(x) = 1$, then $-x \in K$ and $F_s(-x) = 1$, since the S_p^B consists of elements of $O(q^m)$ and $O(2q^m)$.

3. Let $x \in K$ be an identity element, then $-x \in L$ and is an element of $O(2)$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 1$.

It is proved from the above cases that w_1w_2 exhibits an *palindrome pattern.* Similarly, if B contains only elements of $O(q^m)$, where m varies from 1 to $n-1$, the distribution of $B-$smooth numbers show no discernible patterns.

Theorem 5 *Let p be a prime with $p-1 = 2^n q$, where q is a prime. Let the factor base B be chosen such that it contains elements of order $2^m q$, where m varies from 1 to $n-1$. Then the set S_p^B consists of elements of $O(2), O(q), O(2^m q)$ and $O(2^m)$ and the distribution of $B-$ smooth numbers over Z_p^* exhibits a palindrome pattern.*

Proof.

1. Let $x \in Z_p^*$ be an element of $O(2^m q)$. Then,

$$x^{\frac{2^m q}{2}} \equiv -1; -x^{\frac{2^m q}{2}} \equiv -1$$

That is, $-x$ is an element of $O(2^m q)$.

2. Let $x \in Z_p^*$ be an element of $O(2q)$. Then.

$$x^{\frac{2q}{2}} \equiv -1; -x^{\frac{2q}{2}} \equiv 1$$

It shows that $-x$ is an element of $O(q)$.

3. Let $x \in Z_p^*$ be an element of $O(q)$. Then,

$$x^q \equiv 1; -x^q \equiv -1$$

It shows that $-x$ is an element of $O(2q)$.

4. Let $x \in Z_p^*$ be an element of $O(2^m)$. Then,

$$x^{2^m} \equiv 1; -x^{2^m} \equiv 1$$

It shows that $-x$ is an element of $O(2^m)$.

From (1),(2),(3),(4) and (5), it may be seen that if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 1$; if $x \in L$ and $F_s(x) = 1$, then $-x \in K$ and $F_s(-x) = 1$.

5. Let $x \in K$ be an identity element, then $-x \in L$ and is an element of $O(2)$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 1$.

It is proved from the above cases that $w_1 w_2$ exhibits an *palindrome pattern.*

Corollary 1 *Let B contain only elements of $O(2^l)$, where l varies from 1 to n. Then, S_p^B consists of elements of $O(2), O(2^l)$ and the distribution of $B-$smooth numbers exhibits a palindrome pattern.*

The above corollary can be easily proven from (4) of theorem 5. Similarly, if B contains only elements of $O(q)$, the distribution of $B-$smooth numbers show no discernible patterns.

Theorem 6 *Let p be a prime with $p-1 = 2^x q^n$, where q is a prime. Let the factor base B be chosen such that it contains elements of order $2^m q^l$, where m varies from 1 to $x-1$ and l varies from 0 to n. Then the set S_p^B consists of $O(2), O(2^m q^l)$ and $O(q^l)$, where m various from 1 to $x-1$ and l various from 0 to n, and the distribution of B-smooth numbers over Z_p^* exhibits a palindrome pattern.*

Proof.

1. Let $x \in Z_p^*$ be an element of $O(2^m q^l)$. Then,

$$x^{\frac{2^m q^l}{2}} \equiv -1; -x^{\frac{2^m q^l}{2}} \equiv -1$$

   That is, $-x$ is an element of $O(2^m q^l)$.

2. Let $x \in Z_p^*$ be an element of $O(q^l)$. Then,

$$x^{q^l} \equiv 1; -x^{q^l} \equiv -1$$

   It shows that $-x$ is an element of $O(2q^l)$.

3. Let $x \in Z_p^*$ be an element of $O(2q^l)$. Then,

$$x^{\frac{2q^l}{2}} \equiv -1; -x^{\frac{2q^l}{2}} \equiv 1$$

   It shows that $-x$ is an element of $O(q^l)$.

4. Let $x \in Z_p^*$ be an element of $O(2q)$. Then.

$$x^{\frac{2q}{2}} \equiv -1; -x^{\frac{2q}{2}} \equiv 1$$

It shows that $-x$ is an element of $O(q)$.

5. Let $x \in Z_p^*$ be an element of $O(q)$. Then,

$$x^q \equiv 1; -x^q \equiv -1$$

It shows that $-x$ is an element of $O(2q)$.

6. Let $x \in Z_p^*$ be an element of $O(2^m)$. Then,

$$x^{2^m} \equiv 1; -x^{2^m} \equiv 1$$

It shows that $-x$ is an element of $O(2^m)$.

From (1),(2),(3),(4),(5) and (6), it may be seen that if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 1$; if $x \in L$ and $F_s(x) = 1$, then $-x \in K$ and $F_s(-x) = 1$.

7. Let $x \in K$ be an identity element, then $-x \in L$ and is an element of $O(2)$. Therefore, if $x \in K$ and $F_s(x) = 1$, then $-x \in L$ and $F_s(-x) = 1$.

It is proved from the above cases that w_1w_2 exhibits an *palindrome pattern.*

Corollary 1 *Let B contain only elements of $O(2^x q^l)$, where l various from 0 to $n-1$. Then S_p^B consists of elements of $O(2), O(q^l)$ and $O(2^m q^l)$, where m various from 1 to x and l various from 0 to $n-1$ and the distribution of $B-$ smooth numbers exhibit palindrome pattern*

This corollary can be proven from (1) to (7) of theorem 6.

Corollary 2 *Let B contain only elements of $O(2^m)$, where m various from 1 to x. Then, S_p^B consists of elements of $O(2^m)$ and the distribution of $B-$smooth numbers exhibits a palindrome pattern.*

The above corollary can be proven from (6) of theorem 6. Similarly, if B contains only elements $O(q^m)$, where m from 1 to n. Then, the distribution of B-smooth numbers does not exhibit any pattern.

The following table 4.1 summarize the patterns that are generated from different types of primes.

Table 4.1: Patterns generated from different types of primes

Type of the Problem	Order of factor base elements	Pattern
Type-1($2q$)	q	Inverse Palindrome
Type-2($2q_1q_2$)	q_1q_2	Inverse Palindrome
–	$2q_1$ or $2q_2$	Palindrome
Type-3($2q^n$)	q^n	Inverse Palindrome
–	$2q^m$,where m varies from 0 to $n-1$.	Palindrome
Type-4(2^nq)	2^mq, where m varies from 1 to $n-1$	Palindrome
–	2^m, where m various from 1 to n	Palindrome
Type-5(2^xq^n)	2^mq^l,where m varies from 1 to $x-1$ and l varies from 0 to n.	Palindrome
–	2^xq^l, where l various from 0 to $n-1$	Palindrome
–	2^m, where m various from 1 to x	Palindrome

4.2.2 New algorithms for DLP

In this section, we present two new algorithms based on the results in the previous section for solving DLP. The following equations show the relationship between the discrete logarithms of y and $-y$

$$x = q \pm 2v \bmod (p-1) \quad \text{when } y \text{ is } O(2q) \tag{4.5}$$

$$v = (x \pm q)/2 \bmod q \quad \text{when } y \text{ is } O(q) \tag{4.6}$$

where q is the order of the subgroup; and, v is the discrete logarithm of $-y$ with respect to q if x is that of y and vice-versa.

Equations 4.5 and 4.6 allow us to solve the DLP for exponents lying near the middle of the prime-order subgroup i.e., $q \pm d$.

Similarly, the exponents within the approximate distance $\frac{q-1}{2} \pm d$ can be solved in reduced time. To solve the logarithm of y i.e., $(q-1)/2 \pm d$, the logarithm of $-y$ i.e., $-1 \pm 2d \; mod \; p-1$ to be solved. Since,

$$\begin{aligned} x &= q \pm 2v \; mod \; p-1 \\ x &= q \pm 2((q-1)/2 \pm d) \; mod \; p-1 \\ x &= q+q-1 \pm 2d \; mod \; p-1 \\ x &= 2q-1 \pm 2d \; mod \; 2q \end{aligned}$$

The logarithm of $-y$, which could be solvable in minimum computational power is vulnerable. The following section presents the study on the exponents in a specific range to solve the logarithm in reduced cost.

4.3 Computing the discrete logarithm in a given range

In the current technology, it is considered as infeasible to compute the DLP in a group of order ≈ 1024 bits. Var Oorshot and Wiener analyzed the computation of the DLP for an exponent of size ≈ 160 combined with the random prime p of size ≈ 1024 bits. They suggested the use of prime order subgroup along with the short

exponents or the use of safe primes. Lim and Lee have shown a specific interest in investigating the computation of the DLP in a prime order subgroup of order ≈ 160 bits of p of size ≈ 1024 bits by extracting $x \ mod \ O(\beta)$. The β denotes the product of elements of smooth order and the prime p is assumed to be of random and the $p-1$ has many small factors apart from the large one with 160 bits.

In both the cases random primes are used for the computations and assumed to have many small factors apart from the large one for $p-1$. The difference is that, the former solved the problems with generators of order $|p-1|$ and the later solved the problems with generators of order $|q|$ for the DLP and computations are restricted to prime order subgroup q. From the above discussion, the following problems are formulated and investigated with the help of the algorithms discussed in the section 4.2.2 .

- Finding the exponent in a specific range in a group of order $|2q|$ and the order of y is $|2q|$ or $|q|$.

- Finding the exponent in a specific range in a prime order subgroup of a safe prime or random prime with $p-1$ as 2ρ.

- A random prime with $p-1$ as 2ρ and one of the factor of $p-1$ is a large prime order subgroup.

4.3.1 Exponents in a specific range in a group of order $2q$

The use of safe prime p of the form $p = 2q+1$ where q is also a prime is considered as safe. As discussed earlier Var Oorshot and Wiener [98] proposed an attack on short exponents with the combination of random prime. The algorithm is the combination of Pohlig-Hellman and Pollard-Lambda method to solve the DLP with the above constraints and it works as follows. Let $y = g^x$ be an element of a group G of order $n = zQ$, where $z = B_r$ is the product of smooth factors, and has bit length approximately k. Compute V where $V = x \ mod \ z$, by a partial

Pohlig-Hellman decomposition. Write $x = Az + V$, where $0 \leq V < z$ with A as yet unknown. Then $y = g^x = g^{Az+V}$. Now $A \in [0, 2^c]$, where $c = u - k$ bits of x remain unknown after finding V. Computing g^V and $y^* = y/g^V = g^{Az} = h^A$, where $h = g^z$ is known. Now V to be computed from the lambda method. Since A and V are known, x can be calculated as $x = Az + V$.

The safe primes precludes the Var Oorshot attack, since the partial Pohlig-Hellman decomposition yields only a single bit of information about an exponents x. The advantage of using safe primes is that, all group elements of Z_p^* other than ± 1 are known to have order either q or $2q$.

In this type of primes, the set of smooth numbers over Z_p^* generates an inverse palindrome pattern. If the order of y is known, then the order of $-y$ is to be decided by using this pattern. Since the problem is of safe prime, the order of y is either $2q$ or q.

(i) y is of order $|2q|$

In this section we presented a method to retrieve the logarithm that is known to lie in an interval $[q \pm 2 * d]$, where d is the distance. The method first converts the element, y of order $2q$ into $-y$ and computes the logarithm of $-y$ in the prime order subgroup using the Pollard-Lambda method. The logarithm of y is obtained as follows.

Let us consider $g^x = y$, where g and y are known, x to be solved and the order of g is $p - 1$. The above equation can be rewritten as follows using the properties discussed in the previous sections.

$$g_1^i = -y;$$

where g_1 is the generator of order q .
Since,

$$g^x = y$$
$$g^{q+2i} = y$$
$$g^q.(g^2)^i = y$$

$$(g^2)^i = -y$$
$$g_1^i = -y$$

Here i is assumed to lie in an interval [1, d]
Solve i by Pollard-Lambda method.
Thus,

$$x = q + 2 * i \ mod \ 2q;$$

. (ii) y is of order q.

When the order of y is q. The logarithm of y is calculated as follows. Let us assume $g^x = y$. Since the order of y is q, it can be generated from g_1. Thereby, the equation can be rewritten as follows:

$$g_1^{x'} = y;$$

where g_1 is the generator of q and x' assumed to lie within the range $[(\frac{q-1}{2} \pm d]$. Now, the following equation is obtained by using the properties discussed in the previous sections

$$g^{x_1} = -y;$$

where x_1 is $-1 \pm 2d$

$$g^{x_1} = -y$$
$$g^{-1 \ \pm \ 2d} = -y$$
$$g^{-1} g^{\pm 2d} = -y$$
$$(g^{\pm 2})^d = -y.g$$
$$g_1^d = -y.g$$

Solve d using Pollard-Lambda method and solve x_1 using the equation $-1 \pm 2d$. Then,

$$x' = (x_1 - p)/2 \ mod \ q$$
$$x = 2x'$$

Since,

$$g^x = y$$
$$(g^2)^{x'} = y$$
$$g_1^{x'} = y$$

The equation for $-y$ is

$$g^{x_1} = -y$$

4.3.2 Exponent in a specific range in a prime order subgroup of a safe prime of the form $2q+1$ or a random prime of $2\rho+1$

Use of large prime order subgroup is considered as safe, since all computations are restricted within the subgroup. The popular attack on prime order sub group is proposed by Lim and Lee [56]. They discussed the problem of size ≈ 1024 bits with prime order subgroup of ≈ 160 bits. The prime p solved in the above attack is a random prime p with $p-1$ has many small factors apart from the one with 160 bits. The attack is explained below. Let g be a generator of order q and $O(\beta)$ denotes the order of β. If $z = \gamma^x \ mod \ p$ can be retrieved by attacking the protocol, then $x \ mod \ O(\gamma)$, where $\gamma = \prod \beta_i$(a product of distinct smooth order elements $mod \ p$) can be obtained by using Pohlig-Hellman decomposition and finally the remaining part of x could be found from the public key y using Pollard-Lambda method.

In this section we considered the problem of solving the exponents in a prime order subgroup of a safe prime and random prime of the form $2\rho+1$ along with the assumption that the exponents lie in a specific range. In this type of primes, the set of smooth numbers over Z_p^* generates an inverse palindrome pattern for

safe primes with respect to the factor base of elements of order q. For random primes the set is belongs to inverse palindrome pattern with respect to the factor base of elements of order ρ and to palindrome pattern with respect to the factor base of elements of order $2X$, where X is a prime or product of primes. If the order of y is known, then the order of $-y$ is to be decided by using these patterns. The logarithm can be solved as follows:

(i) Prime order subgroup of safe prime

In the same way as discussed in the previous section the logarithm of y is obtained as follows. Let us assume $g^x = y$; where g is the generator of prime order subgroup(q) of safe prime, y is of order q and x is assumed to lie within the range $[(\frac{q-1}{2} \pm d]$. This equation can be rewritten as follows using the property, which relates y and $-y$.

$$g_1^{x_1} = -y \ mod \ 2q;$$
$$g^{x_1} = -y^2 \ mod \ q.$$

Since,

$$g^x = y$$
$$g_1^{x_1} = -y$$
$$g_1^{2x_1} = -y^2$$
$$(g_1^2)^{x_1} = -y^2$$
$$g^{x_1} = -y^2$$

where x_1 is $-1 \pm 2d$, g_1 is the generator of order $2q$ and g is the generator of order q.

$$g^{x_1} = -y^2;$$
$$g^{-1 \ \pm \ 2d} = -y^2$$
$$g^{-1} g^{\pm 2d} = -y^2$$
$$(g^{\pm 2})^d = -y^2.g$$
$$g_x^d = -y^2.g$$

The value for d can be obtain using Pollard-Lambda method and x_1 can be solved using the equation $-1 \pm 2d$.
x can be solved as follows:-

$$x = (x_1 - p)/2 \ mod \ q$$

(ii) Prime order subgroup of random prime

Let us assume q is a prime order subgroup of random prime and the computations are restricted within the prime order subgroup. When the exponent is assumed to lie within the range $[\frac{q-1}{2} \pm d]$, it can be retrieved as follows:

$$g^x = y \ mod \ q$$
$$g^{x_1} = (-y)^2 \ mod \ p$$

Since,

$$g^x = y \ mod \ q$$
$$g_2^{x_1} = -y$$

where g_2 is the generator of order $2q$

$$g_2^{2x_1} = (-y)^2$$
$$g^{x_1} = (-y)^2$$

where x_1 is $-1 \pm 2d$

$$g^{x_1} = -y^2;$$
$$g^{-1 \ \pm \ 2d} = -y^2$$
$$g^{-1} g^{\pm 2d} = -y^2$$
$$(g^{\pm 2})^d = -y^2.g$$
$$g_x^d = -y^2.g$$

Solve d using Pollard-Lambda method and x_1 is solved using the equation $-1 \pm 2d$. Once the logarithm of x_1 is known, the equation $x_1 = q + 2 * x$ is to be solved to obtain the value of x.

