ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 6, Special Issue 5, March 2017

National Conference on Advanced Computing, Communication and Electrical Systems - (NCACCES'17)

24th - 25th March 2017

Organized by

C. H. Mohammed Koya
KMEA Engineering College, Kerala- 683561, India

High-Performance Architecture of Elliptic Curve Scalar Multiplication Over GF (2^m)

Shahana K S [1], Prof. Riyas K S [2]

P.G Student, Department of Electronics and Communication Engineering, Rajiv Gandhi Institute of Technology, Kottayam, Kerala, India [1]

Associate Professor, Department of Electronics and Communication Engineering ,Rajiv Gandhi Institute of Technology, Kottayam, Kerala, India [2]

ABSTRACT: In the present era, the communication field has grown tremendously. Applications such as online banking, wireless sensor networks, mobile communication etc. increased the need for security in resource constrained environment. These applications can be secured using several cryptographic algorithms. Cryptography is used to encrypt and decrypt data. It helps people to store or transmit sensitive information via insecure network. The two main categories of cryptography are public-key cryptography and symmetric key cryptography. Elliptic curve cryptography (ECC) acts as a perfect cryptographic tool. It is a public key cryptographic algorithm. It provides security with shorter key sizes compared to other standard public key algorithms. Transactions over internet have been increasing day by day. Securing such electronic transactions has become an important issue. This paper explores a high-performance architecture for elliptic curve scalar multiplication over Galois field GF (2^m). It relies on cryptographic algorithm and is used to secure data transfer over insecure networks.

KEYWORDS: Elliptic curve cryptography (ECC), field-programmable gate array (FPGA), Karatsuba-Ofman multiplier (KOM), scalar multiplication.

I.INTRODUCTION

Enormous data such as credit card numbers and social security numbers are transmitted over the Internet during transactions. Thousands of transactions take place over the world-wide web day by day. Securing electronic transaction has become a very important problem. Cryptography can be used to provide and assure confidentiality and integrity of such transactions and it is an efficient way to protect and secure information.

Cryptology is the science that deals with providing secure communications. The aim of cryptology is to construct schemes that allow only authorized access to information. All malicious attempts to access secure information are prevented. A cryptographic key is used to identify an authorized access. All the users are not allowed to access the information while a user having the right key will be able to access hidden information. Cryptology consists of cryptography and cryptanalysis. Cryptography involves the study and application of various techniques through which information may be rendered unintelligible to all but the intended receiver. Cryptanalysis is the science of breaking crypto systems and recovering the secret information.

There are basically two types of cryptographic algorithms. They are symmetric key and asymmetric key algorithms. Symmetric key cryptographic algorithms have a single key for both encryption and decryption. In asymmetric key cryptographic algorithms two keys are involved. A private key and a public key. The private key is kept secret while the public key is known to everyone. The encryption is done using the public key, and the encrypted message can be decrypted by using the corresponding private key. Security of these algorithms relies on the hardness of deriving the private key from the public key. Asymmetric key has immense advantages even though it is slow and highly complex.

The most used asymmetric key crypto algorithm is RSA (Ron Rivest, Adi Shamir and Leonard Adleman). ECC has gained popularity due to their higher level of security with lower key sizes.

In [14] Christoforus Juan Benvenuto described about Galois field in cryptography. Elliptic curves are usually defined over Finite Fields (FFs). The most commonly used FFs are prime fields GF (p) and binary fields GF (2^m). Both provide the same level of security. ECC over GF (2^m) is more efficient for hardware implementation because arithmetic in GF (2^m) is carry-free. Galois Field (GF) is useful in translating computer data as they are represented in binary forms. The computer data consist of combination of two numbers, 0 and 1, which are the components in Galois field whose number of elements is two. Galois field is widely used in Cryptography.

