VOL. 10, NO. 5, MARCH 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

IMPLEMENTATION OF MESSAGE AUTHENTICATION SCHEME WITH ELLIPTIC CURVE CRYPTOGRAPHY

G. Indumathi and T. Kiragapushpam
Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi, India

ABSTRACT

Transmission of private information over the public channels requires security or data protection against unauthorized access. Elliptic Curve Cryptography (ECC) is one of the efficient encryption technique can be used to secure the private data. High level security requirement of Restricted Services of Indian Regional Navigation Satellite System (IRNSS) to transmit the navigation data through wireless channel, can be achieved by ECC with minimum key size.ECC is based on Elliptic Curve Scalar Multiplication (ECSM) which is the process of multiplying a point on elliptic curve by a scalar value. The operations has been performed on National Institute of Standards and Technology (NIST) recommended elliptic curves over binary field E (2^{233}).The performance of ECC algorithm is influenced by the implementation of elliptic curve finite field operations. Therefore, field operations play vital role in ECC. Among finite field operations such as squaring, multiplication and inversion, multiplication is very important in cryptosystem. Karatsuba algorithm with polynomial multiplication is more efficient for large numbers. The encryption algorithm, point operations and field operations have been implemented in Xilinx Virtex-5 FPGA board.

Keywords: cryptography, data security, elliptic curves, public key.

1. INTRODUCTION

Elliptic Curve Cryptography is an effective public key cryptography or asymmetric key cryptography. It uses elliptic curves for encryption and decryption. The points on elliptic curve satisfy the properties of finite field elements. Thus the set of points on elliptic curve can be considered as a finite field. In ECC, the cryptographic operations are performed over this finite field. With ECC high level of security can be achieved even for smallest key size [1]. The encryption is performed after encoding the message to the points in the elliptic curve finite field. Elliptic curve encryption can be implemented efficiently in two different finite fields. One is prime field E(p) and another one is binary field E(2^m) [2] and [3].

The message is encrypted to a point in elliptic curve finite field by mapping the message to the points in the same elliptic curve finite field. The encryption is performed by Elliptic Curve Scalar Multiplication (ECSM) or point multiplication. ECSM multiplies a scalar value with a point in an elliptic curve finite field to obtain another point in the same field. ECSM can be performed by repeated point addition and point doubling. It is called "Add and Double Method" [4].

Many hardware implementations of ECC have been proposed to achieve optimized efficient encryption. A high performance ECC processor for general curves over GF (p) based on systolic arithmetic unit has been proposed in [5].

The pipeline stalls are avoided by delaying the conditional operations and the communication mismatch is avoided by distributing the register to individual PE (Processing Elements). Elliptic curve processor can be used in RFID to provide privacy, authentication and protection against tracking of RFID tags. Due to the property of high security for smallest key size ECC can be used in RFID [6]. A hardware design using Motgomery scalar multiplication based on "add and double method" has been presented in [7]. An elliptic crypto processor has been implemented over 256-bit prime field with minimum clock cycles using new unified modular inversion algorithm in [8].

There are various standard bodies leading to the implementation of security protocols for the industrial applications. Some of the organizations involved in standard activities are Internet Engineering Task Force (IETF), American Bankers Association, International Telecommunication Union, IEEE and National Institute of Standards and technology (NIST) [9]. There are ten NIST -recommended elliptic curves. Among them five is for prime field and another five is for binary field. The processor which supports all five NIST-recommended primes of sizes 192, 224, 256, 384 and 521 bits has been designed in [10]. The characteristics of NIST-recommended prime fields can be analyzed by its software implementation [11].

ECC can also be used to encrypt the image efficiently. There are many research have been run for image encryption. Image can be encrypted by mapping the image pixels into the points on the elliptic curve. Even text can also be encrypted efficiently by converting it into image [12]. Aim of cryptography is to achieve secured communication through insecure channel. Image can be securely transmitted using ECC over finite field [13].

In different fields, ECC can be used for secured data communication. Combination of ECC and AMP (Authentication via Memorable Password) can produce stronger authentication protocol [14]. ECC can be used to provide high authentication and security to the messages with small computation time [15]. The large number of security vulnerabilities in Wireless Sensor Networks (WSN) can be avoided by ECC [16].

Multiplication is the basic building block of ECC systems. In practice, there are different algorithms employed for multiplication [17-20]. With the usage of

Hybrid Karatsuba multiplier high speed and optimized design can be achieved.