On the other hand, the x_1 can be solved if the value $x_1 \ mod \ \gamma$ is leaked through the protocol design. The γ is the product of primes less than some small bound B. In such case, the computation needed for the adversary is relatively less. For example a random prime of size ≈ 1024 bits with factors of $p-1$ as many smooth factors along with a prime of size ≈ 160 bits is chosen for testing. Since one of the factors of $p-1$ is a large prime order subgroup of size 160 bits, the $x_1 \ mod \ q$ can not be solvable by using Pollard-Rho method. The Pollard-Rho method needs $O(\sqrt{}(2^{160}))$ group operation, which is $O(2^{80}))$ group operations. If $x_1 \ mod \ \gamma$ is known, then the logarithm of remaining bits of x_1 is obtained as follows. Suppose $x_1 \ mod \ \gamma$ is ≈ 40 bits and $x_1 \ mod \ q$ is within the interval $[1, d]$ and ≈ 80 bits. Then the computation of remaining bits requires only $O(\sqrt{}(2^{40}))$ time and space using Pollard-Lambda method, which is a feasible computation power for the adversary.

4.3.3 Random prime with $p-1$ as factors 2ρ

As discussed in the previous section in this type of primes the smooth numbers over Z_p^* belongs to inverse palindrome pattern with respect to the factor base of elements of order ρ and to palindrome pattern with respect to the factor base of elements of order $2X$, where X is a prime or product of primes. If the order of y is known, then the order of $-y$ is to be decided using these patterns. The logarithm can be solved as follows:

The random primes with factors of $p-1$ as many smooth factors along with a large prime is considered for the exponents of specific range. The logarithm can be solved by using Pohlig-Hellman method. The method solves the logarithm $x \ mod \ p$ in the subgroups by obtaining $x_i \ mod \ p_i$ and combining the results using the Chinese Remainder Theorem. The $x_i \ mod \ p_i$ can be solved by using either Shanks or Pollard-Rho method. Since one of the factor of $p-1$ is a large prime order subgroup q, the $x_i \ mod \ q$ can not be solvable by using Pollard-Rho method. Another way to get the solution is, if $x_i \ mod \ \gamma$ is leaked through the protocol design as discussed above, then the logarithm of remaining bits of x_i is obtained by using Pollard-Lambda method.

Another solution to the above problem is, If the logarithm $x_i \ mod \ q$ is assumed

to lie within the range $[(\frac{q-1}{2} \pm 1d]$, then it can be computed in $(O(\sqrt{(}d))$ group operation as follows:

$$g^x = y$$
$$g^{\frac{p-1}{q}} = y^{\frac{p-1}{q}}$$
$$g_1^{x_i} = y_1$$

$$g_1^{x_i} = y_1 \ mod \ q$$

; where x_i assumed to lie in the range $[(\frac{q-1}{2} \pm 1d]$, g_1 is the generator of subgroup q and the reduction from g^x to $g_i^{x_i}$ is by using Pohlig-Hellman decomposition.

$$g_1^{x_i} = y_1 \ mod \ q$$
$$g_1^{x_1} = (-y_1)^2$$

Since $g_1^{x_i} = y_1 \ mod \ q$. This can be rewritten as follows using the property which relates y and $-y$.

$$g^{x_1} = -y_1$$

where g is the generator of order $|2q|$

$$g^{2x_1} = (-y_1)^2$$
$$(g^2)^{x_1} = (-y_1)^2$$
$$g_1^{x_1} = (-y_1)^2$$

where x_1 is $-1 \pm 2d$

$$g_1^{x_1} = -y_1^2;$$
$$g_1^{-1 \ \pm \ 2d} = -y_1^2$$
$$g_1^{-1} g^{\pm 2d} = -y_1^2$$
$$(g_1^{\pm 2})^d = -y_1^2.g_1$$
$$g_x^d = -y_1^2.g_1$$

Once the logarithm of x_1 is known, the equation $x_1 = q + 2 * x_i$ is to be solved to obtain the value of $x_i \ mod \ q$. The logarithm $x \ mod \ p$ can be solved using the Chinese Remainder Theorem on $x_i \ mod \ p_i$. The following section presents the results to support the claims made in the present section.

4.4 Experimental results

We implemented the methods for problems of different types discussed in the previous section and conducted experiments using an appropriate database. In this section we describe these experiments and give a representative selection of our experimental results. The purpose of our experiments is to produce the data on which we can base reliable statements about the expected running time of the proposed methods to solve the DLP. Let us describe clearly, First we generated a database of approximately 100 problems in each type described below along with the necessary information to carry out the experiments:

- Problems of prime p with $p-1$ as $2q$ (safe prime) with the generator of order $|2q|$ and the order of y as $|2q|$, safe primes with the generator of order $|2q|$ and the order of y as $|q|$
- Problems of safe primes with generator of order $|q|$, all computations are restricted within this prime order subgroup.
- Problems of primes with $p-1$ as 2ρ, where ρ consist of small primes along with a big prime q with the generator of order $|q|$, and the computations are restricted within this prime order subgroup.
- Problems of primes with $p-1$ as 2ρ, where ρ consist of small primes along with a big prime q with the generator of order $|2\rho|$.

A data file is produced with approximately 6-tuple (p, d, g, y, t, F) of each of the above primes with the following properties:
p is the prime to be tested of size between 100 to 1024 bits.
d is the approximate distance from the order of the subgroup q.
g is the generator, it may be a generator for a group or for a prime order subgroup.
y is g^x .
t is the type of the prime.
F is an array of factors of $p-1$.
The following algorithm is used to produce the database.

- Select a prime q with an appropriate size according to the type of the problem to be tested.
- Check the $2q + 1$ is a prime or not for a safe prime or $2\rho + 1$ is a prime or not for random primes, where ρ is a product of small primes less than some small bound B along with a big prime q.
- Repeat the above steps till a prime of required type is found.
- Find the generator of order $2q$ for the safe primes, the generator of order $|q|$ for the prime order subgroup computations of safe primes, the generator of order $|q|$ for the prime order subgroup computations of random prime with $p - 1$ as 2ρ and the generator of group order $|p - 1|$ for the random prime with $p - 1$ as 2ρ.
- The distance for the above problems to be selected within the range from 2^{10} to 2^{40} and the y is calculated using this distance as a reference.
- Finally the factors of $p - 1$ and the type of the problem is to be stored.

Having built up the file the following algorithm is executed.

- Read a tuple (p, d, g, y, t, F)
- Find the type of the problem.
- Find the order of y using the factors of $p - 1$.
- Use the methods discussed in the previous section according to the order of y and the type of the problem.
- Use Pollard-Lambda method to solve the logarithms in a range [1, d]
- Keep the track of the computed run time.
- Repeat the above steps until all tuples are calculated.

The following tables report the results on the problems of size up to 1024 bits and the average running time of the methods discussed in the previous Section.

Table 4.2 tabulates the average running time of solving DLP on different size problems of safe primes from 100 to 1024 bits. The variable d indicates the approximate distance. Table 4.3 shows the average running time of problems with computations are in prime order subgroup of $p-1$. We considered the prime order subgroup of safe primes and random primes.

Table 4.4 presents the running time of random prime with the generator of order $p-1$. The group order $p-1$ is assumed to have many small factors apart from the large one q and the γ is the product of these many small factors. The DLP $x \; mod \; p-1$ is to be obtained by computing $x_i \; mod \; \gamma$ and $x_i \; mod \; q$ and finally combine these results with Chinese Remainder Theorem. $x_i mod \gamma$ can be solved using Pohlig-Hellman method. The problems are tested under two assumptions, when the γ bits of $x_i \; mod \; q$ are not leaked and γ of $x_i \; mod \; q$ bits are leaked. In the former case the $x_i \; mod \; q$ can be obtain only from Pollard-Lambda method. In the later case the remaining bits are solved using Pollard-Lambda method, since $x_i \; mod \; \gamma$ bits are leaked. In all the above cases the logarithm in an interval [1, d] is solved using the Pollard-Lambda method.

The Pollard-Lambda method involves computing two sequences of points T and W. T computes the sequence $y_0^{'}, y_1^{'}, ...y_N^{'}$, where $y_{i+1} = y_i^{'}.g^{f(y_i^{'})}$ using a random function whose values takes the output from the set R. At y_N, the T halts and "set a trap" hoping to catch W should W land at this point during its own trail $y = y_0, y_1....y_N$, this will occur if W's trial hits any $y_i^{'}$. The T trails begins at $y_0^{'} = g^{b+w}$, and proceed to $y_N{'}$ and the distance traveled is $d_N^{'} = \sum_{i=0}^{N-1} f(y^{'}) \; mod \; n$. The W trials begins at $y_0 = y$, when $y_M = y_N{'}$ for some y_M in W's trial, at which point the logarithm of y is computed as $x = b + w + d_N^{'} - d_M \; mod \; n$. If no collision occurs before d_M exceeds $w + d_N^{'}$, then the failure probability is controlled by a parameter θ. The sequences T and W are deterministic paths and stepped by the random values obtained from the set R of mean m. For $\theta = N/m$ and m large, the probability of success is $PS = (1 - (1 - m^{-1})^{\theta m} \approx 1 - e^{-\theta}$. For optimal performance, set $m = \alpha.w^{\frac{1}{2}}$ for α as optimized below. For example, for $\theta = 4, PS = .98$, and the total expected work, $N + M$, minimized by $\alpha = \frac{1}{4}$, is $O(w^{\frac{1}{2}})$ steps. Pollard suggests computing and storing g^s for all $s \in R$ used, and therefore choosing $|R| \leq w^{\frac{1}{2}}$. $R = \{2^0, 2^1, 2^2, ...2^{L-1} \}$ is one suggestion, with

$f(y_i) = 2^j$, where $j = y_i \ mod \ L$, which is followed in the present work.

4.4.1 Selected list of solved problems

This section presents the selected list of solved problems using the methods discussed in the previous section. In the following examples, p represents the prime, g represents the generator of the group $p-1$, $g1$ denotes the generator of the prime order subgroup and y is the element for which the DLP is solved. The experimental results are conducted on a Pentium 1V machine with 256MB RAM capacity

Safe primes:-

y of order $2q$:-

$p = 20000000000000000000000000000000000000000000000000000000000000000$
$0000000000000000000000000000000000000000000000000000000000000000000$
$0000000000000000000000000000000000000000000000000051447$

$g = 777$

$y = 769367604420241674542275365954981646937100692721704838463174 9$
$2212653021234719597652837781690853930120962452644045823166679885 4$
$1954672097721872032646888627078423175416914084650394 7$

y of order q:-

$g = 777$

$g1 = 603729$

$y = 5648738405318464140368797118091985246714031633537023632088462 7$
$0247933286188389331652249213178744595795241807450245339115750049 3$
$5796723304813242625495950599752383615070543571186482 3$

Prime order subgroup of safe primes:-

$p = 20000000000000000000000000000000000000000000000000000000000000000$
$0000000000000000000000000000000000000000000000000000000000000000000$
$0000000000000000000000000000000000000000000000000000000000000000000$

0000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000501759

$g1 = 60493815061729$

$y = 2078460713195865422772440389954893159002412188138022102300277209$
6522895486632852356664687259364940745982566779505138312112376005108 1
7718542158329100644942837818163244169562060457796803031347503416980 3
6049702721018738429578544623530313311410627386475009324308858356350 8
1993919035843621557263441375831

Prime order subgroup of Random primes :-

$p = 3530596818954837594058363570277060586834520000000000000000000000$
0000000000000000000000000000000002298948118662442499371103438785908
001117297699

$g = 2$

$g1 = 332317183564122258213280212875609733819427440070345162979120714$
4404019442347666146004851900430305932537615128033273456315817255295
3907790655112

$y = 245505689811074246209897912132321429111654923220163565643653125 24$
89977023983947006356459131339580436008715638751116866708872574
9006168938241077

Random primes :-

$p = 4423203904733101730044905437206937176754800000000000000000000000$
0000000000000000000000000000000002880169222566959191518740175437297
14264388803

$g = 2$

$g1 = 300633768193089947353734726305109766168 03$
415127777894317398309099313057213722079753712 7
353192198583530692480230514413880625807007631 9
949023406

$y = 2558846458805501631041178176496220079153865378269711255449457263 6$
980556028930096742787908956094526373675778913947520327051399363687

98463146663

Prime order subgroup of Random primes:-

$p = 28721874389328679940809578569558567881693352764839668121224746091115198472247901457171$

$g1 = 84356340542889230018958878775437020141734107647551864529981512717196296200418796030$

$y = 19116509438627235277907417664901897191473327409276161171266242838693887181351558216 6$

4.5 Conclusion

In the present study we investigated the following problems.

- Finding the exponent in a specific range in a group of order $2q$ and the order of y is $2q$ or q.
- Finding the exponent in a specific range in a prime order subgroup of a safe prime or random prime (prime with $p-1$ as 2ρ).
- A random prime with $p-1$ as 2ρ and one of the factor of $p-1$ is a large prime order subgroup.

Safe primes and prime order subgroups are consider as safe in the literature, since the set elements of safe prime are known to have order $p-1$ or $\frac{p-1}{2}$ other than ± 1. On the other hand in the prime order subgroup, the computations are restricted within the subgroup, this property does not allow to solve the partial logarithm through the small factors of $p-1$. In the present study we analyzed and solved the problems defined in the above groups. The DLP that is known to lie in an interval is investigated through the distribution of the smooth numbers over Z_p^*. Many cryptographic systems are based on the hardness of solving the DLP. The private keys and ephemeral keys of many systems are the exponents x. The analysis will help the systems to generate safe keys. This leads to avoid the keys that

are vulnerable to the methods reported in the present study. Even though the probability of getting the exponents within the specific range is less, caution to be taken, while generating the keys. The properties and the investigation reported in the present study are on specific instances of prime fields with $p-1$ as 2ρ. The work can be extended for the primes of the form $2^n\rho + 1$.

Table 4.2: Running time of methods to solve DLP on safe primes

Problem size in bits	Safe prime with y as order $2q$				Safe prime with y as order q			
	2^{10}	2^{20}	2^{30}	2^{40}	2^{10}	2^{20}	2^{30}	2^{40}
100	–	–	–	–	–	–	–	–
256	–	2	9m	4 h	–	1s	10m	4h 2m
512	–	3s	15m	5 h	–	2s	15m	4h 30m
1024	–	3s	1h 12m	13h 15m	–	3s	1h 30m	15 h

Table 4.3: Running time of methods to solve DLP on prime order subgroups

Problem size in bits	Prime order subgroup of safe prime				Prime order subgroup of random prime			
	2^{10}	2^{20}	2^{30}	2^{40}	2^{10}	2^{20}	2^{30}	2^{40}
100	–	–	–	–	–	–	–	–
256	–	2s	11m	5 h	–	–	1s	9m
512	–	3s	16m	4h 30m	–	–	5s	13m
1024	–	2s	50s	15h	–	–	4s	4h

Table 4.4: Running time of methods to solve DLP on random primes

Problem size in bits	Random prime with generator of order p-1				Random prime with generator of order p-1 and γ bits leaked			
	2^{10}	2^{20}	2^{30}	2^{40}	2^{10}	2^{20}	2^{30}	2^{40}
100	–	–	–	–	–	–	–	
256	–	1s	14m	3h	–	–	14s	1h 10m
512	–	–	13m	4h	–	–	17s	2h 15m
1024	–	2s	57s	12h 30m	–	1s	35s	6h

Chapter 5

Ephemeral Key Recovery using Index Calculus Method

A special case of ICM, analogous to Pohlig-Hellman, when the factors of $p-1$ are small is studied in this chapter. In the literature, the Pohlig-Hellman is the best known method to solve the DLP, when the factors of $p-1$ are small and known, while ICM is an efficient method for general DLP. Two algorithms are proposed to improve the efficiency of the pre-computation step of the ICM by using the equivalence classes formed from a special case of B-smooth numbers over Z_p^*. The two algorithms proposed here are useful in recovering ephemeral keys often used for session-based security. Similarly an another approach followed to built the ICM by using the property of generators to recover the ephemeral keys is also presented.

