

EIJO Journal of Engineering, Technology And Innovative Research (EIJO-JETIR)

**Einstein International Journal Organization (EIJO)** Available Online at: www.eijo.in Volume - 1, Issue - 2, May - June 2016, Page No. : 09 - 18

# FPGA Based Elliptic Curve Cryptography for LAN Security

G. Indumathi<sup>1</sup>, S.Sathyakala<sup>2</sup> <sup>1</sup>Professor, <sup>2</sup>PG Student Department of ECE, MEPCO Schlenk Eng. College, Sivakasi, India E-Mail: <sup>1</sup>sssk92@gmail.com

## ABSTRACT

Cryptography protects the data stored in the hardware from unauthorized access. At present many authentication schemes have been developed. One of those schemes is authentication based on elliptic curves with the advantage of high security, small key size, and small bandwidth. Elliptic curve cryptography has evolved vast field for public key cryptography systems. In public key cryptography system, we use separate keys to encrypt and decrypt the data. In this project, a secured public key cryptography system has been designed and implemented. The encrypted data is transferred between two systems through Ethernet cable. In this technique, a public key is generated with the help of ring oscillator PUF which oscillates with unique frequency and produces random outputs. This resulted in speed, high throughput, area efficiency and lesser hardware requirements on FPGA. In the generalized ECC, cryptographic operations are performed over the points in Elliptic curve finite field and the data is mapped to those points. In this proposed method, the data varies each and every time. Hence arbitrary mapping (direct mapping) is used to easily map the data for elliptic curve points. Elliptic curve cryptography operations are programmed and synthesized in Xilinx ISE 14.6. Simulations have been done by "ModelSim Altera6.4a (Quartus-II 9.0) starter Edition". The public key cryptography system have been implemented in two Virtex-5 FPGA board, where the plain text is taken as input in one device and cipher text is obtained at the output of the device. The encrypted cipher text is transferred through LAN (Ethernet cable) and received by another device. this device decrypts the plain text. It provide confidentiality, authentication and message integrity in a LAN by including various attacks like brute-force attack, chosen-cipher text attack, chosen-plaintext attack.

Keywords: Elliptic curves, Public key Cryptography system, Montgomery Point multiplication.

# 1. Introduction

Elliptic curve cryptography schemes are public key based mechanisms that provide encryption, digital signatures and key exchange algorithms [1]. To achieve the primary goal of increasing the Hardware Speed, a new ECC processor is proposed. This processor supports all five NIST-recommended primes of sizes 192, 224, 256, 384, and 521 bits [2]. The mapping of message to the points on Elliptic curves is a major part in ECC. It is very difficult to generate points and to map the message to it. A deterministic method for mapping data to points in elliptic curve binary field is proposed in [13]. The Secured data transmission through Ad-hoc on demand distance vector algorithm using elliptic curve cryptography has low Efficiency and reliability as presented in [15]. In general, Security is critical for a variety of sensor network applications. There exist a large number of security vulnerabilities in (Wireless Sensor Networks) WSN, which cause many kinds of attacks. Elliptic curve cryptography offers practical implementation possibilities in resource constrained devices [3]. Wireless devices are rapidly becoming more dependent on security features such as the ability to do secure email, secure Web browsing and virtual private networking to corporate networks. ECC allows more efficient implementation of all of these features. The PUF based ring oscillator circuit is designed for encryption of RFID system, to generate a permanent secret key storage in FPGA. It is low hardware cost and used to protect the third-party attack as proposed in [4], [5]. The ring oscillator circuit [14] is used to compare all the possible inputs and select the random value which is a key for elliptic curve cryptography. Various attacks are performed in public key cryptosystem. It is used to check the security level of elliptic curve cryptography. To provide a detailed examination of the leading attacks against the ECDLP, and to use the knowledge of these attacks in an attempt to generate cryptographically strong elliptic curves. [7], [8], [11].

## 2. Elliptic Curve over Binary Field F (2 m)

The equation of elliptic curve over binary field F m 2 is given by:  $Y^2 + xy = x^3 + ax + b$ .....1

#### Where $b \neq 0$ .

The elements of finite field are integers of length at most m bits. These numbers can be considered as the binary polynomial of degree (m-1). In binary polynomial the coefficients can only be 0 or 1 otherwise reduced to 0 or 1 by modulo 2 operations. All the operations such as addition, subtraction, multiplication and division involve polynomial of

#### .....

degree (m-1) or lesser. If in any operation the degree of polynomial greater than or equal to *m*, degree of the result will be reduced to less than m using irreducible polynomial also called reduction polynomial. The value of *m* is chosen such that there is finitely large number of points on elliptic curve to make the cryptosystem more secure. Domain parameters of F ( $2^m$ )

The domain parameters for Elliptic curve over binary field F m 2 are m, f(x), a, b, G and n. Where:

- *m* is an integer defined for finite field  $F(2^m)$
- f(x) is the irreducible polynomial of degree m.
- *a* and *b* are the parameters defining the curve

 $\mathbf{Y}^2 + \mathbf{x}\mathbf{y} = \mathbf{x}^3 + \mathbf{a}\mathbf{x} + \mathbf{b}.$ 

- G is the generator point  $(x_G, y_G)$ , a point on the elliptic curve.
- *n* is order of the finite field that is number of points in elliptic curve field.

#### 3. Proposed System Model

At the sender side of public key cryptography system the input data captured from the local area network using wire shark software. The plaintext is directly mapped to the elliptic curve points using Deterministic Mapping algorithm. Key is generated with the help of PUF based ring oscillator and encryption is performed which results in two ciphers (C1, C2). The same process repeats at the receiver side but with inverse mapping resulting in the original plain text as shown in fig 1.



