ISSN 2348 – 9928
IJAICT Volume 2, Issue 3, March 2015 Doi:01.0401/ijaict.2014.02.11 Published Online 05 (03) 2015

DESIGN AND IMPLEMENTATION OF EC BASED CRYPTOSYSTEM ON FPGA

Ms. S.P. Arya
PG Scholar, Applied electronics,
Electronice and Communication Engineering,
Vivekanandha College of Engineering for Women,
Tiruchengode, Tamilnadu, India.

Prof. A. Muruganandham
Head of the Department,
Electronice and Communication Engineering,
Vivekanandha College of Engineering for Women
Tiruchengode, Tamilnadu, India.

Abstract — ***As computing and communication devices are equipped with increasingly versatile wireless connection capabilities, the demand for security increases. Cryptography provides a method for securing and authenticating the transmission of information over the insecure channels. Elliptic Curve [EC] Cryptography is a public key cryptography which replaces RSA because of its increased security with lesser number of key bits .EC point multiplication module will be available in majority of secure communication systems. The most crucial operation in Elliptic Curve Cryptosystem is the computation of point multiplication, i.e., computation of kP for given integer k and point P on elliptic curve. This work aims to design and implement elliptic curve based crypto system on a single field programmable gate array (FPGA).The hardware complexity is reduced using normal basis representation of GF and projective co-ordinate representation of elliptic curves.***

Keywords—Elliptic Curve Cryptosystems, Gaussian Normal Basis, Finite Fields, FPGA.

I. INTRODUCTION

Elliptic Curve Cryptography is a public key cryptography which is now replacing the commonly used RSA. It uses lesser number of key bits as compared to RSA. Shorter key implies less memory need and lower power consumption. Elliptic curve cryptography (ECC) is proposed by Miller and Koblitz. EC point multiplication module will be available in majority of secure communication systems. Since, EC point multiplication is replacing RSA as a standard for digital signature and key exchange, the hardware module for the implementation of elliptic curve (EC) point multiplication will be available in the hardware structure of majority of the secure communication systems.

Because of all these factors the architecture for key exchange and message encryption using a single module of EC point multiplication on a time sharing basis becomes highly suitable for battery powered handheld devices This work aims to design and implementation of elliptic curve based cryptosystem on a single field programmable gate array (FPGA). It proposed the use of a single module of elliptic curve point multiplication in a time sharing basis. The hardware complexity is reduced using normal basis representation of GF(2^m) and projective co-ordinate representation of elliptic curves. The flexibility and high speed capability of FPGAs make them a suitable platform for cryptographic applications. In this paper, we propose a high performance elliptic curve cryptosystem over GF(2^m). The proposed architecture is based on standard elliptic curve point multiplication algorithm and uses GNB for GF(2^m) field arithmetic. It uses fast arithmetic units based on a word-level multiplier. It adopts a parallelized point doubling and point addition unit with uniform addressing mode. It utilizes benefits of GNB representation. The proposed architecture leads to a considerable reduction of computational delay time compared with previously proposed hardware implementations.

The remainder of the paper is organized as follows: In Section 2, the mathematical background of elliptic curves is and point multiplication in elliptic curve is discussed. In Section 3, key generation based on EC is presented and the hardware structure for EC point multiplication is discussed. Section 4 is about the algorithm for EC arithmetic. Finally Section 5 describes the FPGA implementation and the experimental result, Concluded in Section 6.

II. MATHEMATICAL BACKGROUND

Elliptic curves have been intensively studied in algebraic geometry and number theory. They are used in constructing efficient and secure cryptosystems. Elliptic Curves are so named because they are described by cubic equations, similar to those used for calculating the circumference of an ellipse. An elliptic curve E over a field F is given by the Weierstrass equation

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (1)$$

ISSN 2348 – 9928
IJAICT Volume 2, Issue 3, March 2015 Doi:01.0401/ijaict.2014.02.11 Published Online 05 (03) 2015

The variables x, y and the constants *a1, a2, a3, a4* & *a6* range over any given algebra that meet field axioms.[1],[2],[3] An elliptic curve over a field *F* is the set of points (x, y) with x, y *F* that satisfy equation (1). In the definition of an elliptic curve is a single element denoted by *O* and called the 'point at infinity'.

