An Enhanced Mobile Security Technique using Elliptic Curve Cryptography

K.S.Mohanasathiya
Assistant Professor, Department of Computer Science,
Navarasam Arts & Science College for Women, Erode
Sathya_vinu87@yahoo.com

Abstract— Mobile phones are most common way of communication and accessing internet based services. The Public key cryptography is effective security solution to provide secure the mobile communications. In this research work describe an ECC module to secure data encryption and decryption using public key cryptography. The implementation of ECC module can provide various security services in the form of key exchange, communication privacy through encryption, authentication of sender and digital signature to ensure message integrity. Elliptic curve cryptography is an asymmetric key cryptography. It includes (i) public key generation on the elliptic curve and its declaration for data encryption and (ii) private key generation and its use in data decryption depended on the points on two dimensional elliptical curve. The implementation of ECC on two finite fields, prime field and binary field and overview of ECC implementation on two dimensional representations of plaintext coordinate systems and data encryption through Elgamal Encryption Technique are discussed. Much attention has been given here on the mathematics of elliptic curves starting with their derivations and the proof of how points upon them form an additive abelian group for cryptographic purposes, specifically results for the group formed by an elliptic curve over a finite field, E (Fp), E (F2m) and showing how this can form public key cryptographic systems for use in both encryption and key exchange. Finally, to encrypt the data with the alphabetical table.

Keywords— ECC,Mobile,Encryption,Decryption.

1. INTRODUCTION

Cryptography is commonly employed security concepts and terminology. The concern for security in practice is addressed by choosing a security protocol, which achieves all the required security objectives. Security protocols realize the security objectives through the use of appropriate cryptographic algorithms.

Symmetric Algorithms

These algorithms use the same key for encryption and decryption. They rely on the concepts of "confusion and diffusion" to realize their cryptographic properties and are used mainly for confidentiality purposes. Also known as secret key cryptosystems.

Asymmetric Algorithms

These algorithms use different keys known as the public key and the private key for encryption and decryption. They are constructed from the mathematical abstractions which are based on computationally intractable number-theoretic problems like integer factorization, discrete logarithm, etc.. They are primarily used for authentication and non-repudiation. They called as public key Cryptosystems (PKC).

Elliptic Curves in Cryptography

Mobile phones are most common way of communication and accessing Internet based services. Currently, mobile phones are not only used for formal communication but also, sending and receiving sensitive data. The security of mobile communication has topped the list of concerns for mobile phone users. So public key cryptography is effective security solution to provide secure the mobile communications. A ECC module to secure data encryption and decryption using public key cryptography. The implementation of ECC module can provide various security services in the form of key exchange, communication privacy through encryption, authentication of sender and digital signature to ensure message integrity.

The basic idea of Elliptic Curve Cryptography (ECC) and its implementation through co-ordinate geometry for data encryption. Elliptic curve cryptography is an asymmetric key cryptography. It includes (i) public key generation on the elliptic curve and its declaration for data encryption and (ii) private key generation and its use in data decryption depended on the points on two dimensional elliptical curve. The implementation of ECC on two finite fields, prime field and binary field. An overview of ECC implementation on two dimensional representations of plaintext coordinate systems and data encryption through Elgamal Encryption Technique. Much attention given to the mathematics of elliptic curves starting with their derivations and the proof of how points upon them form an additive abelian group for cryptographic purposes, specifically results for the group formed by an elliptic curve over a finite field, E (Fp), E (F2m) and showing how this can form public key cryptographic systems for use in both encryption and key exchange. Finally, we describe how to encrypt the data with the alphabetical table.

A new digital signature based on elliptic curves is presented. Here established its efficiency and security. The method, derived from a variant of ElGamal signature scheme can be seen as a secure alternative protocol if known systems are completely broken. A developed efficiency and security new digital signature based on elliptic curves. The method, derived from a variant of ElGamal signature scheme can be seen as a secure alternative protocol if known systems are completely broken.

