A NOVEL AND EFFICIENT ALGORITHM TO ENHANCE THE COMPLEXITY OF ELLIPTIC CURVE CRYPTOGRAPHY

Siddharth Sharma **Akshay Atrey** **Saurabh Jain**
School of Computing Sciences, Vellore Institute of Technology University, Vellore, Tamilnadu, India

Abstract *- In this paper, we propose a novel and efficient way to improve the computational complexity of the Elliptic Curve Cryptography [ECC] algorithm. ECC is a public key cryptography system, where the underlying calculations are performed over elliptic curves. The security of ECC is based on solving the Elliptic Curve Discrete Logarithm Problem [EDCLP]. We propose an algorithm to double the computational complexity of the conventional algorithm. The proposed algorithm generates two ECDLP opposed to one problem that was generated by the conventional algorithm being used till now. With the same key size, the proposed algorithm provides more security when compared to public key cryptography systems like RSA and ECC. It can be implemented efficiently in even less time when compared to ECC. The paper discuses the underlying protocol and proves how the enhancement in security and reduction in implementation time is achieved, thereby making it well suited for wireless communication.*

Keywords: *Discrete Logarithm Problem, Public Key Cryptography, Elliptic Curve Cryptography*

1 Introduction

In 1976 Diffie and Hellman [7] introduced the concept of Public key cryptography. They revolutionized the world of cryptography by developing a key exchange system popularly known as the Diffie Hellman Key Exchange which introduces applicability of discrete log in cryptography .The aim of the algorithm is to enable two users to securely exchange a secret key. It depends for its effectiveness on the difficulty in computing discrete logarithms.This concept involves finding discrete logarithm in a multiplicative group of integers modulo a prime number. But this idea can also be extended extended to other groups. Elliptic curves were introduced in cryptography in 1985 independently by Koblitz[5] and Miller[6],they proposed them as groups over which all calculations are to be performed. This generated more complex discrete logarithm problems over elliptic curve as underlying finite fields when compared to discrete logarithm problem[DLP] generated using multiplicative groups of integers defined over primes. Another advantage of ECC over systems based on the multiplicative group defined over finite field (and also over systems based on the intractability of integer factorization) is the absence of a sub exponential-time algorithm (such as those of "index-calculus" type) [3].Many public key cryptographic systems like RSA rely on the difficulty of factorizing large integers into their prime factors. Due to the enhancement in computing power of processors and the availability of cheaper hardware, many techniques have been discovered that compromise the efficacy of RSA .For example , the 1024 bit keys that seem to provide acceptable security levels today, could be threatened in near future. Over the years the requirement of key length for secure RSA system has increased significantly and is bound to increase further in time to come. The major cause is the development of fast computing processors, which reduces the time taken in brute force computation. This significant increase in the key length has resulted in increased computational overhead on systems employing the RSA cryptography method. Jurisic and Menezes[2] made a comparison based on the study between RSA and ECC. They compared the security achieved by both methods versus the key length used .The results showed that ECC outperforms RSA. Vivek Kapoor and Vivek Sonny[1] further discussed the advantages of ECC over RSA. Another fact that establishes the superiority of ECC over RSA is the existence of sub exponential time complexity attacks on RSA, whereas only few algorithms exist for solving the Elliptic Curve Discrete Logarithm Problem[ECDLP],that too for some particular classes of elliptic curve. Table 1[1] compares computational complexity achieved with different key lengths between RSA and ECC.

RSA key size (bits)	ECC key size (bits)	MIPS years
512	106	10^{4}
768	132	10^{8}
1024	160	10^{11}
2048	210	10^{20}

TABLE 1MIPS year represents computation power of a computer executing a million instructions per second, when used for one year.

