Nonce Based Elliptic Curve Cryptosystem for Text and Image Applications

S. Maria Celestin Vigila[1] and K. Muneeswaran[2]
(Corresponding author: S. Maria Celestin Vigila)

Department of Information Technology, Noorul Islam College of Engineering[1]
Kumaracoil-629 180, TamilNadu, India
Department of Computer Science and Engineering, Mepco Schlenk Engineering College[2]
Sivakasi-626 005, TamilNadu, India.
(Email: {celesleon, kmuni12}@yahoo.com)

(Received Aug. 5, 2010; revised and accepted Nov. 28, 2010)

Abstract

With the explosion of networks and the huge amount of data transmitted along, securing data content is becoming more and more important. Data encryption is widely used to ensure security in open networks such as the Internet. With the fast development of cryptography research and computer technology, the capabilities of cryptosystems such as of RSA and Diffie-Hellman are inadequate due to the requirement of large number of bits. The cryptosystem based on Elliptic Curve Cryptography is becoming the recent trend of public key cryptography. This paper presents the implementation of Elliptic Curve Cryptography by first transforming the message into an affine point on the Elliptic Curve, over the finite prime field. In this paper we illustrate the process of encryption/decryption of a text message and image files in spatial domain by enhancing security using Comparative Linear Congruential Generator for better random number generation. This enables the breaking of cipher text almost impossible for the brute force attack.

Keywords: Comparative linear congruential generator, discrete logarithm, elliptic curve cryptography, nonce

1 Introduction

With the popularity of computers and Internet, real time multimedia data is represented in digital forms to be transmitted on Internet. Digitized data can be texts, images, audios/videos. It is important to send the digital data securely. Cryptography is the science of converting data in non understandable form for the unintended viewers for securely transmitting messages between a sender and a receiver. The objective is to encrypt the message in a way such that an eavesdropper would not be able to read it. A cryptosystem is a system of algorithms for encrypting and decrypting the messages for this purpose. Randomness is a key ingredient for cryptography. Random bits are necessary not only for generating cryptographic keys, but are also often an integral part in the steps of cryptographic algorithms. The cryptosystem requires the generation of a new random number each time a new message is encrypted. Pseudo random generator plays a major role in a cryptographic algorithm for the overall security.

The strength of a public key cryptosystem is directly related to the type of the one-way function it uses and the length of the cryptographic keys. With the computing power and the theoretical knowledge available today, we find that inverting a one-way function by which the scalar multiplication has been done for the case of Elliptic Curve Cryptography (ECC) is a practically intractable problem. The major objective for public key cryptosystem is to enhance the security based this intractable computability.

The use of Elliptic Curves (EC) in public key cryptography was independently proposed by Koblitz and Miller in 1985 [8] and since then, an enormous amount of work has been done on elliptic curve cryptography. ECC based security offers a similar level of security that can be achieved with shorter keys than existing methods which are based on the difficulties of solving discrete logarithms over integers or integer factorizations.

Elliptic curve cryptography uses elliptic curves which are not ellipses in which the variables and the coefficients are all bound to elements of a finite field. In general EC is represented by a set of points in two dimensional Cartesian co-ordinate system. The transformation of the original message is converted into set of points in EC by which the encryption and decryption is achieved. The state space complexity is very high so that the intruders will not be able to interpret the encrypted message so easily. Also we have introduced a novel random number selection process based on Comparative Linear Congruential Generator (CLCG), which is secure and inexpensive

method to generate nonce for the ECC. For both encryption and decryption the keys are represented over the EC field, promising maximum security.

2 Related Works

In the literature, many authors have tried to exploit the features of EC field to deploy for security applications. We have outlined some of the highlights of the relevant work in this section. Aydos *et al.* [1] has implemented ECC on an 80 MHz, 32 bit RAM microprocessor over the field $GF(p)$ and demonstrated the results. We have implemented the text based cryptosystem using ECC over the field $GF(p)$ which is presented in [17] along with the results. Lauter has focused on the use of ECC for the performance advantages in the wireless environment over the traditional RSA cryptosystem [10]. Ray has discussed the design of a generator, for producing the customized ECC hardware meeting user-defined requirements automatically in [3]. The importance of ECC is explained by Cilardo *et al.* as a complex interdisciplinary research field encompassing such fields as mathematics, computer science and electrical engineering in [4]. A novel hardware architecture for ECC over $GF(p)$ was introduced by McIvor *et al.* [14]. Chen presents a high performance EC cryptographic process for general curves over $GF(p)$ [2]. The standard specifications for public key cryptography are defined in [16].