5.1 Introduction

The distribution of a special case of smooth numbers over Z_p^* on different types of primes is studied in the present study. A set of quadruples are generated from Z_p^* using the characteristics of smooth numbers over Z_p^*. Furthermore, a special case of ICM, analogous to Pohlig-Hellman, when the factors of $p-1$ are small is studied in this chapter. Two approaches are followed to develop the ICM. One is by using the characteristics of smooth number over Z_p^* and another is by using

the property of generators of Z_p^*.

The ephemeral keys based on the DLP in the public key cryptosystems are to ensure the security of the system. The life time of these keys is short. In the present study, the problem of solving the DLP for retrieving the ephemeral keys is studied. These 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 key, "by the same party across the different sessions" [61]. One way of retrieving this key is to solve the mathematical hard problem such as DLP. These keys are dynamic and changes for every session between Alice and Bob while the static keys remains the same and lives longer. Since the life time of ephemeral keys is short, it is hard to recover these keys within the short span of time by using the attacks with the target of solving the DLP.

In ephemeral key security, the underlying field and generator of the cryptosystem are held static but each session uses different keys. In some systems, the ephemeral keys are changed periodically(e.g once every 5 minutes). For such systems, the use of ICM allows the pre-computation to be performed once with one search each for the individual logarithm of the ephemeral key. For ephemeral key systems, the security requirements are consider less secure as the keys change frequently [20]. For Pohlig-Hellman, to solve the DLP for every ephemeral key is a different problem. All steps have to be re-computed. On the contrary, in ICM, only the individual logarithm step has to be recomputed. Therefore, the efficient way of developing the individual logarithms step is investigated in the present study. Also, the pre-computation step is studied to built an efficient individual logarithm step.

Traditionally, the ICM is viewed as a general purpose algorithm with respect to solve the DLP. The conventional way of computing the logarithms of factor base is to obtain the logarithms of first-t primes by using the pre-computation step. The DLP of y is solved by using the individual logarithm step, which is compatible with pre-computation step.

The present variant of the ICM is viewed as a special purpose method and investigated, when the factors of $p-1$ are small. An another way of choosing and computing the logarithms of factor base is proposed. The pre-computation step is built to compute the logarithms of newly chosen factor base. The first-t primes along with the equivalence classes of the special case of B-smooth numbers over Z_p^* and the subgroup elements along with the equivalence classes are considered as the factor base elements in the first and second methods of first approach respectively. In the second approach only the subgroup elements are considered as the factor base. Similarly, in the first approach, the logarithms of factor base are computed with respect to the given generator g and the generators of subgroups, while in the second approach the logarithms are computed with respect to the given generator g. The individual logarithm step is one among the following, such as the traditional individual logarithm step, Pohlig-Hellman and Chinese Remainder Theorem in the first approach and using the property of generators of Z_p^* in the second approach. The newly proposed individual logarithm (computation) step outperforms the Pohlig-Hellman on some special cases. This leads to recover the ephemeral keys used in the DLP based schemes in reduced time.

5.2 Ephemeral key recovery using the properties of a special case of smooth numbers over Z_p^*

5.2.1 Smoothness

An integer is called 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 6 (R_1 and R_2 relations) *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^* elements such that for any $a \in Z_p^*$, $b \in Z_p^*$ $ab \equiv -1 \ mod \ p$.*

Definition 7 (Quadruple) *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 $a_1 R_1 a_2$ $a_3 R_1 a_4$ $a_1 R_2 a_3$ $a_2 R_1 a_4$*

Definition 8 (B smooth on over Z_p^*) *Let Z_p^* is a multiplication group of prime field and let B be a factor base ($\subset Z_p^*$). Then a quadruple s is B smooth over Z_p^* iff it satisfies the following relations.*
Let $a \in B$ and $b, e, f \in Z_p^*$ then $ab \equiv 1$, $ef \equiv 1$, $ae \equiv -1$, $bf \equiv -1$.

The smooth element in the above definition is a quadruple and the set of all B-smooth numbers over Z_p^* is a collection of quadruples. Let S denotes the set of all B-smooth numbers over Z_p^* and defined as follows:-
$S = \{s | s$ is B smooth over Z_p^* $\}$.

Theorem 7 *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

Hence $xR_1y; xR_2 - y$, $yR_1x; yR_2 - x$ and $-xR_1 - y$. Since the group elements have unique inverse, the quadruples are disjoint sets. This shows a set of equivalence classes can be derived from Z_p^* using the relations R_1 and R_2. We denote the set of equivalence classes as $Z_p^*/\sim$. The identity element forms the class with -1 and the other $\lceil (p-1-2)/4 \rceil$ classes are of size almost 4, where $p-1$ is the order of the group.

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

Proposition 2 *Let $x \in Z_p^*$ and $y \in Z_p^*$. If $x \neq y$ and xR_1y then $xR_2 - y, yR_2 - x, -xR_1 - y$.*

Proof. Let $xy \equiv 1$
(i) Let $A \in Z_p^*$

$$x.A \equiv -1$$
$$A \equiv \frac{-1}{x} \equiv -1.x^{'} \equiv -y$$
$$A \equiv -y$$

Hence $xR_2 - y$

(ii)$yA \equiv -1$

$$A \equiv -1.y^{'} \equiv -x$$

Hence $yR_2 - x$

(iii)$-xA \equiv 1$

$$A \equiv 1. - x^{'}$$
$$1. - 1.x^{'}$$
$$1.y \equiv -y$$

Hence $-xR_1 - y$

Proposition 3 *Let $x \in Z_p^*$ and $y \in Z_p^*$. If $x = y$ and $xy \equiv 1$ mod p, then $xR_1xxR_2 - x$.*

Proof.

$$xy \equiv 1$$
$$x \equiv \frac{1}{y} \equiv x^{'} \tag{5.1}$$

Hence $x \equiv x^{'}$

$$xz \equiv -1$$
$$z \equiv -1.x^{'} \equiv -x \tag{5.2}$$

Hence $z \equiv -x$

From 5.1 and 5.2 xR_1x and $xR_2 - x$

Next we require a function from the equivalence classes 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 equivalence class and a map, say ϕ on $Z_p^*/\sim$. Let us denote $[g]$ as a class of the generator and the experiments show that the map

$$\phi : [g] \rightarrow Z_p^*/\sim \tag{5.3}$$

can be defined by

$$g^i \rightarrow R \tag{5.4}$$

Where R is a representative of a equivalence class and i is in the range $1 \leq i \leq \lceil (p-1-2)/4 \rceil$.

The above mapping function ϕ maps the classes other than ± 1 and this rotates in a cyclic fashion from an arbitrarily generator. The following section describes the characteristics of smooth numbers over Z_p^*.

5.2.2 Distribution of smooth numbers over Z_p^* on different types of primes

The distribution of smooth numbers over Z_p^* is studied on different types of primes. The primes are classified based on the factors of $p-1$ as described in chapter 4.

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 8 *Let p be a prime of the form $2q+1$. Let S be the smooth number set over Z_p^*. Let Q be an equivalence class $\in S$. If one pair of elements in Q consists of generators then the other pair contains non generators of $O(q)$. 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

$$x^q \equiv 1 \bmod p$$

$$-x^q \equiv -1$$

This shows $-x$ is generator.

Let y be a generator

$$y^q \equiv 1 \bmod p$$

$$-y^q \equiv -1$$

$-y$ is a non generator.

2. Let x be a 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 the equivalence class $\in S$. 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 $O(q_1q_2)$.*
2. *If one pair of elements in Q consists of $O(q_1)$ or $O(q_2)$ then the other pair contains non generators of order $O(2q_1) or O(2q_2)$.*
3. *If one pair is 1 another pair is -1.*

This can be easily proven from 1 and 2 of theorem 8.

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 the equivalence class $\in S$.

1. *If one pair of elements in Q consists of $O(q^i)$, where i varies from 1 to $n-1$ then the other pair contains $O(2q^i)$.*

2. *If one pair of elements in Q consists of generator then the other pair contains* $O(q^n)$.

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

This also can be proven from 1 and 2 of theorem 8.

Theorem 9 *The prime p of the form* $2^n q + 1$.
Let S be the smooth number set over Z_p^*. *Let Q be the equivalence class* $\in S$.

1. *If one pair of elements in Q consists of* $O(q)$ *then the other pair contains non generators of* $O(2q)$.

2. *If one pair of elements in Q consists of* $O(2^i q)$ *where i varies from 2 to n then the other pair is also of* $O(2^i q)$.

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

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

Proof. 1. This follows the proof 1 of theorem 8
2. Let x be the element of order $2^i q$ where i varies from 2 to n

$$x^{\frac{2^i q}{2}} \equiv x^{2^{i-1} q} \equiv -1 mod p$$
$$-x^{\frac{2^i q}{2}} \equiv -1$$

This shows $-x$ is an element of $O(2^i q)$.
Let y be the element of order $2^i q$ where i varies from 2 to n

$$y^{\frac{2^i q}{2}} \equiv x^{2^{i-1} q} \equiv -1 mod p$$
$$-y^{\frac{2^i q}{2}} \equiv -1$$

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

$$x^{\frac{2^i}{2}} \equiv x^{2^{i-1}} \equiv -1 mod p$$
$$-x^{\frac{2^i}{2}} \equiv -1$$

This shows $-x$ is an element of $O(2^i)$.

Let y be the element of order 2^i where i varies from 1 to n

$$y^{\frac{2^i}{2}} \equiv x^{2^{i-1}} \equiv -1 mod p$$
$$-y^{\frac{2^i}{2}} \equiv -1$$

This shows $-y$ is an element of $O(2^i)$.

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

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

1. *If one pair of elements in Q consists of $O(2^i)$, where i varies from 1 to n then the other pair contains $O(2^i)$.*
2. *If one pair of elements in Q consists of $O(q^i)$, where i varies from 1 to m then the other pair contains $O(2q^i)$.*
3. *If one pair of elements in Q consists of $O(2^i q^k)$, where i varies from 2 to n and k varies from 1 to m. then the other pair contains $O(2^i q^k)$.*
4. *If one pair is 1, another pair is -1.*

1. This is easily proven from 3 of theorem 9.
2. This follows from 1 of theorem 8.
3. This follows from 2 of theorem 9.

5.2.3 Results and discussion

From the above theorems, it is observed that, in the first type of primes, the equivalence classes are formed by the pair of generators and non generators. In the second types of primes, when one pair of the class 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 class is a generator, then the another pair is non generator of order $2q_1q_2$. In the third type of problems the classes 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^iq forms quadruple with non generators of order 2^iq, 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^iq^k), where i varies from 2 to n and k varies from 1 to m then the another pair is of order 2^iq^k.

The main conclusion is the equivalence classes exhibit different characteristics on different types of primes. Through our experimental results we found that the equivalence classes 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, classes with $4m$ elements, logarithms of m unknowns to be solved. This principle is used in the following methods to solve the DLP by using the ICM.

In the present study, two ways of performing the pre-computation step and three ways of computing individual logarithm step is studied. In the finite field Z_p^*, the pre-computation is to compute the logarithms of first-t primes i.e., primes

less than some bound. The first type of pre-computation step in the present study computes the logarithms of first-t primes and their quadruples. The first-t primes are treated as m unknowns and the logarithms of first-t primes (m unknowns) are computed using Pohlig-Hellman method and the quadruples are formed by using the definition of smooth numbers over Z_p^*. Since the logarithms of m unknowns are known, the logarithms of $4m$ elements in the quadruples are computed. The individual logarithm step is performed either by using Pohlig-Hellman or the traditional individual logarithm step of ICM. In the second type, the logarithms of elements of each subgroups are stored in a list using the distribution of smooth numbers over Z_p^* and the individual logarithm is only the Chinese Remainder Theorem step. The following table 5.1 presents the details of methods studied in the present work.

Table 5.1: Methods for Index Calculus Method

Methods	Pre-computation step	Individual logarithm phase		
		General Individual logarithm phase	Pohlig-Hellman method	CRT
Method1	Pohlig-Hellman	$\surd$	$\surd$	$\times$
Method2	Using a special case of smooth numbers over Z_p^*	$\times$	$\times$	$\surd$

5.2.4 Algorithms for ICM

In this section the above discussed methods for ICM are addressed. The algorithm 2 presents the method-1 and algorithm 3 addresses the method-2. Algorithm 2 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 3 takes the advantage of the distribution of smooth numbers 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 $\frac{O(p_i)}{4}$, when the subgroup is of order 2^n and $\frac{O(p_i)}{2}$ for other cases. 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.

The individual logarithm step of ICM in method-1 depends 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.

The table 5.2 shows the difference in running time between method-1 and method-2. The table 5.3 presents the running time of individual logarithm step of method-1, method-2, traditional individual logarithm step of ICM and the conven-

Algorithm 2 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. {Known information is g : generator of Z_p^* and Factors of $p-1$ which are small }

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 O(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 FB

Algorithm 3 To find logarithm of all small subgroup elements in the order of subgroup using the distinct set of quadruples formed with relation 1 and -1

Input: Problem of size p and factors of $p-1$.

Output: logarithm of all small subgroup elements in the order of subgroup. {Known information is g : generator of Z_p^*, y : element of Z_p^* and factors of $p-1$.}

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 $A = O(p_i)/4$
5: **else**
6: Assign $A = O(p_i)/2$
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**

Table 5.2: Running time of method-1 and method-2

Problem	Method-1 Running time in sec	Method-2 Running time in sec
43241221044344476653 62908248086967822904859	1597	55
40500691568928903388 503314943591776516203	1393	88
10548813247704246266 317485054480132114947	808	31
22750475822981512251 147389834477659827887	861	29
31023376122247516706 110077047014990674391	987	26
16483888118310633603 1117884004666831843	315	6
29565696133579269116 146450939411987039	306	3
17566082229403199021 60131316475990998066411	2516	203
11604511787937964282 8484543996278641093	1715	181
96834800300461810039 999000830418556963	2327	651
40687382369048479475 232114904989637283	1891	633
24774781676535030702 2689158418187059	1194	171
38174514779333277109 099448511590627	1141	161

Table 5.3: Individual logarithm step

Problem size in digits	Method-1	Method-2	General individual logarithm step of ICM	Pohlig-Hellman method
19	88ms	6ms	104ms	22ms
20	255ms	10ms	300ms	31ms
21	315ms	173ms	407ms	103ms
22	790ms	18ms	605ms	42ms
23	5s	8ms	3s	54ms
24	4s	4ms	4s	13ms
25	6s	16ms	7s	54ms
26	36s	.4ms	37s	14ms
27	55s	400ms	61s	51ms
28	210s	.5ms	194s	16ms

tional Pohlig-Hellman. The general individual logarithm step of ICM in the above table is computed with known precomputed factor base of first-t primes. They are computed using the conventional pre-computation step of ICM. Hence the structure of $p-1$ is not considered in this case. Next, the traditional Pohlig-Hellman is implemented by considering the structure of $p-1$. Since the factors are small, Pohlig-Hellman is a well suitable method for this class of problems. The individual logarithm step of method-1 mentioned in the table 5.3 is the traditional individual logarithm step of ICM with the additional advantage of precomputed logarithms of equivalence classes of first-t primes. The Chinese Remainder Theorem(CRT) is the individual logarithm step of method-2.

From the table 5.3, it is observed that the running time reported for method-1 and general individual logarithm step of ICM are similar. The logarithms of quadruples apart from the first-t primes is the advantage of method-1 compared with the general individual logarithm step of ICM. In the similar line the run-

ning time of method-2 and the Pohlig-Hellman are similar. The running time of method-2 outperforms Pohlig-Hellman method when the factors of $p-1$ are small.