In [1] Lijuan Li *et al.* proposed a pipelined architecture of elliptic curve scalar multiplication (ECSM) over GF (2^m). The architecture uses a bit-parallel finite field (FF) multiplier accumulator (MAC) based on the Karatsuba-Ofman algorithm. Zia *et al.* [2] introduced a high-speed implementation of ECC on FPGA over GF (2^m). Such design and implementations are highly desirable for applications requiring high speed cost effective and flexible implementations of public key cryptography. Hussein *et.al* [13] proposed a new and highly efficient architecture for elliptic curve scalar point multiplication. For achieving the maximum architectural and timing improvements, reorganize and reorder the critical path of the Lopez Dahab scalar point multiplication architecture.

Gustavo D. Sutter *et.al* [3] presented a new high-speed point multiplier for elliptic curve cryptography using field programmable gate array. They explored various digit-serial approaches in Galois field multiplication and division in order to construct the crypto processor. Shuai Liu *et.al* [5] proposed high performance hardware implementation architecture of elliptic curve scalar multiplication over binary fields. The architecture is based on the Montgomery ladder method and uses polynomial basis for finite field (FF) arithmetic. Designs [9]-[11] presented a method for producing efficient hardware designs for Elliptic Curve Cryptography (ECC) over binary fields.

In this paper, we use a bit-parallel Finite Field multiplier accumulator (MAC) based on KOM, along with one FF squarer, to implement ECSM. A Pipelined FF MAC based on KOM is also introduced to compare the performance. The rest of this paper is organized as follows. Section II describes Elliptic Curve Cryptography in detail along with key generation, encryption, decryption and Elliptic Curve Scalar Multiplication. Section III elucidates the pipelined and non-pipelined architectures of ECC. Section IV present the comparison result and discussion part. Chapter V concludes the paper.

II. ELLIPTIC CURVE CRYPTOGRAPHY (ECC)

Elliptic curve cryptography (ECC) was proposed by Neal Koblitz and Victor Miller in 1985. Providing an equivalent level of security with smaller key size is an advantage of ECC compared to RSA. It is very efficient to implement ECC. Lower power consumption and faster computation are the other advantages of ECC over RSA. It also provides small memory and bandwidth because of its key size length. Wireless devices and smart cards present a good example for the constrained devices with limited resources. ECC relies mainly on abstract algebra for its construction.

Arithmetic in elliptic curves requires a number of modules to calculate ECC operations such as modular multiplication, modular division, and modular addition/subtraction [14]. ECC provides unique properties in mathematical structure which is adding two points on a specific curve and getting the result point on the same curve. This specific feature gives us an opportunity to use ECC in the cryptography due to the difficulties of finding our private key and breaking this protocol needs the advanced mathematics. Elliptic curves over GF (2^m) are particularly appealing due to the fact that the binary field operations are more efficiently implemented in hardware and software. Fig.1 shows the hierarchy of ECC.

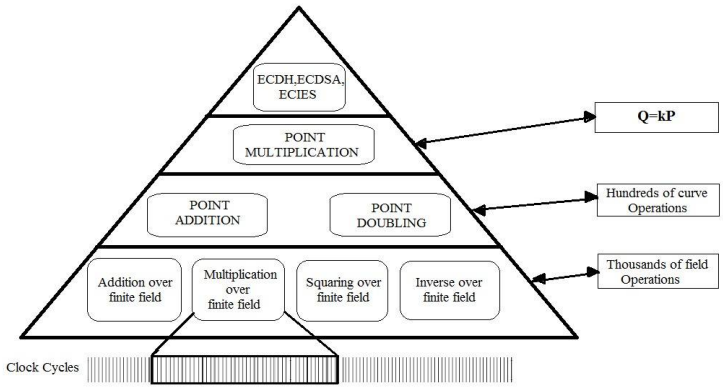

Fig. 1: Hierarchy of ECC

Key Generation: Key generation is an important part in ECC where we have to generate both public key and private key. The sender will be encrypting the message with receiver's public key and receiver will decrypt its private key. Let 'E' represents the elliptic curve 'P' a point on the curve and 'n' be a number which is prime which indicates the key length. Now select a number'd' within the range 'n'. Using the following equation we can generate the public key,

$$Q = d \square P \quad (1)$$

d = random number within the range (1 to n-1). 'Q' is the public key and 'P' is the private key.