Stallings *et al.* [15] has presented a simple tutorial of ECC concept which is very well documented and illustrated in his book. A brute-force attack on ECC implemented on UC Berkley's Tiny OS operating system for wireless sensor networks is explained by Finnigin *et al.* in [5]. The short period of the pseudorandom number generators used by cryptosystem to generate private keys was exploited. Moon [13] proposed an efficient and novel approach of a scalar point multiplication method which is an efficient version of the existing double and add by eliminating redundant recoding. This method was originated from radix-4 Booths algorithm. Lee [11] presented three algorithms to do scalar multiplication on EC defined over higher characteristic finite fields such as Optimal Extension Field. Yongliang [12] demonstrated that the protocol proposed by Aydos *et al.*' could be subjected to man-in the-middle attack from any attacker but not restricted on the inside attacker. In-depth mathematical treatment with a comprehensive coverage of EC field is given in [9]. Owing to these existing works on ECC and its popularity, it is proposed to implement the nonce based crypto system based on ECC for text and image based application.

3 Mathematical Background

The hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP) enables ECC operates on groups of points over EC for security. While sub-exponential algorithms are suitable for solving the integer factorization problem, only exponential algorithms are known for the ECDLP. Hence ECC is able to achieve the same level of security with smaller key sizes and higher computational efficiency.

3.1 Elliptic Curves

An elliptic curve takes the general form as:

$$E : y^2 = x^3 + ax + b, \tag{1}$$

where x, y are co-ordinates of $GF(p)$, and a, b are integer modulo p, satisfying

$$4a^3 + 27b^2 \neq 0(\mathrm{mod}p). \tag{2}$$

Here "p" is modular prime integer which makes the EC of finite field. An elliptic curve E over $GF(p)$ consist of the points (x, y) defined by Equations (1) and (2), along with an additional point called O (point at infinity) in EC. These points are said to be affine points.

3.2 Elliptic Curve Arithmetic

Point addition and point doubling are the basic EC operations. ECC primitives [16] require scalar point multiplication. Let P is a point with the co-ordinates x, y on an EC, and one needs to compute kP, where k is a positive integer. This scalar multiplication can be done by a series of doubling and addition of P. For example, given $k = 13$, entails the following sequence of operations, by which the efficiency of the scalar multiplication of the points is improved (See Table 1).

Let us start with $P(x_P, y_P)$. To determine $2P$, P is doubled. This should be an affine point on EC. Use the following equation, which is a tangent to the curve at point P.

$$S = \lfloor (3x_P^2 + a)/2y_P \rfloor \bmod p.$$

Then $R = 2P$ that has affine coordinates (X_R, Y_R) given by:

$$\begin{aligned} X_R &= (S^2 - 2X_P) \bmod p \\ Y_R &= (S(X_P - X_R) - Y_P) \bmod p. \end{aligned}$$

In order to determine $3P$, we use addition of points P and $2P$, treating $2P = Q$. Here P has coordinates (x_P, y_P). $Q = 2P$ has coordinates (x_Q, y_Q). Now the slope is:

$$\begin{aligned} S &= \lfloor (Y_Q - Y_P)/(X_Q - X_P) \rfloor \bmod p \\ P + Q &= -R \\ X_R &= (S^2 - X_P - X_Q) \bmod p \\ Y_R &= (S(X_P - X_R) - Y_P) \bmod p. \end{aligned}$$

Therefore we apply doubling and addition depending on a sequence of operations determined for "k". Every point (x_R, y_R) evaluated by doubling or addition is an affine point (points on the Elliptic Curve).

Table 1: An example (given $k = 13$)

P	2P	3P	6P	12P	13P
	Doubling	Addition	Doubling	Doubling	Addition

4 Pseudo Random Number Generation

4.1 Linear Congruential Generator

The recurrence relation $x_{i+1} = ax_i + b \bmod m$, where a, b and m are known and x_0 is secret defines a Linear Congruential Generator (LCG). If the length of the sequence generated by LCG is m then it is said to be full period. Also the LCG is said to have a fixed point (this implies that there exists i such that $x_{i+1} = x_i$) when $(1-a)^{-1} \bmod m$ exists. When this occurs the maximum period of the sequence is $m-1$, if the fixed point is not used as an initial condition. The maximum period occurs when the following conditions are satisfied.

