International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39799-39802

Efficient Multiplication in Finite Field for FPGA Design of EC Cryptosystem

Ashkar Mohammed M

Research Scholar Department of Electronics and Instrumentation Engineering NICHE, KK District Tamil Nadu
ashkarmohammed@yahoo.co.in

Dr. S. Suresh Babu

Principal Sree Budha College of Engineering Alappuzha District Kerala
drssbtkm@gmail.com

Abstract
The Elliptic Curve Cryptography (ECC) was introduced in the 1980s. It has superior strength per bit compared to the existing public key cryptosystems such as RSA. This fact made ECC a popular one. It can provide a secured data communication among the portable devices using small key sizes. The scalar multiplication carried out in ECC is expensive in terms of time, power and area. Major time consuming operations are reduced with projective coordinate. The proposed Elliptic Curve Cryptosystem over GF(2^{160}) is designed with various point multiplication schemes in the Galois Field. The operations in the elliptic curve are explored using the finite field arithmetic in generating the keys and digital signatures, based on Elliptic Curve Digital Signature Algorithm (ECDSA). The ECC protocol algorithm is parallelizable and well adapted to FPGA design of Elliptic Curve Cryptosystems.

Keywords: ECC, RSA, ECDSA, FPGA, Lopez-Dahab point multiplication, Montgomery multiplication.

Introduction

The use of elliptic curves in cryptography was invented independently by Neal Koblitz and Victor.S.Miller. Major discussions regarding the security and efficiency of the system were carried out followed by the introduction of Elliptic Curve Cryptosystem. Since ECC features a superior strength per bit compared to other public key cryptosystems, it offers the same level of security with small key size and saves bandwidth. These desirable features made ECC more attractive. The key distribution and digital signature schemes based on ECC is recognized by the IEEE standard. National Institute of Standards and Technology (NIST) of the U.S Government details a list of secured elliptic curves. In the hardware realization of Elliptic Curve Cryptosystem, considering the computation based operations, the Field Programmable Gate Array Technology can serve for the purpose better. Certain elements such as cost effectiveness, performance and re-configurability of the cryptographic algorithm, the hardware realization of ECC supports for the same [7]. Many cryptographic processors set the objective of reducing the latency due to the scalar multiplication with respect to the number of cycles required. Some ECC design methods adopted advanced processors in designing an Application Specific Instruction set Processor (ASIP). Major efforts were spent in optimizing the algorithm as well as betterment in arithmetic architectural designs. The proposed implementation of GF(2^{160}) is designed with suitable point multiplication schemes that reduces the latency due to computational complexity in key pair generation. The paper analyzed the protocol, ECDSA that yielded better confidentiality and integrity in data communication.

Elliptic Curve Over GF(2^m)

The Elliptic Curve Cryptography is carried out under the prime field GF(P) or the binary field GF(2^m). Both Galois fields can offer the same security levels. In this work, the operations in GF(2^m) is focused as it supports for the hardware implementation in a better way with mod-2 operations [3]. The elliptic curve defined under the Galois field, GF(2^m) satisfies the solutions of the equation,

$$y^2 + xy = x^3 + ax^2 + b \qquad \text{v (1)}$$

where a and b are elements of GF(2^m) and $b \neq 0$.

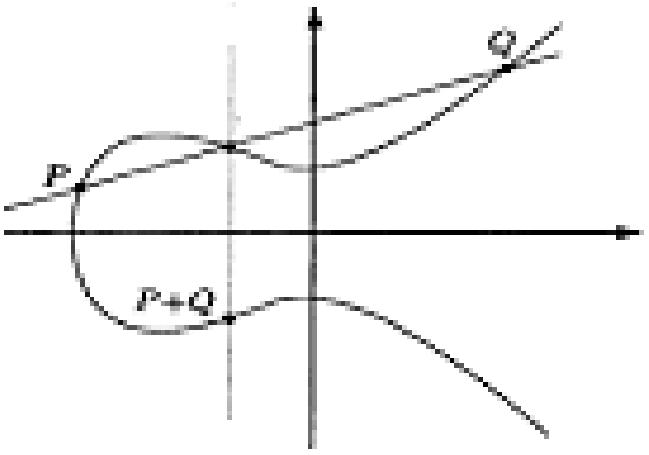

Figure 1: Elliptic Curve

Let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are points on the elliptic curve, then summing the points is given by $P_1 + P_2 = P_3$; where $P_3 = (x_3, y_3)$ is a point on the elliptic curve such that,

$$x_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)^2 + \frac{y_1 + y_2}{x_1 + x_2} + x_1 + x_2 + a \text{; if } P_1 \neq P_2.$$

