Volume 2, No. 4, July-August 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Study and Analysis of Hard Problem in Elliptic Curve Cryptography

Amita Rathee
Master of Technology (Computer Engineering)
P.D.M. College of Engineering
Bahadurgarh, Haryana, India.
amita_rathee12@yahoo.com

Abstract: Hard Problems are the problems infeasible to solve in some sense. In cryptography the difficulty deals with computational requirements in finding solution. In this paper we will discuss how hard problems are base of cryptography. We are going to study the hard problems in Elliptic Curve Cryptography. What we going to discuss is which hard problem makes Elliptic Curve Cryptography more infeasible to break in the existing hard problems. Most suitable hard problem for Elliptic Curve Cryptography is Elliptic Curve Discrete Logarithm Problem.

Keywords: Hard Problem, Elliptic Curve Cryptography, Elliptic Curve Discrete Logarithm Problem, P Class, NP Class.

I. INTRODUCTION

Hard Problems are the problems infeasible to solve and used in cryptography to maintain the difficulty that deals with computational requirements in finding solution so as to ensure the security. Some of hard problems are Factoring, Theorem Proving, Travelling Salesman, and Discrete Logarithm Problem.

In this paper we are going to study and analyze some of major hard problems those are used in public key cryptosystems. We are going to compare them and analyze which existing hard problem best suits Elliptic Curve Cryptography and make it much more secure. It is not that we can use any hard problem for security of any cryptographic system. In this paper I am going to study and analyze the existing major hard problems and the most suitable one for Elliptic Curve Cryptography.

The structure of remaining paper is as follows: Section 2 provides an overview to the existing hard problems in Elliptic Curve Cryptography. Section 3 introduces the study of Elliptic Curve Discrete Logarithm Problem.

II. LITERATURE SURVEY

All cryptographic systems are based on cryptographic algorithms which are based on hard problems to maintain the security. The two basic and famous hard problems are Factoring Big Number Problem and Discrete Logarithm Problem [1].

Factoring Big Number Problem (FBNP) states that given a big number it is computationally infeasible to factorize number into prime factors.

Discrete Logarithm Problem (DLP) states that given a big number N and y, g in interval [1, …, N] where gcd(g, N) = 1. It is hard to find x that satisfies $y = g^x \pmod N$.

The best known protocol that employs the hardness of DLP is Diffie Hellman Key Exchange. Elliptic Curves over finite fields contain finite cyclic groups that we can use for cryptography. There is no factorization problem for elliptic curves but what is used is discrete logarithm problem.

It has been proved by Aaron Blumenfeld [3] in his paper "Discrete Logarithms on Elliptic Curves" published in 2010 that hardness of discrete logarithm problem on elliptic curves has offered an advance in cryptography in cryptography and shown computational evidence those suggested that it is more secure then classical techniques. It is assumed to be secure because of the belief that discrete exponentiation behaves like random map. Lots of Mathematicians have given their best efforts to break the Elliptic Curve Discrete Logarithm Problem but till now the best known attack on ECDLP takes exponential time whereas sub-exponential time algorithms do exists to solve Factoring Big Number Problem [2].

For most cryptography algorithm it is impossible to prove that they are secure. But there are certain algorithms those can help to ensure a level of security. However, even hard problem in mathematics are broken every once in a while. The only true test of an algorithm is time. The best algorithms are those that have been published for entire world to see and have stood the test of time. If an algorithm has been published for a while and still no one has been able to break the algorithm in practical manor, algorithm is assumed to be secure.

III. COMPARISON OF HARD PROBLEMS

Generally the line between tractable and intractable has been taken to be polynomial or exponential line. That means a problem is considered tractable if the running time of an algorithm to solve it is $O(n^k)$ for some constant k, and a

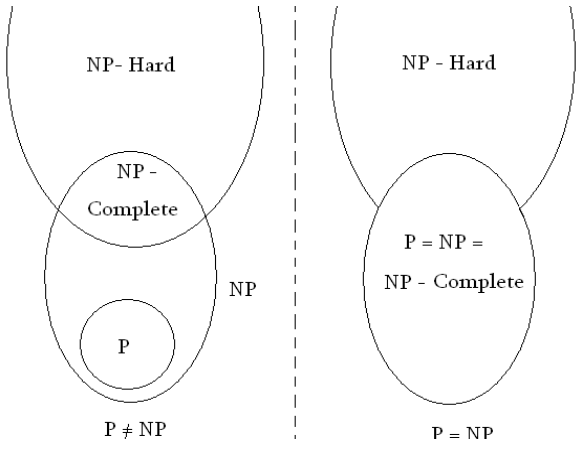

Figure: 1

problem is considered intractable if it cannot be bound by such a limit. Though there are classes between polynomial and exponential by far the most commonly discussed super polynomial bound is exponential time.