5.3 Ephemeral key recovery using the properties of generators

In this thesis, we propose an another approach of retrieving the ephemeral keys by using the property of generators of Z_p^*. In the similar line as discussed above, the ICM is viewed as a special purpose method and investigated, when the factors of $p-1$ are small. Further, a new way of choosing and computing the logarithms of factor base is proposed. An efficient way of solving the individual logarithm step is presented. A pre-computation step compatible with the individual logarithm step is reported. The individual logarithm (computation) step outperforms the Pohlig-Hellman method based on the order of y. This leads to recover the ephemeral keys used in the DLP based schemes in reduced time.

5.3.1 Our approach

The property of generators of finite field Z_p^* motivated the new approach of solving the DLP for ephemeral keys. The individual logarithm step of ICM is designed using this property. The following section discusses the property of generators.

5.3.2 Motivation to develop the new methodology

The main intuition of the present study is the property of generators of finite field Z_p^*. The logarithm of a generator g_a, with respect to the given generator g can be obtain by using the logarithms of subgroup elements. If $p-1 = p_1^{e_1} p_2^{e_2}, \cdots, p_k^{e_k}$, then the subgroup elements, such as a_1 of order $p_1^{e_1}$, a_2 of order $p_2^{e_2}$, a_3 of order $p_3^{e_3}, \cdots, a_k$ of order $p_k^{e_k}$ can be generated from the generator as follows:-

$$g_a^{\frac{p-1}{p_1^{e_1}}} = a_1 \tag{5.5}$$

$$g_a^{\frac{p-1}{p_2^{e_2}}} = a_2 \tag{5.6}$$

$$\cdots$$

$$g_a^{\frac{p-1}{p_k^{e_k}}} = a_k$$

Also, the logarithm of a generator g_a with respect to g can be obtained from the logarithms of elements a_1, a_2, $a_3, \cdots, a_k$ as follows:-

$$\begin{aligned} g_a^{\frac{p-1}{p_1^{e_1}}} &= a_1 \\ \log_g(g_a^{\frac{p-1}{p_1^{e_1}}}) &= \log_g(a_1) \\ \frac{p-1}{p_1^{e_1}} \log_g g_a &= \log_g(a_1) \\ \log_g g_a &= \log_g(a_1)(\frac{p-1}{p_1^{e_1}})^{-1} \end{aligned} \tag{5.7}$$

$$\begin{aligned} g_a^{\frac{p-1}{p_2^{e_2}}} &= a_2 \\ \log_g(g_a^{\frac{p-1}{p_2^{e_2}}}) &= \log_g(a_2) \\ \frac{p-1}{p_2^{e_2}} \log_g g_a &= \log_g(a_2) \\ \log_g g_a &= \log_g(a_2)(\frac{p-1}{p_2^{e_2}})^{-1} \end{aligned} \tag{5.8}$$

$$\cdots$$

$$\begin{aligned} g_a^{\frac{p-1}{p_k^{e_k}}} &= a_k \\ \log_g(g_a^{\frac{p-1}{p_k^{e_k}}}) &= \log_g(a_k) \\ \frac{p-1}{p_k^{e_k}} \log_g g_a &= \log_g(a_k) \\ \log_g g_a &= \log_g(a_k)(\frac{p-1}{p_k^{e_k}})^{-1} \end{aligned} \tag{5.9}$$

The equations 5.7, 5.8, and 5.9 obtain the logarithm of g_a mod $p_1^{e_1}$, g_a mod $p_2^{e_2}$, and g_a mod $p_k^{e_k}$ respectively. The logarithm g_a mod $p-1$ can be obtained by using the Chinese Remainder Method.

The symbols g_a represents the generator and g refers to the given generator, the logarithm of y to be computed with respect to g. The above equations show that the logarithm of a generator g_a can be obtained when the logarithms of

elements $a_1, a_2, a_3, \cdots, a_k$ with respect to g are known. This leads to formulate a new algorithm for ICM. From the pre-computation point of view, the logarithms of subgroup elements are computed with respect to g. From the individual logarithm step, if y is the generator then the logarithm of y can be obtained by using the equations 5.7, 5.8, $\cdots$ 5.9 and Chinese Remainder Theorem. On the other hand if y is not a generator then the following procedure is to be followed.
for any k,

$$yg^k = \prod_{i=1}^{t} p_i^{d_i}, \tag{5.10}$$

where p_i are generators.
Then, $\log_g y = (\sum_{i=1}^{t} d_i \log_g p_i - k) \mod n$

This property of generators leads to device a new methodology for ICM and it is useful only when the factors of $p-1$ are small.

5.3.3 Algorithms for ICM

In this section we presents the algorithm developed for the pre-computation step and individual logarithm step of ICM. The procedure followed in the algorithm is to compute the logarithms of subgroup elements with respect to the generator g and obtain the logarithm of y by using the individual logarithm step. Algorithm-4 represents pre-computation step and Algorithm-5 is for individual logarithm step. In Algorithm-4, the subgroup elements are considered as factor base and the logarithms of this factor base are computed. The logarithm of y is computed in Algorithm-5 from the pre-computed logarithms of subgroup elements.

In pre-computation algorithm (Algorithm-4), the $g^{\frac{p-1}{p_i}}$ computes the generator of the subgroup p_i. The logarithm of subgroup elements of order p_i with respect to the generator g are obtained. In individual logarithm step algorithm (Algorithm-5), if the element y in the equation $g^x = y$ is a generator then $\log y$ can be obtained in reduced time. The Chinese Remainder Theorem is used on the pre-computed logarithms of subgroup elements, which are derived from y as in the equation 5.6 and $\log y$ is obtained. The steps 18 to 27 are to solve $\log y$. Since y is a generator, g_j

Algorithm 4 Algorithm for the pre-computation step of Index Calculus Method

Input: : p (Problem to be solved) and the factors of $p-1$

Output: : RESULT (The discrete logarithm of subgroup elements.)

1: **for** each subgroup of order $O(p_i)$ **do**
2: Assign e as $g^{\frac{p-1}{p_i}}$
3: **for** j in $1..O(p_i)$ **do**
4: Assign A as e^j
5: Find the logarithm of A as $\log A = j * \frac{p-1}{p_i}$
6: Store the elements and logarithms in a list
7: **end for**
8: **end for**

is y and steps from 19 to 25 are to find the logarithms of subgroup elements derived from y, step 26 is the Chinese Remainder Theorem to combine the results and step 31 is for logarithm of y. In this case the individual logarithm step outperforms the Pohlig-Hellman method. The algorithm is still efficient, when yg^k is a generator in first few trails, even in this case the $\log y$ is computed as above. If y is not a generator or yg^k is not turned as a generator, then the factors of yg^k is checked for generators. In such case, the equation $yg^k = g_1 g_2 g_3 \cdots$ retrieves the logarithm of y as follows $\log_g y = \log_g g_1 + \log_g g_2 + \log_g g_3 + \cdots k$. The steps 18 to 31 are used to find the $\log y$. For each g_j in the above equation, the steps from 19 to 26 are used to compute the logarithm of g_j by using the subgroup elements which are derived from the equation $g_j^{\frac{p-1}{p_i}}$. From the logarithms of g_j the logarithm of y is obtained.

Performance study and numerical results

We implemented the above algorithms to ensure the efficiency of the algorithms and conducted experiments on a data base of random primes with small factors.

Algorithm-3

- Select k between 15 to 50.

Algorithm 5 Algorithm for individual logarithm step

Input: : Problem of size p, pre-computed logarithms of subgroup elements, g,y and m is the number of factors in $p-1$.

Output: : logarithm of y.

1: **if** y is the generator **then**
2: $g_j = y; j = 1; n = 1$
3: Go to step 18
4: **end if**
5: Repeat the following steps
6: Randomly choose the value for k
7: Compute yg^k
8: **if** yg^k is a generator **then**
9: $g_j = yg^k; j = 1; n = 1$
10: Go to step 18
11: **end if**
12: Factorize yg^k and obtain the factors
13: Store the number of factors as n and the factors as g_j, where j varies from 1 to n.
14: **if** the factors (g_j) are generators **then**
15: break the loop
16: **end if**
17: Go to step 6
18: **for** j in 1..n **do**
19: **for** i in 1..m **do**
20: Assign A as $g_j^{\frac{p-1}{p_i}}$
21: Find the logarithm of A in the list stored before and obtain $\log A$
22: Find the logarithm of g_j as below
23: $\frac{p-1}{p_i} \log g_j \equiv \log A mod p_i$
24: store the logarithm of $g_j \bmod p_i$
25: **end for**
26: Solve the logarithm of $g_j \bmod p-1$ using logarithms of $g_j \bmod p_i$ of each subgroup using Chinese Remainder Theorem
27: **end for**
28: **for** j in 1 .. n **do**
29: $\log y = \log y + \log g_j$
30: **end for**
31: $\log y = \log y - k$

- Select small primes.
- Compute the product of primes(X).
- Check $2X+1$ is a prime using probabilistic primality test algorithm and the size of the prime is k digits.
- Find the generator g for the above prime.
- Create a 3-tuple as (p, ρ, g), where p is a prime, ρ is an array of factors and g is a generator.

Having built up the database the following algorithm is executed to test the pre-computation and individual logarithm algorithms discussed in the previous section.

Algorithm-4

- Read the 3-tuple (p, ρ, g).
- Compute the logarithm of elements of subgroups of order of factors in the array ρ using the pre-computation algorithm proposed in the previous section.
- The logarithm of y is obtained by using the individual logarithm step discussed in the previous section.
- Keep the running time for each tuple.
- Repeat the above procedure for all the tuples.

The above algorithm is executed on a data base of primes and the reports are tabulated in table 5.4. Table 5.4 list the average running time for problems of size $\approx$ 44 digits. The table 5.5 reports the difference in running time between the Pohlig-Hellman and the individual logarithm step of newly proposed Index calculus algorithm. This will leads to break the ephemeral keys. If the pre-computed logarithms of subgroups elements are known then the ephemeral keys can be retrieved through the individual logarithm step. The table 5.5 shows

that when y is a generator, the individual logarithm step outperforms the Pohlig-Hellman. On the other hand, if y is not a generator or yg^k is not turned as generator in one or fewer trials then the Pohlig-Hellman outperforms the individual logarithm step.

Let us assume that the ephemeral key to be solved is $y = g^k$. To recover k , the DLP of y to be solved. Apart from the public keys g and y the additional information regarding the factors of $p-1$ on the prime field p are known and small. With the sufficient introduction of ephemeral keys, it is well known that the DLP for ephemeral keys can be solved in reduced time when the logarithms of a subset of a group Z_p^* is known. Then the individual logarithm step can be used to recover the key. The table 5.5 reports the running time of the individual logarithm step and the Pohlig-Hellman method. For example the value of k is recovered in 12ms for problem of size $\approx$ 27 digits by using the individual logarithm step and 14ms by using the popular Pohlig-Hellman. The Pohlig-Hellman is executed without considering the pre-computation step. This shows the potential impact of the analysis made in the present study.

5.4 Conclusion

In the present work a special case of smooth numbers over Z_p^* is defined. The properties are analyzed on different types of primes and found to exhibit different characteristics that correspond to the order of the group. Also, in this chapter a new methodology to recover the ephemeral keys by using two approaches are discussed. The approaches are depend on smooth number over Z_p^* and the property of generators.

A new algorithm is proposed for ICM to recover these keys. In particular an efficient way of solving the individual logarithm step is proposed based on the newly developed pre-computation step in the present study. The new algorithm of ICM is investigated on a class of problems in which the Pohlig-Hellman is an efficient method. Through the experimental results we have shown that the newly proposed individual logarithm step outperforms the Pohlig-Hellman method on some special cases and this leads to recover the ephemeral keys in reduced time.

Table 5.4: Running time of pre-computation and individual logarithm steps of ICM

Problem	Running time in sec
432412210443444766536290 824808696782290485 9	56
405006915689289033885033 14943591776516203	90
105488132477042462663174 85054480132114947	71
227504758229815122511473 89834477659827887	31
310233761222475167061100 77047014990674391	55
164838881183106336031117 884004666831843	10
295656961335792691161464 50939411987039	11
175660822294031990216013 13164759909980 66411	252
116045117879379642828484 543996278641093	179
968348003004618100399990 00830418556963	633
406873823690484794752321 14904989637283	671
247747816765350307022689 158418187059	274
381745147793332771090994 48511590627	180

Table 5.5: Running time of individual logarithm step and Pohlig-Hellman

Problem	Individual logarithm phase	Pohlig-Hellman
19	10ms	22ms
20	88ms	41ms
21	195	113
22	18ms	42ms
23	8ms	54ms
24	14ms	13ms
25	15ms	15ms
26	400ms	51ms
27	12ms	14ms

Chapter 6

Cryptanalysis on the DLP using Random method, on the ECDLP using Pollard-Rho and the DLP $\alpha^a\beta^b$

In this chapter, a few other computational approaches to solve the DLP are presented. One improves the performance of a primitive method of ICM, known as Random method, second is solving ECDLP efficiently using Pollard-Rho and the third approach is the improved cryptanalysis on the DLP $\alpha^a\beta^b$.

6.1 Performance analysis of Index Calculus Method

The performance of one of the primitive method of ICM known as Random method is enhanced by more than 50% using two approaches. The first approach is by using the partial linear sieve method. The second approach is by using the characteristics of smooth numbers over Z_p^*.

In the first approach, a new method (partial linear sieve method) is derived from the well known linear sieve and Pollard-Rho method and analyzed. The partial linear sieve and Random method are compared based on their running time. It is observed that a range parameter i.e., range of field elements checked for smooth-

ness, is influencing the running time of the partial linear sieve method. This leads to outperform the partial linear sieve method than the Random method. The range parameter is introduced in the Random method, due to which the probability of numbers getting smoothing is improved. It is shown that the performance of Random method is enhanced by more than 50% for problems of size ≈ 20 digits .

Similarly, the property based on the characteristics of smooth numbers over Z_p^* allows to map the field elements from one subset to another. Based on this concept an improved Random method by increasing the probability of numbers getting smoothing is devised. The efficiency of the improved Random method is analyzed experimentally. Through the experimental results it is observed that the performance is enhanced by $\approx 50\%$. The method is designed by exploiting the distribution pattern of smooth integers over Z_p^* on one of the large subgroup of $p - 1$.

6.1.1 Performance analysis on Random method

The following section presents the empirical analysis on the Random method and the partial linear sieve method.

Random method

In the first step of ICM, the linear relation involving the logarithms of the primes in the factor base is computed. In the field of integers, a factor base denotes a set of primes that are less than some bound, say B. An integer, say r, is said to be B-smooth if it factored into a product of primes such that,

$$r = p_1^{e_1}.....p_k^{e_k}$$

where $p_i^{e_i}, i = 1, 2, 3...k$ should be less than B and $e_i \neq 0$.

The easiest way to find the linear relation is to choose an integer, say u, at random satisfying $1 \leq u \leq p-1$. Then compute the least non-negative residue $r \equiv g^u \bmod p$ and test r for its smoothness. If r is smooth then a linear relation is formed using the factors of r [89].

The experimental analysis of Random method

To analyze this method, first a data file is produced, which contains a list of tuples. A tuple is of the form (p, g, B, m) with the following properties: a problem size m is chosen randomly and $11 \leq m \leq 20$; p is the safe prime of the form $2q + 1$, in which the factor q is also a prime. The size of p is chosen as m; g is the generator of order $p - 1$ and chosen randomly; B is the bound on which the factor base is formed.

Having built up the above data file the following algorithm is implemented to find the running time of the Random method for solving the given problem:-

- Read a tuple (p, g, B, m).
- A factor base is formed with first-t primes, which are less than the bound B.
- A linear relation involving the elements in the factor base has been found by using the Random method.
- The above step is repeated till a linear system of required size is obtained
- This linear system is solved by using the Lanczos method.
- An individual logarithm is found by using the fourth step in general algorithm of ICM as explained in the chapter 2 and the section 2.3.1.

The above steps are repeated and the running time is noted till all the tuples are calculated. In this method, the element to be checked for smoothness is assumed to be a random number lies between the range from 1 to $p - 1$. The element arising for the smooth test lies almost in the range of numbers with magnitude nearer to p and this leads to minimize the probability of getting a smooth number from the field element. This is due to the fact that the probability of getting a smooth integer from a large size field element is always less. Thus, the collection of required number of relations needs substantial amount of time in this method. The table 6.1 tabulates the running time of the Random method for problem size between 10 and 21. It is observed that the running time and the bound parameter are drastically increasing with respect to the problem size.