Figure1: Proposed Public Key Cryptosystem

## A .Arbitrary Mapping

To encrypt the message it has to be mapped to some point in the elliptic curve finite field. Arbitrary mapping which allows to map any non-trivial (nonzero) message M (interpreted as a member of the field GF  $(2^n)$ ) to a valid EC point. Message M to the x or y coordinate of an elliptic curve point. Two types of mapping are Probabilistic mapping and Deterministic mapping. The only existing map of plaintext messages to an EC point is a probabilistic algorithm. This method is described as a mapping of a plaintext message to an EC point where the elliptic curve is defined over a prime field Zp.

# 1. Deterministic Mapping of M to an EC Point

There is natural mapping between elements of GF (2n) and non negative integers  $<2^{n}$ .an- $1x^{n-1}+a_{n-2}x^{n^2}+...+a_{1x}+a_0 \in GF(2^{n})$  is associate with the integer. Therefore message (plaintext) can be interpreted as an element of GF (2n) and it can be mapped as follows:

• Select  $x1 \in \mathbb{R}$  GF(2n) such that x1 is an x-coordinate of an EC point.

© 2015 EIJO, All Rights Reserved

S.Sathyakala, et al. Einstein International Journal Organization (EIJO)

• Let PM = (x1, M)

Set 
$$\gamma \leftarrow (\frac{\mathbf{M}}{\mathbf{x}_1})^2 + \frac{\mathbf{M}}{\mathbf{x}_1} + \mathbf{x}_1 + \mathbf{a} + \frac{\mathbf{b}}{\mathbf{x}_1^2}$$
, then  $(x_1, M)$ 

• belongs to  $Ea + \gamma, b$ .

#### **B.** Key generation and Exchange

## 1. Ring Oscillator PUF

Ring oscillator PUF is used to generate the key, which oscillates with unique frequency and produces random outputs. This is given to the input of multiplexer where one pair of ring oscillator is selected. The counter counts the number of oscillations for a fixed time interval, after comparison the counter generates the response. The output of the comparator is set 0 or 1 based on selection of the ring oscillator in accordance to its response time. The block diagram of ring oscillator PUF is shown in fig 2.



Figure 2: Proposed Ring Oscillators PUF

A public key is a point on the curve and the private key is a random number. Public key can be obtained by multiplying the private key with the generator point "G". Steps in key generation are,

- Initially the curve C is selected (i.e. the selection of a, b and p) by A and it is sent to B.
- A and B generate points in elliptic curve finite field.

• A selects generator point G which presents in the generated elliptic curve finite field. A sends generator point G to receiver B.

• Using generator point G and private keys (nA and nB), A and B generates their public keys separately.

The public keys are exchanged between A and B as shown in Fig 3.



#### Figure 3: Key generation and exchange

#### C. Encryption and Decryption

To encrypt the message point PM, A selects a random integer k and computes the cipher text as a pair of points PC using public key of B.

$$P_{\rm C} = \left[ kG, P_{\rm M} + kP_{\rm B} \right] \dots 2$$

Where *PB* is public key of *B*.

After receiving the cipher text pair of point's *PC*, *B* multiplies the first point, (kG) with its private key, n<sub>B</sub> and then subtracts the result from the second point in the cipher text pair of points as given by:

$$(P_{M} + kP_{B}) - n_{B}kG = P_{M} + kn_{B}G - n_{B}kG = P_{M}$$
 .....3

P is the plaintext point, corresponding to the plaintext message M. The encryption operation which generates a pair of points {C1, C2}.

## D. cryptographic attacks

A cryptographic attack is a method for circumventing the security of a cryptographic system by finding a weakness in a code, cipher, cryptographic protocol or key management scheme. This process is also called "cryptanalysis". Various attacks are performed in public key cryptosystem. It is used to check the security level of elliptic curve cryptography. To provide a detailed examination of the leading attacks against the ECDLP, and to use the knowledge of these attacks in an attempt to generate cryptographically strong elliptic curves. Some of attacks are performed in proposed elliptic curve cryptosystem are listed below.

#### **1.** Brute Force Attack

Brute force attack or exhaustive key search is a type of strategy which can be applied to any type of encrypted data. In this type of attack all possible keys are tried systematically until the correct key is found. This method is used when any other weakness is not useful. The key length used in the encryption process specifies the practical feasibility of brute force attack, with longer keys exponential more difficult to crack as compared to smaller keys one of the measured strength of the encryption system depends on theoretically how much time is taken to mount a successful brute force attack. The resources required for brute force attack grow exponentially with increase in key size, not linearly. The Flow diagram of brute force attack is shown in fig4.



Figure 4: brute force attack

#### 2. Chosen Plaintext attack

The attackers obtain the various cipher text corresponding to an arbitrary set of plain text. The architecture of Chosen Plaintext attack is shown in fig 5.



**Figure 5: Architecture of Chosen Plaintext attack** 

#### 3. Chosen Cipher text attack

The attackers obtain the various plaintexts corresponding to an arbitrary set of cipher text. The architecture of Chosen cipher attack is shown in fig 6.



Figure 6: Architecture of Chosen cipher text attack

## 4. Results and Discussions

The Elliptic curve cryptographic operations have been programmed and synthesized in Xilinx ISE 14.6. Simulations have been done by "ModelSim-Altera6.4a (Quartus-II 9.0) starter Edition". The public key cryptography system has been implemented by using two Vertex-5 FPGA boards. The plaintext (message) is taken as input from the local area network by using Ethernet cable and Wire shark software. By elliptic curve encryption process, cipher text is taken as output. *A. Tri-mode Ethernet MAC* 

The real-time the elliptic curve encryption of plain text taken as local area network using vertex -5 boards. The pockets are captured with the help of Wire shark software.fig.7shows the Ethernet connection between board and Local Area Network.



