Window Method Based Cubic Spline Curve Public Key Cryptography

Addepalli V. N. Krishna[1], Addepalli Hari Narayana[2], K. Madhura Vani[3]
(Corresponding author: Addepalli V. N. Krishna)

The Department of Computer Science & Engineering, Faculty of Engineering, Christ University[1]
Bangalore, Karnataka - 560 029, India
The Department of Electrical Engineering, Indian Institute of Technology[2]
Indore, Madhya Pradesh- 453 331, India
The Department of Computer Science & Engineering, Shreyas Institute of Engineering & Technology[3]
Hyderabad, Telangana - 500 068, India
Email: hari_avn@rediffmail.com

(Received Jan. 16, 2015; revised and accepted Aug. 28, 2015)

Abstract

In this work, a cubic spline curve is considered for Asymmetric mode encrypting data. A steady state, one dimensional equation is integrated over a control volume. The derivatives of above equation form piece wise linear profile, which leads to discretization equation with corresponding weights. These weights are initially solved to generate global variables. Those global variables are used to calculate public key which in turn use the ElGamal mode of encryption. The proposed algorithm supports the features like Authenticity of users, Security & Confidentiality of data transmitted. Going by the construction of the algorithm, Encryption is being done on blocks of data for which it consumes less computing resources. Going by complexity of the algorithm, the key length needed is about 120 bit length to provide sufficient strengths against cryptanalysis.

Keywords: Cubic Spline Curve, Public Key Cryptography, Window Method

1 Introduction

Any symmetric encryption scheme uses a private key for secure data transfer [12]. In their work on a new Mathematical model on encryption scheme for secure data transfer [6], the authors considered not only key but also time stamp and nonce values to increase the strength of sub key generated. In addition the nonce value can also be used for acknowledgement support between participating parties. The model can be further improved by considering a non linear model where the key values vary with the data generated [7].

Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA today [8]. Recently, Elliptic Curve Cryptography has begun to challenge RSA. The principal attraction of ECC, compared to RSA, is that it appears to offer better security for a smaller key size, thereby reducing processing overhead.

Some recent works on application of ECC are cited here. [4] explains the engineering of ECC as a complex interdisciplinary research field encompassing such fields as mathematics, computer science and electrical engineering. [1] presents a high performance EC cryptographic process for general curves over GF(p). [5] specifies the standard specifications for public key cryptography. Encryption to data supports the very important features like security, Confidentiality to data & Authentication of users. [10, 13] discussed the features of Numerical data analysis which helps in building a mathematical model. [14] discusses computing complexity of ECC and [9, 11] identifies additional features with ECC which makes it more secure.

The work is divided into following modules: 1) Converting Mathematical models to Cryptosystems; 2) Generating Public and Private Keys; 3) Encryption and Decryption Process; 4) Cryptanalysis and Complexity of the Model; 5) Conclusion.

2 Converting Mathematical Model to Cryptosystems

2.1 Introduction To Cubic Spline Curve

The basic idea behind cubic spline is to draw smooth curves through number of points [3]. The spline consists of weights attached to a flat surface at the points to be connected. A flexible strip is then bent across each of these weights resulting in a pleasant smooth curve. These weights are the coefficients on cubic polynomials used to interpolate the data. The essential idea is to fit a piece wise function of the form

$$S(x) \quad = \quad \begin{cases} S_1(x) & x_1 \le x \le x_2 \\ S_2(x) & x_2 \le x \le x_3 \\ \vdots & \\ S_{n-1}(x) & x_{n-1} \le x \le x_n \end{cases}$$

where S_i is a third degree polynomial defined by

$$S_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \;\text{ for }\; i = 1, 2, 3, \cdots, n - 1.$$

The first & second derivations of degree $n - 1$ equations are fundamental to this process. They are

$$\begin{aligned} S_i^{'}(x) &= 3a_i(x - x_i)^2 + 2b_i(x - x_i) + c_i \\ S''{}_i(x) &= 6a_i(x - x_i)^2 + 2b_i, \quad \text{for}\;\; i = 1, 2, 3, \cdots, n - 1. \end{aligned}$$

On summing the equation based on the fact that $S'(x)$ & $S''(x)$ are continuous across interval, the coefficients for $(n - 1)$ equations can be calculated from $a_i\, b_i\, c_i\, d_i$ and $S''_i(x_1) = M_i$ which can be represented in matrix form as follows. We can consider three cases,