2. REVIEW OF LITERATURE

Asha et al. [1] proposed methodology is different issues of Wireless Sensor Network (WSN) and the relevance of the Elliptic curve y cryptography. Security in WSN is a greater challenge in WSN due to the processing limitations of sensor nodes and nature of wireless links. Extensive use of WSNs is giving rise to different types of threats.

Aarti Singh et al [2] provide the agent community works on the core idea of cooperation and delegation of tasks, which in turn should be prevented from any malicious usage. In order to avoid this malicious usage, an instrument for ensuring proficient and secure communication among these collaborating agents is trust.

Amounas et al [3] a novel mapping of text message into multiple points on Elliptic Curve by using addition table. Then, we describe a new method for encryption and decryption based on matrices. Further, this paper also attemps to utilize the properties of invertible matrices in encryption and decryption process with more flexible and efficient. The proposed method enhances the security of ECC with multi fold encryption.

Gandhewar et al. [4] IEEE 802.16 provides several security mechanisms, which provides more security by protecting the network against unauthorized access. Many works provides the security improvement mechanism for WiMax. Many sophisticated authentication and encryption techniques have been embedded into WiMAX but it still exposes to various attacks. This provides a mechanism for increasing the efficiency and hence improves the existing model.

Haodong et al. [5] developed an control in sensor networks is used to authorise and grant users the right to access the network and data collected by sensors. Different users have different access right due to the access restriction implicated by the data security and confidentiality. Even though symmetric-key scheme, which has been investigated extensively for sensor networks, can fulfill the requirement, public-key cryptography is more flexible and simple rendering a clean interface for the security component.

Jaspreet Singh et al [6] weaknesses and possible attacks on the RC4 stream cipher in WEP have analyzed and we proposes more secure WEP Protocol that offers secure encrypted communication by using Elliptic Curve Cryptography (ECC) Technique. Point Multiplication is the core operation performed in ECC. NAF (Non Adjacent Form) is the efficient method used for Point Multiplication. They implemented both Standard and Block method for computing NAF of ECC and done the comparative study of these methods by taking several parameters in WEP. The proposed ECC Technique will ensure secure encryption in WEP and will enhance its security.

Kishore Rajendiran et al, [7] security in wireless sensor networks (WSNs) is an upcoming research field which is quite different from traditional network security mechanisms. Many applications are dependent on the secure operation of a WSN and have serious effects if the network is disrupted. Therefore, it is necessary to protect communication between sensor nodes. Key management plays an essential role in achieving security in WSNs. To achieve security, various key predistribution schemes have been proposed in the literature.

Mohammed et al, [8] avoid inversion complexity, the elliptic computations arithmetic utilizes projective coordinates instead of the normal affine coordinates. We adjusted the elliptic curve crypto addition operation with efficient scheduling for this pipelining. To proposed hardware is compared to the previous parallel (non-pipelined models that were similarly designed. All considered architectures have been synthesized for 160-bits operations showing interesting features.

Sumedha Kaushik et al. [9] network Security is the most vital component in information security because it is responsible for securing all information passed through networked computers. Network Security refers to all hardware and software functions, characteristics, features, operational procedures, accountability, measures, access control and administrative and management policy required to provide an acceptable level of protection for Hardware and Software , and information in a network.

Santoshi et al [10] proposed a implementation of Elliptic Curve Cryptography Algorithm. The implementation includes Diffie Hellman Key Exchange and the Digital Signature Algorithm gives an overview of Elliptic Curve Cryptography algorithm. Cryptography (or cryptology) from Greek word kryptos, "hidden, secret" and graph, "writing" is the practice and study of hiding information.

3. METHODOLOGY

The background necessary to understand the cryptographic importance of Binary Edwards curves. The begin with a brief discussion of elliptic curves in general. Since the mostly interested in the application of elliptic curves and pairing computations. To recommend these two books, to readers interested in a more in-depth background.