Figure 7: Ethernet connections between board and network.

The implementation of project utilization is 173%, number LUT usage is 71% and the number Flip flop usage is 35%. The captured real time data packets along with their overhead information can be seen with the help of wire shark software. Such window is shown in fig 8.

Churnel Ditze       Connect Ditze       None       Write's Stringt       Decopition Keys.         0       Time       Source       Decination       Protocal Length Md         1       0.00000000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.16.17 Tell 172.16.28.142         2       0.2755901000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.16.17 Tell 172.16.28.142         3       0.27554000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.16.17 Tell 172.16.28.142         4       0.38555000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.20.1357 Tell 172.16.28.142         5       1.164170000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.20.1357 Tell 172.16.28.142         6       1.364180000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.16.17 Tell 172.16.28.142         7       2.162553000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.16.17 Tell 172.16.28.142         13.179650000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.20.1357 Tell 172.16.28.142         13.179650000       HewNettP_23:bd:4b       Broadcast       APP       42 Mho has 172.16.20.1357 Tell 172.16.28				opression., Dem Apply Sove			
Time         Source         Destination         Protocal Length Ints           1 0.000000000         Hewlettr	211 Downe	Churrent Diffeet	tes All Frames r None	• Wildless Settings Decryption Keys			
1 0.00000000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 3 0.276548000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.20.1357 Tell 172.16.28.142 4 0.389535000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 5 1.16412000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 5 1.16412000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 5 1.16412000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 7 2.162533000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 7 2.162533000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 8 2.16354000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 10 3.17668000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 11 3.10775000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MERCOECE4cp 10 3.17668000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 11 3.10775000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 12 3.457406000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 13 3.57075000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MERCOECE4cp 14 3.9721000 172.16.3.142 172.16.31.255 NBMS 92 Name query NB MERCOECE4cp 15 4.159319000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 14 3.9721000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MERCOECE4cp 15 4.159319000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.16.17 Tell 172.16.28.142 14 3.9721000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MERCOECE4cp 15 4.159319000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.20.1357 Tell 172.16.28.142 14 3.9721000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MERCOECE4cp 15 4.159319000 HewlertP_33:bd:4b Broadcast ARP 42 Who has 172.16.20.1357 Tell 172.16.28.142 16 4.159324000 HewlertP_33:bd:4b Broadcast ARP 42 Who	0,	Time	Source	Destination	Protocol	Length Info	
2 0.773900000 New Netter_23102140 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.26.142 3 0.76548000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 3 1.164170000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 4 0.389535000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 4 1.16418000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.10.137 Tell 172.16.28.142 4 2.162552000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 8 2.162552000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 9 2.443503000 172.16.28.142 172.16.31.253 NBMS 92 Name query NB MEPCOECE40= 10 3.178650000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 11 3.207730000 172.16.28.142 172.16.31.253 NBMS 92 Name query NB MEPCOECE40= 12 3.45740000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 13 3.570730000 172.16.28.142 172.16.31.253 NBMS 92 Name query NB MEPCOECE40= 14 3.9217000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MEPCOECE40= 14 3.9217000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MEPCOECE40= 14 3.9217000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MEPCOECE40= 15 4.159319000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 14 3.9217000 172.16.28.142 172.16.31.255 NBMS 92 Name query NB MEPCOECE40= 15 4.159319000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.10.17 Tell 172.16.28.142 14 3.9221000 182.16.28.142 172.16.31.255 NBMS 92 Name query NB MEPCOECE40= 15 4.159319000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.10.17 Tell 172.16.28.142 16 4.159324000 HewNetter_23105146 Broadcast ARP 42 who has 172.16.10.17 Tell 172.16.28.142 172.16.20.1357 Tell 172.16.28.142		1 0.00000000	HewlettP_23:bd:4b	Broadcast	ARP	42 who has 172.16.16.17 Tell 172.16.28.142	