1) Natural splines, $M_1 = M_n = 0$;

2) Parabolic runout spline:

$$\begin{aligned} M_1 &= M_2; \\ M_{n-1} &= M_n. \end{aligned}$$

3) Cubic runout spline:

$$\begin{aligned} M_1 &= 2M_2 - M_3; \\ M_n &= 2M_{n-1} - M_{n-2}. \end{aligned}$$

2.2 Numerical Models of Steady State Equations

Steady state one-dimensional equation is given by $\frac{\partial}{\partial x}\left(K.\frac{\partial T}{\partial x}\right) + s = 0.0$, where K&s are constants. To derive the discretization equation we shall employ the grid point cluster. We focus attention on grid point P which has grid points E, W as neighbors. For one dimensional problem under consideration we shall assume a unit thickness in y and z directions. Thus the volume of control volume is delx*1*1. Thus if we integrate the above equation over the control volume, we get $\left(K.\frac{\partial T}{\partial X}\right)_e \left(K\frac{\partial T}{\partial X}\right)_w + \int S\,\partial X = 0.0$. If we evaluate the derivatives $\partial T/\partial X$ in the above equation from piece wise line ar profile , the resulting equation will be

$$K\,e\,\frac{(T_e - T_p)}{(\partial X)_e} - K\,W\frac{(T_p - T_W)}{(\partial X)_e} + S^{*}\,\mathrm{del}\,x = 0.0$$

where S is average value of s over control volume.

2.2.1 Linear Data Flow Problem

Dividing the problem area into M number of points, and for simplicity by assuming data of the 1^{st} and M^{th} grid points are considered to be known and constant. For the grid points $2, M - 1$, the coefficients can be represented by considering the conservation equation,

$$\frac{\alpha}{\partial x}(T_{I+1}^{n+1} - T_I^{n+1}) + \frac{\alpha}{\partial x}(T_I^{n+1} - T_{I-1}^{n+1}) = \frac{\partial x}{\partial t}(T_I^{n+1} - T_I^{n}) \tag{1}$$

where T_i represents data value for the considered grid point for the preceding delt, T_{I+1}^{n+1} & T_I^{n+1} represents data values for the preceding and succeeding grid points for the current delt.

Considering α for the given model, the coefficients are obtained for each state (grid point) in terms of $A(I)$ refers to data value of the corresponding grid point, $C(I)$ and $B(I)$ refers to data values of preceding and succeeding grid points for the current delt, $D(I)$ refers to data value of the considered grid point in the preceding delt.

$$\begin{aligned} A(I) &= 1 + 2\alpha \frac{\mathrm{del\,t}}{(\mathrm{del\,x})^2} \\ B(I) &= -\alpha \frac{\mathrm{del\,t}}{(\mathrm{del\,x})^2} \\ C(I) &= -\alpha \frac{\mathrm{del\,t}}{(\mathrm{del\,x})^2} \\ D(I) &= T_I^n, \end{aligned}$$

where α is a constant value, the model generated is a linear model. For $I = 1, 2, 3, \cdots, n_i$. Thus the data value T is related to neighboring data values T_{i+1} and T_{i-1}. For the given problem $C_1 = 0$ and $B_n = 0$. The coefficients are arranged in matrix form. As both sides of the curve are maintained at same data points, the coefficients are represented as

$$\begin{bmatrix} a_2 & b_2 & & & & \\ & a_3 & b_3 & c_3 & & \\ & & a_4 & b_4 & c_4 & \\ & & & \vdots & & \\ & & & & c_{n-1} & a_{n-1} \end{bmatrix} = \begin{bmatrix} d_2 \\ d_3 \\ d_4 \\ \vdots \\ d_{n-1} \end{bmatrix}$$

These conditions imply that T_1 is known in terms of T_2. The equation for $I = 2$, is a relation between T_1, T_2, & T_3. But since T_1 can be expressed in terms of T_2, this relation reduces to a relation between T_2 and T_3. This process of substitution can be continued until T_{n-1} can be formally expressed as T_n. But since T_n is known we can obtain T_{n-1}. This enables us to begin back substitution process in which $T_{n-2}, T_{n-3}, \cdots, T_3, T_2$ can be obtained. For this tri-diagonal system, it is easy to modify the Gaussian elimination procedures to take advantage of zeros in the matrix of coefficients [2].