ECC when compared to RSA offers equivalent security for much smaller key sizes. The benefit of smaller key size is reduced computational overhead during implementation. These benefits, though applicable in all scenarios, are especially attractive for security applications in battery powered mobile devices with limited computational capabilities.,this finds immense application in devices with limited Integrated Circuit space like smart cards, cell phones and PDAs.

$$\lambda = \begin{cases} \frac{y2-y1}{x2-x1} & \text{if} P != Q \\ \frac{3x12+a}{2y1} & \text{if} P=Q \end{cases} \qquad (5)$$

The remainder of the paper is organized as follows. Section 2 presents an insight into the EDCLP. The arithmetic details behind elliptic curve are discussed in section 3. Section 4 describes procedure for cipher text generation, security of ECC is discussed in section 5 and the proposed method to increase the computational complexity achieved by ECC is explained in section 6. The algorithmic implementation is explained in section 7. Section 8 explains the enhancement in security achieved by proposed method. The implementation efficiency of the proposed algorithm is discussed in Section 9.

2. Elliptic curve discrete logarithm problem

The discrete logarithmic problem originally discussed by Diffie and Hellman is defined as the problem of finding logarithms with respect to a generator in the multiplicative group of the integers modulo a prime. But the problem of finding discrete logarithms can be extended to other groups when group arithmetic is performed over elliptic curves, the computational complexity increases many folds .[3] explains the discrete logarithm problem as, let G be a finite group of order n, and let α be an element of G. The discrete logarithm problem for G is the following: given an element β€ G, find an integer x, $0 \leq x \leq n - 1$, such that $\alpha^x = \beta$. Various groups have been proposed over the years for cryptographic purposes like Agnew et al[4] proposed multiplicative groups of characterstics two over finite field. Koblitz[5] and Miller[6] used the group of points on an elliptic curve defined over a finite field.

3. Arithmetic behind ECC

An elliptic curve used for cryptographic purposes is defined as follows,

$$y^2 \bmod p = (x^3 + ax + b) \bmod p \qquad (1)$$

here a and b are integer constants.The set of points E (a, b) is a set all points x and y satisfying the above equation. For an elliptic curve over a finite field Z_p , in (1) all variables and coefficients take on values in the set of integers from 0 and p − 1 for some prime p, and the calculations are performed modulo p .

Assume first that F_q has characteristic greater than 3. An elliptic curve E over F_q is the set of all solutions (x, y) € $F_q \times F_q$ to an equation

$$y^2 = x^3 + ax + b \qquad (2)$$

here a,b € F_q and 4a + 27b = 0, together with a special point ∞ called the point at infinity.

Addition on the curve defined by (1) is followed according to the below defined method. Let P = (x1 , y1) € E; then −P = (x1 , −y1). If Q = (x2, y2) € E, Q = −P, then P + Q = (x3 , y3), where

$$x3 = \lambda^2 - x1 - x2 \qquad (3)$$

$$y3 = \lambda(x1 - x3) - y1 \qquad (4)$$

and

$$\lambda = \begin{cases} \frac{y2-y1}{x2-x1} & \text{if} P != Q \\ \frac{3x12+a}{2y1} & \text{if} P=Q \end{cases} \qquad (5)$$

4. Procedure for generating cipher text

Let us assume that E is an elliptic curve over field F_q and G (generating point) is a point on the curve that is mutually agreed upon by the users of the system. Let there be two users, Alice and Bob, who desire to secretly exchange data. Alice selects a random number n_A and keep it secret, which acts as her private key. Similarly, Bob selects his private key n_B. Both will now generate their public keys p_A and p_B as follows

$$p_A = n_A * G \qquad (6)$$

$$p_B = n_B * G \qquad (7)$$

Here * is the multiplication operation defined over the elliptic curve E. Now we will use MAP2 Group method to map a message to a point on the elliptic curve E .Now message m is converted to point P on E. Alice will now select a secret parameter k. Alice will calculate $k*p_B$ and k*G. Cipher text Cm will be generated as follows

$$C_m = \{kG, P+k*p_B\} \qquad (8)$$

In eq.3 kG will act as trapdoor parameter and will help Bob in decryption. The adversary can access G and pB as they are in the public domain. Given G and kG, it is computationally hard to kind k, this is called as ECDLP. To decrypt, Bob will multiply his private key with trapdoor kG and subtract it from the second argument. The plain text, P_m will be generated as following

