ISSN:0975-9646

Application of Elliptic Curve Method in Cryptography: A Literature Review

Samta Gajbhiye,Dr. Sanjeev Karmakar,Dr. Monisha Sharma,Dr. Sanjay Sharma,Dr. M K Kowar
Shri Shankaracharya Group of Institutions
Junwani, Bhilai , Distt. Durg, Chattisgarh

Abstract- Application of Elliptic Curve Method (ECM) in cryptography popularly known as Elliptic Curve Cryptography (ECC) has been discussed in this paper. Finally the performance of ECC in security and moreover, its recent trends has been discussed.

Keywords: Elliptic Curve Cryptography(ECC), Integer Factorization Problem (IFP) , Finite Field Discrete Logarithm Problem(FFDLP) , Elliptic Curve Discrete Logarithm Problem(ECDLP)

1. INTRODUCTION

Elliptic Curve Method(ECM) was applied on cryptography known as ECC was discovered in 1985 by Victor Miller (IBM) and Neil Koblitz (University of Washington) as an alternative mechanism for implementing public-key cryptography (PKC) [Vm85]. Elliptic Curve Cryptography (ECC) has the special characteristic that to date, the best known algorithm that solves it runs in full exponential time. Its security comes from the Elliptic Curve Logarithm, which is the Discrete Logarithm Problem (DLP) in a group defined by points on an elliptic curve over a finite field. These results in a dramatic decrease in key size needed to achieve the same level of security offered in conventional public key cryptography schemes.

In constrained environments such as mobile phones, wireless pagers or personal digital assistant(PDA), the resources like bandwidth, memory and battery life are highly limited. Thus, a suitable public key scheme would be one that is efficient in terms of computing costs and key sizes[Dmd04]. To date, the ECC has the highest strength-per-bit compared to other public key cryptosystems. Small key sizes translate into savings in bandwidth, memory and processing power. This makes ECC the obvious choice in this situation.

Examinations of the different mathematical problems that underlie the majority of the public key cryptosystems in use recently along with the algorithms that are used to overcome them have been discussed in the following sections. This will give us a better understanding of the security on which different types of public key cryptosystems are based. An overview of comparisons in the performance of ECC with other PKC applications is provided followed by ECC applications in constrained devices, as well as in powerful computers.

Initially, we begin by introducing the three mathematical problems and the various algorithms that are used to overcome them followed by comparisons in the performance of ECC with other PKC applications.

2. HARDNESS OF MATHEMATICAL TECHNIQUES IN PUBLIC KEY CRYPTOGRAPHY

PKCs security essentially is based on the difficulty of solving an integer factorization problem (IFP). Nowadays, there are three shortcomings that are believed to be both secure and practical after years of intensive studying. They are the (1) IFP, (2) Finite Field Discrete Logarithm Problem (FFDLP) and the (3) Elliptic Curve Discrete Logarithm Problem (ECDLP). While this by no means proves that they are unbreakable, it is highly unlikely that anyone will find an efficient algorithm to solve them in the near future.

2.1. IFP

Multiplication is easy: Given *p* and *q*, it's easy to find their product, $n = pq$. There are many efficient ways to multiply two large numbers, starting with the "grade-school" method that multiplies one number by the other digit-by-digit, and sums the tableau of intermediate results.

Factoring is hard: Given such an *n*, it appears to be quite hard to recover the prime factors *p* and *q*.

Despite hundreds of years of study of the problem, finding the factors of a large number still takes a long time in general. The fastest current methods are much faster than the simple approach of trying all possible factors one at a time. (Such a method would take on the order of *n* steps.) However, they are still expensive. For instance, it has been estimated recently that recovering the prime factors of a 1024-bit number would take a year on a machine costing US $10 million. A 2048-bit number would require several billion times more work.