1) b and m are relatively prime.

2) $(a-1)$ is divisible by every prime factor of m.

3) $(a-1)$ is divisible by 4 if 4 divides m.

Shamir and Hastad [6] have argued that LCG is an insecure method. It is possible to recover the seed x_0 if at least 1/3 of the leading bits of three consecutive numbers in the sequence are known.

4.2 Coupled Linear Congruential Generator

Coupled or Comparative LCG (CLCG) [7] is defined as follows:

$$\begin{aligned} x_{i+1} &= ax_i + b \bmod m \\ y_{i+1} &= cy_i + d \bmod m \\ z_{i+1} &= \begin{cases} 1, & \text{if } x_{i+1} > y_{i+1}; \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Example 4.1 *Let $a = 5$, $b = 5$, $c = 3$, $d = 2$, and $m = 8$. Both sequences, x_i and y_i have a period of 8 and are hence full period. If the initial condition (or the seed) is (x_0, y_0) = (3, 6), then the sequences are,*

$$\begin{aligned} \{x_i\} &= (4, 1, 2, 7, 0, 5, 6, 3) \\ \{y_i\} &= (4, 6, 4, 6, 4, 6, 4, 6). \end{aligned}$$

The bit sequence z_i therefore is

$$\{z_i\} = (0, 0, 0, 1, 0, 0, 1, 0).$$

Assume that a, b, c, d, m are known and the seed (x_0, y_0) is secret. Let us analyze the way of guessing the initial condition or seed (x_0, y_0) of coupled LCGs given the bit sequence z_i. The numbers (x_i, y_i) are taken to be positive integers between 0 and $(m-1)$ for the computation of z_{i+1}. This makes the computation of x_{i+1} and y_{i+1}, from z_{i+1} very difficult. It is easy to see that the kth output of an LCG $x_{i+1} = ax_i + b \bmod m$, is given as:

$$x_k = a^k x_0 + b \sum_{i=0}^{k-1} a^i \bmod m.$$

This implies that if the kth output of the coupled LCGs is z_k, then the following inequality holds based on whether z_k is 1 or 0.

$$\begin{aligned} a^k x_0 + b \sum_{i=0}^{k-1} a^i \bmod m \quad &> \quad c^k x_o + d \sum_{i=0}^{k-1} c^i \bmod m \\ & \qquad if \quad z_k = 1 \\ a^k x_0 + b \sum_{i=0}^{k-1} a^i \bmod m \quad &\leq \quad c^k x_o + d \sum_{i=0}^{k-1} c^i \bmod m \\ & \qquad if \quad z_k = 0. \end{aligned} \tag{3}$$

The number of inequalities that can be set as above can be the maximum number of bits that are known from the z_k. For example if u bits of the output z_k are known, then only u number of inequalities can be set up as E_k, $1 \leq k \leq u$, where E_k is an inequality of the form described above.

4.3 Advantage of the CLCG problem

Solving of the CLCG problem is difficult since it requires solving inequalities of the form specified by Equation (3). If the inequalities are converted into equalities then lattice like methods can be used to obtain the solutions. The main difficulty in solving with inequalities is the fact that there is no ordering over integers modulo m. This is because $x \bmod m$ can be both less than and greater than another integer y. The reason for this is the fact that y and $y - m$ are congruent.

Another difficulty with an inequality of the form given in Equation (3) is the fact that it cannot be manipulated. This implies that this inequality cannot be converted to the following form:

$$ax + b - cy - d \bmod m > 0 \bmod m.$$

Even if is converted it would lead to incorrect solutions for (x, y). Also lattice methods for solving modular equalities lead to exponential complexity with respect to the size of the problem ($\log m$ is the input size).

In the following section, we demonstrate how the CLCG system can be extended to enhance the security of the elliptic curve cryptosystem further.

5 ECC and CLCG

To do operations with EC points in order to encrypt and decrypt the points are to be generated first. The algorithm 'genPoints' describes the process of generating the points for the given parameters 'a', 'b', and 'p'. Also the algorithm 'ECC' describes the process of encryption and decryption on EC field. Here, the nonce in the elliptic curve cryptosystem is generated using a CLCG.