$x_3 = x_1^2 + \frac{b}{x_1^2}$; if $P_1 = P_2$. Also,

$$y_3 = \left(\frac{y_1 + y_2}{x_1 + x_2}\right)(x_1 + x_3) + x_3 + y_1 \text{; if } P_1 \neq P_2.$$

$$y_3 = x_1^2 + \left(x_1 + \frac{y_1}{x_1}\right)(x_3) + x_3 \text{ if } P_1 = P_2.$$

Therefore, the point doubling is carried out for the condition, $P_1 = P_2$. The operation will be a point addition if $P_1 \neq P_2$. These operations have major role in developing the algorithm for ECC[1]. They constitute the scalar multiplication or the point multiplication, kP.

The computation of inversion operation is expensive when compared to the multiplication. To minimize the inversion operation, various methodologies based on projective coordinate were proposed that employed typical fractional methods in the finite field arithmetic. The proposed work used a generic algorithm that is based on projective coordinate method gives performance enhancement. The algorithm was proposed by Lopez-Dahab that resulted an efficient realization method for the point multiplication, kP. This method does not require any pre-calculations or additional field properties.

Lopez and Dahab Multiplication

The effective realization of point multiplication needs the optimization of the elliptic curve operation as well as the operations in the Galois field. The optimization over the curve arithmetic is focused here that employed projective coordinate methods. Both point addition and point multiplication operations requires the inversion operations. The usage of projective coordinates granted the elimination of inverse operation by means of a few additional field multiplications [4]. The efficiency can be measured as the ratio of, time for the completion of inversion to the time taken for the completion of multiplication.

Several methods of projective coordinates were proposed during the previous years. The popular among them are; Jacobian, Standard and Lopez-Dahab projective coordinate. The projective coordinate system by Lopez-Dahab is used as it offers high performance for both point doubling and addition.

The steps for the operation of point addition and doubling were based on typical formulae that use the x-coordinates of the points. When used with the projective coordinate from formulae, the x-coordinate is given by, $\frac{X_i}{Z_i}$, where *i* takes values 1, 2, 3. The computation for point addition and doubling is shown in equations (2) to (5). They are used by the algorithm for the point multiplication in the projective coordinate.

$$x(2P_i) = X_i^4 + bZ_i^4 \quad (2)$$

$$z(2P_i) = Z_i^2 X_i^2 \quad (3)$$

$$Z_3 = (X_1 Z_2 + X_2 Z_1)^2 \quad (4)$$

$$X_3 = xZ_3 + (X_1 Z_2)(X_2 Z_1) \quad (5)$$

Algorithm for Lopez-Dahab Multiplication

Input: An integer, $k \geq 0$; Base point, $P = (x, y) \in E$.

Output: $Q = kP$.

If *k* or $x = 0$; then output is (0, 0); stop.

Let $k \leftarrow (k_{l-1} k_{l-2} .. k_0$.

Let $X_1 \leftarrow x; Z_1 \leftarrow 1; X_2 \leftarrow x^4 + b; Z_2 \leftarrow x^2$.

From $l - 2$ down to 0; for *j*, do;

If $k_j = 1$;

$ADD(X_1, Z_1, X_2, Z_2), DBL(X_2, Z_2)$

Else,

$ADD(X_2, Z_2, X_1, Z_1), DBL(X_1, Z_1)$

Return, $(Q = \Pr oj - Aff(X_1, Z_1, X_2, Z_2))$.

Karatsuba Multiplication

The Two m-bit numbers, whose multiplication can be carried out with reduced bit complexity of less than $O(m^2)$. This was discovered by Ofman and Karatsuba in 1963. The algorithm is named as Karatsuba multiplication. For the multiplication in Galois field, the Karatsuba multiplication can be applied where, a polynomial A in GF (2^m) is divided into two.