3 0.70540000       HewlettP_2100140       Broadcast       ABP       42 WHO has 172.10.10.71 Tell 172.10.28.142         4 0.38955000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.10.17 Tell 172.10.28.142         6 1.164180000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.10.17 Tell 172.10.28.142         7 2.10255000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.10.17 Tell 172.10.28.142         8 2.102562000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.10.17 Tell 172.10.28.142         9 2.443503000       172.10.28.142       172.10.31.255       NBMS       92 Name query NB WERCOECE4c>         10 3.17960000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.10.17 Tell 172.10.28.142         11 3.207750000       172.10.28.142       172.10.31.255       NBMS       92 Name query NB WERCOECE4c>         12 3.45706000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.0.117 Tell 172.10.28.142         13 3.50779000       172.10.28.142       172.10.31.255       NBMS       92 Name query NB WERCOECE4c>         12 3.45706000       HewlettP_231bd:4b       Broadcast       ABP       42 WHO has 172.10.00.117 172.10.28.142         13 4.5079000       HewlettP_231bd:4b <td></td> <td>2 0.275901000</td> <td>HewlettP_23:bd:40</td> <td>Broadcast</td> <td>ARP</td> <td>42 who has 172.16.16.17 Tell 172.16.28.142</td> <td></td>		2 0.275901000	HewlettP_23:bd:40	Broadcast	ARP	42 who has 172.16.16.17 Tell 172.16.28.142	
* 0.33333000       HewlettP_31:b0:4b       Broadcast       ARP       42 Who has 172.16.20.137 Tell 172.16.28.142         6 1.164180000       HewlettP_31:b0:4b       Broadcast       ARP       42 who has 172.16.20.1337 Tell 172.16.28.142         7 2.162553000       HewlettP_31:b0:4b       Broadcast       ARP       42 who has 172.16.20.1337 Tell 172.16.28.142         8 2.162542000       HewlettP_31:b0:4b       Broadcast       ARP       42 who has 172.16.10.11 Tell 172.16.28.142         9 2.463502000       172.16.28.142       172.16.31.235       NBMS       92 Name query NB WERCOECE-dc>         10 3.17660000       HewlettP_23:b0:4b       Broadcast       ARP       42 who has 172.16.10.17 Tell 172.16.28.142         11 3.207750000       172.16.26.142       172.16.31.235       NBMS       92 Name query NB WERCOECE-dc>         12 3.457460000       HewlettP_23:b0:4b       Broadcast       ARP       42 who has 172.16.16.17 Tell 172.16.28.142         13 3.570750000       172.16.26.142       172.16.31.255       NBMS       92 Name query MB WERCOECE-dc>         14 3.972170000       HewlettP_23:b0:4b       Broadcast       ARP       42 who has 172.16.16.17 Tell 172.16.28.142         14 3.972170000       H22.16.26.142       172.16.31.255       NBMS       92 Name query MB WERCOECE-dc>         15 4.15913000       HewlettP_23:		3 0.276548000	Hewietty_23:00:40	Broadcast	ARP	42 WHO Fills 1/2.10.20.1337 1013 1/2.10.28.142	
6 1.164180000       Hewlettr_21:b0:4b       Broadcast       ARP       42 who has 172.16.10.17 Tell 172.16.28.142         7 2.162553000       Hewlettr_21:b0:4b       Broadcast       ARP       42 who has 172.16.10.17 Tell 172.16.28.142         8 2.16256200       Hewlettr_21:b0:4b       Broadcast       ARP       42 who has 172.16.10.17 Tell 172.16.28.142         9 2.443503000       172.16.28.142       172.16.31.253       NBMS 92 Name query NB MERCOCCEC4c>         10 3.176680000       Hewlettr_23:b0:4b       Broadcast       ARP       42 who has 172.16.16.17 Tell 172.16.28.142         11 3.207730000       172.16.28.142       172.16.31.253       NBMS 92 Name query NB MERCOCCEC4c>         12 3.457406000       Hewlettr_23:b0:4b       Broadcast       ARP       42 who has 172.16.16.17 Tell 172.16.28.142         13 3.570730000       172.16.38.142       172.16.31.253       NBMS 92 Name query NB MERCOCCEC4c>         14 3.97217000       172.16.38.142       172.16.31.255       NBMS 92 Name query NB MERCOCCC4c>         14 3.97217000       172.16.38.142       172.16.31.255       NBMS 92 Name query NB MERCOCCE4c>         15 4.159319000       HewlettP_23:b0:4b       Broadcast       ARP       42 who has 172.16.20.1357 Tell 172.16.28.142         14 3.97217000       172.16.38.142       172.16.31.255       NBMS 92 Name query NB MERCOCCCE4c> </td <td></td> <td>5 1 164170000</td> <td>Hewiette_25:00:40</td> <td>Broadcast</td> <td>100</td> <td>42 who has 172 16 20 1357 Tell 172 16 38 143</td> <td></td>		5 1 164170000	Hewiette_25:00:40	Broadcast	100	42 who has 172 16 20 1357 Tell 172 16 38 143	
7       2.162353000       Hewlettr_21:bit.4b       Broadcast       ARP       42 Who has 172.16.20.1337 Tell 172.16.28.142         8       2.162362000       Hewlettr_21:bit.4b       Broadcast       ARP       42 Who has 172.16.20.1337 Tell 172.16.28.142         9       2.443503000       172.16.28.142       172.16.21.253       NBMS       92 Name query NB MERCOECE.cp.         10       3.178680000       Hewlettr_23:bit.4b       Broadcast       ARP       42 Who has 172.16.16.17 Tell 172.16.28.142         11       3.207730000       172.16.28.142       172.16.31.253       NBMS       92 Name query NB MERCOECE.cp.         12       3.457460000       Hewlettr_23:bit.4b       Broadcast       ARP       42 Who has 172.16.16.17 Tell 172.16.28.142         13       3.50730000       Hewlettr_23:bit.4b       Broadcast       ARP       42 Who has 172.16.16.17 Tell 172.16.28.142         13       3.50730000       Hewlettr_23:bit.4b       Broadcast       ARP       42 Who has 172.16.20.137 Tell 172.16.28.142         14       3.972170000       172.16.28.142       172.16.31.255       NBMS       92 Name query NB MERCOECE.cp         15       4.139313000       Hewlettr_23:bit.4b       Broadcast       ARP       42 Who has 172.16.10.17 Tell 172.16.28.142         14       4.972170000       172.16.28.		6 1 164180000	Hewlette 23:bd:4b	Broadcast	APP	42 who has 172 16 16 17 Tell 172 16 28 142	