Elliptic curves defined over GF (2^m) has great significance since they allow binary operations and are very suitable for hardware implementation. A Galois Field (finite field) GF(2^m)consists of 2^m. elements for some integer 'm' together with addition and multiplication operations that can be defined over polynomials in GF(2). Elliptic curves over GF(2^m) are defined by a cubic equation in which the variables and coefficients take on values in GF(2^m). So, all mathematical operations on EC are performed using the rules of arithmetic in GF(2^m) [8],[3]. Since the characteristic of the finite field GF(2^m) is 2, the equation (1) can be transformed by suitable change of variables to get the following forms

$$y^2 + xy = x^3 + a_2 x^2 + a_6 \quad (2)$$

$$y^2 + a_3 y = x^3 + a_4 x + a_6 \quad (3)$$

The set **E** (a_2, a_6) consisting of all pairs (x, y) that satisfy equation (2) together with the point at infinity ***O*** form an abelian group if $a_6 \neq 0$. This type of curves is obtained if *a1* in equation (1) is non zero. These curves are non-super singular elliptic curves. The set **E** (a_3, a_4, a_6) consisting of all pairs of (x, y) that satisfy equation (3) together with the point at infinity ***O*** form an abelian group if $a_3 \neq 0$. These curves are super singular elliptic curves. This type of curve is obtained if $a_1$in equation (1) is zero. Here we will be considering the non-super singular elliptic curves only as they provide the highest security in GF(2^m).

Rules for addition over non-super singular curves over GF(2^m) can be stated as follows:[2]

For all points **P**, **Q** **E** (a_2,a_6),

1. **P** + **O** = **P**
2. If **P** = (x_1, y_1), then **–P** = (x_1, x_1+y_1)
3. Addition formula: If **P** = (x_1, y_1) and **Q**=(x_2, y_2), then **P**+**Q** = **R** = (x_3, y_3) is given by the 'tangent and chord' method

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2$$
$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \quad (4)$$

where = $(y_1 + y_2)/(x_1 + x_2)$

4. Doubling formula: If **P** = (x, y), then **2P** =**R** = $(x_3, y_3)$is given by

$$x_3 = \lambda^2 + \lambda + a_2$$
$$y_3 = x_1^2 + \lambda x_3 + x_3 \quad (5)$$

Where $\lambda = x_1 + y_1/x_1$

Thus, adding two elliptic curve points as well as doubling an elliptic curve point requires one inversion and two multiplications each over the underlying finite field GF (2^m) . Computing inverses is relatively expensive in comparison to multiplication in GF(2^m). In order to avoid computing inverses the point P(x,y) in affine coordinates can be converted to projective coordinate as (x,y,1). A point P(X, Y, Z) in projective coordinates can be converted to affine coordinates as (X/Z, Y/Z) provided Z ≠ 0. Z = 0 implies a point at infinity. For projective coordinates representation of the affine points, the common denominator for X and Y coordinates are taken as Z .The projective equation of the EC is given by

$$Y^2Z + XYZ = X^3 + a_2X^2Z + a_6Z^3 \quad (6)$$

2.1 Elliptic curve addition

Let P = (x_1,y_1, z_1),Q = (x_2,y_2, z_2) such that P ≠ ±Q then P + Q = R = (x_3, y_3, z_3) is given by

A = $y_1z_2 + z_1y_2$,,
B = $x_1z_2 + z_1x_2$,
C = B^2,
D = z_1z_2,
E = $(A^2 + AB + a_2C)D + BC$

Thus for EC addition,

$$x_3 = BE,$$
$$y_3 = C(Ax_1 + y_1B)z_2 + (A + B)E$$
$$z_3 = B^3D. \quad (7)$$

2.2 Elliptic curve doubling

If P = (x_1,y_1, z_1) then 2P = R = (x_3, y_3, z_3) is given by

A = x_1^2
B= $A + y_1z_1$,
C= x_1z_1,
D = c_2,
E=$(B^2+BC + a_2D)$

Thus for EC doubling,

$$= CE,$$
$$= (B + C)E + C,$$
$$= CD. \quad (8)$$

In projective coordinates, no inversion is needed.

2.3 Point Multiplication on Elliptic Curves

If P is a point on the elliptic curve and 'k' is any integer, computing a new point 'kP' returns another point on EC. This operation is called point multiplication operation on EC. The EC point multiplication is computed by repeated point additions which is same as adding the point P to itself 'k' times and point doubling which is same as multiplying with 2. The point multiplication operation can be implemented with a number of point addition and doubling operations. For example, 5P can be written as 2(2P) + P, which can be implemented as a combination of two doubling and one addition operation. Thus for a given point P on the EC and any integer 'k', computation of kP is easy and also at the same time, computing k from R and P is extremely difficult. This is called the *elliptic curve discrete logarithm problem* .The EC operations in turn are composed of basic operations in the underlying finite field (FF or GF)[4],[7].