2. ELLPTIC CURVE CRYPTOGRAPHY

The mathematical operations in ECC are based on the equation of elliptic curve.

$$y^2 = x^3 + ax + b \quad (1)$$

For different values of a and b different elliptic curves can be obtained. The values of a and b should satisfy the following condition,

$$4a^3 + 27b^2 \neq 0 \quad (2)$$

If the condition is not satisfied, curve will be non smooth and which is called singular curve. The non smooth curve is not safe for cryptography.

a) Layers of ECC

Figure-1 shows the layers of ECC which include finite field arithmetic, point addition, point doubling, point multiplication and protocols. The operations in the top layers are influenced by the operations in the lower layers.

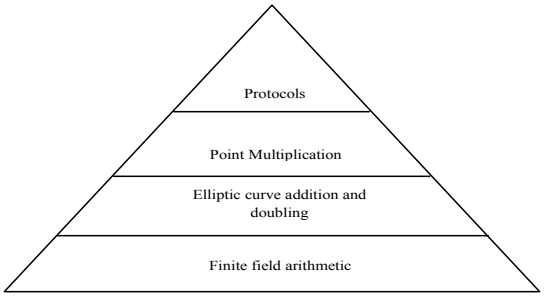

Figure-1. Layers of ECC.

b) Protocols

The first layer of ECC is protocols which include key generation and Exchange protocol, encryption algorithm, decryption algorithm, signature generation and signature verification.

c) Point multiplication

Point multiplication or Elliptic Curve Scalar Multiplication (ECSM) is the process of multiplying a scalar value (k) with the point on the elliptic curve (P) to obtain another point (Q) on the same curve.

$$Q = kP \quad (3)$$

Point multiplication can be performed by two fundamental elliptic curve operations.

Point Addition (L = J + K)

Point Doubling (L = 2J)

It uses point addition and point doubling repeatedly so it is called “double and add” method.

If k = 47 then kP = 23P = 2(2(2(2(2P)+P)+P)+P)+P

d) Point addition

Figure-2 explains the point addition which is the process of adding two points in the elliptic curve finite field to obtain the third point in the same field. Point addition of J (x_J , y_J) and K (x_K , y_K) gives a third point L (x_L , y_L).

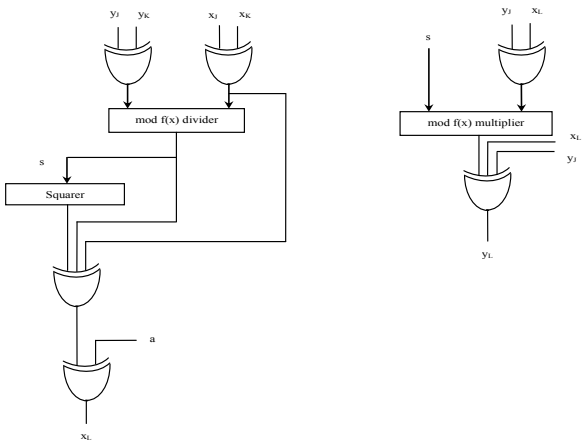

Figure-2. Point addition.

e) Point doubling

Figure-3 explains point doubling which is the process of adding a point in the elliptic curve finite field itself to obtain another point in the same field. Point double of J (x_J , y_J) gives another point L (x_L , y_L).

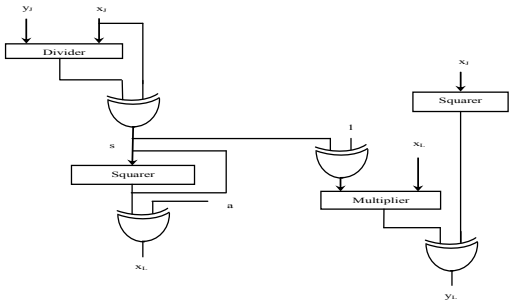

Figure-3. Point doubling.

f) Finite field arithmetic

Cryptographic operations on real numbers are not fast and less accurate due to truncation error. To make the cryptographic operations more efficient and accurate two finite fields are used.

Prime field E(p)

Binary field E(2^m)

If a value exceeds the limits then it wrapped around to finite value by modulo reduction by prime p or irreducible polynomial.

3. PROTOCOLS

A. Key generation and exchange

A public key is a point or element in the elliptic curve finite field and the private key is a random number. Public key is the multiplication of private key and generator point 'G'. The key generation and distribution is explained in Figure-4.

