International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 11 (2015) pp. 29041-29049
© Research India Publications
http://www.ripublication.com

Karatsuba Multiplication In Galois Field For The Implementation of Elliptic Curve Cryptography

Ashkar Mohammed M[1], Dr. S Suresh Babu[2], M Navas chokli[3]
[1] *Research Scholar, NICHE, KK District, Tamilnadu.*
ashkarmohammed@yahoo.co.in
[2] *Principal, Sree Budha College of Engineering, Alappuzha, Kerala.*
drssbtkm@gmail.com
[3] *Instructor, Indian Naval Academy, Ezhimala, Kerala.*
mohammednavas@gmail.com

Abstract

From the initial stages of public key cryptography, mainly two types of cryptosystems were used to overthrow the attacks. Therefore these two cryptosystems called as RSA and El Gamal are mainly preferred and commonly used. They can be used for encryption and decryption as well as digital signatures. In 1985, Victor Miller and Neal Koblitz invented the Elliptic Curve Cryptography that received more attention widely and formed a suitable substitution for the traditional public key cryptosystems like RSA in the application level. For realizing the protocols such as Elliptic Curve Digital Signature Algorithm(ECDSA), Diffie-Hellman key Exchange, Elgamal Encryption and Decryption etc the Elliptic Curve Cryptography is preferred. An algorithm based on modular multiplication called as Karatsuba multiplication is used here for developing the elliptic curve cryptosystem in the Galois field. The finite field operations in the elliptic curve is used in implementing the Elliptic Curve Digital Signature Algorithm. The algorithm confirmed the suitability of the VLSI implementation of the Elliptic Curve Cryptography.

Keywords- RSA, ECC, ECDLP, ECDSA, SHA, Karatsuba multiplication.

Introduction

In the present situation, the developments in the field of information technology is progressing day by day. The mobile phones and the smart handhelds are very popular now a days and what we need is a secured communication with these devices. But when this aspect is considered, some limitations also taken into account; like requirement for band width, memory requirement, computations in power

requirement etc[1]. After the introduction of elliptic curve cryptography, major discussions were carried out regarding the efficiency and the level of security of the system.

The efficiency in the encryption and decryption scheme need not to be account for more as such operations are usually carried out with private key cryptosystem. A cryptosystem with minimum signature size and less number of keys is appreciably demanding. The elliptic curve cryptography holds these favorable features. These properties are gained from the fact that for the elliptic curve discrete logarithm problem, there are no sub exponential algorithms. This ensures that comparing with other cryptosystems, only less number of keys required with high level of security.

The finite field operations in the elliptic curve is carried out to implement the elliptic curve cryptography protocol, ECDSA. The major steps in the elliptic curve digital signature algorithm includes the generation of the key pair, generation of the signature and its verification. The karatsuba multiplication is carried out for the two points selected from the elliptic curve. The resulting point is called the generating point, G. A random integer, d is selected from the finite field in the interval [0, n-1]. The scalar multiplication is carried out between d and G. The resultant is Q, that forms the public key. The signing function of the signer and the hash function is utilized for creating the digital signature typically. The message is bonded with the two generated signatures r, s and send. In the message receiving end, the verification of the signature is carried out and the data is retrieved. The sender's public key is known by the receiver. Therefore the receiver can check the authenticity of the signature with his private key. Therefore the elliptic curve cryptography realized with elliptic curve digital signature algorithm confirms the authenticity and integrity of message communication.

Operations In The Galois Field, Gf (2^m)

The operations in the Galois field, GF(2^m) plays an important role in the implementation of hardware in cryptography. The field elements are used in the form of a basis. The normal basis or a polynomial basis is usually used in the implementation. For the implementation of hardware in an efficient way, the normal basis is found to be the good choice[2].

The field operation includes addition, squaring as well as multiplication. The addition in the Galois field operation is easily carried out through the XOR operations as bit wise. The operation with rotate left yields the squaring operation. For the finite field operations, different types of point multiplications such as comba multiplication, karatsuba multiplication etc are used. Here the point multiplication is carried out by karatsuba multiplication.

The Karatsuba Multiplication

The two n bit numbers, when subjected for multiplication, the operation can be accomplished with a lesser bit complexity of even less than $O(n^2)$. This can be achieved if the karatsuba multiplication is used for the algorithm. The karatsuba

multiplication is discovered by A. Karatsuba and Y. Ofman in 1963. The karatsuba point multiplication can be used for the finite field operations in cryptography.