3.1 ELLIPTIC CURVES

Elliptic curves were proposed for use as the basis for discrete logarithm-based cryptosystems almost 20 years ago, independently by Victor Miller of IBM and Neal Koblitz of the University of Washington. At that time, elliptic curves were already being used in various cryptographic contexts, such as integer factorization and primarily proving.

3.2 Weierstrass Curves

Broadly speaking, elliptic curves are curves of genus one having a specified base point". After appropriate scaling, such curves are usually written in generalized coordinates in the homogeneous form

$$Y^2Z + a1XY\,Z + a3Y\,Z^2 = X^3 + a2X^2Z + a4XZ^2 + a6Z^3$$

Where X; Y and Z are taken to be projective coordinates from P2 over some base field K and a1,…,a6 are scalars from the algebraic closure K (though often they're just taken to be elements of K itself). The ease of notation, often work in non-homogeneous are coordinates instead, taking

$$x = X=Z \text{ and } y = Y=Z \qquad \ldots.. 3.1$$

$$y2 + a1xy + a3y2 = x3 + a2x2 + a4x + a6$$

Two forms are interchangeably called the Weierstrass form of the curve. If char(K) =2 f2; 3g, then usually simplify further to

$$y^2 = x^3 + Ax + B \quad \text{....3.2}$$

After a further change of coordinates (though of course we won't be able to do this when working with binary curves, i.e. curves over finite fields of characteristic two). It specify a special point, denoted by 1 or O, with the projective coordinates (0 : 1 : 0).1 For fields K with char (K) = 2, Weierstrass curves are usually written in the form

$$y2 + xy = x3 + a2x + a6 \quad \text{.....3.3}$$

The typically only work with non-singular curves. This allow the curve to have multiple roots; we choose our constants such that

$$4A^3 + 27B^2\ 6= 0 \quad \text{.....3.4}$$

3.3 ECC IMPLEMENTATION

To implement ECC cryptosystem on Telos-B mote powered by MSP430 micro controller. The MSP430 incorporates an 8 MHz, 16-bit RISC CPU, 48 K bytes flash memory (ROM) and 10 K bytes RAM. This architecture provides 27 instructions and 7 addressing modes. The CPU also provides sixteen 16-bit registers. The first four are dedicated for special-purpose, such as programmed counter, stack pointer and status register. The rest of the twelve are available for general use. Besides, the MSP430 also provides a peripheral hardware multiplier, which is capable of conducting up to 16 × 16 bits multiplication. Given the limited processor resources, concentrate most of efforts on computation optimization. The fundamental ECC operation is large integer arithmetic over either prime number finite field GF *(p)* or binary polynomial field GF *(2m)* (where *m* is a prime). Because the two heavily used operations: multiplication and modular reduction, can be more effectively optimized if pseudo-mersenne primes are picked up for elliptic curves compared with those of binary field, limit the discussion in prime number finite field *GF(p)*.

3.4 LARGE INTEGER OPERATIONS

To implement a suite of large integer arithmetic operations, including addition, subtraction, shift, multiplication, division and modular reduction. Due to the space limit, only present three of the most important functions: multiplication, division and modular reduction.

3.4.1 Multiplication

The efficiency of large integer multiplication dominates the overall performance of ECC operation. It shows that as much as 85% of execution time is spent on multiplication for a typical point multiplication in ECC. That means the optimization on multiplication is critical for overall performance of our implementation. To ease the explanation, we use three large integers as the examples for our following discussion: *A(an*−1, *an*−2, . . . , *a*1, *a*0*)*, *B(bn*−1, *bn*−2, . . . , *b*1, *b*0*)*, and *C(n*2*n*−1, *c*2*n*−2, . . . , *c*1, *c*0*)*, where *C* = *AB*. *A* and *B* both have length of *n* words, each word has *k*-bit size. The product *C* has 2*n* words.