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.

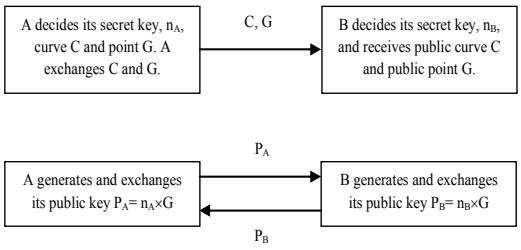

Figure-4. Key generation and exchange.

- 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.

B. Encryption and decryption

To encrypt plain text into cipher text using ECC the plain text M is encoded into the points PM in the finite set of points EP (a,b). A selects a random integer k and computes the cipher text as a pair of points Pc using public key of B as shown in Figure-5.

$$P_c = [kG, P_M + kP_B] \quad (4)$$

where PB is public key of B.

After receiving the cipher text PC, b Multiplies the first point with its private key nB and subtracts the result from the second point.

$$(P_M + kP_B) - n_B kG = P_M + kn_B G - n_B kG = P_M \quad (5)$$

PM is the point on elliptic curve to the corresponding plain text message M.

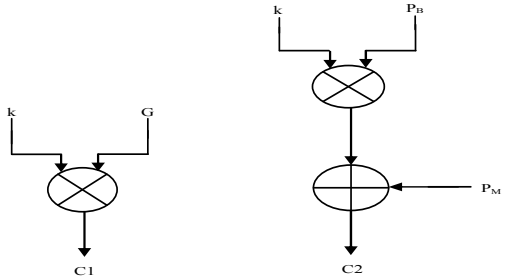

Figure-5. Encryption.

4. FINITE FIELD ARITHMETIC

Implementation of ECC over binary field is more efficient than prime field if the chosen irreducible polynomial is trinomial or pentanomial.

a) Squaring unit

The squaring operation on binary field is as easy as addition. The square of the polynomial $a(x) \in GF(2^m)$ is given by

$$a(x)^2 = \sum_{i=0}^{m-1} a_i x^{2i} \bmod p(x) \quad (6)$$

The squaring spreads the input bits by inserting zeroes in between two bits as shown in Figure-6.

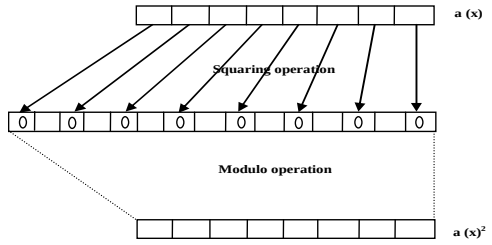

Figure-6. Squaring unit.

b) Hybrid karatsuba multiplication

For high performance crypto processor, a multiplier called Karatsuba multiplier with small delay is suited. 233 bit multiplication is implemented using the hybrid of simple and general karatsuba multiplication. The hierarchy of implementation is shown in Figure-7.

The larger multiplications are performed by the smaller multiplication.

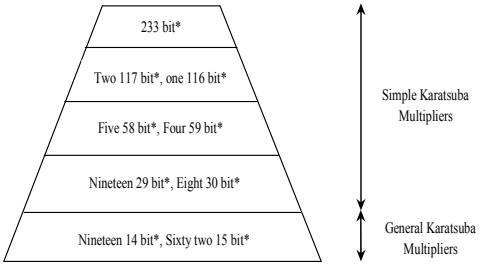

Figure-7. Hybrid Karatsuba multiplication.

c) Modulo reduction

In finite field operation, modulo reduction is required to make the result of squaring and multiplication within the finite field. In binary field the reduction is performed by irreducible polynomial as illustrated in Figure-8. Trinomial or pentanomial makes the reduction easier.

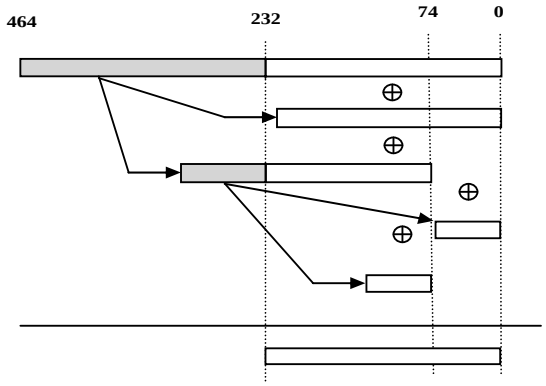

Figure-8. Modulo reduction.