2.2.2 Modelling of Cubic Spline Curve Problem

Global Parameters:

$$\begin{aligned} T_1 &= T_N = \text{First \& last data points} \\ w &= \text{wit=dht of the curve considered} \\ G &= \text{base Sequence considered} \\ t &= \text{private key considered} \\ r &= \text{random number generated} \\ p &= \text{field considered.} \end{aligned}$$

For the 2 points on the curve,

$$\begin{aligned} B_2 &= -\alpha \frac{\Delta t}{\Delta x^2} \bmod P \\ A_2 &= 1 + 2\alpha \frac{\Delta t}{\Delta x^2} \bmod P \\ D_2 &= Y(2) + \alpha \frac{\Delta t}{\Delta x^2} * D(1) \bmod P. \end{aligned}$$

For the 3 to $n-2$ points on the curve,

$$\begin{aligned} B(I) &= -\alpha \frac{\Delta t}{\Delta x^2} \bmod P \\ C(I) &= B(I) \\ A(I) &= 1 + 2\alpha \frac{\Delta t}{\Delta x^2} \bmod P \\ D(I) &= Y(I). \end{aligned}$$

For the $n-1$ point on the curve,

$$\begin{aligned} C(N-1) &= -\alpha\frac{\Delta t}{\Delta x^2} \bmod P \\ A(N-1) &= B(I) \\ A(I) &= 1+2\alpha\frac{\Delta t}{\Delta x^2} \bmod P \\ D(N-1) &= Y(N-1)+\left(\alpha\frac{\Delta t}{\Delta x^2}\right)D_N. \end{aligned}$$

These conditions imply that T_1 is known in terms of T_2. Thus the point 2 is a relation between T_1, T_2 & T_3. But since T_1 is known, this relation reduces to a relation between T_2 and T_3. This process of substitution can be continued until T_{n-1} can be formally expressed as T_n. But since Tn is known we can obtain T_{n-1}. This enables us to begin back substitution process in which T_{n-2}, T_{n-3}, $\cdots$, T_3, T_2 can be obtained for one iteration. The process is repeated for t iterations.

2.2.3 Scalar Multiplication

Left to Right Multiplication:

The iteration value of the process is represented by its equivalent binary value. The reading starts from left to right positions. It starts with G as its initial input stream. If the process encounters a 0 in the bit pattern, the output of earlier iteration forms the input for the current iteration else the output of earlier iteration is summed up with Value of G which in turn forms the input of current iteration (See Algorithm 1).

1) Data Doubling: The output of earlier iteration forms the input of current iteration.
2) Data Addition: The output of earlier iteration is summed up with Value of G which in turn forms the input of current iteration.

Algorithm 1 Scalar Multiplication

```
Q := 0
for i from m to 0  do
  Q := 2Q ()(Using Data Doubling)
  if di = 1 then
    Q := Q + G (using Data addition)
  end if
end for
Return Q
```

I.E. If t is 13, its equivalent binary bit pattern is 1101. Then the iteration process continues with G as input in the first iteration, the output of earlier iteration is summed up with Value of G which in turn forms the input of current iteration, then followed with the output of earlier iteration forms the input for the current iteration and finally the output of earlier iteration is summed up with Value of G which in turn forms the input of final iteration.

Window Method (See Algorithm 2):

Algorithm 2 Window Method

```
Q := 0
for i from m to 0 do
  Q := $2^w$Q (using repeated data doubling)
  if $d_i > 0$ then
    $Q := Q + d_1P$ (using a single data addition with the pre-computed value of $d_1P$)
  end if
end for
Return Q
```

Sliding windowed method (See Algorithm 3):

Algorithm 3 Sliding Windowed Method

```
Q := 0
for i from 0 to m do
  if $d_i = 0$ then
    Q := 2Q (point double)
  else
    Grab up to $w - 1$ additional bits from $d$ to store into
    $Q := Q + d_1 P$ (using a single data addition with the pre-computed value of $d_1 P$ (including $d_i$) $t$ and decrement $i$ suitably)
    if (If fewer than w bits were grabbed) then
      Perform double-and-add using $t$
      Return Q
    else
      $Q := 2^w Q$ (repeated point double)
      $Q := Q + tP$ (point addition)
    end if
  end if
end for
Return Q
```

3 The Proposed Scheme

There are two phases of the proposed scheme: Generating Public and Private Keys and Encryption & Decryption Processes.