$$P_m = (P_m+k*p_B)-(kG*n_B) \quad (9)$$

5. Security in ECC

The basis for the security of elliptic curve cryptographic systems such as the ECDSA is the apparent intractability of the following ECDLP. Given an elliptic curve E defined over F_q , a point P Є $E(F_q)$ of order n, and a point Q Є $E(F_q)$, such that Q = l P here $0 \leq l \leq n - 1$.Several possible methods for solving the ECDLP have been discussed in [3]. The determination of l is reduced to the determination of l modulo each of the prime factors of n by using the algorithm proposed by Pohlig–Hellman [9]. The best known algorithm developed to counter the ECDLP is Pollard ρ-method [10] .The previous method was made more efficient by Gallant, Lambert and Vanstone [11] that takes about $\sqrt{(\pi n)}/2$ elliptic curve additions. Semaev[12], Smart[13], Satoh–Araki[14] designed an algorithm which efficiently computes isomorphism between $E(F_p)$,where E is a prime-field-anomalous curve, and the additive group of F_p .This gives a polynomial-time algorithm for the ECDLP in $E(F_p)$. Menezes, Okamoto and Vanstone [15] developed the famous MOV attack,it uses the concept of weil pairing to transform the elliptic curve group to a multiplicative group, defined over field F_q^k for some integer k. Due to this transformation the Elliptic curve discrete logarithm problem gets reduced to finding discrete logarithm problem (DLP) in F_q^k. Miller in [16] discussed the implementation of index calculus method in elliptic curve groups.

6. Proposed method to increase the complexity of ECC

In this paper we propose a new method of enhancing the security of elliptic curve crypto system by generating two discrete log problems, for which each has to be solved independently. In ECC the curve on which computation is to be performed and the underlying algorithm are kept in public domain because it reduces the implementation overheads. After doing some computations on this base curve i.e. the curve available in the public domain using techniques mentioned in section 7, we rotate it along the perpendicular axis to some appropriate value. Now this curve becomes hidden from the adversary. To know this curve the adversary has to solve an ECDLP. After rotating this curve we map the public keys and generating point (G) from base curve to the rotated curve. This can be performed by developing a mapping function that performs a one to one mapping between the two curves. Now all the desired group arithmetic on the rotated curve is performed using conventional methods. The user receiving the encrypted message has to use his/her private key with a given trapdoor provided along with encrypted message in the cipher text to find the position of the curve. We are increasing the security by producing two discrete log problems using keys of same size as used earlier. Now the adversary using any of the sub exponential time algorithms explained in section 5, has to apply those to two separate ECDLP problems. The algorithm is defined in the next section.

7. Algorithm

An elliptic curve E1 over Fq is the set of all solutions (x, y) Є $F_q \times F_q$ to an equation

$$y^2 = x^3 + ax + b \quad (10)$$

Where a, b Є F_q and 4a + 27b = 0, together with a special point ∞ called the point at infinity. Let Alice and Bob have their private key and public key as (n_A, p_A) and (n_B, p_B) . Now in the following subsections the various steps involved are discussed *reference page numbers in the text.*

7.1 Rotating the curve

The below mentioned steps are followed to generate the curve on which final computation is to be performed from the base curve.

1. Alice will take a secret number k1 and multiply the Bob's Public key p_B per the group laws defined above.
$Q=k1* p_B$
Let Q's coordinates be (x', y').
2. The angle of rotation, θ is evaluated as
$\theta=\tan^{-1}(y/x)$
3. The curve is rotated by θ along the axis perpendicular to the plane of the curve which generates the curve E2.
4. Alice will map public keys p_A, p_B and generating point G to the rotated elliptic curve, using the mapping function F defined in subsection 7.5. Let us assume that the mapped images are p_A', p_B' and G'.
5. Now Alice will select another secret parameter k2 and compute k2*G' and $k2*p_B'$.
6. Alice will map the plain text to P to point on the curve P_m, using the Map2Group algorithm.