The partial linear sieve method

This method is derived from the well known linear sieve and the popular Pollard-Rho method. The procedure followed in this method is discussed below:-

Initially set $H = \sqrt{p}$; where p is the prime number. Compute the least non-negative residue of $(H + c_1)(H + c_2)$, where c_1 and c_2 lies in the range $0 \leq c_1 < c_2 \leq C$ and C is the sieve length i.e., the range of numbers to be checked for smoothness. If for a given c_1 and c_2 the residue is smooth it can be factored as $(H+c_1)(H+c_2) \equiv p_1^{e_1}.....p_k^{e_k} \mod$ p and the linear relation $\log_g^{(H+c_1)} + \log_g^{(H+c_2)} = e_1.\log_g^{p_1} + ...e_2.\log_g^{p_k} \mod$ p-1 is formed.

A non-negative residue of $(H+c_1)(H+c_2)$ is calculated and checked for smoothness by using Pollard-Rho method.

The procedure involved in this partial linear sieve method is given below:-

1. A linear system is formed as given in the previous paragraph.
2. The linear system is reduced by using the structured Gaussian elimination method.
3. The reduced linear system is solved by using the Lanczos method.
4. An individual logarithm for a given problem is found as explained in the general algorithm given in the section 2.3.1.

The main difference between the Random method and the partial linear sieve method is the range of numbers that need to be checked for smoothness. Here, the numbers selected for the smooth test will exist between some range, say from X to Y.

To analyze this partial linear sieve method, first a data file is produced, which contains a list of tuples. A tuple is of the form (p, g, B, C, y) with the following properties:-

A problem size m in chosen randomly and $21 \leq m \leq 30$; p is the safe prime of the form $2q + 1$, in which the factor q is also a prime; The size of p is chosen as m; g is the generator of order $p - 1$, such that g is a prime and less than the bound B; B indicates the bound based on which the factor base is formed. C indicates the sieve length i.e., the range of numbers to be checked for smoothness; y is the

element for which the discrete logarithm to be found. Having built up the above data file the following algorithm is implemented to find the running time of the partial linear sieve method for solving the given problem:-:-

- Read a tuple (p, g, B, C, y) .
- A factor base is formed with first-t primes less than the bound B.
- Set H as $\sqrt{(}p)$.
- Compute a residue $(H + c_1)(H + c_2) \bmod p$
- Check for the smoothness of the above residue by using the Pollard-Rho method.
- If the above element is smooth, then a linear relation has to be formed by involving the elements in the above factor base along with the unknowns $(H + c_1)$ and $(H + c_2)$. This procedure has to be repeated until the required number of relations are obtained.
- This linear system has to be reduced by using the structured Gaussian elimination method.
- The reduced linear system is of the form $Ax = 0$, where A is the coefficient matrix and x is a one dimensional unknown matrix. This linear system is transformed into the form $Ax = B$, where B is formed by using the logarithm of the generator. This reduced linear system is solved by using the Lanczos method.
- An individual logarithm phase is computed to solve the discrete logarithm of y by using the fourth step of general algorithm explained in the section 2.3.1.

The above steps are repeated and the running time is noted till all the tuples are calculated. The following table 6.2 tabulates the running time of the partial linear sieve method. The last column in table 6.2 represents the size of Y in

digits. However, the size of X is based on H. It is observed that the running time and the bound parameter are increasing with respect to the problem size but not drastically as observed in table 6.1. The following section discusses the comparative analysis between the Random method and the partial linear sieve method based on the running time. The result is tabulated in tables 6.1 and 6.2. Further, the improvement in the performance of Random method through the parameter range is analyzed.

Comparison analysis

Let us denote the base factor base consists of first-t primes less than the bound B. The number of relations generated to form the linear system in Random method is equal to the number of unknowns i.e., the primes in the base factor base. On the other hand, in the partial linear sieve (PLS) method, the factor base is expanding i.e, the number of unknowns is increasing, such as four times the number of primes in the base factor base. This is due to the fact that the addition of $(H + c_1)$ and $(H + c_2)$ also as unknowns. Thus the number of relations needed to form the linear system is also increasing to four times the number of primes in the base factor base. The PLS method outperforms than the Random method, even with the penalty of more number of relations to be generated for solving the given problem. The major conclusion from the above discussion is, the probability of numbers getting smoothing depends on the magnitude of numbers. The Random method checks the field elements nearer to the field size for smooth test, whereas, the PLS method checks the field elements in a range X to Y, where $X \approx \sqrt{p}$ and $Y \approx \sqrt{p}+d$. The PLS method checks the field elements of lesser magnitude for the smooth test than the Random method, this leads to achieve a better performance in the PLS method. For example the range of numbers checked for smoothness of a 28 digit problem is between 14 and 18 digits as reported in the table 6.2.

Similarly, the running time of Random Method is reduced by filtering the field elements of larger magnitude from the smooth test. Generally, in the Random method as explained in the section 6.1.1, u is chosen randomly and $g^u \bmod p$ is checked for smooth test. The smooth test is conducted on all elements ir-

respective of their magnitude. In the proposed approach, the field elements of approximately near to the field size are eliminated from the smooth test. Since the larger magnitude elements are eliminated from the smooth test, a more number of smaller magnitude elements are tested for smoothness to obtain required number of relations. The probability of getting smooth elements from smaller magnitude elements is always high and this leads to achieve $\approx 50\%$ improvement in the performance of Random method.

A comparison analysis is performed between the traditional approach and the new approach for different size problems. For each problem, the number of elements ($g^u \bmod p$) arising for smooth test and the number of smooth elements obtained in the above approaches are categorized into different groups with respect to their magnitude. Table 6.3 tabulates the results for problem size ranges from 14 digits to 20 digits. The first two columns in the table 6.3 represents the number of field elements ($g^u \bmod p$) generated for the smooth test and the number of smooth elements. The last two columns in the table 6.3 represents the number of field elements ($g^u \bmod p$) arising for smooth test after filtering larger magnitude elements and the number of smooth elements obtained for the same problem, respectively. The results are presented in terms of groups, classified based on their magnitude. The ratio between the number of smooth elements to number of field elements for each group in every problem can be calculated from this table. This ratio shows that the probability of getting smooth elements from smaller size elements is always high.

For example, for a 14 digits problem the first two columns represents the traditional approach and the last two columns represents the new approach. The required number of smooth elements to form the linear system is 95 and the size of field elements generated for the smooth test is between 14 and 16. In the traditional approach the 95 smooth elements are obtained in the following way. Out of 95, 29 smooth elements are obtained from 68481 field elements of size 14, 51 smooth elements from 61582 of size 13, 11 from 6250 of size 12, 2 from 610 of size 11, 1 from 58 of 10 and 1 from 1 of 6 digits elements, respectively. Similarly, the same 95 smooth elements are obtained in the following way after the filtration of 14 digit elements from the smooth test. Out of 95, 51 smooth elements are

obtained from 48020, 35 from 10788, 7 from 1089, 1 from 101 and 1 from 1 of 13, 12, 11, 10 and 8 digits, respectively.

In the similar line, for a 15 digits problem, the required number of smooth elements is 168 and the size of field elements generated for smooth test is between 15 and 13. The 168 smooth elements are obtained as follows: Out of 168, 46 smooth elements are obtained from 94267, 97 smooth elements from 84829 and 25 from 9328 of 15, 14 and 13 digits elements, respectively. Similarly, the same 168 smooth elements are obtained in the following way after the filtration of 15 and 14 digits elements. Out of 168, 127 smooth elements are obtained from 55354, 28 from 5628, 13 from 645 of 13, 12, 11 digits elements, respectively. The other problems such as 16, 17, 18 and 19 in the Table-3 also reports the similar statistics on the Random method.

From the above discussion it can be said that, the range of numbers to be checked for the smooth test can be reduced from the original field size to some extent. Thus the required number of smooth elements can be obtained from the smaller magnitude elements itself in comparison with the larger magnitude elements. This leads to improve the performance of the Random method. Based on this concept the following table 6.4 and table 6.5 present the running time of the Random method with or without reduction in the range with respect to the problem size.

Table 6.1: Running time of Random method

Problem size in digits	Time(sec)	bound
11	1	500
12	4	800
13	8	1000
14	35	3000
15	81	5000
16	153	7000
17	464	10000
18	943	13000
19	1865	16000
20	4001	20000

Table 6.2: Running time of partial linear sieve method

Problem size in digits	Time(sec)	bound	range
21	128	5000	15
22	200	7000	15
23	262	9000	16
24	409	12000	16
25	682	14000	17
26	1266	16000	17
27	2060	18000	18
28	3705	20000	18
29	6520	23000	19
30	11200	27000	19

Table 6.3: The difference in the number of smooth integers before and after the reduction

size of random elements	Number of random elements	smooth elements	size of random elements after filtration	smooth elements after filtration
14	68481	29	–	–
13	61582	51	48020	51
12	6250	11	10788	35
11	610	2	1089	7
10	58	1	101	1
9	6	0	4	0
8	0	0	1	1
7	0	0	–	–
6	1	1	–	–
15	94267	46	–	–
14	84829	97	–	–
13	9328	25	55354	127
12	0	0	5628	28
11	0	0	645	13
16	117542	128	–	–
15	105237	155	129970	220
14	10746	26	29146	85
13	1078	7	2931	20
12	107	1	294	2
11	0	0	33	0
10	0	0	1	0
17	465715	387	–	–
16	11270	33	–	–
15	1626	10	124687	399
14	–	–	2350	25
13	–	–	258	6
18	660530	326	–	–
17	198479	185	459618	429
16	19462	28	58849	93
15	1962	9	5821	22
14	2177	2	604	6
13	215	0	64	0
12	21	0	3	0
19	913165	532	–	–
18	91595	109	667174	403
17	8978	22	6721	23
16	924	6	770	4
15	88	0	–	–
14	9	0	–	–

Table 6.4: The running time with or without reduction in the range with respect to problem size in Random method

Digits	Time(sec)	Time (reduction in range)
13	9	9
14	38	33
–	39	29
–	53	46
–	60	49
–	91	75
–	41	39
–	45	43
–	48	35
–	49	41
–	51	42
15	58	46
–	96	64
–	99	81
–	126	86
–	139	101
–	150	107
–	134	110
–	184	119
–	170	122
–	101	76
16	110	79
–	148	122
–	179	132
–	206	141
–	227	150
–	264	171
–	229	149
–	287	168
–	299	181

Table 6.5: Difference in running time with or without reduction in the range with respect to problem size

Digits	Time(sec)	Time (with reduced range)
17	265	173
–	368	278
–	479	295
–	495	313
–	566	309
–	567	354
–	601	361
–	737	425
–	752	439
18	632	391
–	845	599
–	1028	603
–	1239	702
–	1456	866
–	1475	813
–	1575	910
–	1887	938
–	1920	976
19	2274	880
–	2346	1297
–	2853	1678
20	4460	2010

6.1.2 Improved Random method using the properties of smooth number over Z_p^*

As discussed earlier, the ICM is the most effective attack on the DLP. The pre-computation step of ICM is based on the probability of field elements getting smoothing i.e., the factors of field elements are less than a prescribed bound. In the present work the ICM is improved through the property of smoothness concept over the the prime field Z_p^* for certain instances of primes, such as $p = 2q+1$, where q is a prime. The property based on the characteristics of smooth numbers over Z_p^* allows to map the field elements from one subset to another. Based on this concept an improved ICM by increasing the probability of numbers getting smoothing is devised. The efficiency of the improved ICM is analyzed experimentally. Through the experimental results it is observed that the performance of ICM is enhanced by $\approx 50\%$.

From the discussions of chapter 4, it is observed that, in certain primes, such as $p = 2\rho+1$, where ρ is a prime or product of prime, when there is a mapping between a subgroup of order $\frac{p-1}{2}$ and the underlying group an inverse palindrome pattern is generated. On the other hand for a mapping between a subgroup of order $2x$ and a group, where x is a prime or product of primes yields a palindrome pattern. The proposed method is based on inverse palindrome pattern. The relationship between the discrete logarithm of an element, say y, and the corresponding $-y$ in this pattern is given below.

$$x = q \pm 2v \bmod (p-1) \quad \text{when } y \text{ is } O(2q) \tag{6.1}$$

$$v = (x \pm q)/2 \bmod q \quad \text{when } y \text{ is } O(q) \tag{6.2}$$

where q is the order of the subgroup; and, v is the discrete logarithm of $-y$ with respect to order q if x is that of y and vice-versa. The above equations are presented as discussed in chapter 4 for completeness.

Equations 6.1 and 6.2 allow us to improve the probability of numbers getting smoothing. This property of relating y and $-y$ allows to map a field element of one subset to another and increases the probability of getting smooth numbers. The main intuition behind the work is the probability of getting smooth numbers from

elements of smaller magnitude is higher. The above equations aid in mapping the higher magnitude elements to smaller magnitude elements. For example, a subset of the field elements $\{1, 2, ..., p-1\}$, such as $\{\frac{p-1}{2}+1, ..., p-1\}$ can be mapped to $\{-\frac{p-1}{2}+1, ..., -(p-1)\}$ for a problem p. Here the set $\{\frac{p-1}{2}+1, ..., p-1\}$ is considered as elements of larger magnitude and $\{-\frac{p-1}{2}+1, ..., -(p-1)\}$ i.e., $\{1, 2, ..., \frac{p-1}{2}\}$ is considered as elements of smaller magnitude. The margin between the smaller and larger elements can be chosen arbitrarily.

The **Algorithm-6** explains the improved ICM using the Random method for solving the DLP. The mapping of one subset to another subset is exploited to device an improved ICM. Initially the margin between the smaller and larger elements is to be chosen arbitrarily. The set $\{1, 2, ..., p-1\}$ is partitioned into two sets such as S_1, and S_2 of arbitrary size based on the margin and the magnitude of elements are in increasing order. The procedure for generating relations from the random elements belongs to S_1 is similar to that of traditional random method, whereas, for elements of set S_2, the relations are formed by transforming S_2 to S_1. The smaller magnitude elements belongs to S_1 are tested for smoothness in place of larger magnitude elements belongs to S_2. Since the elements of S_1 are lesser in magnitude in comparison with S_2 and it is known that the probability of lesser magnitude elements getting smoothing is high, the performance of Random method is improved. The transformation is possible only if the logarithms of S_2 can be related with the logarithms of S_1. This is obtained by using the equations 6.1 and 6.2 that are evolved from the inverse palindrome pattern of B- smooth numbers over Z_p^*

The **Algorithm-6** presents the problems of primes of the form $p = 2q + 1$, where q is a prime. In general, the DLP is to be computed with respect to the order $p - 1$ i.e., $2q$. The probability of field elements getting smoothing can be improved through the inverse palindrome pattern. Since the pattern is obtained from the elements of order q, the algorithm is designed by implementing the Random method with respect to prime order subgroup of order q. This leads a way to compute the DLP, say u, with respect to q. This DLP, u mod q is transformed to u mod $p - 1$ by computing $u = u * 2$.

The important steps involved in **Algorithm-6** are as follows :

- Transform the generator of order $p-1$ into generator of one of the large subgroup.
- Map the field elements of S_2 into S_1 which are arising in the Random method.
- Relate the logarithms of S_2 into S_1 using the equations 6.1 and 6.2.
- Form the linear system, reduce the system into smaller size and solve the system.

Algorithm 6 Algorithm for pre-computation step

Input: : Problem of size p, factors of $p-1$ is $2q$, a primitive element g and FB of bound B.

Output: : logarithms of elements of factor base.

1: Set N=no of unknowns in the FB, and R=0
2: Partitioned the set $0, 1, 2, ...p-1$ into two sets S_1 and S_2 of arbitrary size.
3: Find g_1 as $g^{\frac{p-1}{q}}$
4: Repeat the following steps 4 to 15 until N equals R
5: Choose randomly u from $1, 2, ...q$
6: Compute $s = g_1^u \ mod q$
7: **if** $s \in S_2$ **then**
8: $s = p - s$
9: $u = q + 2 * u$
10: **else**
11: $u = u * 2$
12: **end if**
13: Factorize s and check for smoothness
14: Form the relation $u = e_1 \log_g^{p_1} + e_2 \log_g^{p_2} + e_3 \log_g^{p_3} + ...$; where $p_1^{e_1} p_2^{e_2} p_3^{e_3}$ are factors of s
15: $R = R + 1$
16: Reduce the linear system into smaller system using the structured Gaussian elimination method
17: Solve the linear system using the Lanczos method

6.1.3 Experimental results

This section presents the results and analysis of the ICM discussed above. First a data file is produced, which contains a list of tuples. A tuple is of the form (m, p, B) with the following properties. m lies between 13 and 20 digits, p is a safe prime of the form $p = 2q + 1$ and B is the bound of the factor base. Based on these properties the tuples are computed as follows:

1. Choose k as 100.
2. m is selected between 13 and 20 digits.
3. A prime number is selected of size m and checked for safe prime.
4. A bound B is calculated.
5. Repeat the above steps (2 to 4) for k number of times.