The general integer factorization problem is defined as follows. Given a positive integer n, write $n = p_1^{e_1} p_2^{e_2} p_3^{e_3} \ldots p_k^{e_k}$, where the p_i are pairwise distinct primes and each $e_i > 1$ [Mov97]

Factoring algorithms

A special-purpose factoring algorithm's running time depends on the properties of the number to be factored or on one of its unknown factors: size, special form, etc. Exactly what the running time depends on varies between algorithms. Trial division, Wheel factorization, Pollard's rho algorithm, Algebraic-group factorisation algorithms(among which are Pollard's $p - 1$ algorithm, Williams' $p + 1$ algorithm, and Lenstra elliptic curve factorization) , Fermat's factorization method, Euler's factorization method, Special number field sieve all these are special purpose factoring algorithms.

Kraitchik family algorithm, general purpose algorithms has a running time depends solely on the size of the integer to be factored. This is the type of algorithm used to factor RSA numbers. Most general-purpose factoring algorithms are based on the congruence of squares method. Dixon's algorithm, Continued fraction factorization (CFRAC), Quadratic sieve, General number field sieve, Shanks' square forms factorization (SQUFOF) belongs to the family of Kraitchik. Heuristically expected running time of these algorithms is $Ln\left[\frac{1}{2}, 1 + o(1)\right] = e^{(1+o(1))\,(\log n)^{\frac{1}{2}} (\log \log n)^{\frac{1}{2}}}$

The Schnorr-Seysen-Lenstra probabilistic algorithm has been rigorously proven by Lenstra and Pomerance to have expected running time $\boldsymbol{Ln\left[\frac{1}{2}, 1 + o(1)\right]}$ [Len92].

General number field sieve (GNFS) is the most efficient classical algorithm known for factoring integers larger than100digits. Heuristically, its complexity for factoring an integer n (consisting of $\log_2 n$ bits) is of the form

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}} + o(1)\right)(\log n)^{\frac{1}{3}}(\log \log n)^{\frac{2}{3}}\right) = Ln[\frac{1}{3}, \sqrt[3]{\frac{64}{9}}]$$

On a quantum computer, to factor an integer N, Shor's algorithm runs in polynomial time .The time taken is polynomial in $\log N$, which is the size of the input. Specifically it takes time $\underline{O}((\log N)^3)$, demonstrating that the integer factorization problem can be efficiently solved on a quantum computer. This is exponentially faster than the most efficient known classical factoring algorithm, the general number field sieve, which works in sub-exponential time—about $O(e^{1.9 (\log N)1/3 (\log \log N)2/3})$. [Shp97, Shp98]

Typically, in practical cryptographic applications, only two factors are used for the modulus n. A larger number of factors for n does not seem to offer any additional security in the IFP. The best-known public key cryptosystem that bases its security on the difficulty of the IFP is RSA. Named after its inventors: Ron Rivest, Adi Shamir and Len Adleman who developed it at MIT in 1978, it was the first practical implementation of public key cryptography since the introduction of the concept[Rsa79]. Another example is the Rabin-Williams cryptosystem. It is similar to RSA, but it uses an even public exponent .

Two of the most extensively used factoring algorithms today are the quadratic sieve (QS) and number field sieve (NFS). They are both based on the idea of finding a factor base of primes to generate a system of linear equations[Mov97].

2.2 FFDLP

In mathematics, specifically in abstract algebra and its applications, discrete logarithms are group-theoretic analogues of ordinary logarithms. In particular, an ordinary logarithm $\log_a(b)$ is a solution of the equation $a^x = b$ over the real or complex numbers. Similarly, if g and h are elements of a finite cyclic group G then a solution x of the equation $g^x = h$ is called a discrete logarithm to the base g of h in the group G.

Discrete logarithms are perhaps simplest to understand in the group $\underline{(\mathbf{Z}_p)^\times}$. This is the set $\{1, \ldots, p-1\}$ of congruence classes under multiplication modulo the prime p.

Discrete exponentiation is finding kth power as an integer of the group and then finding the remainder after division by p. For example, consider $(\mathbf{Z}_{17})^\times$. To compute 3^4 in this group, we first compute $3^4 = 81$, and then we divide 81 by 17, obtaining a remainder of 13. Thus $3^4 = 13$ in the group $(\mathbf{Z}_{17})^\times$.