8 2.162962000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.16.17 Tell 172,16.26.142         9 2.443502000       172,16.28.142       172,16.31,255       ABMS       92 Name query NB MERCOCCE-dc>         10 3.179650000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.16.17 Tell 172,16.26,142         11 3.207730000       172,16.28.142       172,16.31,255       ABMS       92 Name query NB MERCOCCE-dc>         12 3.457406000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.16.17 Tell 172,16.28,142         13 3.570790000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.20,1397 Tell 172,16.28,142         14 3.972170000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.20,1397 Tell 172,16.28,142         14 3.972170000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.20,1397 Tell 172,16.28,142         14 3.972170000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.20,1397 Tell 172,16.28,142         16 4.159324000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.20,1357 Tell 172,16.28,142         16 4.159324000       Hewlettr,23:bd:4b       Broadcast       ARP       42 who has 172,16.20,1357 Tell 172,16.28,142         17 16		7 2 162553000	Hewlette 23:bd:4b	Broadcast	APP	42 who has 172 16 20 1337 Tell 172 16 28 142	
9 2.443503000       172.16.28.142       172.16.31.255       NBMS       92 Name query NB MEPCOECE.1c;         10 3.178680000       HewlettP_321bd14b       Broadcast       ARP       42 who has 172.16.16.17       TP1 172.16.28.142         11 3.207730000       172.16.36.142       172.16.31.253       ABP       42 who has 172.16.16.17       TP1 172.16.28.142         12 3.457040000       HewlettP_231bd14b       Broadcast       ARP       42 who has 172.16.16.17       TP1 172.16.28.142         13 3.570790000       HewlettP_231bd14b       Broadcast       ARP       42 who has 172.16.16.17       TP1 172.16.28.142         14 3.97210000       172.16.31.42       172.16.31.255       NBMS       92 Name query NB MEPCOECE.dc;         14 3.97210000       172.16.8.142       172.16.31.255       NBMS       92 Name query NB MEPCOECE.dc;         15 4.159319000       HewlettP_23:bd14b       Broadcast       ARP       42 who has 172.16.20.1357       Te11 172.16.28.142         16 4.159322000       HewlettP_23:bd14b       Broadcast       ARP       42 who has 172.16.20.1357       Te11 172.16.28.142         17       HewlettP_23:bd14b       Broadcast       ARP       42 who has 172.16.20.1357       Te11 172.16.28.142         16 4.159322000       HewlettP_23:bd14b       Broadcast       ARP       42 who has 172		8 2.162562000	HewlettP 23:bd:4b	Broadcast	ARP	42 who has 172, 16, 16, 17 Tell 172, 16, 28, 142	
10       11       3.179650000       HerletP_23:bit4b       Broadcast       ARP       42 who has 1/2.16.16.1?       Tell 1/2.16.26.142         11       3.207730000       172.16.28.142       172.16.31.253       NBMS       92 Name query MB HERCOECE-dcs         12       3.45740000       HewlettP_23:bit4b       Broadcast       ARP       42 who has 1/2.16.16.1?       Tell 1/2.16.28.142         13       3.570790000       HewlettP_23:bit4b       Broadcast       ARP       42 who has 172.16.10.137       Tell 1/2.16.28.142         14       3.972170000       172.16.28.142       172.16.31.255       NBMS       92 Name query MB MERCOECE-dcs         15       4.199319000       HewlettP_23:bit4b       Broadcast       ARP       42 who has 172.16.10.17       Tell 1/2.16.28.142         16       4.199324000       HewlettP_23:bit4b       Broadcast       ARP       42 who has 172.16.10.17       Tell 1/2.16.28.142         16       4.199324000       HewlettP_23:bit4b       Broadcast       ARP       42 who has 172.16.20.1357       Tell 1/2.16.28.142         16       4.199324000       HewlettP_23:bit4b       Broadcast       ARP       42 who has 172.16.20.1357       Tell 1/2.16.28.142         17       Brane 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0       Tetrarr		9 2,443503000	172.16.28.142	172.16.31.255	NBNS	92 Name query NB MEPCOECE<1c>	
11 3.207730000       172.16.28.142       172.16.31.255       MBMS       92 Name query MB MEPCOECE<1c>         12 3.45705000       HewlettP_2310614b       Broadcast       ARP       42 Who has 172.16.16.17       Tell 172.16.28.142         13 3.570730000       HewlettP_2310614b       Broadcast       ARP       42 Who has 172.16.017       Tell 172.16.28.142         14 3.972170000       172.16.28.142       172.16.31.255       NBMS       92 Name query NB MEPCOECE<1c>         14 3.972170000       172.16.28.142       172.16.31.255       NBMS       92 Name query NB MEPCOECE<1c>         15 4.139313000       HewlettP_23:0614b       Broadcast       ARP       42 Who has 172.16.16.17       Tell 172.16.28.142         16 4.159324000       HewlettP_23:0614b       Broadcast       ARP       42 Who has 172.16.20.1357       Tell 172.16.28.142         16 4.159324000       HewlettP_23:0614b       Broadcast       ARP       42 Who has 172.16.20.1357       Tell 172.16.28.142         16 4.159324000       HewlettP_23:0614b       Broadcast       ARP       42 Who has 172.16.20.1357       Tell 172.16.28.142         17 frame       Si arcsin the interface 0       Interface 0       Interface 0       Interface 0         Extination: Broadcast (fiff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff		10 3.179680000	HewlettP_23:bd:4b	Broadcast	ARP	42 who has 172.16.16.1? Tell 172.16.28.142	