Hybrid multiplication is the combination of Row-wise multiplication and column-wise multiplication. The row-wise method fixes the multiplier *bi* ($0 \le i \le n$) and multiplies it with everyword of multiplicand A. Partial results are stored in *n* + 1 accumulator registers. Every time one row is finished, the last accumulator register can be stored in memory as part of the final results. On average, one memory load is required for each $k \times k$ multiplication. When integer size *n* is increased (integer size is 10 for curve secp160r1), the required number registers increase linearly in Row-wise method. The column-wise method, on the other side, computes the partial results of *aibj* (where $i + j = l$) for column *l*. After one column finishes, the last word of accumulator registers is stored as the part of final result.

The column-wise method only requires three accumulator registers and two more for operands. However, two memory load operations are required for each $k \times k$ multiplication. Considering a large number of data in ECC operations, unnecessary memory operations would lower the performance. The Hybrid method takes advantage of row-wise and column-wise strategies.

To optimize the memory operation, the hybrid method merges a number (*d*) of columns together and then conducts row-wise multiplication in each merged column. When *d* equals to 1, the hybrid method becomes the Column-wise multiplication. When *d* equals to *n*, then it equals to Row-wise method. Therefore, a single memory load operation can be used for several multiplications. A larger *d* leads to fewer memory operations, but requires more registers. Since the MSP430 microcontroller only has 12 general registers, only implement the hybrid method with column size $d = 2$, which requires 5 accumulator registers, 3 operand register and other 4 registers for pointer, temporary storage and loop control. To achieve better performance and enable flexible control over registers, we implement the hybrid multiplication in assembly language.

The experiments show that the performance of point multiplication improves about 5% with the hybrid multiplication compared with the column-wise method and improves another 5% with assembly language compared with original implementation with C.

3.4.2 Division

Modular division is another expensive operation in ECC. In affine coordinate, each ECC operation of PADD and PDBL requires a modular inversion. The integer inversion is also required for ECC digital signature generation and verification. Given a denominator *x* and numerator *y*, to compute the modular division *y/x* over *GF (p)*. This is equivalent to find *r*, so that

$$r \equiv y/x \pmod q \quad (10)$$

To find *r* efficiently, maintain following two invariant relationships

$$Ay \equiv Ux \text{ and } By \equiv Vx \quad (11)$$

Where A,B,U and V are four auxiliary registers and assigned with initial values *x*, *q*, *y* and 0, respectively. The second invariant relationship is true even for $v = 0$ because algebraically the value of modulus is equivalent to zero in finite field. The division procedure repeatedly reduces the values of *A* and *B* in the following way. In each iteration, if either *A* or *B* is even, divide by 2 both sides of the equation. If *U* or *V* is not even at that time, can make it even by adding modulus *q*. If both *A* and *B* are odd, add two equations together and then divide by 2 at both sides. *A* or *B* reduces one bit in one iteration. The procedure stops when $A = B = 1$ the first equation becomes

$$y \equiv Ux \quad (12)$$

The value of U is our final result. If we initialise U with 1, this routine can be used to calculate an inversion of x. This algorithm works when x and q are relatively prime. Otherwise, the routine would return the greatest common divisor of A and B. The great divide finishes division or inversion operation in $2(log(x) - 1)$.

3.4.3 Reduction

The modular reduction operation is as important as modular multiplication. Each multiplication must be followed by a reduction operation. Great Divide algorithm does not work for modular reduction. Since we choose to use pseudo-mersenne primes as specified in NIST/SECG curves, the modular reduction can be optimised by conducting a fixed number of integer additions. Because the optimization is curve specific, in more detail in the section of ECC operation. The modular reductions in digital signature generation and verification. In most cases the order of an elliptic curve is not a pseudo-mersenne prime, the optimization cannot be applied for those reduction calculation. To choose the classic long division method to implement this operation. It may not be the most efficient algorithm, but it does not affect the overall performance much because very limited number of modular reductions is required in digital signature algorithm. The long division method as follows. Given an integer x, to calculate

$$r \equiv x \bmod p \quad (13).$$ Where p is a prime.