There are two basic classes of problems: P class and NP class. P class problems are those which can be solved in polynomial time whereas NP problems are those problems in which proposed solution can be checked in polynomial time.

Different types of intractable problems are: Provably intractable problems, presumably intractable problems, and Conjectured intractable problems [5].

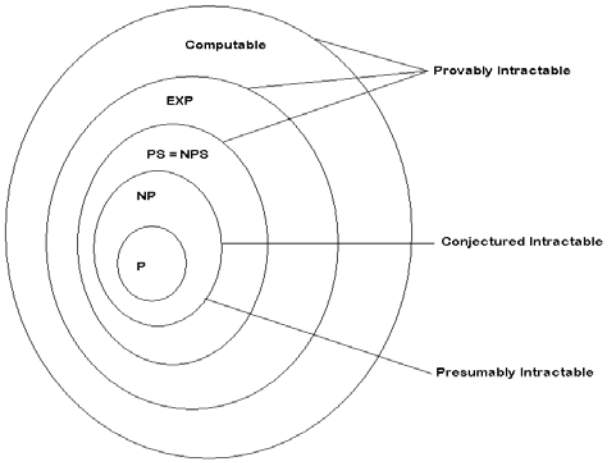

Figure: 2

Provably Intractable Problems are Turing computable and can be in PS (P Space), NPS (NP Space) and EXP (Exponential Time). Presumably Intractable problems are the problems in NP but outside P. These are NP – Complete such as Travelling salesman, Knapsack etc. Conjectured Problems are those problems which are in NP – complete But they may not always be NP-complete. They may become P problems if efficient algorithm would be implemented. Some of such problems are Factoring Problem, DLP, and ECDLP etc.

The security of modern public key cryptography is based on the intractability of hard problems. Any Problem beyond P class is intractable.

A. Factoring Big Number Problem

The security of some of public key cryptosystems is based on hardness of finding the prime factor f of a large number N that lies in between 1 to N. This is Factoring Big Number Problem (FBNP). This problem has been proven to be hard but lots of mathematicians are working on the validation of hardness of FBNP.

Factoring has become easier over last decade because the new systems are much powerful, inexpensive and better factoring algorithm has emerged. So we need to increase the key size as the demand for security increases. Better factoring algorithms help attacker to harm the system. Still factoring is hard problem. But have lot of challenges in front. Increasing the key size have kept pace in algorithm efficiency, resulting in no net loss of security. But this raise the demand of increasing key size tremendously but that will slow down the speed and also would increase the computational overhead. Those can cause the system slowdown.

FBNP has been believed that it is hard problem but have not yet proved. So there may be some algorithm that can solve it. All problems those haven't been proved to be hard have some solution. The thing is that till now no one could find out the solution to prove that they are not hard. But that maintain the security till solution for the problem would not be found.

B. Elliptic Curve Discrete Logarithm Problem

The Security of Elliptic Curve Cryptosystems is based on computation of k, given a curve E defined over F_q, point $P \in E(F_P)$ of order n and Q such that $Q = kP$ so $k = \log_P(Q)$ which is hard to compute. This is Elliptic Curve Discrete Logarithm Problem.

ECDLP is almost same as basic DLP but it is over the elliptic curves. ECDLP have the same issues as with DLP. The hardness of DLP on Elliptic Curves enhance the security, the reason is that discrete exponentiation behaves like random map. The most general attack on ECDLP can be Exhaustive Search. But it takes exponential time and low velocity. Another attack can be Pohlig- Hellman and Pollard's rho that have exponential running time.

For DLP there no polynomial time algorithm but some sub-exponential algorithms is there, but ECDLP has no sub-exponential algorithm that makes it most secure hard problem [5].

IV. STRONGEST HARD PROBLEM

The hard problems FBNP and ECDLP can be solved in future if a quantum computer would be developed. Peter Shor, a mathematician, gave a quantum algorithm in 1994 that can solve FBNP in polynomial time on a quantum computer. In 2001, Shor's algorithm was demonstrated by a group at IBM using NMR(Nuclear Magnetic Resonance) implementation of a quantum computer with 7 qubits. However it was not believed to be a true demonstration.