7.2 Cipher text generation

Alice sends cipher text Cm to Bob as
Cipher text

$$C_m=\{k1G, k2G',P_m+k2*p_B'\} \quad (11)$$

Here k1 is the parameter which Alice chooses for rotation of the curve, G is the generating point used initially on the curve E1 given in the public domain to find E2,k2 is the secret parameter used by Alice to calculate k2G and k2pB,G' is the mapped generating point ,P_m is the plain text. The first two arguments of the cipher text act as trapdoors while the third argument is the encrypted message.

7.3 Decryption

Bob will multiply his private key n_B with the first trapdoor to discover the curve E2 as follows

$$n_B*k1G=k1(n_B*G)=k1*p_B=Q=\{x', y'\} \qquad (12)$$

From this data Bob will calculate the angle by which curve is rotated .Now all the calculations have to be performed on the curve E2. Bob will multiply his private key with the second trapdoor i.e. k2*G' and subtract the result form the third argument of cipher text.

$$\{P_m+k2*p_B'\}-\{k2G'*n_B\}=\{P_m+k2*p_B'\}-\{k2*(n_B*G')\}=\{P_m+k2*p_B'\}-\{k2*p_B'\}=P_m \qquad (13)$$

7.4 Development of a one to one function

Let F be a one to one function mapping points from E1 to E2 i.e. F: E1→E2. Let there be a point (x,y) on E1. After the rotating the curve by θ we obtained the curve E2. Let there be another point having abscissa x' and ordinate y' on E2, which is the image of the point (x,y) on E1. F will find a mapping scheme between the two groups as

$$x'=x\cos\theta-y\sin\theta \qquad (14)$$

$$y'=x\sin\theta+y\cos\theta \qquad (15)$$

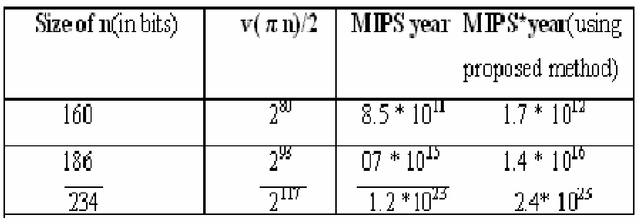
The images thus generated will also form a group structure and will obey the underlying laws of group arithmetic. Since the new points are mapped on an Elliptic curve the EDCLP will still be generated when calculations are formed over this elliptic curve.

8. Enhancement in security

If the proposed method is implemented then it would be necessary for the adversary to solve an ECDLP to find the curve E2 only then adversary can perform the required computation on it. After that he/she has to solve another ECDLP to obtain plain text from the cipher text. Thereby we are able to enhance the security twofold without increasing the key length. The degree of security achieved can be compared with that achieved using conventional methods against various key lengths. If we keep the key length same, the security is increased two folds e.g. the index calculus algorithm takes sub exponential running time

$$\exp((c + O(1))(\log qk)1/3(\log \log qk)2/3) \qquad (16)$$

Here we assume that the elliptic curve is defined over the field F_q^k. If the proposed method is implemented, the adversary has to apply the index calculus algorithm twice to obtain the plain text. Majority of the algorithms solving the ECDLP exploit the field properties and field size on which the elliptic curve is defined. As an example, we will show the security level reached by incorporating the above proposed method in comparison to the conventionally used ECC algorithm .One of the best known algorithms to solve ECDLP is Pollard p method[10]. Let us assume that over the curve E is defined over the field F_2^k. We will depict the field size parameter 2^k by n bits. Pollard p method will take $\sqrt{}(\pi n)/2$ steps to solve ECDLP, here a step represents elliptic curve addition. Let us denote the computing power of the computers solving the EDCLP by Pollard p algorithm in MIPS i.e. million instructions per second. Table 2 compares the complexity of conventional ECC algorithm and proposed algorithm when solved by Pollard p method .