Discrete logarithm is just the inverse operation. For example, take the equation $3^k \equiv 13 \pmod{17}$ for k. As shown above k=4 is a solution, but it is not the only solution. Since $3^{16} \equiv 1 \pmod{17}$, it also follows that if n is an integer then $3^{4+16n} \equiv 13 \times 1^n \equiv 13 \pmod{17}$. Hence the equation has infinitely many solutions of the form $4 + 16n$. Moreover, since 16 is the smallest positive integer m satisfying $3^m \equiv 1 \pmod{17}$, i.e. 16 is the order of 3 in $(\mathbf{Z}_{17})^\times$, these are the only solutions. Equivalently, the solution can be expressed as $k \equiv 4 \pmod{16}$.[Wpd12]

More sophisticated algorithms exist like Baby-step giant-step (The running time of the algorithm and the space complexity is $\underline{O}(\sqrt{n})$), Pollard's rho algorithm(The running time is approximately $O(\sqrt{p})$ where p is n's smallest prime factor.[Plo78]), Pollard's kangaroo algorithm also known as Pollard's lambda algorithm (Time complexity of which is $O(\sqrt{b-a}) = o\,(e^{\frac{1}{2}\log(b-a)})$ is exponential in the problem size[Pka00]). For this reason, Pollard's lambda algorithm is considered an exponential time algorithm. The worst-case time complexity of the Pohlig–Hellman algorithm is $O(\sqrt{n})$ [Phe78]). Assuming an optimal selection of the factor base, Index calculus algorithm the expected running time of the index-calculus algorithm can be stated as $Ln\left[\frac{1}{2}, c\right]$ c >0 [Adl79].The running time of the number field sieve is super-polynomial but sub-exponential in the size of the input.[Len92, Len93]. ,Function field sieve inspired by similar algorithms for integer factorization. These algorithms run faster than the naive algorithm, but none of them runs in polynomial time (in the number of digits in the size of the group).

The origin of using the discrete logarithm problem in cryptographic schemes goes back to the seminal paper of Diffie and Hellman[Dh76]. It is there that they proposed the discrete logarithm problem as good source for a "one way function" . Other cryptographic applications that base their security on the intractability of the DLP include the ElGamal encryption scheme and the digital signature algorithm (DSA). DLP in a prime field is considered to be harder than DLP in fields of characteristic two. The current record for computing discrete logarithms in GF(p) is a 120-digit prime p[Kap99]

The most powerful algorithm known for computing the DLP is the index calculus method. It is a probabilistic algorithm that applies only to finite fields. Examples of finite fields that are commonly used in practical applications are GF(p) and GF($2m$). The index-calculus method is currently the only known algorithm that solves the DLP in sub-exponential time, making it the champion of all DL algorithms. All the other Algorithms that solve the DLP for arbitrary groups run in full exponential time.

There exist groups for which computing discrete logarithms is apparently difficult. In some cases (e.g. large prime order subgroups of groups $(\mathbf{Z}_p)^\times$) there is not only no efficient algorithm known for the worst case, but the average-case complexity can be shown to be about as hard as the worst case using random self-reducibility. Popular choices for the group *G* in discrete logarithm cryptography are the cyclic groups $(\mathbf{Z}_p)^\times$; Newer cryptography applications use discrete logarithms in cyclic subgroups of elliptic curves over finite fields[Tel85]

2.3 ECDLP

In 1985, Neal Koblitz and Victor Miller independently proposed the concept of ECC[Kob87,Mv85]. Other work on the security and implementation of elliptic curve cryptosystems (ECC) was reported in Menezes, Okamoto and Vanstone [Mov93, Asv93]. It is based on the DLP in a group defined by points on an elliptic curve over a finite field. The discrete logarithm problem has been adapted to elliptic curves in the hopes of providing even more security[Ame93]. The basic idea is that, for any prime *p*, there is only one field, F*p*. For elliptic curves, however, the number of possible elliptic curves over F*p* is extremely large, even for small values of *p* [Mov93]. The security of Elliptic Curve Cryptosystems relies on the difficulty of the ECDLP[Asv93]. The ECDLP is elucidated as follows:

Let E (Fp) denote an elliptic curve taking values in finite field Fp, and $B \in E$ (Fp) denote a point on the curve E. Then, given the additive structure of the points, and $kB = B + B + \cdot\cdot\cdot + B$ (k times). ECDLP is defined as : Given a basepoint B, and elliptic curve E, and a point $P \in E$ (Fp) such that $P = kB$ .Calculate the value of k. While it is customary to use additive notation to describe an elliptic curve group, some insight is provided by using multiplicative notation. Specifically, consider the operation called "scalar multiplication" under additive notation: that is, computing kB by adding together k copies of the point B. Using multiplicative notation, this operation consists of multiplying together k copies of the point B, yielding the point Bk.

In the multiplicative group Zp*, the discrete logarithm problem is: given elements r and q of the group, and a prime p, find a number k such that $r = q^k$ mod p. If the elliptic curve groups is described using multiplicative notation, then the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number that $P^k = Q$; k is called the discrete logarithm of Q to the base P. When the elliptic curve group is described using additive notation, the elliptic curve discrete logarithm problem is: given points P and Q in the group, find a number k such that Pk = Q [Mwp09]

The discrete logarithm problem is the basis for the security of many cryptosystems including the ECC. More specifically, the ECC relies upon the difficulty of the ECDLP. The ECDLP is based upon the intractability of scalar multiplication products. Implementations of ECC include elliptic curve analogs of DSA (ECDSA), ElGamal and Diffie-Hellman [Dah04].

ECDLP algorithms

The most attractive feature of ECC is that at present, the fastest known algorithm that solves it run in full exponential time. Despite the fact that index calculus methods can compute conventional logarithms in sub-exponential time, they cannot be applied to the case of discrete logarithms over elliptic curves. This is a claim made by Miller in his 1986 paper, which was later backed by theoretical study and computational experiments by J. H. Silverman and Suzuki in their paper published in 1998.

ECC security consists in the difficulty to calculate logarithms in discrete fields (discrete logarithms problem): being given *A* (an element from a finite field) and *Ax*, it is practically impossible to calculate *x* when *A* is big enough.[Kra09,Krb09]

For elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly-known base point is infeasible. The size of the elliptic curve determines the difficulty of the problem. The primary benefit promised by ECC is a smaller key size, reducing storage and transmission requirements[Ita09]—i.e., that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key[Rsh09] —e.g., a 256bit ECC public key should provide comparable security to a 3072bit RSA public key. The entire security of ECC depends on the ability to compute a point multiplication and the inability to compute the multiplicand given the original and product points.

Several discrete logarithm-based protocols have been adapted to elliptic curves, replacing the group Zp* with an elliptic curve like the elliptic curve Diffie–Hellman (ECDH) key agreement scheme is based on the Diffie–Hellman scheme, the Elliptic Curve Integrated Encryption Scheme (ECIES), also known as Elliptic Curve Augmented Encryption Scheme or simply the Elliptic Curve Encryption Scheme, the Elliptic Curve Digital Signature Algorithm (ECDSA) is based on the Digital Signature Algorithm, the ECMQV key agreement scheme is based on the MQV key agreement scheme. the ECQV implicit certificate scheme.

To date the best attack on ECCs are Pollard's ρ or λ method; both of which have expected exponential running times and hence are infeasible given today's technology[Cw97]. This suggests that elliptic curve cryptosystems are superior to currently deployed public key cryptosystems since not only do they offer a greater level of security when the underlying parameters are chosen correctly, but they offer a greater advantage due to factors including shorter key sizes, faster generation of systems, smaller space requirements and efficient implementation techniques. Elliptic curve cryptography is vulnerable to a modified Shor's algorithm for solving the discrete logarithm problem on elliptic curves

Weak curves: There are certain types of elliptic curves in which a successful attack could take place in sub-exponential time. If identified, these curves can easily be tested for and avoided. So far, several classes of curves have been identified and prohibited in all drafted standard specifications for public key cryptography such as IEEE P1363, ANSI X9.62 and ANSI X9.63 . Such curves are called the supersingular curves and anomalous curves[Cw97].