For polynomials A, B ∈ GF (2^m); the n bit multiplication, A.B is sub-divided into n/2 bit multiplications. By defining some additional polynomials, that results in a total of three n/2 bit multiplications and some extra additions using XOR operations that performs one m-bit multiplication. The Karatsuba Multiplication for polynomials in Galois field is based on the approach of divide and conquer[2]. Here the operands are separated into two segments, which follow an attempt for generalizing the approach by sub-dividing the operands to yield greater two segments. The multiplication over finite field is computed using AND operation. The multiplication of two polynomials of degree-m can be carried out with three m/2-bit multiplications and a few XOR operations to find the interim results and then to accumulate the final result [5]. This leads to a recursive development process, which builds Combined Karatsuba Multipliers (CKMs) of width m =2^i for arbitrary i ∈ N.

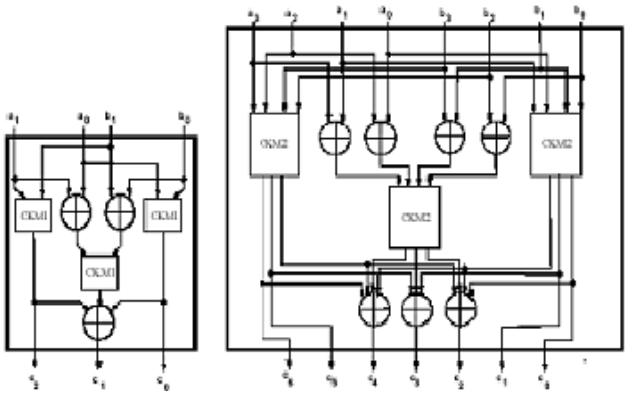

Figure 2: 2-bit & 4-bit Combined Karatsuba Multiplier

Montgomery Multiplication

The architecture proposed for Elliptic Curve Cryptosystem used the Montgomery scalar multiplication with projective coordinate for the realization of fast point scalar multiplication. The point operation with affine coordinate (x, y) includes finite field inversion [8]. This is more expensive compared to multiplication. The usage of projective coordinate (*X, Y, Z*) allows effective replacement of field inversion through a sequence of multiplications. The Montgomery scalar multiplication proposed here with projective coordinate algorithm needs only the computation with x and z coordinates for the point multiplication. The computation with y coordinate is needed for the coordinate conversion (M_{xy}) only which does the conversion from projective coordinate to affine coordinate. Moreover, the finite field inversion is only necessary in M_{xy}.

The Montgomery point multiplication can be realized efficiently through operation scheduling. Initially, the functions M_{add} & M_{double} can be disintegrated into Galois field multiplications, additions and squares. The operation scheduling of M_{add} and M_{double} can be made better by using a dedicated squarer and three AUs, applying the reduction technique. The AU1 & AU2 together yields M_{add}, meanwhile AU3 and the squarer find outs M_{double}. The reduction technique resulted in a high performance Cryptosystem architecture using the operation schedule for the Montgomery point multiplication algorithm as well as bit parallel Modular reduction [9]. The proposed Elliptic Curve Cryptosystem architecture that explored the Montgomery scalar multiplication with projective coordinate designed for fast point multiplication is discussed in the algorithm. The proposed design method outperforms other Elliptic Curve Cryptosystem designs significantly in terms of performance as well as cost-effectiveness.

Algorithm for Montgo Multiplication with Projective Coordinate

Input: A point $P = (x, y)$, an l-bit integer $k = (k_l-1, ..., k_1, k_0)$.
Output: Q = kP.