Elliptic curve point multiplication is a one-way function because the computation in one direction is easy while that in the opposite direction is difficult,. This is the underlying mathematical problem that provides security and strength to EC-based cryptosystems. Non-supersingular curves are considered to be more secure compared to supersingular elliptic curves.

2.4 Finite Field Arithmetic

1. Galois Field Addition

If A = (, ,…. ,) and B = (, ,…. ,) are elements of GF( ), then the sum C =A+B = (, ,…. ,) , where $ci = ai \oplus bi$. Therefore sum can be obtained as bitwise XORing of A and B.

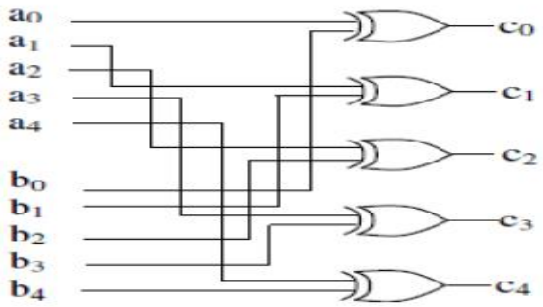

Fig.1: GF Addition Unit

2. Galois Field Squaring

Squaring of an element A in the normal basis representation is a cyclic shift operation. Hence, the hardware implementation of squaring operation requires only a shift register

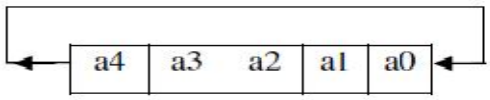

Fig. 2: GF Squaring Unit

3. Galois Field Multiplication

It is very costly in terms of hardware requirement. The number of clock cycles required for its computation depends on the particular architecture of the FF multiplier. So, an attempt is being made to reduce hardware complexity required for multiplication.

III. KEY GENERATION

Private key cryptosystem and public key cryptosystem are the two kinds of cryptosystems that implement cryptographic algorithms. In a private key cryptosystem both communicating entities share a secret key through a secure and authenticated channel. This secret key is used for both encryption and decryption of data. Private Key cryptography is used for the encryption of data due to its speed and reduced complexity of operations. However, it has certain shortcomings.[5],[7]

Key Management Problem :In a broadcast communication scenario, each user will have to communicate with many different ones. Thus, communication on a public network is not restricted to one-on-one. For a network of *n* users, $n(n-1)/2$ private keys need to be generated. When *n* is large, the number of keys becomes unmanageable.

Key Distribution Problem: With such a large number of keys that need to be generated on a network, the job of generating the keys and finding a secure channel to distribute them becomes a burden.

No digital signatures possible: A digital signature is an electronic analogue of a handwritten signature. If Alice sends an encrypted message to Bob, Bob should be able to verify that the received message is indeed from Alice. This can be done with Alice's signature. Private key cryptography does not allow such a feature. But, public key

cryptography uses two keys. Each user on a network publishes a public encryption key that anyone can use to send them messages, while keeping the private key secret for decryption. On a network of *n* users, it only needs *n* public and *n* private keys. Furthermore, it allows the use of digital signatures. However, public key cryptography does have its drawbacks.[9]

Public and private key cryptography work best together. Public key cryptography is ideal for key distribution and management, while private key cryptography is ideal for ensuring *confidentiality*, such as encrypting data and communication channels. Thus in this hardware implementation public key cryptography is used for key exchange and private key cryptography is used for message encryption.[2]

3.1 Hardware Structure for Ec Point Multiplication

This work concentrates on hardware implementation of encryption system, integrity verification system and key exchange system. In the design, the point multiplication operation is realized through point addition and point doubling operations on EC. As it is clear from the formulae for point addition equation (7) and point doubling equation (8), these operations are performed through addition, multiplication and squaring operations. Here the used variables are elements of the basic field over which the EC is defined. The hierarchy of arithmetic for EC point multiplication for a point **P** on an elliptic curve is as shown in Fig 3.

Since EC over GF( ) are more suitable for hardware implementation, the basic operations are to be done in GF( ).