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

- Read a tuple (m, p, B)
- Set the factor base from B.
- Execute the traditional random method for the above tuple.
- Execute the new random method for the same tuple.
- Keep track of the computed run time.
- Repeat the above steps until all the tuples are calculated.

The tables 6.6 and 6.7 show the difference in running time of traditional ICM using Random method with improved ICM.

6.2 DLP on elliptic group using Pollard-Rho

This section presents the elliptic curve cryptosystem and the popular attacks to break this system. Let p be a prime number and F_p denotes the field of integer modulo p. An elliptic curve, say, E over F_p is defined by the following equation

Table 6.6: Difference in the running time of traditional and improved Index Calculus Method

Problem size in digits	Running time of Traditional Index Calculus Method	Running time of improved Index Calculus Method
13	9	8
14	38	30
–	39	27
–	53	43
–	60	41
–	91	67
–	41	35
–	45	40
–	48	33
–	49	37
–	51	47
15	58	48
–	38	32
–	96	61
–	99	77
–	126	76
–	139	98
–	150	97
–	134	104
–	185	115
–	170	112
–	101	70
16	110	76
–	148	112
–	179	123
–	206	136
–	227	152
–	264	163

Table 6.7: Difference in the running time of traditional and improved Index Calculus Method

Problem size in digits	Running time of Traditional Index Calculus Method	Running time of Improved Index Calculus Method
16	229	147
–	287	163
–	299	176
17	479	290
–	495	310
–	566	301
–	567	350
–	601	358
–	737	429
–	752	430
18	632	386
–	845	594
–	1028	600
–	1239	698
–	1456	841
–	1475	802
–	1575	890
–	1887	918
–	1920	956
19	2274	860
–	2346	1207
–	2853	1628
20	4460	1969

$$y^2 = x^3 + ax + b$$

where $a, b \in F_p$ satisfy $4a^3 + 27b^2 \not\equiv 0$ mod p. A pair (x, y), where $x, y \in F_p$, is a point on the curve, if (x, y) satisfies the above equation. The point at infinity is denoted as ∞. The elliptic group is formed by the set of points on the curve and infinity with addition operation. Elliptic Curve Discrete Logarithm Problem (ECDLP) is defined as: For a given elliptic curve $E(F_p)$ on a field F_p, a generator P and another point Q, the problem of finding x, such that $Q = xP$.

Pollard-Rho is a popular method to solve the ECDLP, since it solves the given problem in any group structure. Pollard [73] first showed that the application of the Pollard-Rho method to solve the DLP in expected running time $O(\sqrt{G})$, where G is the order of the group. The space requirements of algorithms using the Rho method are negligible, whereas the Shanks baby step and giant step method needs both runtime and space $O(\sqrt{G})$. In the present study Pollard-Rho method is considered to solve the ECDLP. In comprehensive experiments, the Pollard-Rho method is applied to solve DLP on the Elliptic curves over $F(2^m)$. The following section presents the Pollard-Rho method in detail.

Pollard-Rho Method for ECDLP

If Z is a point then a function $f(Z)$ is defined as

$$f(Z) = \begin{cases} 2Z & : \quad Z \in S_1 \\ Z + P & : \quad Z \in S_2 \\ Z + Q & : \quad Z \in S_3 \end{cases} \tag{2}$$

Where S_1, S_2 and S_3 are roughly equal set of points

The procedure for this method is as follows:

1.ChooseA_o and $B_o \in [1, n-1]$ at random

2. Compute the starting point, say, $Z_o = A_o P + B_o Q$

3. Compute the sequence $Z_i = f(Z_i)$

4. Keep the track of A_i and B_i such that $Z_i = A_i P + B_i Q$

For the next sequence the above equation (2) can be written as follows:

$$Z_{i+1}, A_{i+1}, B_{i+1} = \begin{cases} (2Z_i, 2A_i, 2B_i) & : \quad Z \in S_1 \\ (Z_i + P, A_{i+1}, B_i) & : \quad Z \in S_2 \\ (Z_i + Q, A_i, B_{i+1}) & : \quad Z \in S_3 \end{cases} \tag{3}$$

For a particular case $Z_i = Z_j$; then $l = \frac{A_i - A_j}{B_j - B_i}$, which is the required solution to the ECDLP.

The following section describes the experimental work based on Pollard-Rho method to solve the ECDLP.

6.2.1 Experimental results

Teske [92] conducted experiments on the Elliptic group based on prime field i.e., $E_{a,b}(F_q)$. In the present work the experiments are conducted on finite field of $E_{a,b}(F_{2^m})$, since most of the cryptographic applications prefer this field for ECC as the computations are faster.

The problem is described as follows : First a data file is produced, which containing a list of tuples, namely, (m, a, b, n, p) with the following properties: m lies between 16 and 80 bits, $E_{a,b}$ is an elliptic curve over $F(2^m)$, the corresponding elliptic curve group has group order n and p is the largest prime factor of n.
Based on these properties the tuples are computed as follows:

- m is selected randomly between 16 and 80 bits.
- An irreducible polynomial of m degree is selected.
- Two constants, namely, a and b are selected randomly belongs to $F(2^m)$, such that $4a^3 + 27b^2 \neq 0$.
- The order of an elliptic curve (n) on group $E_{a,b}(F(2^m))$ is found using Weil's algorithm .
- Factorization of n and the largest prime factor p is obtained.

Having built up the above file, the following algorithm is computed:
Step1 : Read a tuple (m, a, b, n, p)

Table 6.8: Results on Pollard-Rho method

Problem size in bits	x	Running time
41	6	2h37m27s
–	76	13m22s
–	6637	4m17s
–	576866	3m23s
–	60000066	2m34s
65	555588545325	3h43m15s
–	4955	3h50m13s
–	456	16 hours

Step2 : Find a fixed point P of order p.

Step3 : choose a random integer k value between 10 and 100.

Step4 : Choose a point Q generated from P.

Step5 : Use the Rho method to compute $\log\ Q = \log_F Q$.

Step6 : Keep track of run time computed until a match has been found.

Step7 : Repeat Steps 4,5 and 6 for k number of times.

Step8 : Repeat the above steps until all the tuples are calculated.

Table 6.8 and 6.9 show the problems that are solved for the ECDLP. The problem size is represented based on the bit size of $E_{a,b}(F(2^m))$, the private key x and the the time taken to solve the problems. It is interesting to see that irrespective of the group order, the smaller size private key x takes more time to get solved, whereas the higher size x get solved comparatively in reduced time bound.

Table 6.9: Problems solved using Pollard-Rho method

Problem size in bits	x	No of group operation
32	4935	1759
–	99	1983
–	88	4486
–	55	7231
–	f	19179
–	b	25615
–	4	60844
–	2	100309
20	9008	837
–	8888	433
–	546	592
–	88	116
–	8	882
–	5	1324
–	4	1583
–	3	1966
–	2	2605

6.3 Cryptanalysis on the DLP $\alpha^a\beta^b$

Recently Kashyap et al., [43] proposed a cryptosystem based on the DLP, $\gamma = \alpha^a\beta^b$, where α and β are generators, a and b are distinct integers. The above cryptosystem and the corresponding cryptanalysis are further analyzed in the present study. Sramka [87] claimed that for the cipher text (c_1, c_2, c_3) of the cryptosystem discussed in [43], the random integer, say k, can be obtained by the simple relation such as $c_1c_2 = (\alpha\beta)^k$. Further, he reported an attack, in obtaining the plain text from a valid cipher text, by computing the single traditional DLP. In the present study, it is shown that the above k can be obtained using above relation only for some special cases and those cases are discussed. Also an improved version of the encryption scheme based on the cryptosystem published by Kashyap et al., [43] is proposed. It is observed that the proposed cryptosystem is invulnerable to the attack, in obtaining the plain text from a valid cipher text, by computing the single traditional DLP.

6.3.1 The encryption scheme based on the DLP $\alpha^a\beta^b$

Kashyap et al., [43] defined the discrete logarithm with two different exponentiation and with two distinct integers. The DLP is defined as follows: For a finite cyclic group G with generators of α and β and two distinct integers a and b, it is hard to recover a and b, with known γ, α and β, such as $\gamma = \alpha^a\beta^b$, where $\alpha \neq \beta^i$ and $a \neq b^i$. Based on the above DLP, they [43] proposed a public-key encryption scheme. They claimed that to recover a and b, a double DLP to be solved. The description of the scheme between two communicators, namely, Alice and Bob is given below:

Key generation:

Alice selects an efficient description of a cyclic group G of order p with generators α and β , such that $\alpha \neq \beta^i$. Alice chooses random integers, say a and b from $\{$ $0, ..., p-1$ $\}$, such that $a \neq b^i$. Alice computes $\gamma = \alpha^a$ and $\delta = \beta^b$. Then, Alice publishes γ and δ along with the description of G, p α and β as her public key.

Also, Alice retains a and b as her private key.

The encryption algorithm :

For encrypting a message, say m, to Alice, with her public key $(G, p, \alpha, \beta, \gamma, \delta)$, the following algorithm is adopted.

- Bob converts the m into an element of G.
- Bob chooses a random integer, say k, from $\{0, ..., p-1\}$, then calculates $c_1 = \alpha^k, c_2 = \beta^k$ and $c_3 = m\gamma^k\delta^k$.
- Bob sends the cipher text (c_1, c_2, c_3) to Alice.

The decryption algorithm :

For decrypting a cipher text (c_1, c_2, c_3), received from Bob, Alice computes $\frac{c_3}{c_1^a c_2^b}$ as the plain text message with her private keys a and b. Sramka [87] reported an attack on the above discussed scheme and it is explained in the following section.

The cryptanalysis on the cryptosystem based on the $\gamma = \alpha^a\beta^b$

Sramka [87] described that the random integer, k used in the encryption can be obtained using the following equation

$$c_1 c_2 = \alpha^k \beta^k = (\alpha\beta)^k \tag{6.3}$$

Here k can be solved with the single traditional DLP. Once the value of k is known, the plain text can be obtained as $\gamma^k\delta^k c_3$. Also, he [87] proposed an attack, which is explained below. Since α is a generator, β can be represented as α^m. Then $\gamma\delta = \alpha^a\beta^b = \alpha^{a+mb}$. The logarithm $a + mb$ can be solved and the plain text x, can be retrieved as follows

$$\begin{aligned} c_3 &= x\gamma^k\delta^k \\ c_3 &= x\alpha^{ak}\beta^{mbk} \\ c_3 &= x\alpha^{k(a+mb)} \end{aligned} \tag{6.4}$$

Since $c_1 = \alpha^k$

$$x = x\alpha^{k(a+mb)} c_1^{-(a+mb)} \tag{6.5}$$

$$x = x\alpha^{k(a+mb)} \alpha^{-(a+mb)k} \tag{6.6}$$

$$x = x \tag{6.7}$$

The following section describes that the plain text cannot be recover with the knowledge of k and an improved version of encryption scheme based on the cryptosystem published by Kashyap et al., [43] is proposed.

The analysis on the cryptanalysis

Sramka [87] claimed that the random integer, k, can be obtained simply by multiplying c_1 and c_2. When the underling group is Z_p^*, k be a random integer in the range $0, ..., p-1$. The value of k to be obtained from the equation $c_1c_2 = (\alpha\beta)^k$ is k mod $O(\alpha\beta)$, where $O(\alpha\beta)$ denotes the order of an element $\alpha\beta$ mod p . In some cases the product of $\alpha\beta$ may get confine to a small subgroup, then the plain text cannot be recover with the knowledge of k. For example, if β is an inverse of α, then the product of $\alpha\beta$ yields an identity element, which in turns produces the value of k as zero. For simplicity, let us consider a field F_{11}. Suppose α, which is discussed in the above paragraph is 2 and β is 6, since they are inverse of each other, the product $(\alpha\beta)$ mod p yields one, which, in turns produces the value of k as zero. The value of k mod $p-1$ cannot be retrieved with the information of k mod one. On the other hand if the product $\alpha\beta$ is an element of a large prime order subgroup, then the value of k mod p-1 can be obtained by using the Pohlig-Hellman method, with the additional knowledge of the factorization of $p-1$. For example, consider a field F_{11}. Suppose α is 2 and β is 7, then the product is 3, which is an element of prime order subgroup of order 5. The k mod 10 is obtained by solving the DLP on the prime order subgroup of order 5, which is k mod 5 and combining the result with k mod 2 using Pohlig-Hellman method. From the above discussion it is known that the value of k cannot be retrieved simply by the product of c_1c_2, only the value of k mod $O(\alpha\beta)$ can be obtained. However the value of k mod p-1 can be obtained simply from c_1 or c_2 with the knowledge of α

and c_1 or β and c_2 , since both are public, k can be obtained by computing a single DLP and the plain text can be retrieved from cipher text with the knowledge of k.

6.3.2 Improved version of cryptosystem

In this section an improved encryption scheme is discussed. The modified encryption and description procedures of proposed scheme [43] of Kashyap et al., are presented here.

Key generation:

Alice selects an efficient description of a cyclic group G of order p with generators α and β.
Alice chooses random integers, say a and b from $\{0, ..., p-1\}$, such that $a \neq b^i$.
Alice computes $\gamma = \alpha^a$ and $\delta = \beta^b$. Then, Alice publishes γ and δ along with the description of G, p, α and β as her public key. Also, Alice retains a and b as her private key.

The encryption algorithm :

For encrypting a message, say m, to Alice, with her public key $(G, p, \alpha, \beta, \gamma, \delta)$ the following algorithm is adopted.

- Bob converts the m into an element of G.
- Bob chooses a random integer, say k, from $\{0, ..., p-1\}$, then calculates $c_1 = \alpha^{k_1}, c_2 = \beta^{k_2}$ and $c_3 = m\gamma^{k_1}\delta^{k_2}$.
- Bob sends the cipher text (c_1, c_2, c_3) to Alice.

The decryption algorithm :
For decrypting a cipher text (c_1, c_2, c_3), received from Bob, Alice computes $\frac{c_3}{c_1^a c_2^b}$ as the plain text message with her private keys a and b. Now, the proposed is analyzed for the resistance of Sramka attack
As described by Sramka [87] the double discrete logarithm problem can be written

as $\gamma\delta = \alpha^a\beta^b = \alpha^{a+mb}$. The discrete logarithm of $a+mb$ can be solved by the single traditional DLP. The hardness of solving a and b is based on finding the solution of $a + mb = t \bmod p - 1$. The cipher text (c_1, c_2, c_3) represents $c_1 = \alpha^{k_1}, c_2 = \beta^{k_2}$ and $c_3 = m\gamma^{k_1}\delta^{k_2}$.

Now,

$$c_3 = x\gamma^{k_1}\delta^{k_2} \tag{6.8}$$

$$c_3 = x\alpha^{k_1}\alpha^{mbk_2} \tag{6.9}$$

$$c_3 = x\alpha^{k_1+mbk_2} \tag{6.10}$$

From the above equation 6.10, it can be inferred that it is possible to recover the plain text with the knowledge of $a + mb$, only when $k_1 = k_2$ as explained by Sramka [87], whereas, it is hard to recover the plain text with the knowledge of $a + mb$, when $k_1 \neq k_2$. This proves that the proposed system is invulnerable to Sramka attack.

6.3.3 Security analysis of newly proposed system

Apart from the fact, that the proposed system is invulnerable to Sramka attack, it has two levels of security. The first level is a one way function known as the DLP. The second level is, even if DLP is solved and t is known the equation $a + mb$ mod p-1 is to be solved. With known t, it is not possible to retrieve a and b from the above equation. For a cryptosystem to be semantically secure, it must be infeasible for a computationally-bounded adversary to derive significant information about a message when given only its cipher text and the corresponding public encryption key. Semantic secure doest not consider the chosen cipher text attack, where the attacker is able to request the decryption of chosen cipher text. El-Gamal is unconditionally malleable and therefore is not secure under chosen cipher text attack.