We have studied both the problems FBNP and ECDLP. The Pollard-Strassen method has been proved to be deterministic algorithm for FBNP. But the Most prominent algorithm is the Number Field Sieve(NFS) which has been used to factor numbers up to 576 bits long. But its PC-based implementation cannot do much better. So number of algorithms has been proposed. But it also has been proposed that if we use custom hardware with sieving process, that is advanced step of NFS, it would be able to tract factors of 1024 bit long integers. This is why currently existing systems using RSA need to increase the size of the key [4].

ECDLP is DLP on Elliptic curves. The main advantage of ECC is that for suitably chosen curves there is no known sub-exponential algorithm, like Number Field Sieve (NFS) algorithm for integer factorization, to solve ECDLP [7].

Based on the study done by jithra Adhikari [8], an instance of IFP is an integer n which is a product of two l/2-bit primes. The number field sieve (NFS) which is the fastest known algorithm for factoring n has a sub-exponential expected running time of L_n [1/3, $(64/9)^{1/3}$ for O(l)-bit input [24]. An instance of DLP has parameters p (l-bit prime) and q (t-bit prime divisor of p-1). We observe sub-exponential expected running time L_n [1/3, $(64/9)^{1/3}$ for O(l)-bit input to solve DLP with NFS and an expected running time $(\pi q/2)^{1/2}$ for Pollard's rho algorithm. The choice of method for solving DLP depends on the sizes of parameters p and q. Generally, they are selected such that L_n [1/3, $(64/9)^{1/3}$] and $(\pi q/2)^{1/2}$ are equal.

Assuming both NFS and Pollard's rho algorithm are the fastest algorithms and take same expected running time to solve DLP, we observe that NFS sub-exponential expected running time is L_n [1/3, $(64/9)^{1/3}$] for O(l)-bit input and

expected running time of Pollard's rho algorithm $(\pi q/2)^{1/2}$ are equal.

Next we consider ECDLP. n is a t-bit prime and P is a point of order n on an elliptic curve defined over Fp. Assuming that $n \approx p$ (which is the general case) and that the fastest algorithm to solve ECDLP is Pollard's rho algorithm, we have an expected running time of $\sqrt{\pi n} / 2$.

Based on these observations, NIST has concluded that the elliptic curve discrete logarithm problem is the hardest to break among the known intractable problems.

V. CONCLUSION

In this paper we studied what hard problems are? And the two most famous hard problems: Factoring Big Number Problem and Elliptic Curve Discrete Logarithm Problem. We briefly introduce the basics of Elliptic Curve Cryptography. We have studied that Elliptic Curve Discrete Logarithm Problem has no sub-exponential algorithm that is what makes it secure. We also analyzed that it is important to consider right domain parameters for Elliptic Curve Discrete Logarithm Problem to avoid all attacks. We also have discussed basic classification of the problems. We also have studied the comparison between the two major problems separately. The improvement of security is due to higher complexity of solving Discrete Logarithm Problem over Elliptic Curve. This is applicable to all existing system needing a secure system.

VI. ACKNOWLEDGMENT

I would like to thank my guide Prof. Paramjit Singh for his helpful suggestions throughout this work of piece.

VII. REFERENCES

[1] H.L. Nguyen, "RSA Threshold Cryptography," Department of Computer Science, University, May 4, 2005.

[2] Eric Cole, Ronald Krutzl, James W. Conley, "Network Security Bible", Wiley Publishing Inc. USA, Canada.

[3] Blumenfeld, Aaron, "Discrete Logarithm on Elliptic Curves", Rose Hulman Institute of Technology, Univesity of Rochester, July 31, 2010.

[4] Kostas Bimpikis, Ragesh Jaiswal, "Modern Factoring Algorithms", University of California, San Diego

[5] Song Y. Yan, Glyn Jame, Gongyi Wu, "Polynomially Uncomputable Number- Theoretic Problems in Cryptography and network security",

[6] Song Y. Yan, "Cryptoanalytic attack on RSA", Springer, University of Bedfordshire, UK, Massachusetts Institute of Technology, USA, 2008.

[7] P. K. Mishra, " Introduction to Elliptic Curve Cryptography and Side Channel Attacks", Cryptography Simplified, July, 2009.

[8] Jithra Adikari, "Efficient Algorithms for Elliptic Curve Cryptography", University of Calgary, Alberta, January, 2011.