3.5 Plaintext encryption

From the above basic theory on elliptic curve cryptography, in this section to describe the concept of plaintext encryption by defining a two-dimensional alphabetic table. It is worth noting that in the case of elliptic curve cryptography there is no specified rule and or algorithm to specify the letters of the English alphabet as well as special symbols. For a 6x5 table(Table 3.1) formed here for both the upper case and lower case letters of the English alphabet along with some of the other symbols like **,** , **.** , **?** and space for illustration purpose only. Other symbols of punctuation marks and special characters can also be considered in a similar way. The tables play some important role in ECC as two-dimensional plaintext co-ordinate representation requires adding with any point on the elliptic curve. Now, for any plaintext to be encrypted add or multiply coordinates of a given character with selected points on the elliptic curve.

	0	1	2	3	4
0	A a	B b	C c	D d	E e
1	F f	G g	H h	I i	J j
2	K k	L l	M m	N n	O o
3	P p	Q q	R r	S s	T t
4	U u	V v	W w	X x	Y y
5	Z z	,	.	?	

Table 3.1 Alphabetic table

For this purpose we consider the respective co-ordinates of the respective character. All the coordinate points should be on the surface of the elliptic curve**.** The process with these alphabetic tables.

3.5.1 Algorithm: Alphabetic table_Value_Assign

Step 0 : *Generate appropriate alphabetic table.*
Step 1: *Use an appropriate data structure to store the text to be encrypted.*
Step 2: *Read the table in row-major from and find the Character stored in that position.*
Step 3: *Note the row and column values.*
Step 4: *Assign these values to the same character in Positions it appears.*

ENCRYPTION
Step 1: User A selects P, a point on the curve, as a plaintext
Step 2: Then calculates a pair of points on the text as Cipher texts: C1=r x e1 and C2=P + r x e2.

DECRYPTION
Step 1: User B, after receiving C1 and C2, calculates P,the plaintext using the following formula, P=C2-(d x C1). The Minus sign here means adding the inverse.
Step 2 : Prove that the P calculated by Bob is the same as that sent by Alice. P, C1, C2, e1, e2 are all points on the curve. Note the Result of adding two inverse points on the curve is the zero point.

Fig 3.5 Overall System

The security levels which are given by RSA can be provided by smaller keys of elliptic curve Cryptosystem as compared to RSA, which offers 1024 bit security strength, ECC offers the same in 160 bit key length. Efficiency of ECC is depends upon factors such as computational outlay, key size, band width, ECC provides higher-strength per- bit which include higher speeds, smaller power consumption, bandwidth reserves, storage efficiencies, and smaller certificates. For providing security mechanism will require fundamental basic security services such as authentication, confidentiality, non-repudiation and message integrity. The implementation ECC shows that it offers complete security solution.

4. EXPERIMENTAL AND RESULTS

4.1 Key generation

First, select the point value E(a,b)with an elliptic curve over GF(p) or GF(2n). Then choose the point on the alphabetic table corresponding to the letter as plain text and select the private key value as d. To calculate the point as

e2=(x2, y2) using the formula d*e1. Finally, announce e1, e2 as public key and keep "d" as a private key.

4.2 Encryption

User A select the point value p as plaintext and private key r for sender. Then calculate a pair of points on the text as cipher text. The cipher texts are as c1 and c2.

Hence, c1=r * e1 and c2=p + r * e2

4.3 Decryption

User B after receiving c1 and c2, calculates p, the plaintext using the formula as p=c2-(d*c1). Here minus sign means adding the inverse. To prove the point p calculated by receiver is the same as that by sender.