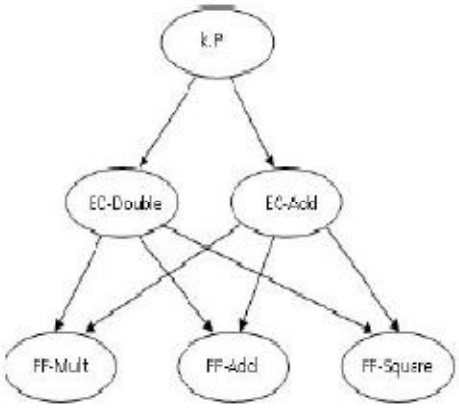

Fig. 3 : EC arithmetic hierarchy

The finite field addition (FF-Add) and finite field squaring (FF-Square) operations are quite simple. These operations can be done with very few clock cycles. But finite field multiplication (FF-Mult) is very costly in terms of hardware requirement. The number of clock cycles required for its computation depends on the particular architecture of the FF multiplier.

3.2 Algorithm for Ec Arithmetic

The Elliptic Curve point multiplication computed by repeated point additions and point doubling.

Algorithm:

Input :An integer $k>0$, Point P on EC.

Output: $Q = k.P$

Step1: Set $k=$

Step2: Set $Q \leftarrow P$,

Step3: for i from $l-1$ down to 0 do

$Q \leftarrow 2Q$,

If =1

$Q \leftarrow P+Q$,

end if

end for

Step4: Return(Q)

The architecture for the EC point multiplication can be obtained by combining the EC addition and EC doubling architectures. If is '1' then first an EC doubling operation and then an EC addition operation is to done. If is '0' then only an EC doubling operation is to done.

IV. FPGA IMPLEMENTATION

The complete structural block diagram of the implementation of generation is shown below. The hardware requires three different clock frequencies which can be generated either using an independent clock generation circuit or using the Digital Clock Manager in FPGA. The hardware consists request and grant signals. Once the initial handshaking is done the process of key exchange is initiated. The hardware randomly selects an integer and stored in the memory. Initial seed point is the point P. EC point multiplication block computes the result according to the algorithm. The control signals to the various blocks are generated by the controller. A controller generates various control signals to synchronize operations within the EC point multiplication unit. These operations

ISSN 2348 – 9928
IJAICT Volume 2, Issue 3, March 2015 Doi:01.0401/ijaict.2014.02.11 Published Online 05 (03) 2015

include loading a new integer input into the point multiplication block and passing control between doubling and addition units. Finally loading the result into output lines. The counter synchronizes the starting and ending of a point multiplication. After the completion of point multiplication in the projective coordinates, the z coordinate of the result is passed into an inversion block.. Then the inverse of z is multiplied with x and y coordinates of the result and we get the point multiplied result in the affine form. The integer for point multiplication, k, is entered serially into the multiplication block in the order of MSB to LSB. The serial input is the i th bit of the number k. For the first non-zero , the point P is stored as such in projective co-ordinate representation.

Depending on the value of the following 's, the controller enables the EC-addition module for EC addition operation and EC-doubling module for EC doubling operation. The ADD sequential machine does the sequential actions. It generates the necessary control signals for the EC addition operation. Similarly the DOUBLE sequential machine generates the necessary control signals for the EC doubling operation. The intermediate results are stored in the register/RAM array. Clock division is provided for the GF multiplication. The EC addition operation and EC doubling operation requires additional one or more clock cycles.

4.1 Experimental Result

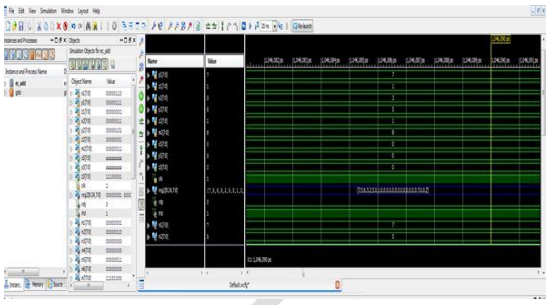

Fig 4 : Simulation Result of elliptic curve addition

TABLE.1 Synthesis Result of Hardware for Key Generation

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slices	480	5888	8%
Number of Slice Flip Flops	175	11776	1%
Number of 4 input LUTs	925	11776	7%
Number of bonded IOBs	37	311	11%
Number of GCLKs	3	24	12%

Maximum output required time after clock: 5.248ns.

Timing summary

Minimum period: 16.107ns

Maximum frequency: 62.084MHz.

Minimum input arrival time before clock: 6.034ns.