Encryption: Let 'm' be the message that we are sending. We have to represent this message on the curve. Consider 'm' has the point 'M' on the curve 'E'. Randomly select 'k' from [1 - (n-1)]. Two cipher texts will be generated let it be C1 and C2.

$$C_1 = k \square P \quad (2)$$

$$C_2 = M + k \square Q \quad (3)$$

C_1 and C_2 will be sending. During decryption we get back the message 'm' that is send to us where M is the original message that we have send.

$$M = C_2 - d \square C_1 \quad (4)$$

Elliptic Curve Scalar Multiplication: The underlying operation in elliptic curve cryptosystems is scalar point multiplication Q = k·P, i.e., the multiplication of an elliptic curve point P by a scalar k to give the resultant point Q. Scalar multiplication is performed by repeated point addition and doubling, which essentially rely on a series of FF arithmetic's, such as multiplication, square, addition, and inversion. Point addition and doubling in affine coordinate involve inversion each time. As inversion is the most complex and time-consuming operation in FF, it is more practical to represent the curve points using a projective coordinate [1]. In this way, only one inversion is needed for the entire scalar multiplication at the expense of more other FF operations. The equation of an elliptic curve is given as,

$$y^2 + xy = x^3 + ax + b \quad (5)$$

Algorithm for Scalar Multiplication: The basic algorithm for ECC is scalar multiplication. The Montgomery Ladder Scalar Multiplication algorithm for scalar multiplication is shown in Algorithm 1. The input to the algorithm is elliptic curve point 'P' and n bit scalar 'k'. The result is scalar product k P. The Montgomery ladder algorithm computes both point doubling and point addition for every bit of k (denoted as k_i). The x-coordinate (X and Z) is computed and the y-coordinate is recovered in the end. Each iteration of the main loop performs the same operation no matter k_i is 0 or 1.

Algorithm 1 Montgomery Ladder Scalar Multiplication Over GF(2^m)

Input : Basepoint P = (xp,yp) and Scalar k = (km−1,km−2,.....k0)2 where km−1=1
Output : Point on the curve Q = kP = (x3,y3)
/*Initialisation: Affine to Projective*/
1: X1← xp, Z1 ← 1, X2 ← xp4 + b, Z2 ← xp2
2: for i from m − 2 down to 0 do
3: if ki = 1 then
4: T ← Z1, Z1 ← (X1Z2 + X2Z1)2, X1 ← xpZ1 + X1X2TZ2.
5: T ← X2, X2 ← X24 + bZ24, Z2 ← T2Z22
6: else
7: T ← Z2, Z2 ← (X1Z2 + X2Z1)2, X2 ← xpZ2 + X1X2TZ1
8: T ← X1, X1 ← X14+ bZ14, Z1 ← T2Z12
9: end if
10: end for
11: x3← X1/Z1
12: y3 ← (xp + X1/Z1)[(X1 + xpZ1)(X2 + xpZ2) + (xp2 + yp)(Z1Z2)](xpZ1Z2)−1 + yp
13:return (x3,y3)

III.ARCHITECTURE OF ELLIPTIC CURVE CRYPTOPROCESSOR

In this section the architecture of elliptic curve crypto processor is described. Figure.3 shows the block diagram of the proposed architecture. It consists of an arithmetic unit (AU), a control unit, a register bank and a ROM. The processor use Montgomery Ladder Scalar Multiplication algorithm described in Algorithm 1. The output is produced as a result of scalar multiplication and the product is k P. P is the base point on the curve and is stored in ROM.

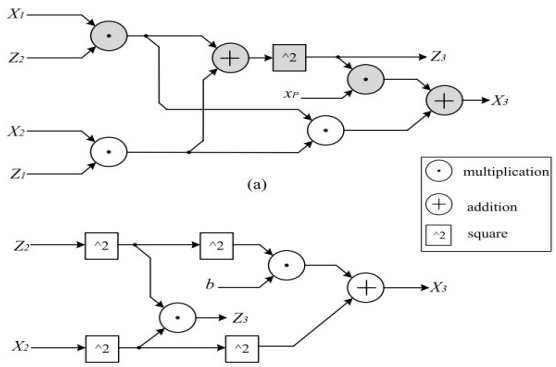

Fig. 2: Data dependence graph of (a) point addition and (b) point doubling in the Montgomery ladder algorithm.