The proposed results were built on a java platform and implemented public key cryptography are connected in a ring network. The following algorithm takes file size of v kb as an input and it will calculate the execution time as the output. Here input message will be construct into the encrypt and decrypt. If it's not accepted its output it's failed. After this calculate the execution time if the text file is less than or equal to 99 kb. While it's calculate the execution time for a file if the file size is greater than or equal to 100 kb. The algorithms its depend on the process or speed. The execution time for different algorithms is as follows.

5. PERFORMANCE ANALYSIS

The proposed results were built on a java platform and implemented public key cryptography are connected in a ring network. The following algorithm takes file size of v kb as an input and it will calculate the execution time as the output. Here input message will be construct into the encrypt and decrypt. If it's not accepted its output it's failed. After this calculate the execution time if the text file is less than or equal to 99 kb. While it's calculate the execution time for a file if the file size is greater than or equal to 100 kb. In a table 2.1 the algorithms its depend on the process or speed. The execution time for different algorithms is as follows.

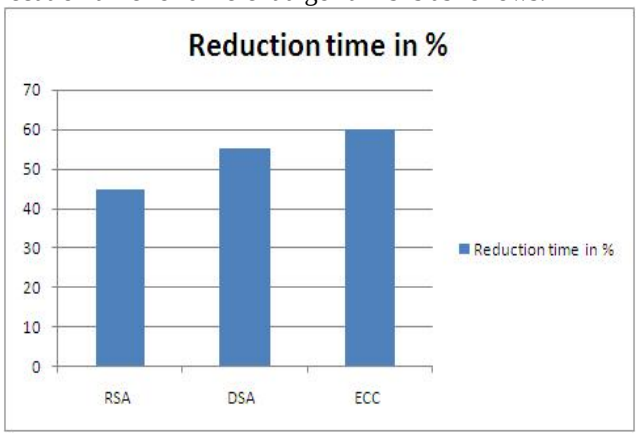

Fig 5.1 Performances Improvements of Various Algorithms

In the fig 5.1 results shows that the proposed system for implementation of various cryptography algorithms using java application programming interface has been reduced the execution time for algorithm from 40-60% for different algorithms. From the implementation results above three algorithms. It is taking very less time for execution time process compared to other algorithms. First, it is evident from Figure 5.2 that for the same key size, ECC generates keys much slower than RSA.

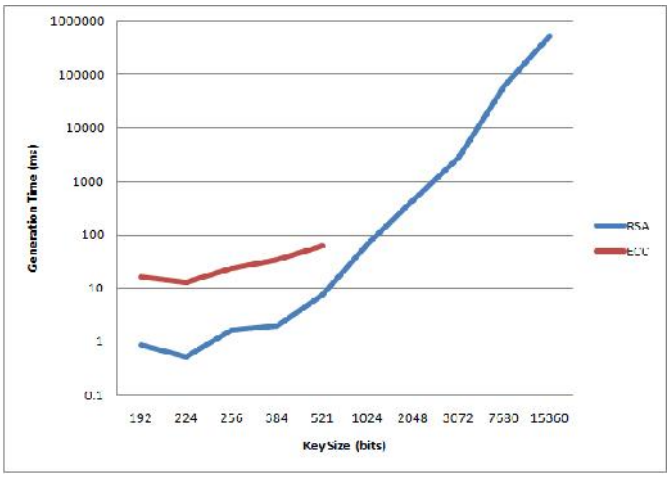

Fig 5.2 Key Generation Times by Key Size

The comparative strength of the keys is taken into consideration, as shown in Figure 5.3, ECC is not only faster than RSA in all cases but also shows a much shallower rate of increase in key generation times as opposed to RSA and the logarithmic scale for generation times. The definite shows that ECC is much faster than RSA in real world scenarios where key strength is more important than the key size itself.