The new scheme proposed at section 6.3.2 is also vulnerable to chosen cipher text attack. This can be precluded with the following feature incorporated in the proposed system.

Key generation:

Alice selects an efficient description of a cyclic group G of order p with generators α and β. Alice chooses random integers, say a and b from $\{0, ..., p-1\}$, such that $a \neq b^i$. Alice computes $\gamma = \alpha^a$ and $\delta = \beta^b$. Then, Alice publishes γ and δ along with the description of G, p, α and β as her public key. Also, Alice retains a and b as her private key.

The encryption algorithm :-

For encrypting a message, say m, to Alice, with her public key $(G, p, \alpha, \beta, \gamma, \delta)$, the following algorithm is adopted.

- Bob converts the m into an element of G.
- Bob chooses a random integer, say k, from $\{0, ..., p-1\}$, then calculates $c_1 = \alpha^{k_1}, c_2 = \beta^{k_2}$ and $c_3 = m\gamma^{k_1}\delta^{k_2}$.
- Bob gets λ by using a collision resistance hash function H as $\lambda = H(c_1, c_2, c_3)$ and calculates q as $\gamma^{k_1}\delta^{k_2\lambda}$.
- Bob sends the cipher text (c_1, c_2, c_3, q) to Alice.

The decryption algorithm :-

Alice computes $\lambda = H(c_1, c_2, c_3)$ and verifies $c_1^a c_2^{b\lambda}$ is q or not. If the verification is success then Alice computes the plain text as $\frac{c_3}{c_1^a c_2^b}$ with her private keys a and b.

6.3.4 Limitations

The size of the cipher text generated in the proposed system is larger compared with the message to be encrypted. Therefore, the newly proposed cryptosystem can be used for the transmission of smaller size messages, for example: to exchange a session key. An attack is proposed on El-Gamal system when the message is short. The attack is described as follows: The attack is based on the fact that public key encryption is typically used to encrypt session keys. These keys are typically short i.e. less than 128 bits. The attack shows that when using plain

RSA and plain El-Gamal to encrypt an m-bit key, it is often possible to recover the key in time approximately $2^{\frac{m}{2}}$. When the session- keys are 64 bits, the attack shows that both plain RSA and plain El-Gamal completely insecure systems [16].

- Let $< p, q, y >$ be an El-Gamal public key. When the order of g is at most $\frac{p}{2^m}$, it is possible to recover M from any El-Gamal ciphertext of M in the time it takes to compute $2.2^{\frac{m}{2}}$ modular exponentiation.

- Let $< p, q, y >$ be an El-Gamal public key. Suppose $p-1 = qs$ where $s > 2m$ and the DLP for the subgroups of Z_p^* of order s is tractable i.e. takes time T for some small T. When the order of g is $p-1$ it is possible to recover M from any cipher text of M in time T and $2.2^{\frac{m}{2}}$ modular exponentiation.

- Let $< p, q, y >$ be an El-Gamal public key. Suppose $p-1 = qs$ where $s > 2m$ and the DLP for the subgroups of Z_p^* of order s takes time T for some small T. When the order of g is $p-1$ or at most $\frac{p}{2^m}$, it is possible to recover M from any cipher text of M in time T plus one modular exponentiation and $2.2^{\frac{m}{2}}$ additions.

The above attack can be avoided in the newly proposed system by choosing an appropriate prime field and the generator i.e., a safe prime of the form $2q + 1$, where q is also prime or a prime order subgroup , where the computations in discrete logarithm based schemes are restricted within the prime order subgroup. These types of primes does not allow the leakage of bits from the DLP . The DLP for the subgroups of Z_p^* of order s is not helpful for the computation of the DLP in the prime order subgroup. This is due to the fact that in the above subgroup, the DLP of y to be computed with respect to the prime order instead of the DLP with respect to the order $p-1$, which is the common practice. On the other way , the safe primes are also considered as safe in the literature, since it does not allow any partial computation of the DLP. It leaks only one bits of information through the factor 2 apart from q. Therefore the security implication of newly proposed system is based on the prime field and the generators chosen for the computations.

6.4 Conclusion

In the present study the performance of Index Calculus Method is improved through the Random method. A parameter, which is influencing the running time of this method is identified and the performance is improved through the same parameter. A new method, namely, partial linear sieve method is introduced and analyzed in the present study. It is observed that the partial linear sieve method outperformed than the Random method based on the running time with respect to the problem size. This leads to introduce the range parameter in the Random method. Hence the probability of elements getting smoothing is improved. It is also shown that the probability of numbers getting smoothing depends on the magnitude of numbers and this leads to achieve a better performance in the Random method. This result is achieved, since the Random method needs lesser number of relations to form the linear system than the other popular methods and hence the performance of this method is analyzed and improved. The characteristics of smooth number over Z_p^* lead to develop a new method for pre-computation step of index calculus method. The probability of getting smooth number in a given range is improved through the new method. Our experimental results show that the performance of index calculus method is improved by more than 50%.

The problem of solving the DLP is extensively studied due to its numerous applications. The cryptosystem based on the DLP, $\gamma = \alpha^a \beta^b$, and the cryptanalysis of the above mentioned cryptosystem are reported in the literature. The present study described here shows that even though the condition $\alpha \neq \beta^i$ does not holds in any cyclic group, the equation $a + mb \bmod p - 1$ has to be evaluated to obtain the value of a and b with the knowledge of $a + mb$. It is also shown that $a + mb$ does not have any impact on the improved version of cryptosystem proposed in the present study. Further the methods of improving the security of newly proposed cryptosystem are also discussed.

Chapter 7

Discussion and Conclusions

In this thesis, the problem of computational analysis on the Discrete Logarithm Problem (DLP) is considered. The unreliable communication media rely on many cryptographic protocols for the security. These cryptographic protocols in turn depend on the hard problems such as the DLP. From computational point of view, there is no algorithm that solves the DLP in polynomial time. This motivated the present study on the DLP and the computation of DLP is improved from various aspects such as developing new techniques, improving the traditional methods and identifying the trap doors.

7.1 Discussion

The main focus in the present study is the most effective attack on the DLP namely ICM. Our approach to improve the performance of ICM is viewed from two perspectives such as to improve the traditional methods involved in ICM and developing a new procedure to solve the problem using ICM. Traditionally the ICM is viewed into two steps such as generation and solving. The solving consists of reduction and solving the reduced matrix. In the present study, reduction is moved alongside generation and it enabled the selection of parameters in such a way that a smaller system of equations is generated overall. Although it is shown to be useful for solving the DLP through ICM it may also be useful for many linear algebra techniques that are rely on the sparsity of the matrix to be solved.

A special case of ICM, analogous to Pohlig-Hellman, when the factors of $p-1$ are small is studied in this thesis. In the literature, the Pohlig-Hellman is the best known method to solve the DLP, when the factors of $p-1$ are small and known, while ICM is an efficient method for general DLP. Two algorithms are proposed to improve the efficiency of the pre-computation step of the ICM by using the equivalence classes formed from a special case of B-smooth numbers over Z_p^*. This variant of ICM is shown to be particularly useful on ephemeral keys. In ephemeral key security, the underlying field and generator of the cryptosystem are held static but each session uses different keys. For such systems, the pre-computation of ICM can be performed once using the new algorithms in reduced time with individual logarithm step for every ephemeral key. For ephemeral key systems, the security requirements are considered less stringent as the keys change frequently [20]. Pohlig-Hellman solves every ephemeral key as a different (new) problem. On the contrary, in ICM, only the individual logarithm step, which is usually the easier step, has to be recomputed. An efficient way of developing the individual logarithms step is developed in the present study.

In Chapter 4, primes (p) are classified into five types based on the factors of $p-1$. This classification aids in analyzing the characteristics of new variants of smoothness concept of integers in Z_p^* on different types of primes. This leads to formulate new techniques to solve the DLP in reduced time. In the literature smooth integers are known as B-smooth numbers. In the present study B-smooth numbers over Z_p^* are introduced and found to exhibit different characteristics on different types of primes. Patterns are identified from the distribution of B-smooth numbers over Z_p^*. A new way of looking at cryptanalysis is presented in this thesis through the identification of patterns from the prime field. The identified patterns are used to solve the DLP, which are in the range near the middle of the group on safe prime, prime order subgroup and random primes. The DLP defined on safe primes, primes order subgroup and random primes with one large factor of $p-1$ are considered as safe in the literature. In this thesis we proved that the above problems are vulnerable to the techniques that are developed by using the properties of B-smooth numbers over Z_p^*. The analysis made in the present work aids in computing the system parameters for various cryptosystems securely.

A special case of smooth numbers over Z_p^* that helps to divide Z_p^* into partitions is proposed. These partitions are shown to be useful in developing an efficient variant of ICM to solve the DLP for ephemeral keys. A new pre-computation method is proposed for ICM based on the characteristics of special case of smooth numbers over Z_p^*. The characteristics are studied on different types of primes based on the factors of $p-1$.

Also, the performance of Random method, which is one of the primitive methods of ICM is analyzed and improved through the concepts discussed above. The partial linear sieve method, newly introduced in the present study, is shown to improve the performance of the random method sometimes by over 50% for problem sizes of about 60 bits. A second improvement is through the patterns identified from the smooth number set over Z_p^*. The property evolved from the patterns of smooth numbers over Z_p^* aids in mapping the group elements from the larger magnitude to much smaller magnitude. This led to the development of an improved Random method by increasing the probability of finding field elements that are smooth.

At more preliminary level the ECDLP problem is analyzed through the Pollard-Rho method. New results are obtained though the experimental results conducted on the $E_{(a,b)}F(2^m)$, which is the most widely used field in the Elliptic curve cryptosystems. The results appears counter-intuitive and needs furthermore detailed analysis.

From the above discussion, the algorithms designed to solve the Discrete Logarithm Problem in the present study from a computational perspective is summarized as follows:-

- Efficient algorithm to find the exponent near the middle of a group or a prime order sub group.
- Improved linear sieve based pre-computation step of ICM based on better choice of parameters
- Two algorithms for ICM that are efficient for the special case of $p-1$ having no large factors. These are useful for attacking ephemeral keys.

- Other studies that may lead to more efficient algorithm in certain special cases for ECDLP; improved efficiency of Random method.

At a broader and a more abstract level, we introduced a new approach to cryptanalysis that is based on deriving computationally optimal choices. For example, the choice of R (chapter 3), factor bases and types of primes(chapter 4 and 5) have to the best of our knowledge no direct theoretical basis. The computational approach that we have taken is not related to finding efficient ways(either using hardware or special purpose libraries) of implementing known algorithms; nor is it based on a number-theoretic approach. Thus the approach followed in this thesis is a novel computer-science approach to cryptanalysis in contrast with number-theoretic methods.

7.2 Extensions

This section discusses the various ways of extending the algorithms, methods and techniques presented in this thesis. The algorithm developed for the pre-computation step of ICM is a generic algorithm. The terminating conditions obtained from the heuristic analysis reported in this thesis are independent from the implementation point of view. The generation step of pre-computation step, generates the relations by satisfying the above conditions iteratively. The algorithm presented in this thesis uses crisp values for the parameters. This can be extended by modifying the algorithm using the fuzzy concept. A fuzzy controller can be designed for the above parameters and the required matrices can be obtained by using the fuzzy logic. Furthermore, other soft computing techniques can be attempted to achieve better performance. Application of soft computing techniques are widely used for all types of statistical analysis on the patterns of cipher text. Extending the pre-computation algorithm for ICM by using these techniques in the present study is comparatively new. The factor base, which is the key parameter in the pre-computation step of ICM consists of first-t primes less than the bound B and this is considered as optimal way of selecting the factor base elements in the literature. These factor base elements can be considered as

dependent parameters instead of merely considering independent variables. Further, these variables can be reduced into minimal set using popular dimensionality reduction techniques. This may leads to reduce the number of relations required to form the linear relation and the overall running time of ICM.

The definition of B-smooth number over Z_p^* is studied with emphasis on the order of the elements in the factor base and new techniques are proposed for some instances of primes, such as $p = 2\rho + 1$. These types of primes are considered as Type1 and Type2 problems in the present study. The same concept can be extended for other types of primes, such as Type3, Type4 and Type5 problems. New techniques and comparative analysis can be made on the above type of problems with emphasis on the techniques to solve the DLP. Also, the probability of a randomly chosen prime to be of the required form such as $= 2\rho+1$ and other forms can be obtained by using the statistical analysis. This is useful for the estimation of the security of the cryptosystems based on the DLP.

A special case of smooth numbers over Z_p^* is introduced in this thesis. Equivalence classes are derived from Z_p^* by using the properties of the above smoothness concept. A new algorithm for ICM is proposed using the equivalence classes. The equivalence classes and the mapping function defined in the present study can be used to define a one way function, which may be harder to solve. A detail analysis is needed to define such a one way function and should satisfy the properties reported in the standards. Once such function is defined, a secure public key cryptosystem also can be devised based on this hard problem. The security of this cryptosystem is to be analyzed based on a attack model.

The properties of special case of smooth numbers over Z_p^* combined with the patterns identified in this present study can be used to solve the DLP in reduced time. Further, the patterns can be used when the generator is not known and a generator and the logarithm of a small subgroup of order say x is leaked and the order of y and g are $2x$. This may leads to transform the generator of order x to $2x$ and solves the DLP of y in reduced time. If the factors of $p-1$ are small, then the first-t generators can be stored as a factor base and the factor base may be extended by constructing equivalence classes from the elements in the factor base. The DLP of first-t generators can be solved and the DLP of y can be solved using

the pre-computed DLP of first-t generators.

The Random method, which is a primitive method of ICM is analyzed from two different perspectives. The approach can be extended to other methods, such as linear sieve, cubic sieve and number field sieve methods. Estimation of smooth numbers in a given range using the patterns identified in this thesis is useful to estimate the probability of numbers getting smoothing. This will leads to device an efficient algorithm for solving the DLP. Finally, the proposed cryptosystem based on the DLP $\alpha^a\beta^b$ in the present study offers two levels of security and this system can be analyzed further for the suitability of proxy re-encryption proxy-re signature., to name a few.

7.3 Contributions

We conclude by listing the original contributions made in the thesis.

- A new approach to improve the performance of ICM is developed. This leads to an improved algorithm for the generation step of ICM and new results are reported on the parameters used to generate and solve the relations. The performance of ICM is improved by 30 - 40 % by using this new approach.

- A new idea, partial linear sieve, is introduced and used to improve the performance of one of the primitive method of ICM known as Random method. It is shown that the performance is enhanced by more than 50% for problems of size $\approx$ 20 digits.

- A new concept of B-smooth numbers over Z_p^* is defined in the present work and a detailed analysis on the distribution is presented. Through these properties, the DLP is solved in reduced cost, in particular on safe primes and prime order subgroups.

- The characteristics of smooth numbers over Z_p^* lead to an improved Random method for pre-computation step of ICM and aid in improving the probability of getting smooth numbers. We achieved $\approx$ 50% of improvement in the performance of ICM.

- A special case of smooth numbers over Z_p^* is defined and the properties are analyzed on different types of primes. A variant of ICM is proposed to recover the ephemeral keys based on the properties of smooth numbers over Z_p^* and the generators of Z_p^*.
- At more preliminary level the ECDLP and DLP $\alpha^a\beta^b$ are analyzed.

Chapter 8

Publications

1. Discrete Logarithm Problem over Elliptic group using Pollard Rho, *International Journal on Computing and Mathematical Applications*, v3(4), pp(81-92), (2008).

2. Performance Analysis on Index Calculus Method , *Journal of Discrete Mathematical Sciences and Cryptography*, (Accepted).

3. Improved Cryptanalysis on DLP $\alpha^a\beta^b$, *International Journal of Computer Sciences and Engineering Systems*, v4(4), (2009).

4. Ephemeral Key Recovery using Index Calculus Method *Journal of Discrete Mathematical Sciences and Cryptography*, (under review).

5. Improved Random method using Smooth numbers over Z_p^* *International Journal of Applied Mathematics and Computing*, (Communicated).