Algorithm 1 Algorithm genPoints (a, b, p)

```
{
  x = 0;
  While(x < p)
    y^2 = (x^3 + ax + b) mod p;
    if (y^2 is a perfect square in GF(p))
      output(x, sqrt(y)) (x, -sqrt(y));
    x = x + 1;
}
```

Algorithm 2 Algorithm ECC

```
{
//Key Distribution
  //Let U_A and U_B be legitimate users
    U_A = {P_A, n_A} //Key pair for U_A
    U_B = {P_B, n_B} //Key pair for U_B
  //Send the Public key of U_B to U_A
    Send(P_B, U_A);
  //Send the Public key of U_A to U_B
  Send (P_A, U_B);

//Encryption at A
    P_m1 = aP_m
  //a: Ascii value of text
  //P_m: random point on EC
    P_B = n_B × G
  //G is the base point of EC
   //n_B is the private key
    CipherText={kG, P_m1 + k × P_B}
  //k is nonce generated by CLCG

//Decryption at B
   Let kG be the first point and P_m1 + k × P_B be the
second point
    n_B kG = n_B× first point;
    Calculate P_m1 = P_m1 + kP_B − n_B kG;
     Calculate the P_m value from P_m1 using discrete
logarithm
}
```

6 Implementation of The Proposed Algorithm

For demonstration purposes typical Elliptic Curve is represented by:

$$y^2 \bmod 37 \quad = \quad x^3 + x + 1 \bmod 37,$$

where $a = 1$, $b = 1$ and $p = 37$. The generated points on the curve can be found as shown in Table 2.

Table 2: Set of sample points on EC

(0, 1)	(0, 36)	(21, 25)	(21, 12)
(1, 15)	(1, 22)	(24, 14)	(24, 23)
(2, 14)	(2, 23)	(25, 0)	(25, 0)
(6, 36)	(6, 1)	(26, 18)	(26, 19)
(6, 36)	(6, 1)	(26, 18)	(26, 19)
(8, 15)	(8, 22)	(27, 8)	(27, 29)
(9, 31)	(9, 6)	(28, 15)	(28, 22)
(10, 7)	(10, 30)	(29, 31)	(29, 6)
(11, 14)	(11, 23)	(30, 24)	(30, 13)
(13, 18)	(13, 19)	(31, 36)	(31, 1)
(14, 24)	(14, 13)	(33, 9)	(33, 28)
(17, 11)	(17, 26)	(35, 18)	(35, 19)
(19, 16)	(19, 21)	(36, 6)	(36, 31)

The base point G is selected as (0, 1). Base point implies that it has the smallest (x, y) co-ordinates which satisfy the EC. Pm is another affine point, which is picked out of a series of affine points evaluated for the given EC. We could have retained G itself for Pm. However for the purpose of individual identity, we choose Pm to be different from G. Let $P_m = (1, 15)$. Varying values of Pm can be chosen as part of an exercise to work with ECC process on the given EC.

In the ECC method, we generate a nonce, i.e a random integer k $(k < p)$, which needs to be kept secret. Then kG is evaluated, by a series of additions and doublings, as discussed above. Let us call the source as A and destination as B. Let the private key of the host B be n_B. The values of k and n_B are generated by random a number generator that is CLCG to give credibility.

Let $a = 5$, $b = 4$, $c = 3$, $d = 1$, and $m = 8$. Both sequences, x_i and y_i have a period of 8 and are hence full period. If the initial condition (or the seed) is $(x_0, y_0) = (1, 0)$, then the sequences are:

$$\begin{aligned} \{x_i\} &= (1,1,1,1,1,1,1,1) \\ \{y_i\} &= (1,4,5,0,1,4,5,0). \end{aligned}$$

The bit sequence z_i therefore is

$$\{z_i\} \quad = \quad (0,0,0,1,0,0,0,1).$$

The decimal equivalent values are $n_B = 17$. Similarly the value k is generated as $k = 13$. The public key of user B

is evaluated by

$$P_B = n_B G.$$

Suppose A wants to encrypt and transmit a character to B, it does the following. Assume that host A wants to transmit the character '♯'. Then the ASCII value of the character '♯' is 35. Therefore,

$$\begin{aligned} P_B &= n_B G = 17(0,1) = (21,12) \\ P_{m1} &= 35(1,15) = (2,14). \end{aligned}$$

The computed coordinate (P_{m1}) should fit into the EC. This conversion is done for two reasons. First the ASCII key representation of the text message is mapped into a (x, y) co-ordinate of the EC. Second it will be completely hidden from the hacker. These steps are introduced to add some level of complexity even before the message is encrypted according to ECC.