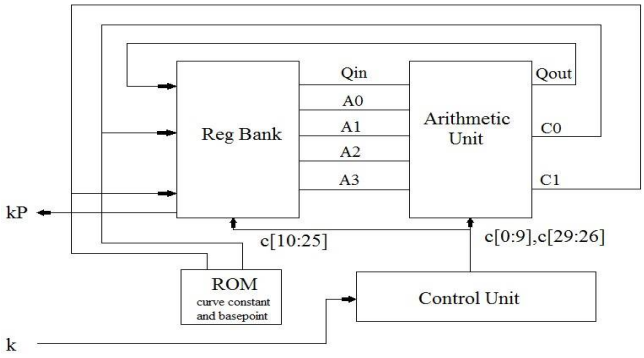

Fig. 3: Block diagram of Elliptic Curve Crypto processor

The scalar multiplication is done using Lopez-Dahab (LD) Projective co-ordinate system. The LD coordinate form of elliptic curve over finite field is,

$$Y^2 + XYZ = X^3 + aX^2Y^2 + bZ^4 \quad (6)$$

The curve constant b and base point P are stored in ROM. a is taken as 1. Initially the curve constant and base points are loaded from ROM into registers. There are two phases of computation. During first phase scalar multiplication takes place. The scalar k is multiplied to base point. During second phase projective point result from first phase is converted to affine co-ordinate kP. Second phase involves inversion which is done using Itoh-Tsujii inverse algorithm. The register bank consists of eight registers of size 233 bits each. The registers are used to store results of computations. Multiplexers associated with register bank determines which of the inputs are to be stored in the register banks. It also determines which output of a register bank gets driven on the output buses.

Karatsuba-Ofman (KOM) algorithm for Finite Field Multiplier Accumulator

The finite field multiplier is the most important component in the elliptic curve crypto processor (ECCP). It occupies the most area on the device. Finite field arithmetic circuits are used to build the arithmetic unit. They are organized for efficient implementation of point addition and point doubling in LD coordinates. There are 5 inputs (A_0 to A_3 and Q_{in}) in Arithmetic Unit (AU) and three outputs (C_0,C_1 and C_{out}) . The main components of the AU are a inversion block named as quad block and a multiplier. The multiplier uses a hybrid Karatsuba algorithm. It is used in both phases (during the scalar multiplication phase and conversion to affine coordinate phase) of the computation. The AU includes several adders and squarer circuits. In the Karatsuba multiplier, the n bit multiplicands $A(x)$ and $B(x)$ represented in polynomial basis are split as shown in Algorithm 2. The multiplication is then done using three n/2 bit multiplications. The Karatsuba multiplier can be applied recursively to each n/2 bit multiplication. This multiplier is best suited when n is a power of 2, the multiplicands are allowed to be broken down until they reach 2 bits. The final recursion consists of 2 bit multiplications and can be achieved by AND gates. Such a multiplier with n a power of 2 is called the basic Karatsuba multiplier. An *n*-bit KOM [1] consists of three (*n*/2)-bit multipliers when n is even, and two $\lceil \frac{n}{2} \rceil$ -bit and one $\lfloor \frac{n}{2} \rfloor$ -bit multipliers when n is odd.