12 3.457400000 HewlettP_23:bd:4b Broadcast ARP 42 Who Has 172.16.16.17 Tell 172.16.28.142 13 3.570790000 HewlettP_23:bd:4b Broadcast ARP 42 Who Has 172.16.20.1337 Tell 172.16.28.142 14 3.97270000 172.16.28.142 172.16.31.255 NBMS 92 Name query M8 MERCOECE-dc- 15 4.19913000 HewlettP_23:bd:4b Broadcast ARP 42 Who Has 172.16.20.1357 Tell 172.16.28.142 16 4.19932000 HewlettP_23:bd:4b Broadcast ARP 42 Who Has 172.16.20.1357 Tell 172.16.28.142 17 Frame 5: 42 bytes on wire (335 bits), 42 bytes captured (336 bits) on interface 0 Frame 5: 42 bytes on wire (335 bits), 42 bytes captured (336 bits) on interface 0 Extendence III. Secalcast (fift:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:f		11 3,207750000	172, 16, 28, 142	172.16.31.255	NBNS	92 Name query NB MEPCOECE<1c>	
13 3.570790000 HewlettP_23:b6:4b Broadcast ARP 42 who has 172.16.20.1357 Tell 172.16.28.142 14 3.972170000 172.16.28.142 172.16.31.255 NBHS 92 Name query NB PEPCOECE-1c- 15 4.159315000 HewlettP_23:b6:4b Broadcast ARP 42 who has 172.16.16.17 Tell 172.16.28.142 16 4.159324000 HewlettP_23:b6:4b Broadcast ARP 42 who has 172.16.20.1357 Tell 172.16.28.142 The set of the set		12 3,457406000	HewlettP_23:bd:4b	Broadcast	ARP	42 who has 172.16.16.1? Tell 172.16.28.142	
14 3.972170000 172.16.28.142 172.16.31.255 NBWS 92 Name query NB MERCORCEGES 15 4.159319000 HewlettP_23:bd:4b Broadcast ARP 42 who has 172.16.16.1? Tell 172.16.28.142 16 4.159324000 HewlettP_23:bd:4b Broadcast ARP 42 who has 172.16.20.1357 Tell 172.16.28.142 Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0 Ethernet 11. Societ HewlettP_23:bd:4b (14.64:a):32:bd:4b) 1 Source: HewlettP_23:bd:4b (34:64:a):32:bd:4b) Type: ARP (0x0806) Address Resolution Protocol (request)		13 3.570790000	HewlettP_23:bd:4b	Broadcast	ARP	42 who has 172,16,20,135? Tell 172,16,28,142	
15 4.159319000 HewlettP_23:bd:4b Broadcast ARP 42 Who has 172.16.10.17 Tell 172.16.28.142 16 4.159324000 HewlettP_23:bd:4b Broadcast ARP 42 Who has 172.16.20.1357 Tell 172.16.28.142 Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0 Estimation: Broadcast (fiff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff		14 3,972170000	172.16.28.142	172.16.31.255	NENS	92 Name query NB MEPCOECE<1c>	
16 4.199324000 MewlettP_23:bd:4b Broadcast ARP 42 Who has 172.16.20.1357 Tell 172.16.28.142 Prame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0 Enterror at a sectime intrace 2000/40 (5164ra1922B00/40) wists sections (interface 10) E Source: HewlettP_23:bd:4b (34:64:a0:23:bd:4b) Type: ARP (0x0800) Address Resolution Protocol (request)		15 4.159319000	HewlettP_23:bd:4b	Broadcast	ARP	42 who has 172,16,16,1? Tell 172,16,28,142	
Prame 5: 42 bytes on wire (336 bits), 42 bytes captured (136 bits) on interface 0         Externance 11. Speck Hewlettle_32b0d/AD (14.065a0222b0d/AD (14.065a0222b0d/AD (14.065a0222b0d/AD (14.065a0222b0d/AD (14.065a0222b0d/AD (14.065a0222b0d/AD (14.065a02202b0d/AD (14.065a0202b0d/AD (14.065a020a02b0d/AD (14.065a020a02b0d/AD (14.065a020a02b0d/AD (14.065a020a02b0d/AD (14.065a020b0d/AD (14.065a020b0d/AD (14.065a020b0d/AD (14.065a020a02b0d/AD (14.065a02b0d/AD (14.065a020a02b0d/AD (14.065a02b0d/AD (14.065a020a02b0d/AD (14.065a02b0d/AD (14.065a020a02b0d/AD (14.065a02b0d/AD (14.065a020a02b0d/AD (14.065a02b0d/AD (14.065a02b0d/AD (14.065a02b0d/AD (14.065a02b0d/AD (14.065a02b0d/AD (14.065a02b0d/AD (14.065a02b0d/AD (14.065a0200000))))))))))))))))))		16 4,159324000	HewlettP_23:bd:4b	proadcast	ARP	42 who has 172,16,20,1357 Tell 172,16,28,142	
Frame 5: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0 Externet 11. Dect Hewlette_33bbitb(de(4).64:a4)-23bbitb(b), Cote Excedence (frifffffffff) = Destination: Broadcast (ff:ff:ff:ff:ff:ff) = Source: Hewlette_33bbitb(de(34:64:a9):23:bd:4b) Type: ARP (0x0806) Address Resolution Protocol (request)							
Externat 11. Soci Hewlett/23:bd/Ab (41.64:a022:bd/Ab), dst. Broadcast (FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	Frase 5	: 42 bytes on wire (336	bits), 42 bytes capture	d (336 bits) on interface 0	22		
Type: ABF (0x0806) Address Resolution Protocol (request)	Encenned	nation: Broadcast (ff:f	f:ff:ff:ff:ff) 8:64:39:73:bd:4b)	), ESC BROWSLASS (FF.FF.FF.FF.FF.FF	1(1)		
Address Resolution Protocol (request)	E Destin	1 HOW LOC TOP: 2 4 DOI: 400 14					
	E Destin E Source Type:	ARP (0x0806)					
	E Destin E Source Type: Address	ARP (0x0506) Resolution Protocol (r	equest)				
	E Destin E Source Type: Address	ARP (0x0806) Resolution Protocol (r	equest)				
	E Cestin E Source Type: Address	ARF (0x0806) Resolution Protocol (r	equest)				
	E Destin E Source Type: Address	, HewYett_22100:40 (3 ARF (0x0506) Resolution Protocol (r	equest)				
	E Destin E Source Type: Address	. Herietty_2:100:40 (3 ARF (000806) Resolution Protocol (r	equest)				
	E Destin E Source Type: Address	, mentect=00.40 (3 ARP (0x1006) Resolution Protocol (r	equest)				
	E Destin E Source Type: Address	o, mey Met D-22100440 (3 ARP (CotOGO) Resolution Protocol (r	equest)				
	Destin Source Type: Address	5. mevretr_2:00:40 (3 ARP (0x000) Resolution Protocol (r	equest)	record generation			
000 00 00 00 00 00 00 01 34 64 - 30 23 bd 4b ac 10 1c 8e	000 000 000 000 000 000 000 000 000 00	5. mey (metro1:00:40 (3 ARF (0x000) Resolution Protocol (r 0000 00 00 01 34 64 a	equest) 7.1.050(05) 0(04) 9 23 bd 4b ac 10 1c 8e 4 37	<mark></mark>			
000         16	E Destin E Source Type: Address	new(retr_2):00:40 (3 ARE/(0x000) Resolution Protocol (r 0 06 04 00 01 34 64 a 0 00 00 00 00 ac 10 1	equest) 9 23 bd 4b ac 10 1c 8e 4 87				
000         10	E Destin Source Type: Address	0 06 04 00 01 34 64 a 10 00 00 00 00 ac 10 1	equest) 9 23 hd 4b ac 10 1c 8e 4 87	<mark>n</mark>			