Next the generated random number using CLCG k and the public key P_B are multiplied, which is carried out a series of doubling and additions, depending on the value of k. Efficient procedure can be adapted for optimal number of doublings and additions.

$$\begin{aligned} kP_B &= 13(21,12) = (21,12) \\ P_{m1} + kP_B &= (2,14) + (21,12) = (30,24) \\ kG &= 13(0,1) = (0,1). \end{aligned}$$

The encrypted message is derived by adding P_{m1} with kP_B, that is, $P_{m1} + kP_B$. This yields a set of (x_2, y_2) coordinates. Then kG is included as the first element (x_1, y_1) of the encrypted version. Hence the entire encrypted version for purposes of storing or transmission consists of two sets of coordinates as follows:

$$\begin{aligned} Cm &= (kG, p_{m1} + kP_b) \\ kG &= x_1, y_1 \\ p_{m1} + kP_B &= x_2, y_2. \end{aligned}$$

Encrypted version of the message is: (0, 1), (30, 24), where $x_1 = 0$, $y_1 = 1$, $x_2 = 30$, $y_2 = 24$. Thus the modified plaintext has been encrypted by application of the ECC method. The selection of random i.e. the secret number dictates the complexity of encryption algorithm for breaking. We have introduced a novel random number selection process based on CLCG, which is not a part of any of the existing work on ECC.

Recall that kG is represented by (x_1, y_1) and $P_{m1}+kP_B$ is represented by (x_2, y_2). In order to pull out P_{m1} from $P_{m1} + kP_B$, B applies his/her secret key n_B and multiplies kG so that, $n_B kG = kP_B$. Subtract this from $P_{m1}+kP_B$, to get P_{m1} that is, $P_{m1} = P_{m1}+kP_B-n_B kG$.

$$\begin{aligned} n_B kG &= 17(0,1) = (21,12) \\ p_{m1} &= (30,24) - (21,12) = (2,14). \end{aligned}$$

This subtraction is nothing but another ECC procedure involving doubling and addition with difference having its y co-ordinate preceded by a minus sign. Hence the determination of the new values of x_R, y_R follows the same procedure. This will yield P_{m1}. Now apply discrete logarithm concept to get the ASCII value of "♯".

♯ (1, 15) = (2, 14). Since the ASCII value of ♯ is 35, we retrieve the character '♯'.

7 Image Encryption

In the previous sections we have demonstrated the generation of EC points, encryption of a character and decryption of the same with an aid of simple example. We selected a random which is lesser than a small prime number. Now, the ECC based text encryption is extended to an image encryption in spatial domain. Here we apply the encryption algorithm in the pixel values of an image directly. NIST standard has listed a collection of recommended elliptic curves, with the private key lengths and underlying fields. It specifies how to represent field elements and provides for random generated curves and selected curves.

The website www.secg.org provides the complete properties of the recommended Elliptic Curve domain parameters over Fp. We have used the parameter as specified in secp256r1 in the standard. Here pseudo random curve 256-bit Elliptic Curve over Fp is chosen for image encryption.

The various EC parameters are p, a, b, S, gx, gy, and n which are implemented as Big Integer since they are all 256 bits wide and are detailed as follows:

p: prime number

a, b: Elliptic Curve parameters

S: Seed used to generate the parameter

g_x, g_y: Coordinates of the base points

n: the order of base point

The software implementation of the elliptic curve cryptosystem is done using Java. Figure 1 shows the result of encryption and decryption for the image by using ECC technique with the parameters specified in secp256r1 and are tabulated in Table 3 as Bird2.jpg.

(a) Image before encryption (b) Image after decryption

Figure 1: Bird2.jpg

The Table 3 illustrates the original image size and execution time i.e., both encryption time and decryption time taken for various image files using ECC in addition to the example shown in Figure 1.