Due to work of Menezes, Okamoto and Vanstone, it is already known that one must avoid elliptic curves which are supersingular, these are the curves which have trace of frobenius equal to zero. Menezes, Okamoto and Vanstone reduce the discrete logarithm problem on supersingular elliptic curves to the discrete logarithm problem in a finite field. They hence reduce the problem to one which is known to have sub-exponential complexity.In practice the method described means that when choosing elliptic curves to use in cryptography one has to eliminate all curves whose group orders are equal to the order of the finite field , in other words curves for which the trace of Frobenius is equal to one[Cw97]. To solve the discrete log problem in a subgroup of order p of an elliptic curve over the finite field of characteristic p one needs O(ln p) operations in this field. When time is measured in terms of the number of basic group operations , the discrete logarithm problem on this curve runs in linear time that one must perform.[Mov93]

The other class of curves, the anomalous curves, allows an even more efficient attack when applicable. Proposed independently in 1998 by Satoh andAraki, Semaev, and the following year by Smart, this type of curves allow the ECDLP to be solved in polynomial time by reducing it to the classical DLP in an additive group GF(*p*) [Kap99]. Further readings can be found in [Sak98, Sia98, Smn99].

NIST-recommended elliptic curves

NIST recommends fifteen elliptic curves. Specifically, FIPS 186-3 has ten recommended finite fields:

- Five prime fields F_p for certain primes *p* of sizes 192, 224, 256, 384, and 521 bits. For each of the prime fields, one elliptic curve is recommended.
- Five binary fields F_2^m for *m* equal 163, 233, 283, 409, and 571. For each of the binary fields, one elliptic curve and one Koblitz curve is recommended.

The NIST recommendation thus contains a total of five prime curves and ten binary curves. The curves were chosen for optimal security and implementation efficiency[SEC00]

THE ECC ADVANTAGE

It is worthy to note that a 160-bit ECC key has about the same level of security as a 1024-bit RSA key. The most important difference between ECC and other conventional cryptosystems is that for a well-chosen curve, the best method currently known for solving the ECDLP is fully exponential, while sub-exponential algorithms exist for conventional cryptosystems. This difference largely contributes to the huge disparity in their respective running times. It also means that ECC keys have much fewer bits than IFP and DLP based applications. The contrast in key lengths of RSA, DSA and ECC are shown in the Figure 1 [Mov97].

Clearly, ECC keys take much more effort to break compared to RSA and DSA keys. Due to this, many people believe that ECDLP is intrinsically harder than the other two problems. While this deduction might be true, we have no way of *proving* it. We do not know if a fast and efficient elliptic curve DL algorithm that runs in sub-exponential time will be discovered, say, in the next ten years, or if another class of weak curves will be identified that could compromise the security of elliptic curve cryptosystems. But one thing is certain. After years of intensive study, there is currently no faster way to attack the ECDLP other than fully exponential algorithms.

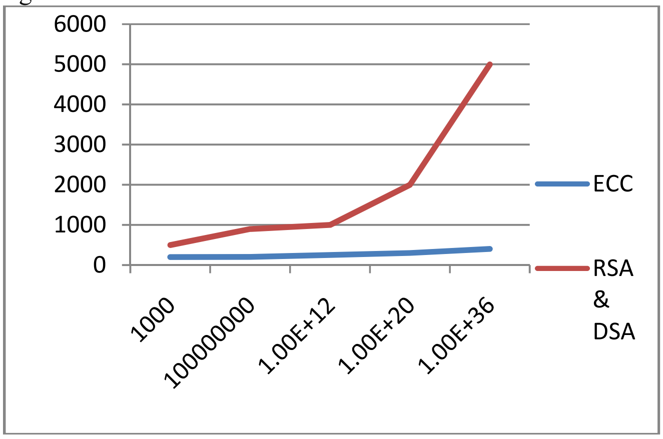

Figure 1: Comparison of Security levels ECC and RSA & DSA[Mov97]