Figure 8: Wire Shark window

# B. Arbitrary data mapping

Arbitrary message mapping is to map message(m) to a point (Pmx, Pmy) on elliptic curve Ea+ $\gamma$ ,b. Simulation of Message mapping is shown in Fig 9. The arbitrary data mapping resource utilization is 88%.

📻 wave - default			+
Messages			
<pre>/ptgen/ck /ptgen/rst /ptgen/Px /ptgen/Py /ptgen/Py /ptgen/Pox /ptgen/Pox /ptgen/Poy /ptgen/Poy /ptgen/P1x /ptgen/P1y</pre>	0 1 0ec4f40c26564f388! 1b5999bf89d1dec2d 000000000000000000000000000000000000	0ec4f40c26564f3885bc68215aec250c298aeeaa971276e04d59.           1b\$999bf89d1dec2d8691c74aee8565ffcc1349f140d2071c230f.           000000000000000000000000000000000000	
Now	3100 ps	2500 ps 3000 ps	'

Figure 9: Simulation result of arbitrary mapping

#### C. Ring Oscillator PUF

Ring oscillator PUF is used to generate the key, which oscillates with unique frequency and produces random outputs. This is given to the input of multiplexer where one pair of ring oscillator is selected. The counter counts the number of oscillations for a fixed time interval, after comparison the counter generates the response. The output of the comparator is set 0 or 1 based on selection of the ring oscillator in accordance to its response time. This is needed to prevent our chip from overheating.PUF circuit is used to generate volatile secret keys and this is taken for encryption and decryption. Simulation of ring oscillator PUF is shown in fig10.



Figure10: Simulation of Ring Oscillator PUF

The implementation of Ring Oscillator PUF resource utilization is 94%. The RTL schematic of ring oscillator PUF is shown in fig11.





#### **D.** Encryption and Decryption

The real time input data or message is captured by using wire shark software. Encryption combines direct mapping of real time data and secret key generation using ring oscillator PUF logically in a single step. The key is shared between sender and receiver and decryption is performed. Figure 12. Shows the simulation result of encryption and decryption of real time data.

wave - default		-							
Messages									
/realtimeencry/dk	0								
/realtimeencry/rst	1								
💼 🔶 /realtimeencry/g	000000000000000000000000000000000000000	000000000	00000000	000000000000000000000000000000000000000	000000000	00000000	0000000	00000000	001
💼 🔶 /realtimeencry/k	000000000000000000000000000000000000000	000000000	00000000	00000000	000000000	00000000	0000000	00000000	010
💼 🔶 /realtimeencry/pm	000000000000000000000000000000000000000	000000000	00000000	000000000	00000000	005e000	0fc3464a9	23bd4b0	806
💼 🔶 /realtimeencry/y	000000000000000000000000000000000000000	000000000	00000000	000000010	101010101	100 10 10	10 10 10 10	00000000	000
🗖 🔶 /realtimeencry/y1	000000000000000000000000000000000000000	000000000	00000000	00101230	1020aeeo	1000000	0000000	12345aed	lede
🖬 🔶 /realtimeencry/w	00000010ae24fff000	000000 10a	e24fff000	00000111	1000000	0234500	0000000	00102031	44
🖬 🔶 /realtimeencry/w1	000000000000000000000000000000000000000	000000000	00000000	00000000	000000111	1111123	654aefe50	0000000	¢04
🖬 🔶 /realtimeencry/x	010000000000000000	010000000	00000101	01010101	0ae00000	00000000	0000000	00000000	0000
🖽 🤶 /realtimeencry/x1	0101aeffaeaff56481	0101aeffac	aff56481	91966961	63463000	00000000	000000000000000000000000000000000000000	00000111	11
	1efefefefefefefefefef	1efefefefe	fefefefef	fefefefef	efefefefef	efefefefe	feffefefff	F	
- /realtimeencry/c1	100000000000000000000000000000000000000	1	rrrrrrrr	tranar	mmmmm	mmmm			
🗖 🔶 /realtimeencry/c2	000000000000000000000000000000000000000	000000000	00000000	00000000	00000006	ad23450	80656aed	a3612364	653
/realtimeencry/en	1								
	encru/c2 @ 2582	pa		-					
sim:/realtime									1
sim:/realtime 00000000000000	000000000000000000000000000000000000000	0006ead2	3450806	56aeda3	6123646		1	1	
sim:/realtime 00000000000000	00000000000000000	0006ead2:	3450806	56aeda3	6123646				
sim:/realtime	5800.05	0006ead2	3450806	56aeda3	6123646				

# Figure 12: Simulation Result of Encryption and Decryption

The implementation of real time data encryption and decryption process utilizes the recourses of about 189%. The RTL schematic of Real time data encryption and decryption is shown in figure 13.