Table 3: Encryption and decryption time of various images

Image File name	Image Size (rows cols, colors)	Enc. Time (ms)	Dec. Time (ms)
Building1.jpg	779×118×3	7078	4125
Building2.jpg	132×93×3	7078	4156
Bird1.jpg	274×373×3	38750	22656
Bird2.jpg	390×545×3	64171	37907

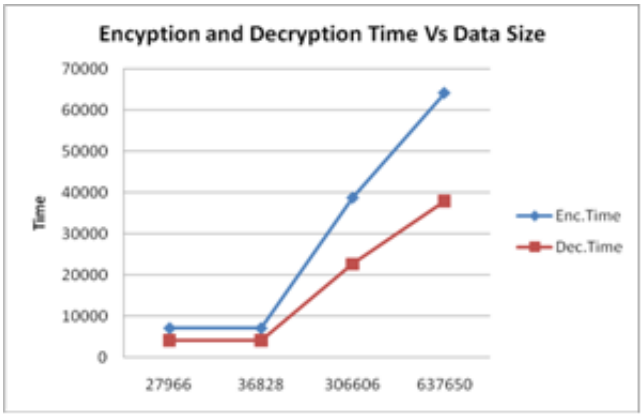

Figure 2: The graphical representation of encryption and decryption time (millisecond) vs. the data size (bytes)

Figure 2 illustrates the actual encryption and decryption time of various images. X-axis shows the size of the image in bytes and Y-axis shows the time taken in milliseconds. The decryption time is lesser than the encryption time due to the high computational complexity involved in the encryption process.

8 Security Analysis

Recent computing power is capable breaking encryption schemes in a real time if the system is not designed to look into these issues. Hence a good encryption scheme should keep away from the possible attacks. The attacks are varying in nature such as statistical attack, brute force attack and so on. Hence analysis of encryption schemes such as statistical analysis, key space analysis ensures right development of the security system.

8.1 Key Space Analysis

The key space that is being used for encryption must be large enough to prevent the brute force attackers to intrude. For the proposed encryption algorithm, key space analysis is carried out as follows.

- **Key Space**
  In our example we have generated the keys as per specification given in secp256r1. It has 2^{256} different combinations of secret key. Hence for this image encryption this large key space (using 256 bits) is sufficient which is immune to all kinds of brute force attacks.

- **Key Sensitivity Test**
  Even a change in a single bit of key will make a completely different encrypted text/image for the intruders to guess the key. This makes the encryption procedure sensitive enough to the secret key.

8.2 Statistical Analysis

Statistical analysis generally depends on the measure of the randomness of the cipher media. Also it works on the relative frequency of the occurred cipher text. Since we have used a novel random number generator based on CLCG the strength of the proposed encrypted system is high compared to the existing methods.

9 Discussions & Conclusion

In this paper, a Nonce based Elliptic Curve Cryptosystem is proposed. The encryption and decryption of the text with simple example is demonstrated. Also the work is extended to the image applications. For the text applications, each character in the message is represented by its ASCII value. Each of these ASCII value is transformed into an affine point on the EC, by using a starting point called Pm. Transformation of the plaintext ASCII value by using an affine point and introducing the modified random number generator are the contributions of our work. The purpose of this transformation is twofold. The ASCII value representation of the character is converted into a set of co-ordinates to fit the EC. This introduces nonlinearity in the character thereby completely disguising its identity. This converted character of the message is encrypted by the ECC technique. The decryption of the encrypted message is itself quite a difficult task, unless the knowledge about the private key 'n_B', the secret integer 'k' and the affine point P_{m1} is known. It is shown that CLCGs are good candidates for nonce generation in the elliptic curve cryptosystem in terms of both security and computational efficiency thereby the drawbacks of the single LCG are removed. Therefore, the task of breaking CLCGs is computationally infeasible for very large value of 'm'.

Hence Elliptic curve based cryptosystem exhibits its power and is suitable for the next generation public key cryptosystem. The literature elucidates the fact that ECC offers same level of security for a smaller key size compared to RSA, thereby reducing processing overhead. ECC based cryptosystem offers benefits such as higher strength per bit leading to faster computation, reduced power consumption and less storage requirements. These features enable the implementation of security system in smart cards, mobile phone and any other tiny devices.

From Table 3, it can be observed that the encryption and decryption time realization is high, in real time

applications such as multimedia data. Hence strategies must be adopted to lower the execution time or parallel/distributed computing environment can be used to enhance the computing power.

Acknowledgements

The authors are grateful to the principal and management of Noorul Islam College of Engineering and MEPCO Schlenk Engineering College for extending all the facilities and constant encouragement for carrying out this research work.