Size of n(in bits)	v(π n)/2	MIPS year	MIPS*year(using proposed method)
160	2^{80}	$8.5 * 10^{11}$	$1.7 * 10^{12}$
186	2^{93}	$07 * 10^{15}$	$1.4 * 10^{16}$
234	2^{117}	$1.2 * 10^{23}$	$2.4 * 10^{25}$

TABLE 2

Since our proposed method generates two separate ECDLP problems, the Pollard p method will have to be applied to both the problems independently,therefore the computational overhead required to breach the security becomes twice. This enhancement is achieved without increasing the key length.

9. Implementation efficiency

ECC has major application in wireless communication. The wireless devices run on battery power, so the limited energy provided by battery acts as a constraint on the processing capabilities .If we use methods like RSA, though they provide high level of security but they will consume the power of the wireless devices at both ends as both encryption and decryption processes will take enough time and energy. ECC overtakes RSA in this respect, as we can use keys of lower size with respect to RSA, and still provide same level of security. ECC takes less time to implement as arithmetic performed i.e. point addition and point doubling over the elliptic curve is relatively fast when compared to arithmetic performed over other fields.

Koblitz, Menezes and Vanstone[3] made a comparison between the time taken by ECC arithmetic performed over by elliptic curve defined over field Fq, whose order is 160 bits with DSA arithmetic performed over field Fp where calculations are performed with a 1024 bit modulus p. Since they both provide same level of security we are comparing the time taken to implement them. Multiplying two n bit numbers takes n^2 bit operation, so modular multiplication is $(1024/160)^2 \approx 41$ times longer than a field multiplication. The arithmetic operations performed over elliptic curves was found to be more time efficient than modular arithmetic operations performed over other groups [3]. This is described in the below mentioned steps.

1. To calculate kP, where $P \in E(F_q)$ and k is a 160 bit integer, we have to do repeated doubling and addition which on average requires 160 elliptic curve doubling and 80 elliptic curve additions. The time taken to perform Elliptic curve addition or doubling requires one field inversion and two field multiplication [17].Also [18] and [19] showed that one field inversion is equivalent to three field multiplications .[3] showed that computing k P requires the equivalent of 1200

field multiplications, or 1200/41 ≈ 29 1024-bit modular multiplications.

2. To calculate a^k mod p, where k is 160 bit random number and p is 1024 bit prime (as performed in DSA) by repeated squaring and multiplying requires an average of 240 1024-bit modular multiplications [3]. It proves that arithmetic operations performed on elliptic curve defined over Fq are nearly 8 times faster then operations performed in case 2.
Let us assume a random number k of 32 bits .
Now cipher text,
C_m= {kG, P_m+k*p_B}
In this mechanism we have to make three calculations
1. Multiplication of G with k
2. Multiplication of p_B with k.
3. One addition operation
Both 1 and 2 will require 240 field multiplications. So in total we have 480 field multiplications. As explained earlier one addition is equivalent to five field multiplications. So the total time taken in producing the cipher text is equivalent of doing 485 field multiplications.
The proposed method may seem to be more time consuming at first look, but if we select the numbers k1 and k2 prudently then we can perform the above calculation in even less time .In our method cipher text,
C_m={k1G, k2G',P_m+k2*p_B'}.
Let us assume that k1 is 32 bit and k2 be any number that can be represented in less than 32 bits. Now, the four operations performed are
1. Multiplication of k1 with G.
2. Multiplication of k2 with G'.
3. Multiplication of k2 with p_B'
4. Addition between P_m and k2*p_B'
Operation 1 is equivalent of doing 240 field multiplications. As the number of bits representing the number becomes less the maximum value attained by the number decreases exponentially since if a number can be represented in ‘n’ bits its maximum value can be 2^n -1. So as we decrease the number of bits representing k2 its value decreases exponentially and so do the field multiplications required in calculations involved. Let us take k2 be of 30 bits. Now step 2 and 3 will take $(240/2^{32)} * 2^{30}$ =60 field multiplications each. Thus a total of 360 field multiplications are made in step 1,2 and 3. Step 4 is an addition operation and is equivalent to five field multiplications .Therefore the total time taken in implementing the proposed algorithm is equivalent of performing 365 field multiplications.