From the advantages of ECC usage, there can be mentioned:

- increased security: cryptographic resistance per bit is much greater than those of any public-key cryptosystem known at present time;
- substantial economies in calculus and memory needs in comparison with other cryptosystems;
- great encryption and signing speed both in software and hardware implementation;
- ECC are ideal for small size hardware implementations (as intelligent cards);
- encryption and signing can be done in separate stages.

APPLICATIONS OF ECC

The intense research done on public-key cryptosystems, based on elliptic curves, demonstrated that ECC are suitable for the vast majority of existing applications. The elliptic curves are suitable in applications where the computing power is limited (intelligent cards, wireless devices, PC boards), memory size on integrated circuit is limited, a great speed of computing is necessary, digital signing and its verification are used intensively, signed messages have to be transmitted or memorized, digital bandwidth is limited (mobile communications, certain computer networks).

CONCLUSIONS AND FUTURE SCOPE

ECM belongs to a general class of curves, called hyperelliptic curves, of which elliptic curves is a special case, with genus, g=1. Hyperelliptic curves were initially candidates, to the next progression, or generalizations, to more secure systems, as they appeared to require even shorter key lengths than elliptic curves for the same level of security. It is found, however, that hyperelliptic curves of genus g=4, or higher, do not have the same level of security, as genus 2 or 3 curves, where attacks of sub-exponential time algorithms have been. Hence elliptic curves are the optimal practical solution from this family of curves.

A class of curves, known as the Koblitz curves, is particularly favorable because it was shown to be very efficient in computing *ord*(P) for arbitrary P on the curve, which can in turn be used to derive #*E*(F*p*) quickly.

In this paper, two attacks against improperly chosen elliptic curves and their underlying fields, but this by no means an extensive list has been covered. There are multiple other attacks against curve over prime fields as well as attacks against curves over binary fields. Suffice to say, anyone implementing an ECC needs to be aware of these potentially harmful curve choices and correctly mitigate them in their system.

This suggests that ECCs are superior to currently deployed public key cryptosystems since not only do they offer a greater level of security when the underlying parameters are chosen correctly, but they offer a greater advantage due to its shorter key sizes, faster generation of systems, smaller space requirements and efficient implementation techniques.

REFERENCES

[Mv85] Miller V , "Uses of elliptic curves in cryptography"., Advances in Cryptology – Crypto '85, Lecture Notes in Computer Science, vol 218. Springer, Berlin Heidelberg New York, pp 417–426, 1986

[Mov97] Menezes, A.., Oorschot, P., and Vanstone, S. "Handbook of Applied Cryptography". CRC Press, 1997.

[Cw97] "Remarks on the Security of the Elliptic Curve Cryptosystem". Certicom, whitepaper. September 1997.

[Kap99] Enge, A., "*Elliptic curves and their applications to cryptography"* . Kluwer Academic Publishers, 1999.

[Sak98] Satoh, T. and Araki, K., "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves". Commentarii Mathematici Universitatis Sancti Pauli 47, 1998: p 81 – 92.

[Sia98] Semaev, I. A, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", Mathematics of Computation 67, p 353 – 356, 1998

[Smn99] Smart. N. ,"The discrete logarithm problem on elliptic curves of trace one", .Journal of Cryptography, vol. 12 no. 3, Springer-Verlag New York,October, p 193 – 196, 1999.

[Sma97] N. P. Smart, "The discrete logarithm problem on elliptic curves of trace one" , October 1999.

[Kob87]N. Koblitz, "Elliptic curve cryptosystems", Mathematics of Computation, 48 , pp. 203-209, 1987.