3.1 Generating Public and Private Keys

The sender executes the following steps (See Figure 1):

1) Sender chooses the Cubic Spline curve with both sides of the curve maintained at same data values i.e $T_S = T_N$.

2) Sender chooses the following parameters $G, T_S, T_N, \alpha, \Delta x, \Delta t$ as Global.

3) Sender chooses an integer t as private key.

4) Sender chooses a large Prime Number P to calculate the Field.

5) Sender calculates (G^t) as Public Key.

6) Private Key considered is t.

7) Public key $= (G)^t = P_E = G_2$.

3.2 Encryption and Decryption Processes

Encryption: Sender selects P_m as her Plain text. He/She then calculates a pair of texts as Cipher texts for chosen random number r.

$(P_m + P_B^r), G_r =$	$(P_m + G_3), G_1 =$	C_1, C_2

Decryption: Receiver after receiving C_1, C_2, Calculates P_m, the Plain Text using the following Formula,

$C_1 - C_2^t =$	$P_m + G_3 - G_3 =$	P_m

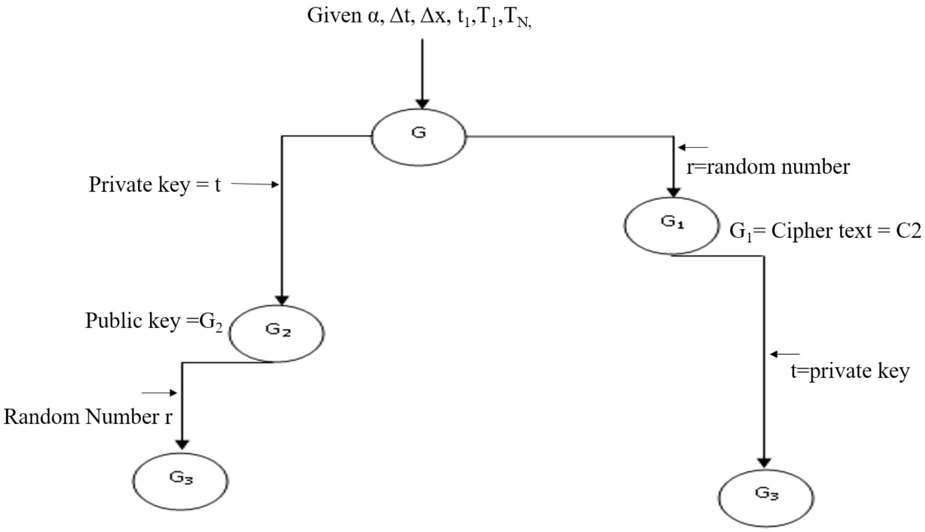

Figure 1: The generating public and private keys phase

Table 1: The parameters of an example

$\alpha = 2$	$\Delta t = 2$	$\Delta x = 2$	$P = 17$

t=2	G	3	12	16	12	6	11	1	16	3
t=4	$C_2 = G_1 = G^r$	3	9	1	15	12	6	8	16	3
t=8	$G_2 = P_B$	3	3	4	16	1	9	5	11	3
t=12	$G_3 = P_B^r = C_2^t$	3	11	14	9	16	10	1	13	3

Example 1. *The following parameters $P, \alpha, \Delta x, \Delta t$ and t are gave as Table 1.*

Mapping the alphabets of English with numerical values, like $a = 0$, $b = 1$, $x = 23$, $y = 24$, $z = 25$.

Encryption: *The plaintext of the example is ASYMMETRY. The ciphertext is in Table 2.*

Table 2: The example of encryption phase

Plain text	A	S	Y	M	M	E	T	R	Y
Alpha numeric	0	19	24	12	12	4	20	18	24
P_B^r	3	11	14	9	16	10	1	13	3
$C_1 = P_m + P_B^r$	3	30	38	21	28	14	21	31	27
Mod 26	3	5	13	21	3	14	21	6	2
C_1 In Alpha numeric	3	5	13	21	3	14	21	6	2
C_1	d	F	n	V	d	o	v	g	C
$G^r = C_2$ In Alpha numeric	3	9	1	15	12	6	8	16	3
C_2	d	J	b	P	m	g	i	q	D