The modulo reduction shown in figure is for the trinomial $x^{233}+x^{74}+1$.

d) Multiplicative inverse

Itoh – Tsujii Multiplicative Inverse (ITMI) algorithm based on Fermat's Little theorem can be used for the efficient implementation of inversing component.

$a \in GF(2^m)$, then inverse of a is given by

$$a^{-1} = a^{2^m-2} = (a^{2^{m-1}-1})^2 \quad (7)$$

$$\beta_k(a) = a^{2^k-1} \in GF(2^m) \quad (8)$$

$$a^{-1} = [\beta_{m-1}(a)]^2 \quad (9)$$

Number of required multiplications is reduced by addition chain. Addition chain for sequence $U = (u_0, u_1, u_2, \ldots u_r)$ satisfies,

1) $u_0 = 1$
2) $u_r = n \ (n \in N)$
3) $u_i = u_j + u_k$ for $k \le j < i$

Inverse of $a \in GF(2^{233})$ is $a^{-1} = [\beta_{232}(a)]^2$ where $\beta_{232}(a) = a^{2^{232}-1}$. Addition chain for 232 is U = (1 2 3 6 7 14 28 29 58 116 232). Inversion has been done in 10 steps with 231 squaring and 10 multiplications.

5. RESULTS AND DISCUSSIONS

Elliptic curve cryptographic and the field operations have been programmed using verilog, synthesised in ISE Design Suite 14.6. Simulations have been done by ISim Simulator and the designs have been implemented in Virtex-5 FPGA board.

a) Point addition

Point addition is the process of adding two points in the elliptic curve finite field to obtain another point on the same field. Point addition is programmed using verilog and its simulation result and RTL schematic are shown in Figure-9 & 10 respectively.

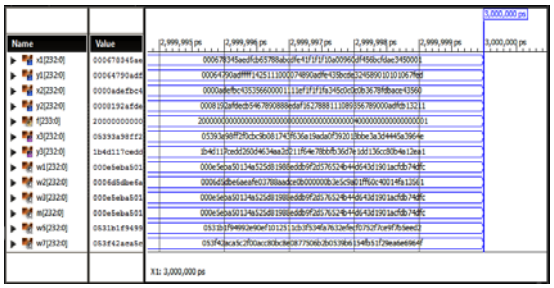

Figure-9. Simulation result of point addition.

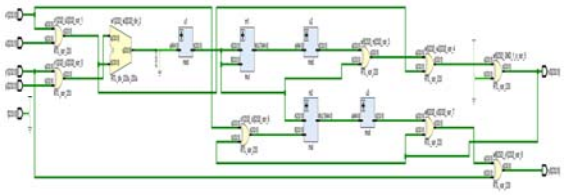

Figure-10. RTL schematic of point addition.

b) Point doubling

Point doubling is the process of adding a point in the elliptic curve finite field to itself to obtain another point on the same field. Point doubling is programmed using verilog and its simulation result and RTL schematic are shown in Figure-11 & 12, respectively.

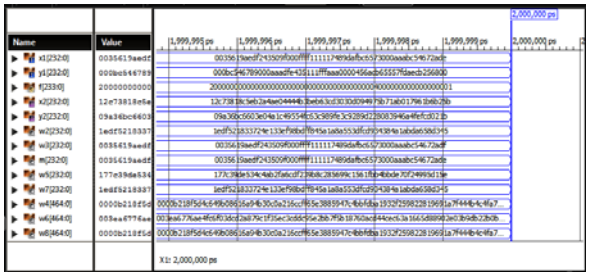

Figure-11. Simulation result of point doubling.

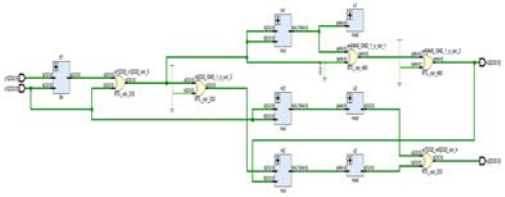

Figure-12. RTL schematic of point doubling.

c) Point multiplication

It is the operation on which the security of elliptic curve cryptosystem relies on. It is more important in key generation and encryption.