[Ame93] A. Menezes, "Elliptic curve public key cryptosystems", Kluwer Academic Publishers, Boston, 1993.

[Mov93] A. Menezes, T. Okamoto and S. Vanstone, "Reducing elliptic curves logarithms to logarithms in a finite field", *IEEE* Transactions on Information Theory, 39 , pp.1639 – 1646, (1993).

[Asv93] A. Menezes, S. Vanstone, "Elliptic Curve Cryptosystems and Their Implementation", Journal of Cryptology, pp. 209-224, 1993.

[Kra09] K. Rabah, "Using Elliptic Curve Cryptography to Secure Online Data&Content", Information Security Research Journal, Vol. 1, No. 1,July 2009.

[Krb09] K. Rabah, "Using Elliptic Curve Cryptography for Information Security", Information Security Research, Journal Vol. 1, No. 2, July 2009.

[Itu09] I. Tutănescu, "Applications of Elliptic Curves Cryptosystems", MCC'2007 Conference Proceedings, Bonn, Germany, 2007.

[Rsh09] R. Shanmugalakshmi, M. Prabu, " Research Issues on Elliptic CurveCryptography and its applications", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.6, June 2009.

[Dh76]Diffie. W., and Hellman M. "New directions in cryptography" . IEEE Trans. Inform. Theory IT-22, , 644-654, Nov. 1976.

[Dmd04] de Miguel de Santos, "Elliptic curve cryptography on constraint environments", 38th Annual 2004 International Carnahan Conference on Security Technology, pg:212 - 220 [Len92] H.W. Lenstra, and C. Pomerance, Pomerance, Carl "A Rigorous Time Bound for Factoring Integers", Journal of the American Mathematical Society , vol 5 (3) ,p: 483–516. July 1992.

[Rsa79]Rivest, R., A. Shamir, and L. Adleman, January 1979, ''On digital signatures and public-key cryptosystems,'' MIT Laboratory for Computer Science, Technical Report,MIT/LCS/TR-212

[Shq98]P. Shor, "Quantum computing", proceedings of the International Congress of Mathematicians, 1998. http://www.research.att.com/~shor/papers/ICM.pdf

[Shp97] P. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithm problems", SIAM J. Computing 26 , 1484-1509. http://www.research. att.com/ ~shor/papers/QCjournal.pdf, (1997),

[Wpd12]http://en.wikipedia.org/wiki/Discrete_logarithm

[Plm78]Pollard, J. M., "Monte Carlo methods for index computation (mod *p*)". Mathematics of Computation, vol 32 (143): 918–924. , (1978).

[Pka00] J. M. Pollard, Kangaroos, "Monopoly and Discrete Logarithms" , Journal of Cryptology, Vol 13, pp 437-447, 2000

[Phe78] S. Pohlig and M. Hellman, "An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance"., *IEEE* Transactions on Information Theory , vol 24, pp: 106–110, 1978.

[Adl79] L. Adleman, "A subexponential algorithm for the discrete logarithm problem with applications to cryptography" ,20th Annual Symposium on Foundations of Computer Science, 1979

[Len92] H. W. Lenstra, Jr., "Algorithms in algebraic number theory", Bull. Amer. Math. Soc. ., vol 26(2), p: 211–244, 1992.

[Lln93] A. K. Lenstra and J. H. W. Lenstra, "The development of the number field sieve", Lecture Notes in Mathematics 1554, Springer, Berlin, 1993.

[Tel85] Taher ElGamal, "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", *IEEE* Transactions on Information Theory, vol 31 (4): 469–472, 1985.

[Dah04] Darell Hankerson, Alfred Menezees and Scott Vanston, " Guide to Elliptic Curve Cryptography" Springer- Verlag, New York, Inc. 2004

[Mwp09]Marisa W. Paryasto, Kuspriyante, Sarwan *et al,* "Issues in Elliptic Curve Crypyography implementation", Internetworking Indonesial Journal, Vol.1(1), 2009.