Figure13: RTL schematic of Encryption and decryption

## E. Comparison of various attacks

The Comparison of Chosen Cipher Text, Brute Force Attack and Chosen Plaintext Attack detection and recovery time by using Elliptic curve cryptographic approach using PUF based key generation is given in Table 1.

Table 1.comparision of various attacks.

Algorith ms	Brute att	force ack	Chosen cipher text attack	Chosen tex atta	plain t ck
key length	16	64	16	16	64
Attack with work factor	2^4	2^6	2^4	2^4	2^6
Attack detection and recoveri ng	160n s	4 mint 16sec	140ns	80ns	35mints 7 sec

## 5. Conclusion

The feature of Elliptic curve cryptography is smaller key size algorithm which provides fast computations as well as less memory, high speed, efficient bandwidth. It provides higher level of security with lesser key size compared to other Cryptographic techniques. A deterministic method has been employed for arbitrary mapping of data to the elliptic curve points. It reduces the complexity involved in elliptic curve point generation and mapping of data to it. Proposed design includes implementation of elliptic curve key generation using Ring oscillator PUF which oscillates with unique frequency and produces random outputs. This is given to the input of multiplexer where one pair of ring oscillator is selected. The counter counts the number of oscillations for a fixed time interval, after comparison the counter generates the response. The output of the comparator is set 0 or 1 based on selection of the ring oscillator in accordance to its response time. This result in speed, high throughput and area efficiency and lesser hardware requirements on an FPGA. Inverse mapping of data points is used to retrieve the original input data. It provides the three major security aspects confidentiality, integrity and authenticity. Elliptic curve cryptographic operations have been programmed and synthesized in Xilinx ISE 14.6. Simulations have been done by "ModelSim- Altera6.4a (Quartus-II 9.0) starter Edition". The public key cryptography system have been implemented by using two Vertex-5 FPGA board, where the plaintext (message) is taken as input in one device from a LAN Ethernet cable by using wire shark software and cipher text is obtained at the output of the device. The encrypted cipher text is transferred through LAN (Ethernet) and received by another device. This device decrypts the plaintext and three types of attacks are considered for simulation. They are brute force attack, chosen plaintext attack and chosen cipher text attack.

# 6. References

[1] William Stallings, "Cryptography and network security", Pearson Education, Fourth Edition, 2006, India.

[2] Hamad Alrimeih and Daler Rakhmatov, "Fast and Flexible Hardware Support for ECC over Multiple Standard Prime Fields", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, January 2014.

[3] K.S.Abitha, Anjalipandey, and DR.K.P.Kaliyamurthie "Secured Data Transmission Using Elliptic Curve Cryptography" International Journal of Innovative Research inComputer and Communication Engineering, Vol. 3, Issue 3, March 2015.

[4] Jiliang Zhang, Yaping Lin, Yongqiang Lyu, and Gang Qu, "A PUF –FSM Binding Scheme for FPGA IP Protection and Pay-Per-Device Licensing" IEEE Transactions on Information Forensics and Security, Vol.10, No. 6, June 2015.

[5] Wonseok Choi, Sungsoo Kim, Yongsoo Park, and Kwangseon Ahn "PUF -based Encryption Processor for the RFID Systems", IEEE International conference on computer and Information Technology, (CIT 2010).

[6] Ahmad Firdaus Mohamad Razy et\_al, "Investigation and Design of the Efficient Hardware based RNG for Cryptographic Applications", IEEE Magazine, pp: 255-260, 2014.

[7] Mohammed Farik, ABM Shawkat Ali "Algorithm to Ensure and Enforce Brute-Force Attack-Resilient Password in Routers" International Journal of Scientific & Technology Research, Volume 4, Issue 10, October 2015.

[8] Mrs.Santoshi Pote, Mrs. Jayashree Katti "Attacks on elliptic curve cryptography discrete logarithm problem (EC-DLP)" International Journal of Innovative Research In Electrical, Electronics, Instrumentation and Control Engineering, Vol. 3, Issue 4, April 2015.

[9] AL-Marake, "Analysis of MD5 Algorithm Safety against Hardware Implementation of Brute Force Attack "International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 9, September 2013.

[10] Swati R. Shete, Prof. Yogini C. Kulkarni, "ATM Pin Transfer Using Visual Cryptography" "International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization) Vol. 3, Issue 5, May 2015.

[11] Mr.Praful V.Barekar, K. N. Hande "Performance Analysis of Timing Attack on Elliptic Curve Cryptosystem" International Journal of Computational Engineering Research, Vol. 2, No.3, 740-743, May- June 2012.

[12] S.Maria Celestin Vigila and K.Muneeswaran, "Nonce Based Elliptic Curve Crypto system for Text and Image Applications", International Journal of Network Security, Vol.14, No.4, pp. 236-242, 2012.

[13] Brian King, "Mapping an Arbitrary Message to an Elliptic Curve" when Defined over GF(2n)", International Journal of Network Security, Vol.8, No.2, PP.169-176, Mar.2009.

[14] J.Guajardo, S. Kumar, G.-J. Schrijen and P. Tuyls. Physical unclonable functions and Public-key crypto for FPGA ip protection. In Field Programmable Logic and Applications, 2007.FPL 2007. International Conference on, pages 189–195, aug.2007.

[15] Kristin Lauter, Microsoft Corporation "The advantages Of Elliptic Curve Cryptography for Wireless Security", IEEE Wireless Communications magazine, pp.62-67, Feb 2004.

......

S.Sathyakala, et al. Einstein International Journal Organization (EIJO)

[16] Kimmo Jarvinen and JormaSkytta, "On Parallelization of High Speed Processors for Elliptic Curve Cryptography", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 16, No. 9, pp: 1162-1175, September 2008.
[17] Rahat Afreen and S.C. Mehrotra "A review On Elliptic Curve Cryptography For Embedded Systems" International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 3, June 2011.