Decryption: *The ciphertext C_1 of the example is 3 5 13 21 3 14 21 6 2. The plaintext is in Table 3.*

Table 3: The example of decryption phase

C_1	3	5	13	21	3	14	21	6	2
C_2^t	3	11	14	9	16	10	1	13	3
$C_1 - C_2^t$	0	-6	-1	12	-13	4	20	-7	-1
Add 26 if Negative	0	19	24	12	12	4	20	18	24
Plain text	A	S	Y	M	M	E	T	R	Y

The simulation: *The simulation of the example is shown in Figure 2.*

4 Cryptanalysis and Complexity of the Model

In this section, we show that cryptanalysis and complexity of the model.

4.1 Cryptanalysis

1) Depending on the width of the curve considered say n a block of plain text will be converted to block of cipher text. So the computing resources needed will be mapped per block rather than per character. So the amount of computing resources needed will be less when compared with algorithms like RSA & ECC.

2) Going by the construction of the algorithm; Known the first and last data points; For the data points 3 to $n-1$ (TDMA Algorithm),

$$\begin{aligned} R &= C(i)/A(i-1) \bmod P; \\ A(i) &= A(i)R(B(i-1))| \bmod P; \\ D(i) &= D(i)R(D(i-1))| \bmod P. \end{aligned}$$

Since $D(n-1)$ is known in terms of $D(n)$, $D(n-1)$ is calculated as

$$D(j) = \frac{D(j) - B(j) * D(j+1)}{A(j)} \bmod \quad \text{where } j = n-1. \tag{2}$$

By the back substitution process, the data values like $D(n-2)$, $D(n-3)$, $\cdots$, $D(2)$ are calculated.

Thus going by the construction of the algorithm, 4 modulus operations are calculated and 2 inverse operations are calculated per bit value in each step of encryption, where as in ECC it is 3 modulus operations and 2

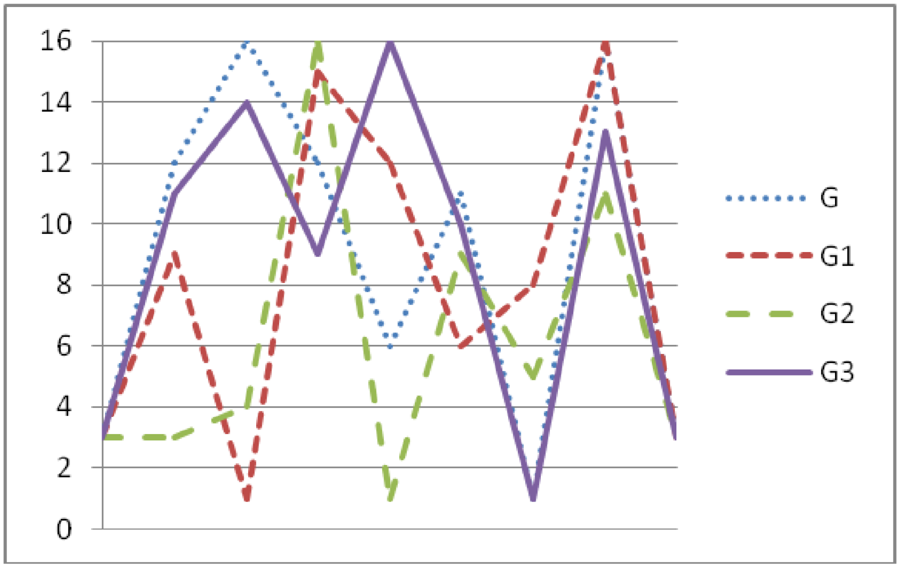

Figure 2: The simulation of the example for $t = 0$, $t = 2$, $\cdots$, $t = 16$

inverse operations per bit value in each step of encryption process. Thus the complexity of Cubic Spline is 4/3 times of ECC. Thus the sufficient key length in Cubic Spline Curve for providing necessary security to the data transmitted is roughly 3/4 of key length of ECC. Thus the key length for cubic spline curve is roughly 120 bit which is necessary to provide sufficient strength against crypto analysis.

3) The windowed method provides the benefit of using fewer point additions.

4) This algorithm has the benefit that the pre-computation stage is roughly half as complex as the normal windowed method while also trading slower point additions for point doubles. In effect, there is little reason to use the windowed method over this approach.