The idea based on divide and conquer approach is used in the point multiplication of polynomials in the Galois field, GF(2^m). The operations are partitioned into two segments. This method can be generalized further by again dividing the operands to above two segments. The multiplication in the Galois field is carried out through the AND operation. Thus the implementation of point multiplication with karatsuba multiplication is carried out through the recursive procedure.

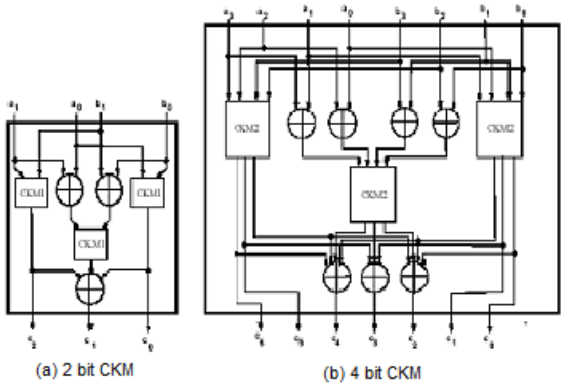

Figure 1: Karatsuba Multiplication

Eliptic Curve Over Galois Field Gf(2^m)

The elliptic curve used here is not a normal ellipse. But the equation used here resembles to that used in finding the circumference of a normal ellipse. The cubic equation used here is much a similar one and therefore called the elliptic curve. The group of points selected from the coordinate plane is characterized by the following equation.

$$y^2[+x.y] = x^3 + ax^2 + b \tag{1}$$

The constants used here are a and b and the variables are x and y. The variables and constants used need not be quantitatively real. They can be any value from the finite field. Therefore the elliptic curve used here represents a non-singular elliptic curve over the Galois field GF(2^m)[3]. This type of elliptic curve therefore forms an abelian group.

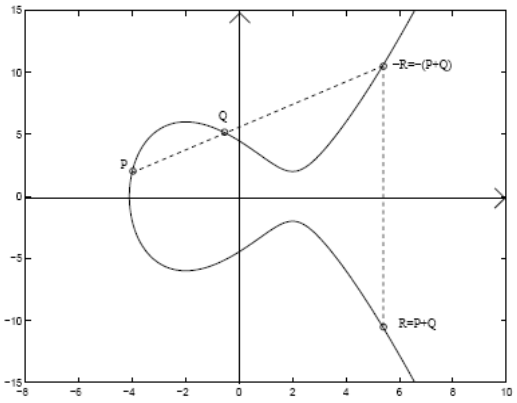

Figure 2: Elliptic Curve

The Curve Operations

The main curve operation includes curve addition and curve multiplication. The elliptic curve holds the property that when two selected points from the curve is subjected to addition then the third point that is the resultant point will be also a point on the curve[4]. The addition rule upholds the normal properties of addition.

The two points in the elliptic curve are $P=(x_1, y_1)$ and $Q=(x_2, y_2)$. The addition rule is as discussed below.

$$(x_1, y_1)+(x_2, y_2)=(x_3, y_3) \tag{2}$$

Such that, $x_3 = L^2 + L + x_1 + x_2 + a$ (3)

$$y_3 = L(x_1 + x_3) + x_3 + y \tag{4}$$

$$L = \frac{(y_1 + y_2)}{(x_1 + x_2)} \tag{5}$$

If $x_1 = x_2$ and $y_1 = y_2$ then we can use as,

$$x_3 = L^2 + L + a \tag{6}$$

$$y_3 = x_1^{\,2} + (L+1)x_3 \tag{7}$$

$$L = x_1 + \left(\frac{y_1}{x_1}\right) \tag{8}$$

Within the addition rules, special cases can be considered. If $y_2 = x_1 + y_1$ and $x_1 = x_2$ then the resultant will be zero. Also the result will be the other operand if either point becomes zero. If the points P and Q are identical then it is known as point doubling. Otherwise if P and Q are non-identical then it is known as point addition[5]. The multiplication in the elliptic curve is carried out through repeated addition.

$$Q = kP \tag{9}$$

$$= P + P + \ldots\ldots\ldots('k' times) \tag{10}$$

These operations can be carried out by point doubling and point addition.