6. Improved Random method of Index Calculus Method, *National Workshop on Cryptology 2008*.

7. Methods to solve Discrete Logarithm Problem for Ephemeral Key, *Advanced Research in Computation and Communication 2009*, (IEEE press), (Accepted).

8. Ephemeral Key Recovery attack on Chang and Chang Key Exchange Protocol, *International Conference on Computer and Network Technology, 2009* (World Scientific press), (Accepted).

Bibliography

[1] Abdalla, M., Bellare, M., Rogaway, P. "The oracle diffie-hellman assumptions and an analysis of DHIES", LNCS, v2020, pp143-158, (2001).

[2] Abhijit, D., Veni Madhavan C. E. "On the cubic sieve method for computing discrete logarithms over prime fields", *International Journal of Computer Mathematics*, v82(12), pp1481-1495, (2005).

[3] Adleman, L. M., Huang, M. D. A. "Function field sieve method for discrete logarithms over finite fields", *Information and computation*, v151(1-2), pp5-16, (1999).

[4] Anderson, R., Vaudenay, S. "Minding your p's and q's" *Asiacrypt'96*, LNCS, v1163, pp26-35, (1996).

[5] Avanzi R.M., Cohen, H., Doche, C., Frey, G., Lange, T., Nguyen, K., Vercauteren, F. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC Press, (2005).

[6] Bellare, M., Desai, A., Pointcheval, D., Rogaway, P. "Relations among Notions of Security for Public-Key Encryption Schemes", *Crypto'98*, LNCS, v(1462), pp549-570, (1998).

[7] Bellare, M., Rogaway, P. "Optimal Asymmetric Encryption – How to encrypt with RSA", *Eurocrypt'94*, LNCS, v950, pp1-19, (1995).

[8] Berstein, D. J. "Arbitrarily tight bounds on the distribution of smooth integers", *Number theory for the Millennium I*, pp 49-66, (2002).

[9] Berstein, D. J. "How to find smooth parts of integers", http://cr.yp.to/papers.html, (2004).

[10] Berstein, D. J. *'Enumerating and Counting smooth integers'*, ph.D Thesis, Department of Mathematics, University of California, (1995).

[11] Biham, E., Shamir, A. *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, (1993).

[12] Biham, E., Shamir, A. "Differential Cryptanalysis of DES-like Cryptosystems", *Crypto'90*, LNCS, v537, pp2-21, (1991).

[13] Biham, E., Shamir, A. "Differential Cryptanalysis of the Full 16-Round DES", *Crypto'92*, LNCS, v740, pp487-496, (1993).

[14] Brickell, E. F., Moore J. H. "Some remarks on the Herlestam-Johannesson algorithm for computing logarithms over $GF(2^n)$", *Crypto'82*, pp15-25, (1983).

[15] Boneh, D. "The Decision Diffie-Hellman Problem", *ANTS 1998*, LNCS, v1423, pp4863, (1998).

[16] Boneh, D., Joux, A., Nguyen, P. Q. "Why Textbook ElGamal and RSA Encryption Are Insecure", LNCS, v1976, pp30-43, (2000).

[17] Buchmann, J. A., Hollonger, C. S. "On smooth ideals in number fields", *Journal of Number Theory* v59(1), pp82-87, (1996).

[18] Buchmann, J., Weber, D "Discrete logarithms: Recent Progress", Technical report, no:T1-12/98.

[19] Certicom elliptic curve challenge, (http://www.certicom.com).

[20] Chang, C. C and Chang, Y. F, A novel three party encrypted key exchange protocol, *Computer Standards and Interfaces*, v26(5), pp471-476, (2004).

[21] Coppersmith, D. "The Data Encryption Standard (DES) and its strength against attacks", *IBM Journal of Research and Development* , v38(3), pp243-250, (1994).

[22] Coppersmith, D. "Fast evaluation of logarithms in fields of characteristic two", *IEEE Transaction on Information Theory*, v30(4), pp587-594, (1984).

[23] Coppersmith, D. "Solving linear equations over GF(2):block Lanczos algorithm", *Linear Algebra Application*, v192, pp33-60, (1993).

[24] Coppersmith, D. "Solving homogeneous linear equations over GF(2) via block-Wiedemann algorithm", *Mathematics of Computation*, v62(205), pp333-350, (1994).

[25] Coppersmith, D., Odlyzko, A., Schroeppel, R. "Discrete logarithms in GF(p)", *algorithmica 1*, pp1-15, (1986).

[26] Cormen, Thomas, H., Charles, E., Leiserson Ronald, L., Rivest., Clifford Stein. *Introduction to Algorithms*, MIT Press and McGraw-Hill, (2001).

[27] Cramer, R., Shoup, V. "A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack", *Crypto'98*, LNCS, v1462, pp13-25, (1998).

[28] Diffie, W. "The First Ten Years of Public-Key Cryptography", *Proceedings of the IEEE*, v76(5), pp560-577, (1988).

[29] Diffie, W., Hellman, M. E. "New Directions in Cryptography", *IEEE Transactions on Information Theory*, v22(6), pp644-654, (1976).

[30] ElGamal, T. "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *IEEE Transactions on Information Theory*, v31(4), pp469472, (1985).

[31] Escott, A., Sagar, J., Selkirk, A., Tsapakidis, D. "Attacking elliptic curve cryptosystems using the parallel Pollard-Rho method", *CrptoBytes - The Technical Newsletter of RSA Laboratories*, v4(2), pp15-19, (1999).

[32] Frey, G., Ruck, H. "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", *Mathematics of Computation* v62(206), pp865-874, (1994).

[33] Gallent, R., Lambert, R., Vanstone, Scott. "Improving the paralyzed Pollard Lambda search on Binary Anomalous Curve", *Mathematics of Computation*, v69, pp1699-1705, 1998.

[34] Gordon, D. M. "Discrete logarithms in GF(p) using the number field sieve", *SIAM Journal of Discrete Mathematics*, v6(1), pp124-138, (1993).

[35] Gordon, D. M., McCurley, K. "Massively parallel computation of discrete logarithms", *Crypto'92*, LNCS, v740, pp312-323, (1992).

[36] Hellman, M. E. "An Overview of Public Key Cryptography", *IEEE Communications Magazine*, v16(6), pp42-49, (2002).

[37] Hellman, M. E., Diffie, B. W., Merkle, R. C. "Cryptographic apparatus and method U.S. Patent #4,200,770", (1980).

[38] Herlestam, T., Johannesson, "On computing logarithms over $GF(2^p)$", *BIT Numerical Mathematics*, v21(3), pp326-334, (1981).

[39] Hildebrand, A., Tenenbaum, G. "On integers free of large prime factors", *Transactions of the American Mathematical Society*, v296(1), pp265-290, (1986).

[40] Hildebrand, A., Tenenbaum, G. "Integers without large prime factors", *J. Theor. Nombres Bordeaux*, v5(2), pp411-484, (1993).

[41] S. Hunter and J. Sorenson, "Approximating the number of integers free of large prime factors", *Mathematics of Computation*, v66(220), pp1729-1741, (1997).

[42] Jacobson, M., Stein, A., Menezes, A. "Solving elliptic curve discrete logarithm problems using Weil descent" *Journal of the Ramanujam Mathematical Society*, v16, pp231-260, (2001).

[43] Kashyap, S. K., Sharma, B. K., Banerjee, "A Cryptosystem Based on DLP $\gamma = \alpha^a \beta^b$", *International Journal of Network Security*, v3(1), pp95-100, (2006).

[44] Knuth, D. E. *The Art of computer programming*, vol.3: Sorting and Searching, Addison-Wesley, (1973).

[45] Koblitz, N. *A course on number theory and cryptography*, Springer-Verlag, Newyork, (1987).

[46] Koblitz, N. "Elliptic Curve Cryptosystems", *Mathematics of Computation*, v48, pp203-209, (1987).

[47] Koblitz, N., Menezes, A. J., Vanstone, S. "The state of elliptic curve cryptography", *Designs, Codes, and Cryptography*, v13(2-3), pp173-193, (2000).

[48] LaMacchia, B. A., Odlyzko, A. M. "Solving large linear systems over finite fields", *Crypto'90*, LNCS, v537, pp109-133, (1991).

[49] LaMacchia, B. A., Odlyzko, A. M. "Computation of discrete logarithms in prime fields", *Designs, Codes, and Cryptography*, v1(1), pp47-62,(1991).

[50] Lambert, R. "Computational aspects of discrete logarithms", Ph.D. thesis, Dept. Electrical Comp. Eng., Univ of Waterloo, (1996).

[51] Lenstra, A. K. "Integer factoring", *Designs, Codes, and Cryptography*, v19(2-3), pp101-128, (2000).

[52] Lenstra, A.K., Verheul, E.R, "Selecting Cryptographic Keysizes", *PKC*, LNCS, v1751, pp446-465, (2000).

[53] Lenstra, A. K., Verheul, E, R, "The XTR public key system", *Asiacrypt'00*, LNCS, v1880, pp1-19, (2000).

[54] Lenstra, A. K., Verheul, E, R, "Key Improvements to XTR", LNCS, v1976, pp220-233, (2000).

[55] Levy, S. *Crypto: How the Code Rebels Beat the Government Saving Privacy in the Digital Age*, Penguin Books, (2001).

[56] Lim, C. H., Lee, P. J. "A key recovery attack on discrete log-based schemes using Prime order Subgroup", *Crypto'97*, LNCS, v1294, pp249-263, (1997).

[57] Matsui, M., Yamagishi, A. "A new method for known plaintext attack of FEAL cipher", *Eurocrypt'92*, LNCS, v658, pp81-91, (1993).

[58] Matsui, M. "Linear cryptanalysis method for DES cipher", *Eurocrypt'93*, LNCS, v765, pp386-397, (1993).

[59] Matsui, M. "The first experimental cryptanalysis of the data encryption standard". *Crypto'94*, LNCS, v839, pp1-11, (1994).

[60] McCurley, K. S. "The discrete logarithm problem, *Symposium of Applied Mathematics*, v42, pp49-74, (1990).

[61] Menezes, A., Berkant, U. "On Reusing Ephemeral Keys in Diffie-Hellman Key Agreement Protocols", www.cacr.math.uwaterloo.ca/ãjmeneze/publications/ephemeral.pdf, (preprint), (2008).

[62] Menezes, A., Okamoto, t., Vanstone, S. "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Information Theory*, v39(5), pp1639-1646, (1993).

[63] Menezes, A., Paul, C., Oorschot, V., Scott, A., Vanstone, *Handbook of Applied Cryptography*, CRC Press, (2001).

[64] Menezes, A., Teske, E., Weng, A. "Week fields for ECC", LNCS, v2964, pp366-386, (2004).

[65] Miller, V. "Uses of Elliptic curves in cryptography", *Crypto 85*, LNCS, v218, pp417-426, (1986).

[66] Montomery, P. L. "A block Lanczos algorithm for finding dependencies over GF(2)", *Eurocrypt'95*, LNCS, v921, pp106-120, (1995).

[67] Odlyzko, A. M. "Discrete logarithms in finite fields and their cryptographic significance", *Eurocrypt'84*, LNCS, V209, pp224-314, (1985).

[68] Odlyzko, A. M. "Discrete logarithms:The past and the future", *Designs codes and cryptography*, v19(2-3), pp129-145, (2000).

[69] Odlyzko, A. M. "On the complexity of Computing Discrete Logarithms and Factoring Integers", *Open problems in communication and computation*, pp113-116, (1987).

[70] Panario, D., Gourdon, X., Flajoet, P. "An analytic approach to smooth polynomials over finite fields", *Algorithmic Number Theory: ANTS-III*, Lecture Notes in Math, v1423, pp226-236, (1998).

[71] Pohlig, S., Hellman, M. "An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance", *IEEE Transactions on Information Theory* v24(1), pp106110, (1978).

[72] Pollard, J. M. "Kangaroos, Monopoly and discrete logarithms", *Journal of Cryptology*, v13(4), pp437-447, (2000).

[73] Pollard, J. "Monte Carlo methods for index computation mod p", *Mathematics of Computation*, v32(143), pp918-924, (1978).

[74] Rivest, R., Shamir, A., Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". *Communications of the ACM*, v21(2), pp120-126, (1978).

[75] Satoh, T., Araki, K. "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", *Commentarii Mathematici Universitatis Sancti Pauli*, v47, pp81-92, (1998).

[76] Schirokauer, O., "Discrete logarithms and local units", *Philosophical Transactions of the Royal Society*, v345(1676), pp409-423, (1993).

[77] Schirokauer, O. "Using number fields to compute logarithms in finite fields", *Mathematics of computation*, v69(231), pp1267-1283,(1999).

[78] Schirokuer, O., Weber, D., Denny, T., "Discrete logarithms: The effectiveness of the index calculus method", *algorithmic Number theory: ANTS-II*, Lecture Notes in Math, v1122, pp337-362 (1996).

[79] Schneir, B. *Applied Cryptography*, 2nd ed., wiley, (1995).

[80] Schnorr, C. P. "Efficient signature generation by smart cards", *Journal of Cryptology*, v4(3), pp161-174, (1991).

[81] Semaev, I. "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", *Mathematics of Computation*, v67(221), pp353-356, (1998).

[82] Semaev, I. A. "A generalization of the number field sieve". *Probabilistic Methods in Discrete Mathematics*, pp45-63, (1997).

[83] Semaev, I. A. "An algorithm for evaluation of discrete logarithms in some nonprime finite fields", *Mathematics of Computation*, v67(224), pp1679-1689, (1998).

[84] Semaev, I. A. "Special prime numbers and discrete logs in prime finite fields", *Mathematics of Computation*, v71(237), pp362-377, (2002).

[85] Shanks, D. "Class number, a theory of factorization and genera", *Symposium of Pure Mathematics*, v20, pp415-440, (1971).

[86] Singh, S. *The Code Book: the evolution of secrecy from Mary Queen of Scots to quantum cryptography*, New York: Doubleday, (1999).

[87] Sramka, M "Cryptanalysis of the Cryptosystem Based on DLP $\gamma = \alpha^a \beta^b$", *International Journal of Network Security*, v6(1), pp80-81, (2008).

[88] Stein, A., Teske, E., "Optimized baby step-giant step methods", *Journal of the Ramanujam Mathematical Society* v20(1), pp132, (2005).

[89] Studholme, C. "Discrete logarithm problem", Research paper requirement (milestone) of the PhD program at the University of Toronto, June 21, (2002).

[90] Sutherland, A. V. *Order computations in generic groups*, PhD thesis, Department of Mathematics, M.I.T, (2007).

[91] Terr, D. C. "A modification of Shanks baby-step giant-step algorithm", *Mathematics of Computation*, v69(230), pp767773, (2000).

[92] Teske, E. "Speeding up Pollard's rho method for computing discrete logarithms", pp. 541-554 *Algorithmic Number theory: ANTS-III*, Lecture Notes in Math, v1423, pp541-554, (1998).

[93] Teske, E. "On random walks for Pollards rho method", *Mathematics of Computation*, v70(234), pp809-825, (2000).

[94] Teske, E. "An elliptic curve trapdoor system", Cryptology ePrint Archive Report 2003/058, (2003).

[95] http:www.hicj.net/files/498/termpaper.pdf.

[96] Vaudenay, S. "The Security of Cryptographic Primitives", Technical Report LIENS-95-10, (1995).

[97] Van Oorschot, P.C., Wiener, M. J. "On Diffie-Hellman Key agreement with short Exponents", *Eurocrypt'96*, LNCS, v1070, pp332-343, (1996).

[98] Van Oorschot, P. C., Wiener, M. J. "Parallel collision search with cryptanalytic applications", *Journal of Cryptology*, v12(1), pp1-28 (1999).

[99] Weber, D. "Computing discrete logarithms with quadratic number rings". *Eurocrypt'98*, LNCS, v1403, pp171-183 (1998).

[100] Weber, D., Denny, T. "The solution of McCurleys discrete log challenge", *Crypto'98*, LNCS, v1462, pp458-471, (1998).

[101] Wells, A. L. Jr. "A polynomial form for logarithms modulo a prime", *IEEE Transaction on Information Theory*, v30(6), pp(845-846), (1984).

[102] Wiedemann, D. H. "Solving sparse linear equations over finite fields", *IEEE Transaction on Infomation Theory*, v32(1), pp54-62 (1986).