Algorithm 2 Karatsuba-Ofman Algorithm

Input: X, Y: n-digit integers.
Output: the product X ∗ Y.
Comment: We assume n = 2k, by prefixing X, Y with zeros if necessary.
1: if n = 1 then use multiplication table to find T: = X ∗ Y
2: else split X, Y in half:
3: X =: 10n/2X1 + X2
4: Y =: 10n/2Y1 + Y2
5: Comment: X1, X2, Y1, Y2 each have n/2 digits
6: U: = KO(X1, Y1)
7: V: = KO(X2, Y2) 8: W: = KO(X1 − X2, Y1 − Y2)
9: Z: = U + V W
10: T: = 10nU + 10n/2Z + V
11: Comment: So U = X1 ∗ Y1,V = X2 ∗ Y2,W = (X1 − X2) ∗ (Y1 − Y2), and therefore
Z =X1∗Y2+X2∗Y1. Finally we conclude that T = 10nX1∗Y1+10n/2(X1∗Y 2+X2∗Y1)+X2∗Y2 = X∗Y
12: return T

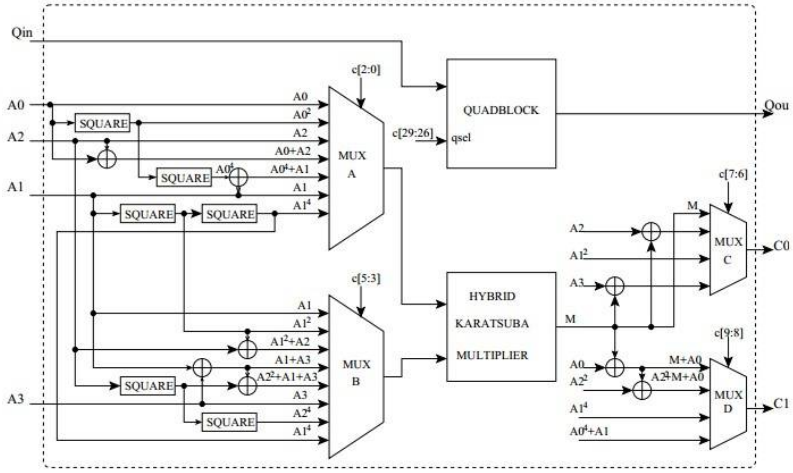

Fig. 4: ECSM using non-pipelined Finite Field Arithmetic Unit (FF MAC).

In this paper a comparison of hybrid KOM and pipelined Hybrid KOM is presented. Fig.5[1] shows pipelined architecture of Elliptic Curve Scalar Multiplier. A comparison between pipelined MAC and non pipelined MAC is made. It is found that pipelined MAC improves the speed of multiplication. Pipelining a combinational logic involves putting levels of registers in the logic to introduce parallelism and, as a result, improve speed. Flip flops introduced by pipelining typically incur a minimum of additional area on FPGAs, by occupying the unused flip flops within logic cells that are already used for implementing combinational logic in the design.

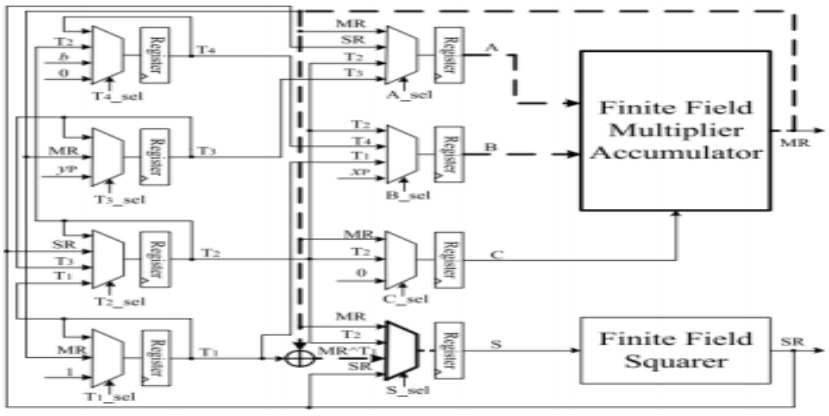

Fig. 5: ECSM using a pipelined FF MAC.

IV. RESULTS AND DISCUSSIONS

Finite Field Multiplication is the most important operation in Elliptic Curve Crypto processor. It performs scalar multiplication as well as inversion from projective co-ordinate to affine co-ordinate. Table.1 shows the comparison results of two designs. The results show that pipelined KOM provide better results than non-pipelined one. Multiplication speed is increased by pipelining.