5) As a random number is used for every block of encryption of data, it is free from differential side channel attacks.

4.2 Complexity

Consider the equation,

$$P_B \equiv g^x (\bmod P) \tag{3}$$

where P_B is the public key generator, g is the generator, P be the field, x is the private key.

Given g, x, and P, it is easy to calculate P_B. But given y, g, P it is very difficult to calculate x (given the differential logarithm problem). Thus the asymptotically fastest known algorithm for taking discrete logarithm modulo prime number of the order of

$$e^{\left((ln\,P)^{1/2} ln(lnP)\right)^{2/3}} \tag{4}$$

Which is not feasible for large primes.

5 Conclusion

In this work a Numerical model based cubic spline curve public key cryptography algorithm is developed. The algorithm is based on ElGamal mode of Encryption & Decryption. This works on asymmetric block cipher mode. Going by the construction of the algorithm, the key length needed for sufficient security to date is less than key length needed for ECC algorithm. Since a random number is used for Encryption process, it is free from differential side channel attacks.

The work is carried out for the boundary condition $T_1 = T_n$, i.e., both the boundaries of curve are maintained at same data values. The work can also be carried out for other boundary conditions of spline curves. The present work handles data encryption at block level of plain text. The work can also carried out for encryption of data at character level of plain text.

Acknowledgment

The authors of this paper would like to acknowledge Dr. Somanchi VSSNVG Krishna Murthy, Head, Department of Applied Mathematics, Defence Institute of Advanced Technology, Pune for supporting this work.

References

[1] R. C. C. C. Cheng, N. J. Baptiste, W. Luk, and P. Y. K. Cheung, "Customizable elliptic curve cryptosystems," *IEEE Transactions on VLSI Systems*, vol. 13, no. 9, pp. 1048–1059, 2005.

[2] S. D. Conte and C. deBoor, *Elementary Numerical Analysis*. New York: Mc Graw Hill, 1972.

[3] CSC-93-09, *The interpolating random spline cryptosystem.*

[4] W. Diffie, "The first ten years of public key cryptography," *Proceedings of IEEE*, vol. 76, no. 5, pp. 560–577, 1988.

[5] IEEE, "Standard specifications for public key cryptography," *IEEE Standard*, pp. 1363, 2000.

[6] A. V. N. Krishna, "A new non linear model based encryption scheme with time stamp & acknowledgement support," *International Journal of Network Security*, vol. 13, no. 3, pp. 202–207, 2007.

[7] A. V. N. Krishna and A. V. Babu, "A new model based encryption scheme with time stamp & acknowledgement support," *International Journal of Network Security*, vol. 11, no. 3, pp. 172–176, 2010.

[8] A. V. N. Krishna and A. V. Babu, "A new non linear, time stamped & feed back model based encryption mechanism with acknowledgement support," *IJANA*, vol. 2, no. 5, pp. 191–198, 2010.

[9] 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, 2006.

[10] R. Ramanna, "Numerical methods," pp. 78–85, 1990.

[11] R. R. Ramasamy, M. A. Prabakar, M. I. Devi, and M. Suguna, "Knapsack based 11 ECC encryption and decryption," *International Journal of Network Security*, vol. 9, no. 3, pp. 218–226, 2009.

[12] W. Stallings, "Cryptography and network security," *Prentice Hall, 4th Edition.*

[13] V. P. Suhas, "Numerical heat transfer and fluid flow," pp. 11–75, 1991.

[14] N. Sun, T. Ayabe, and K. Okumura, "An animation engine with cubic spline interpolation iih msp-08," *Proceedings of 2008 International Conference on Intelligent Information Hiding & Multimedia Signal Processing*, pp. 109–112, 2008.

A.V.N. Krishna has a total of 22 years of teaching and research experience. The author has Published his research work in the National and International Journals f repute. Presently the author is working as Professor in the Department of Computer Science & Engineering, Faculty of Engineering, Christ University, Bangalore, Karnataka, India. The author is presently guiding 4 doctoral students in the field of Cryptography.

Addepalli Hari Narayana is in his 3^{nd} year B.Tech in Electrical Engineering from IIT-Indore, Indore.

K. Madhura Vani is presently working as Assistant Professor, CSE in Shreyas Institute of Engineering & Technology and working in Network Security area.