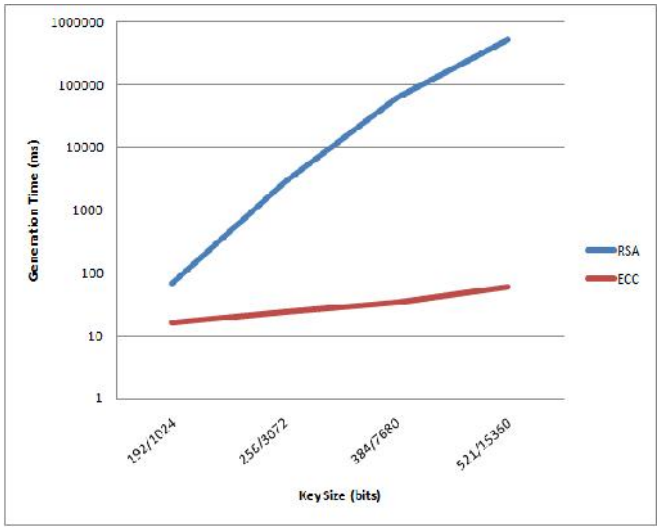

Fig 5.3 Key Generation by Key Strength

6. CONCLUSION

Elliptic curve cryptosystem becomes to be the cryptosystem for the future. One way to improve the performance of such cryptosystem is to use an efficient method for point multiplication which is the most time consuming operation. A study of the scalar multiplication methods to be used in elliptic curve cryptography. The addition-subtraction method decreases number of point additions that speed up the computation. Therefore in the future some efficient methods for point multiplication can be used to speed up the computation. To implement ECC with projective co-ordinate rather than affine co-ordinate system, this system may be fast. Encryption in mobile communication is very crucial to protect information of the subscribers and

avoid fraud. The security by means of elliptic curve cryptographic technique. Actual implementation of encryption and decryption using elliptic curve cryptography on GF (P) shows that a security that security of the proposed system is very hard. It has been mentioned in many literatures that a considerably smaller key size can be used for ECC compared to RSA. Also mathematical calculations required by elliptic curve cryptosystem are easier, hence, require a low calculation power. Therefore ECC is a more appropriate cryptosystem to be used on small devices like mobile phones.

7. REFERENCES

[1] Asha Rani Mishra "Elliptic Curve Cryptography (ECC) for Security in wireless Sensor Network", International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 3, May – 2012.

[2] Aarti Singh "Elliptical Curve Cryptography Based Security Engine for Multiagent Systems Operating in Semantic Cyberspace" International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 2, No. 2, April 2011.

[3] F. Amounas "An Efficient Elliptic Curve Cryptography protocol Based on Matrices" International Journal of Engineering Inventions ISSN: 2278-7461, Volume 1, Issue 9 (November2012) PP: 49-54.

[4] Pranita K. Gandhewar, Kapil N. Hande "Performance Improvement of IEEE 802.16 / Wimax Using Elliptic Curve Cryptography" (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (3) , 2011, 1309-1311.

[5] Haodong Wang, Bo Sheng and Qun Li "Elliptic curve cryptography-based access control in sensor networks" Int. J. Security and Networks, Vol. 1, Nos. 3/4, 2006.

[6] Jaspreet Singh, Er. Sandeep Singh Kang "Security Enhancement in WEP by Implementing Elliptic Curve Cryptography Technique" International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-5, November 2012.

[7] Kishore Rajendiran, Radha Sankararajan, and Ramasamy Palaniappan, "A Secure Key Predistribution Scheme for WSN Using Elliptic Curve Cryptography", ETRI Journal, Volume 33, Number 5, October 2011.

[8] Mohammed Aabed "Implementation of a pipelined modular multiplier architecture for GF(p) elliptic curve cryptography computation" Kuwait J. Sci. Eng. 38(2B) pp 125-153, 2011.

[9] Sumedha Kaushik, "Network Security Using Cryptographic Techniques" Volume 2, Issue 12, December 2012 ISSN: 2277 128X.

[10] Santoshi Ketan pote "Elliptic Curve Cryptographic Algorithm" Proc. of the Intl. Conf. on Advances in Computer Science and Electronics Engineering.