TABLE 1: Result Comparison of Elliptic Curve Scalar Multipliers

Work	Field(n)	Freq(MHz)	Clock cycles	Time(μs)
Hybrid KOM	163 bit	147	1429	9.7
Pipelined Hybrid KOM	233 bit	244	1926	7.9

V. CONCLUSION

This paper focuses on improving the performance of ECSM over GF (2^m) on FPGA. The most important factor that affects the performance of a Elliptic Curve Crypto Processor (ECCP) is the finite field multiplication and finite field inversion. The architecture mainly consists of a bit-parallel FF MAC. The multiplier is based on Hybrid Karatsuba-Ofman Algorithm. The hybrid Karatsuba multiplier is a recursive algorithm. It performs the initial recursions using the simple Karatsuba multiplier, while the final recursion is done using the general Karatsuba multiplier. Hybrid karatsuba multiplication algorithm along with pipelining is found to improve speed of multiplication. Performance improvement can be done using pipelined MAC. Montgomery Ladder scalar multiplication algorithm is used for scalar multiplication. The Itoh Tsujii inversion algorithm is used to find the multiplicative inverse which is required to convert projective co-ordinate to affine coordinate during post process stage.

REFERENCES

[1] Lijuan Li and Shuguo Li, "High-Performance Pipelined Architecture of Elliptic Curve Scalar Multiplication Over GF(2^m)", *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, Vol. 24, No. 4, April 2016

[2] Zia U. A. Khan and Mohammed Benaissa, "High Speed ECC Implementation on FPGA over GF(2^m)", *IEEE Conference*, 2015

[3] Gustavo D. Sutter, Jean-Pierre Deschamps and Jose Luis Imana "Efficient Elliptic Curve Point Multiplication Using Digit-Serial Binary Field Operations" ,*IEEE Transactions on Industrial Electronice* , Vol. 60, No. 1, January 2013

[4] Dimitrios Schinianakis, Athanasios Kakarountas, Thanos Stouraitis and Alexander Skavantzos, "Elliptic Curve Point Multiplication in GF(2n) using Polynomial Residue Arithmetic", IEEE Conference, 2009

[5] Shuai Liu, Lei Ju, Xiaojun Cai, Zhiping Jia, Zhiyong Zhang, "High Performance FPGA Implementation of Elliptic Curve Cryptography over Binary Fields", IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, 2013

[6] Anita Aghaie, "Efficient Elliptic Curve Point Multiplication with Montgomery Ladder Algorithm" ,IEEE ,2013

[7] Arash Reyhani-Masoleh and M. Anwar Hasan, "Low Complexity Bit Parallel Architectures for Polynomial Basis Multiplication over GF(2m)", IEEE Transactions On Computers, Vol. 53, No. 8, August 2004

[8] Huapeng Wu, "Bit-Parallel Finite Field Multiplier and Squarer Using Polynomial Basis", IEEE Transactions On Computers, Vol. 51, No. 7, July 2002

[9] Sujoy Sinha Roy, Chester Rebeiro, and Debdeep Mukhopadhyay, "Theoretical Modeling of Elliptic Curve Scalar Multiplier on LUT-Based FPGAs for Area and Speed", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 5, May 2013

[10] Hero Modares, "A Scalar Multiplication in Elliptic Curve Cryptography with Binary Polynomial Operations in Galois Field", E-CASE conferenceSingapore ,2009

[11] Ray C. C. Cheung, Nicolas Jean-baptiste Telle, Wayne Luk, and Peter Y. K. Cheung, "Customizable Elliptic Curve Cryptosystems", IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 13, No. 9, September 2005

[12] Gang Zhou, Harald Michalik, and Laszlo Hinsenkamp, "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, July 2010

[13] Hossein Mahdizadeh and Massoud Masoumi, "Novel Architecture for Efficient FPGA Implementation of Elliptic Curve Cryptographic Processor Over GF(2163)" IEEE Transactions On Very Large Scale Integration(VLSI)Systems, 2013

[14] Christoforus Juan Benvenuto, "Galois Field in Cryptography", May 31, 2012