1: $X_1 = x$, $Z_1 = 1$, $X_2 = x^4 + \beta$, $Z_2 = x^2$.
2: for i = l−2 to 0 by −1 do
3: if $k_i = 1$ then
4: $(X_1, Z_1) = M_{add}(X_1, Z_1, X_2, Z_2)$, $(X_2, Z_2) = M_{double}(X_2, Z_2)$
5: else
6: $(X_2, Z_2) = M_{add}(X_1, Z_1, X_2, Z_2)$, $(X_1, Z_1) = M_{double}(X_1, Z_1)$
7: end if
8: end for
9: $Q = M_{xy}(X_1, Z_1, X_2, Z_2)$
10: $M_{add}(X_1, Z_1, X_2, Z_2)$ // Point Addition
11: $Z_3 = (X_1 \times Z_2 + X_2 \times Z_1)^2$, $X_3 = x \times Z_3 + (X_1 \times Z_2) \times (X_2 \times Z_1)$
12: return(X_3, Z_3)
13: $M_{double}(X_1, Z_1)$ // Point Double
14: $Z_2 = Z_1^2 \times X_1^2$, $X_2 = X_1^4 + b \times Z_1^4$
15: return(X_2, Z_2)
17: $M_{xy}(X_1, Z_1, X_2, Z_2)$ // Coordinate Conversion
18: $X = X_1/Z_1$, $Y = (x+X) \times (y+x^2 + (X_2/Z_2 + x) \times (X_1/Z_1 + x))/x + y$
19: return(X, Y).

The ECDSA for ECC

The proposed work carried out analysis with the elliptic curve operations of the ECC protocol ECDSA (Elliptic Curve Digital Signature Algorithm). The ECC Transmitter and Receiver section performed the generation and verification of digital signatures based on the ECDSA Algorithm. Both process used the Hash function of the message there by resulting in the message digest [6]. The transmitter sends the message which need not be encrypted, along with the signature to the receiver. The receiver also finds the Hash of the received message and uses the received sign and the sender's public key to verify the signature. The generation and verification of signatures were carried out based on the ECDSA algorithm, thereby it ensured better authentication in secured communication environment.

Conclusion & Future scope

The Elliptic Curve Cryptosystem formed a suitable substitution for the traditional public key cryptosystems in the application level due to its superior strength per bit. This work resulted in the realization of an efficient system through the effective multiplication schemes in the Galois field. The algorithm upholds the enhanced security level of the message with less key size and signature size. The algorithm yielded in improved authentication through verification of signatures at the receiving side. To advance the effectiveness of the system, the ECC operations, including point addition and multiplication is done using typical multiplication schemes. The proposed Elliptic Curve Cryptography processor with 160 bit point multiplication and coordinates Conversion outperforms other EC cryptosystem design in terms of high performance and cost effectiveness, thereby it confirmed the suitability of FPGA implementation of Elliptic Curve Cryptosystem. The improvements in the algorithmic level can be extended further when realized completely in macrocode.

References

[1] N. Koblitz, 1987, Springer-Verlag, "Elliptic Curve Crypto-systems, " *Mathematics of Computation*, vol. 48, pp.203–209.

[2] A. Karatsuba and Y. Ofman, 1963, "Multiplication of multidigit numbers on automata, " *Sov. Phys.-Dokl (Engl. transl.), vol. 7, no. 7*, pp. 595–596.

[3] V. Miller, 1986, "Uses of elliptic curves in cryptography, " in Advances in Cryptology CRYPTO '85, PP. 417{417-426) 6.

[4] J. Lopez and R. Dahab, "Fast multiplication on elliptic curves over GF(2m) without pre-computation", in Cryptographic Hardware and Embedded Systems (CHES), vol. 1717 of LNCS, pp. 316–327, Springer-Verlag, Aug. 1999.

[5] C.K. Koc and C.Y. Hung, July 1998, "Fast algorithm for modular reduction*"*, IEEE Proceedings - Computers and Digital Techniques, 145(4):265-271.

[6] Scheneier. B, 1996, (John wiley and Sons, Inc), "Applied cryptography, algorithms and source code in C, " 2nd ed., p316.

[7] D. Hankerson, A. Mednezes, S. Vanstone, 2004, New York: Springer, "Guide to elliptic curve cryptography, ". PP.1-66.

[8] Lutz. J, Hasan. A, "High Performance FPGA based Elliptic Curve Cryptographic Co-Processor", ITCC-2004, vol. 2, pp 486 – 492, Apr 2004.

[9] C. Paar, P. Fleischmann, and P. Roelse, February 1998, "Efficient multiplier architectures for Galois fields GF (24n)*"* IEEE Transactions on Computers, 47(2):162–170.