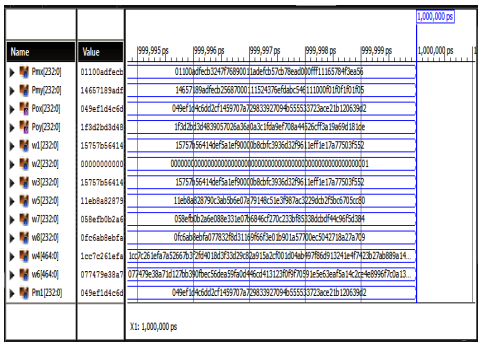

Figure-13. Simulation result of point multiplication.

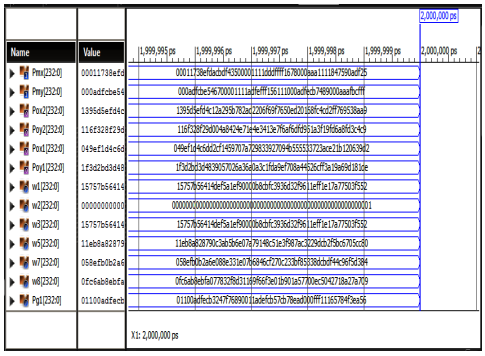

Figure-14. Simulation result of encryption.

d) Karatsuba multiplier

For a 233 bit Hybrid Karatsuba multiplication algorithm the number of LUT slices utilized is only 17%, shown in Figure-16 and the maximum delay obtained is 13.678ns on a Xilinx Vertex 5 FPGA.

The simulation of 233 bit Hybrid Karatsuba Multiplier is performed in ISim Simulator and it is shown in Figure-15 and its RTL schematic is shown in Figure-17.

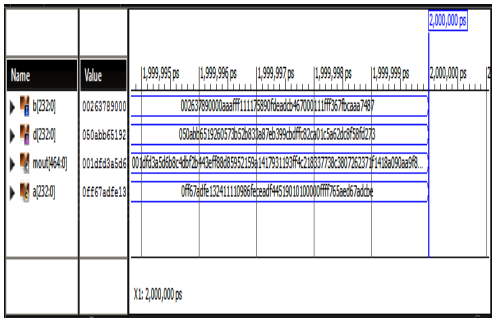

Figure-15. Simulation result of Karatsuba multiplication.

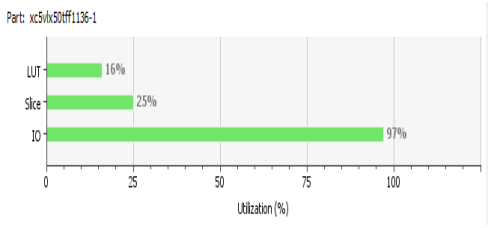

Figure-16. LUT Utilization.

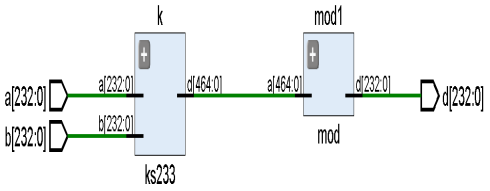

Figure-17. RTL schematic of Karatsuba multiplication.

The comparison of different multiplier designs based on estimated LUT slices utilization and delay is given in TABLE I. The hybrid Karatsuba multiplier uses less LUT slices than other two designs but the delay is more than bit parallel FFM.

Table-1. Comparison table.

	LUT slices	**Delay** (ns)
Massey Omura [20]	127 %	15.91
Bit parallel FFM based on KOM [19]	39 %	7.68
Hybrid Karatsuba multiplier	17 %	13

6. CONCLUSIONS AND FUTURE WORK

a) Conclusions

Elliptic Curve Cryptography is an efficient public key cryptography with minimum key size. The elliptic curve encryption has been implemented using Elliptic Curve Scalar Multiplication (ECSM). The finite field operations such as multiplication, squaring and inversion have been implemented in FPGA. The Hybrid Karatsuba multiplier combines the merits of the Simple and the General Karatsuba algorithms. This resulted in lesser hardware requirements on a FPGA but the delay is more. Delay can be reduced by bit parallel mechanism. The basic operations for ECSM and encryption are point addition and point doubling which have been implemented. ECSM and elliptic curve encryption algorithm have been implemented using point addition and point doubling.