The Discrete Logarithm Problem

To develop the cryptosystem with in the finite field group, the discrete logarithm problem can be applied. If *P* and *Q* are points in the elliptic curve such that the computation of $Q = kP$ is easier. '*k*' is a point with in the finite Galois field. Though the computation of $Q = kP$ is easier, it is not easy to find out '*k*' even if *P* and *Q* values are known. This is due to the reason that so far no sub exponential algorithms are available for its computation. In fact the multiplicative groups in the Galois field were suggested. But the complexity of the system depends on the selection of the group. Compared to the same issues in the multiplicative group of the Galois field earlier, the current issues in groups from elliptic curves are now tedious in several magnitudes[6]. This fact supports the elliptic curve cryptosystem in terms of its effective strength.

The Hash Algorithm

The key process involved in the generation of digital signatures is the Secured Hash Algorithm (SHA 1). Hashing is the process by which the character strings are converted into smaller length size or even the key that constitute the real message. The hash algorithm mainly includes the following process.

A. The Message Digest Formation

The process of message digest formation mainly includes the adding of message with the additional bits. These results in the message digest formation of 512 bits. The main steps involve in the message digest formation is to append padding bits and length. The message is padded such a way that the bit wise length will be congruent to 448 modulo 512. The padded message will have the length 64 bits less than integer multiple of 512 bits. Even when the message is of desired length, padding is carried out always[7]. Suppose the message length is 448 bits, then the 512 bits are added such that the length will be 960 bits. Therefore the number of padding is from 1 to 512 bits. Padding therefore includes 1 bit followed by 0 bits. Padding is followed by the process of append length. The actual message before padding which represents a 64 bit length is appended with the result of previous step such a way that the LSB first. Suppose the actual length is above 2^{64}, in that case lower order bits alone are used. Therefore the field will have the length of actual message modulo 2^{64}. The resultant from the above discussed steps gives a message which is an integer multiple of 512 bits.

B. Buffer Initialization

A buffer of length 160 bits is used to keep the sub stage values and final values of the SHA operation. The buffer is represented with five registers each of 32 bits. These five registers are initialized with thirty two bit integers whose hexadecimal values are shown below.

Table 1: Buffer Values

WORD E	C3 D2 E1 F0
WORD D	10 32 54 76
WORD C	98 BA DC EF
WORD B	EF CD AB 89
WORD A	67 45 23 01

C. The Complete Hash Process

The secured hash algorithm consists of 4 stages of operation. Each stages includes twenty typical steps. The mechanism is demonstrated as in the figure shown below.

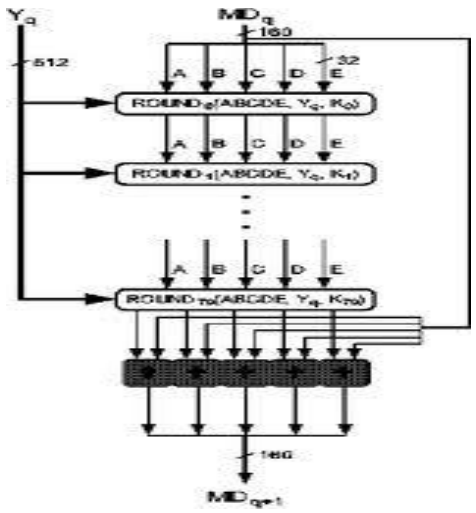

Figure 3: Hash Process

Each stage passes as the present input 512 bit message digest will be processed and the 160 bit value

As the buffer input and the contents of the buffer are updated. Out of the total eighty steps of operation, each stage takes an additive constant, K_t where 't' takes the values from zero to seventy nine. During the stages of operation, only 4 values of K_t are applied.

The elementary operation of hash process assumes twenty steps that is subjected to 4 stages. Therefore the complete operation in SHA 1algorithm includes 80 steps. W_t is

a word that is derived from the present 512 bit. It is of 32 bit length. K_t is an additive type constant that is used.

No matter how much is the message length, finally the output will be of 160 bits. So the hash algorithm thereby carries out a message digest of 512 bits.