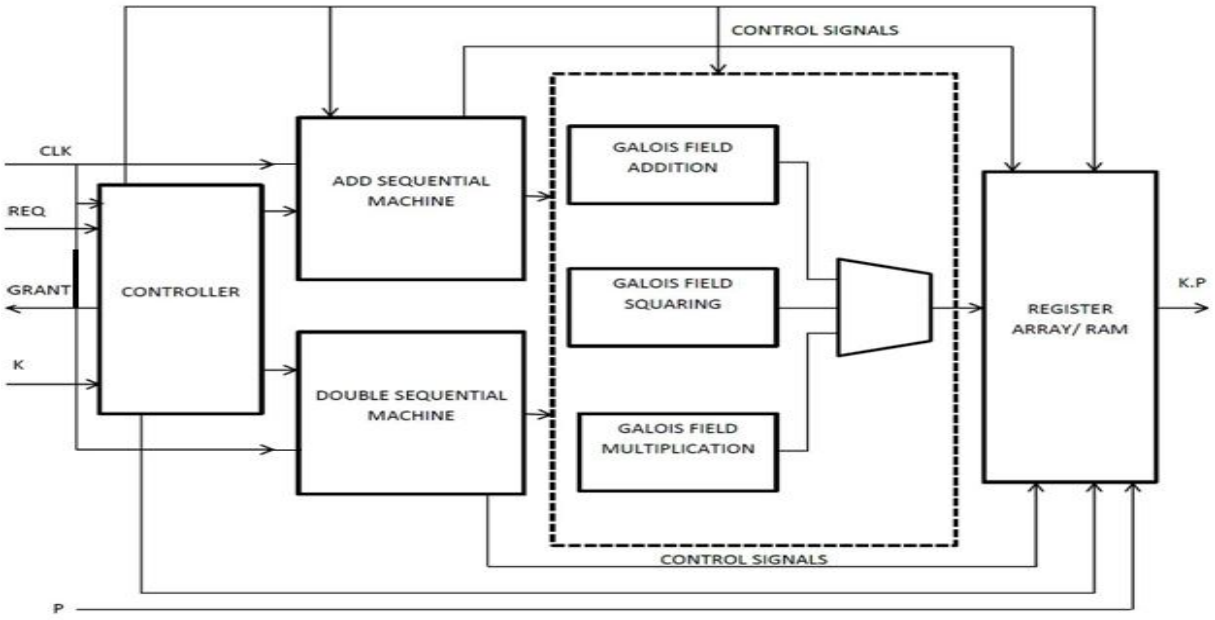

Fig 5 : Structural Block Diagram Of The Hardware Implemented

V. CONCLUSION

We have designed and implemented the EC based Cryptosystem on FPGA Virtex6 (XC6VCX75T). From the results, it can be seen that the increase in hardware for implementing message encryption is 1%. Hence the proposed design is an efficient implementation of EC based encryption system with good security. Since EC based key exchange is a popular option for key exchange in many of the modern communication systems, the proposed design is highly relevant in the implementation of secure communication system.

References

[1] K.S.Lalmohan, P.P.Deepthi and P.S. Sathidevi , "Design and Implementation of Secure Stream Cipher based on Elliptic Curves on Time Shared Basis," International Journal of Computer Applications (0975 – 8887) Volume 68– No.21, April 2013.

[2] Lalmohan, Sreekumari, P. Pattathil Deepthi and JilnaPayingat, "Hardware efficient implementation of encryptionand key exchange based on elliptic curves", Proceedings of the IASTED International Conference ,Signal and Image Processing and Applications (SIPA 2011),June 22 - 24, 2011 Crete, Greece.

[3] IEEE 1363, Standard Specifications for Publickey Cryptography, 2000.

[4] NIST,Recommended elliptic curves for federal government use, May 1999. <http://csrc.nist.gov/encryption>.

[5] Rudolf Lidl and HaraldNiederreiter , "Introduction to finite fields and their applications", Cambridge University Press- 2000.

[6] S. Kwon, K. Gaj, C.H. Kim and C.P. Hong, "Efficient linear array for multiplication in GF(2^m) using a normal basis for elliptic curve cryptography", in: CHES 2004, Lecture Notes in Computer Science, vol. 3156, 2004, pp. 76–91.

[7] Henri Cohen and Gerhard Frey, "Handbook of Elliptic and Hyper Elliptic Curve Cryptography", Chapman & Hall/CRC, 2006.

[8] Neal Koblitz, "A Course in Number Theory and Cryptography" Springer2005,SecondEdition