Thus our proposed method takes even less time in implementation as taken by previous method. It can still be used in wireless communication. On the other hand it provides more security than the conventional elliptic curve cryptographic system.

10. Conclusion

In this paper we have introduced a novel and efficient method to enhance security in ECC cryptography system. A comparison on the basis of computational security achieved and implementation time between our proposed method and previously incorporated technique was also made. We have proved that our proposed technique is more secure than the previously used method and it takes less time in implementation. The method provides more security with less implementation overhead and is well suited for secured wireless communication.

11. References

[1]Vivek Kapoor and Vivek Sonny,Elliptic Curve Cryptography ,ACM Ubiquity, Volume 9, Issue 20 May 20 – 26,2008.
[2]Aleksandar Jurisic and Alfred J. Menezes,.Elliptic curves and cryptography Dr. Dobb’s Journal, 1997.
[3] Neal koblitz.,Alfred J. Menezes , Scott Vanstone , The State of Elliptic Curve Cryptography,ACM Ubiquity, Volume 19, Issue 2-3 (March 2000), Special issue on towards a quarter-century of public key cryptography Pages: 173 - 193
Year of Publication: 2000
[4]Brian A. LaMacchia, John L. Manferdelli , New Vistas in elliptic curve cryptography,information security technical report 11 (2006) 186–192,Elsevier Ltd.
[5]N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, Vol. 48 (1987) pp. 203–209.
[6]V. Miller, Uses of elliptic curves in cryptography, Advances in Cryptology—CRYPTO ’85, Lecture Notes in Computer Science, Springer-Verlag, 218 (1986) pp. 417–426.
[7]W. Diffie and M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, Vol. 22 (1976) pp. 644–654.
[8] G. Agnew, R. Mullin, I. Onyszchuk and S. Vanstone, An implementation for a fast public-key cryptosystem, Journal of Cryptology, Vol. 3 (1991) pp. 63–79.
[9]Pohlig and M. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, IEEE Transactions on Information Theory, Vol. 24 (1978) pp. 106–110.
[10]J. Pollard, Monte Carlo methods for index computation mod p, Mathematics of Computation, Vol. 32 (1978) pp. 918–924.
[11]R. Gallant, R. Lambert and S. Vanstone, Improving the parallelized Pollard lambda search on binary anomalous curves, Mathematics of Computation,69,1699-1750.
[12]I. Semaev, Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p, Mathematics of Computation, Vol. 67 (1998) pp. 353–356.
[13]N. Smart, The discrete logarithm problem on elliptic curves of trace one, Journal of Cryptology,12,193-196.
[14]T. Satoh and K. Araki, Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic

curves, Commentarii Mathematici Universitatis Sancti Pauli, Vol. 47 (1998) pp. 81–92.

[15]A. Menezes, T. Okamoto and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Transactions on Information Theory, Vol. 39 (1993) pp. 1639

[16]Miller, Uses of elliptic curves in cryptography, Advances in Cryptology—CRYPTO '85, Lecture Notes in Computer Science, Springer-Verlag, 218 (1986) pp. 417–426.

[17]G. Agnew, R. Mullin, I. Onyszchuk and S. Vanstone, An implementation for a fast public-key cryptosystem ,Journal of Cryptology,_3(1991), 63-79.

[18]R. Schroeppel, H. Orman, S. O'Malley and O. Spatscheck, Fast key exchange with elliptic curve systems, Advances in Cryptology—CRYPTO '95,_LNCS 963, Springer-Verlag, 1995, pp.43-56.

[19]E. De Win, A. Bosselaers, S. Vandenberghe, P. De Gersem and J. Vandewalle, A fast software implementation for arithmetic operations in G F(2n),Advances in Cryptology—ASIACRYPT '96,_*LNCS 1163*, Springer-Verlag, 1996, pp. 65-76.