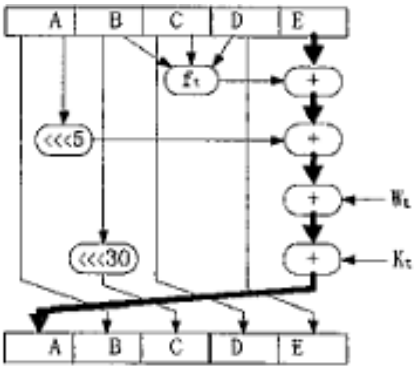

Figure 4: The Single Step Hash Operation

Table 2: Initialization of Additive Constant

Values of the additive constant	Steps
CA62C1D6	60 to 79
8F1BBCDC	40 to 59
6ED9EBA1	20 to 39
5A82799	0 to 19

Table 3: Logical Operations In Hash Algorithm

Ft over (A,B,C,D)	Steps
B XOR C XOR D	60 to 79
(B □ C) □ (B □ (B□ D)) □ (C□ D)	40 to 59
B XOR C XOR D	20 to 39
(B □ C) □ (B` □ D)	0 to 19

Elliptic Curve Digital Signature Algorithm

The Elliptic Curve Digital Signature Algorithm was presented by Vanstone in 1992. This was in reply to National Institute of Standards and Technology's feedback about the initial suggestions on their Digital Signature Schemes[8]. DSS is equivalent to hand written signature. The digital signature is basically a number. This number depends on the secret key(that is known only by the signer) and the data content. The digital signatures should be able to verify even though no access of private key of the signer. The main procedures carried out in Elliptic Curve Digital Signature Algorithm are as follows.

A. Generation of Keys

Assume Hansel needs to pass a digitally signed data to Gretel,

1. Choose a random integer 'd' in the interval [0,n-1]
2. Calculate Q = dG, G is the Generating point on the elliptic curve obtained by Karatsuba Multiplication of two other points on the elliptic curve.
3. The Hansel's key-pair is (d,Q). Such that d is the Private key and Q is the public key.

B. Generation of Signature

1. Select a random number k such that, $1 \leq k \leq n - 1$.
2. Find kG = (x1, y1) and r = x1 mod n. If r = 0 then go to step 1.
3. Calculate k^{-1} mod n.
4. Calculate e = SHA^{-1}(M).
5. Calculate s = k^{-1}(e + dr) mod n. If s = 0 then go to step 1.
6. Hansel's signature for the message M is r, s.

C. Verification of Signature

To validate the received parameters, Gretel carries out the following.

1. Verifiy that r, s are integers in the interval [1, n - 1].
2. Find e = SHA^{-1}(M).
3. Calculate w = s^{-1} mod n.
4. Calculate u1 = ew mod n also u2 = rw mod n.
5. Calculate X = u1G + u2Q. If X = 0, then reject the signature. Otherwise calculate v = x1 mod n such that X = (x1, y1).
6. The signature is accepted only if v = r.

$s = k^{-1}(e + dr)$ (n).

from this datas we have,

$k = s^{-1}(e + dr) = s^{-1}e + s^{-1}rd$ =we + wrd = u1 + u2d (n). Thus u1G + u2Q = (u1 + u2d)G = kG and so v = r as required.

Both the generation and verification process uses the SHA function of the message. These results in the message digest. Hansel passes the data with or without encryption, together with the sign to Gretel. The Gretel also finds the Hash of the received message and uses the received signatures and Hansel's public key to verify the signature. The generation and verification procedures are thereby based on the Elliptic Curve Digital Signature Algorithm. The receiver will be knowing the sender's public key. Therefore the receiver can authenticate the signature using own private key. Therefore the Elliptic Curve Cryptography thereby confirms the secured message communication.

Conclusion & Future Scope

Even with less key size and signature size, the elliptic curve cryptography provides the same security as that of RSA. In this work an efficient multiplication method is used that supports the enhanced security level of message with less key size and signature size. The algorithm used here also confirms the better authentication as the

verification of the signature are carried out at the receiving end. The algorithm thereby confirmed the suitability of the VLSI implementation of the Elliptic Curve Cryptography. A generic projective coordinate algorithm can be used that that offers an improved performance level as this doesn't needs any precomputation or special finite field properties. These can provide the better performance for point doubling and point addition.

References

[1] Martin Feldhofer, Thomas Trathnigg, and Bernd Schnitzer, IEEE 2002, "A Self-Timed Arithmetic Unit for Elliptic Curve Cryptography" Proceedings of the Euromicro Symposium on Digital System Design (DSD'02).PP.476-480.

[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] 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.

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

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

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

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

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