References

[1] M. Aydos, T. Yanik, and C. K. Kog, "High-speed implementation of an ECC based wireless authentication protocol on an ARM microprocessor," *IEE Proceeding Communications*, vol. 148, no. 5, pp. 273-279, Oct. 2001.

[2] G. Chen, G. Bai, and H. Y. Chen, "A high-performance elliptic curve cryptographic processor for general curves over $GF(p)$ based on a systolic arithmetic unit," *IEEE Transactions on Circuits and Systems Part II: Express Briefs*, vol. 54, no. 5, pp. 412-416, May. 2007.

[3] R. C. C. Cheng, N. Jean-Baptiste, W. Luk, and P. Y. K. Cheung, "Customizable elliptic curve cryptosystems," *IEEE Transactions On VLSI Systems*, vol. 13, no. 9, pp. 1048-1059, Sep. 2005.

[4] A. Cilardo, L. Coppolino, N. Mazzocca, and L. Romano, "Elliptic curve cryptography engineering," *Proceedings of the IEEE*, vol. 94, no. 2, pp. 395-406, Feb. 2006.

[5] K. M. Finnigin, B. E. Mullins, R. A. Raines, H. B. Potoczny, "Cryptanalysis of an elliptic curve cryptosystem for wireless sensor networks," *International Journal of Security and Networks*, vol. 2, no. 3/4, pp. 260-271, 2006.

[6] J. Hastad and A. Shamir, "The cryptographic security of truncated linearly related variables," *Proceedings of the 17th annual ACM symposium on Theory of Computing*, pp. 356-362, 1985.

[7] R. S. Katti and R. G. Kavasseri, "Secure pseudorandom bit generation using coupled linear congruential generators," *IEEE International Symposium on Circuits, and Systems*, pp. 2929-2932, 2008.

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

[9] R. V. Kurja, K. Joshi, N. M. Kumar, K. H. Raranape, A. Ramanathan, T. N. Shorey, R. R. Simha, and V. Srinivas, *Elliptic Curves*, International Distribution by American Mathematical Society, 2006.

[10] K. Lauter, "The advantages of elliptic cryptography for wireless security," *IEEE Wireless Communications*, pp. 62-67, Feb. 2006.

[11] J. Lee, H. Kim, Y. Lee, S. M. Hong, and H. Yoon, "Parallelized scalar multiplication on elliptic curves defined over optimal extension field," *International Journal of Network Security*, vol. 4, no. 1, pp. 99V106, Jan. 2007.

[12] Y. Liu, W. Gao, H. Yao, and X. Yu, "Elliptic curve cryptography based wireless authentication protocol," *IJNS*, vol. 4, no. 1, pp. 99-106, Jan. 2007

[13] S. Moon, "A binary redundant scalar point multiplication in secure elliptic curve cryptosystems," *International Journal of Network Security*, vol. 3, no. 2, pp. 132-137, Sep. 2006.

[14] C. J. McIvor, M. McLoone, and J. V. McCanny, "Hardware elliptic curve cryptographic processor over $GF(p)$," *IEEE Transactions on Circuits and Systems*, vol. 53, no. 9, pp. 1946-1957, Sep. 2006.

[15] W. Stallings, *Cryptography, and Network Security*, Prentice Hall, 4th Edition, 2006.

[16] *Standard Specifications for Public key cryptography*, IEEE Standard, P1363, 2000.

[17] S. M. C. Vigila and K. Muneeswaran, "Implementation of text based cryptosystem using elliptic curve cryptography," *IEEE International Conference on Advanced Computing*, pp. 82-85, Dec. 2009.

S. Maria Celestin Vigila completed the B.E. degree in Computer Science and Engineering in 1996 and the M.E. degree in Computer Science and Engineering in 1999. She is currently pursuing her research in the area of Information Security under Anna University, Tiruchirappalli. She is presently Assistant Professor in the Department of Information Technology, Noorul Islam College of Engineering, Kumaracoil and a member of ISTE and IET. Her research interest includes Cryptography and Network Security, Wireless Networks and Information Hiding.

K. Muneeswaran is Professor and Head of the Department of Computer Science and Engineeering, Mepco Schlenk Engineering College, Sivakasi. His area of interest includes image analysis, computer networks, neural networks, security, grid and cloud computing. Seven research scholars are working under his supervison. He contributed to many funded research projects. Also he is the reviewer for the peer reviewed International journals.