Elliptic curve cryptographic operations have been programmed and synthesised in ISE Design Suite 14.6. Simulations have been done by ISim Simulator. The designs have been implemented in Virtex-5 FPGA board.

b) Future work

Future work is to make comparative analysis of different ECSM algorithms based on hardware utilization and optimizing ECC operations using Elliptic Curve Cryptographic algorithms and VLSI Techniques.

REFERENCES

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

[2] Tarun Narayan Shankar. and G. Sahoo. 2009. Cryptography with Elliptic Curves. International Journal of Computer Science And Applications. Vol. 2, No. 1, pp: 38-42.

[3] Sravana Kumar CH. *et al.* 2012. Encryption of Data using Elliptic Curve over Finite Fields. International Journal of Distributed and Parallel Systems (IJDPS). Vol.3, No.1, pp. 301-308.

[4] Rahila Bilal. and M. Rajaram. 2010. High Speed Point Arithmetic Architecture for ECC on FPGA. International Journal on Computer Science and Engineering. Vol. 02, No. 06, pp: 2029-2035.

[5] Gang Chen, Guoqiang Bai. and Hongyi Chen. 2007. 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. Vol. 54, No. 5, pp: 412-416.

[6] Yong Ki Lee. *et al.* 2008. Elliptic-Curve-Based Security Processor for RFID. IEEE Transactions on Computers. Vol. 57, No. 11, pp: 1514-1527.

[7] A. Kaleel Rahuman. and G. Athisha. 2013. Reconfigurable Architecture for Elliptic Curve Cryptography Using FPGA. Article from Hindawi Publishing Corporation. pp: 1-8, DOI: 10.1155/2013/675161.

[8] Ciaran J. McIvor, Máire McLoone. and John V. 2006. McCanny. Hardware Elliptic Curve Cryptographic Processor Over GF(p)", IEEE Transactions on Circuits and Systems—I: Regular Papers. Vol. 53, No. 9, pp: 1946-1957.

[9] Kristin Lauter, Microsoft Corporation. 2004. The advantages Of Elliptic Curve Cryptography For Wireless Security. IEEE Wireless Communications magazine, pp.62-67, February.

[10] Hamad Alrimeih. and Daler Rakhmatov. 2014. Fast and Flexible Hardware Support for ECC Over Multiple Standard Prime Fields. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, to be published in 2014.

[11] Qian Ding, Trey Reece. and William H. 2013. Robinson. Timing Analysis in Software and Hardware to Implement NIST Elliptic Curves over prime Fields. IEEE Magazine. pp.1358-1362.

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

[13] Vinod Kumar Yadav. *et al.* 2012. Public Key Cryptosystem Technique Elliptic Curve Cryptography with Generator g for Image Encryption. International Journal of Computer Technology & Applications. Vol 3 (1), pp. 298-302.

[14] Saed Rezayi, Mona Sotoodeh. and Hojjat Esmaili. 2011. A Password-Based authentication and Key Agreement Protocol for Wireless LAN Based on Elliptic Curve and Digital Signature. International Journal of Computer Science and Information Security. Vol. 9, No. 10, pp: 17-21.

[15] Sri Rangarajan, N. Sai Ram. and N. Vamshi Krishna. 2013. Securing SMS using Cryptography. International Journal of Computer Science and Information Technologies. Vol. 4 (2), pp: 285-288.

[16] Asha Rani Mishra. and Mahesh Singh. 2012. Elliptic Curve Cryptography (ECC) forSecurity in wireless Sensor Network. International Journal of Engineering Research & Technology (IJERT). Vol. 1 Issue 3, pp: 1-6.

[17] Juliano B. Lima. *et al.* 2010. A Karatsuba-Based Algorithm for Polynomial Multiplication in Chebyshev Form. IEEE Transactions on Computers. Vol. 59, No. 6, pp: 835-841.

[18] Laszlo Hars. 2006. Applications of Fast Truncated Multiplication in Cryptography. Hindawi Journal on Embedded Systems. Article ID 61721, pp: 1-9.

[19] Gang Zhou. *et al.* 2010. Complexity Analysis and Efficient Implementations of Bit Parallel Finite Field Multipliers Based on Karatsuba-Ofman Algorithm on FPGAs. IEEE Transactions On Very Large Scale Integration (VLSI) Systems. Vol. 18, No. 7, pp. 1057-1066.

[20] Retheesh D. 2014. Analysis on FPGA Designs of Parallel High Performance Multipliers. International Journal of Communication and Computer Technologies. Vol. 02, No.12, pp. 70-77.