International Journal of Advance Research, IJOAR .org
Volume 1, Issue 7, July 2013, Online: ISSN 2320-9194

# IMPLEMENTATION OF ELLIPTIC CURVE POINT MULTIPLICATION ALGORITHM USING DSP PROCESSOR

1Prof. Renuka H. Korti, 2Dr. Vijaya C.

*1Assistant Professor, Dept. of ECE, SDMCET Dharwad, Karnataka, India*
*2Dean, Academic program, SDMCET Dharwad, Karnataka, India*
*1rhh_korti@yahoo.com, 2vijayac26@yahoo.com*

## ABSTRACT:

Network security is becoming more and more crucial as the volume of data being exchanged on the internet increases. Elliptic curve cryptography(ECC) offers high security for networking and communication. ECC is an approach to public key cryptography based on the algebraic structure of elliptic curves over finite fields. The ECC is advantageous due to the provision of high level of security and the usage of small keys. This paper presents an elliptic curve point multiplication algorithm over GF(P) and implemented using TMS320VC5416 DSP starter kit.

## Keywords:

Elliptic curve cryptography, Finite field, Public key cryptography, Point addition, Point Doubling, Point multiplication.

## 1. Introduction:

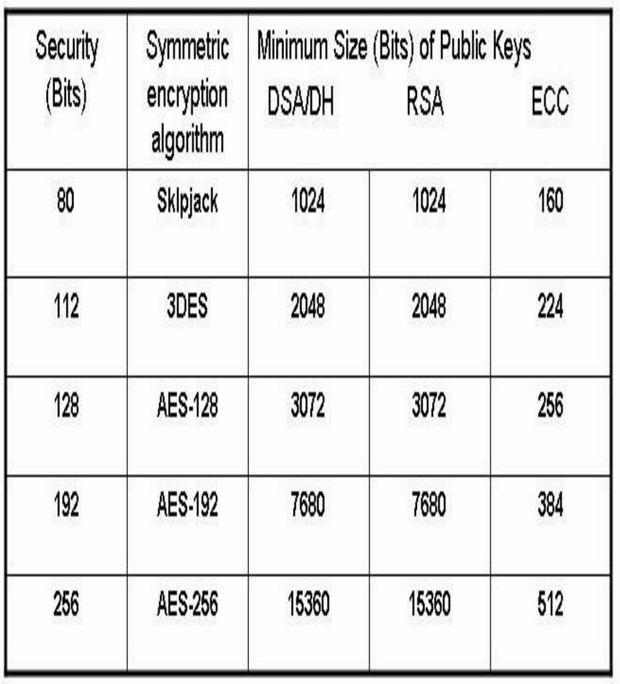
Cryptography, the art and science of hiding data, plays a central role in achieving information security. It has become a fundamental part of communication and commercial applications in the Internet as well as in many other digital applications. Cryptography is deployed with cryptographic algorithms, mathematical functions for hiding messages and retrieving hidden messages. Cryptography is widely applied in everyday life. Governments and militaries also rely heavily on cryptography to keep certain information classified, like attack orders or obtained intelligence. Encryption schemes can be classified in two distinct categories: symmetric and asymmetric. We will call an encryption schemes symmetric, if the decryption key is identical to the encryption key, or can be easily determined from it. Otherwise, we call it asymmetric. Asymmetric encryption schemes give rise to public key cryptosystems, since it causes no problems if the encryption key is published, contrary to symmetric schemes. Public-key cryptography has several advantages compared with non public-key cryptography. In particular, it eliminates the problem of the key distribution and provides some important function such as digital signature. To ensure a high degree of security, most proposed public key cryptographic algorithms require large keys and intensive computations. In recent years, elliptic curves over finite fields have been focused in public key cryptography [1][2]

## 2. Elliptic curve cryptography:

Elliptic curves were first proposed as a basis for public key cryptography in the mid 1980s independently by Koblitz and Miller. Elliptic curve cryptography (ECC) algorithm is practical than existing security algorithms[3][4]. Because of this fact, it showed real attraction to portable devices (handheld devices) manufacturers and the security of their systems. In fact, through these devices, any one can access either email, or do bank transaction or buy an thing oninternet using credit cards with high security standards. Elliptic curve algorithm is promising to be the best choice of these handhelds or similar devices because of low computing power (low battery consumption) and fast execution. ECC further gives very high security as compared to similar crypto systems with less size of key. For example, 160 bit ECC system is believed to provide same level of security as 1024 bit RSA [3][4].

Also, the rate at which ECC key sizes increase in order to obtain increased security is much slower than the rate at which integer based discrete logarithm (DL) or RSA key sizes increase for the same level increase in security [5].

Table 1. Nist Guidelines for Public-Key Sizes with Equivalent Security Levels



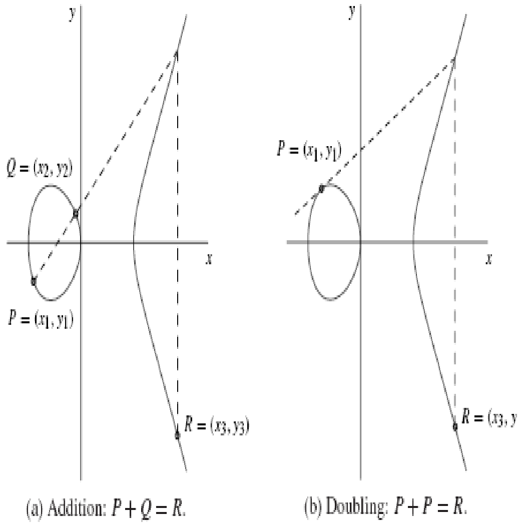
| Security (Bits) | Symmetric encryption algorithm | Minimum Size (Bits) of Public Keys DSA/DH | RSA | ECC |
|---|---|---|---|---|
| 80 | Sklpjack | 1024 | 1024 | 160 |
| 112 | 3DES | 2048 | 2048 | 224 |
| 128 | AES-128 | 3072 | 3072 | 256 |
| 192 | AES-192 | 7680 | 7680 | 384 |
| 256 | AES-256 | 15360 | 15360 | 512 |

ECC is included in the following major standards: ANSI X9.62 and X9.63, IEEE 1363, FIPS 186-2 (NIST), ISO/IEC 15946-1/2/3/4 and 18033-2 , and SECG. The DSA standard also includes an elliptic curve variant of DSA called Elliptic Curve Digital Signature Algorithm(ECDSA).

Elliptic curves provide a public key crypto-system based on the difficulty of the elliptic curve discrete logarithm problem, which is so called because of its similarity to the discrete logarithm problem (DLP) over the integers modulo a prime p [6]. This similarity means that most cryptographic procedures carried out using a cryptosystem based on the DLP over the integers modulo p can also be carried out in an elliptic curve cryptosystem. ECCs can also provide a faster implementation than RSA or DL systems, and use less bandwidth and power [7]. These issues are crucial in lightweight applications, i.e. smart cards [8]. An elliptic curve over a Galois field with p elements, GF(p), where p is prime and p > 3 may bedefined as the points (x,y) satisfying the curve equation *E:* $y^2=x^3+ax+b \ (mod\ p)$ , where a and b are constants satisfying $4a3+27b^2\neq 0 \ (mod\ p)$. In addition to the points satisfying the curve equation E, a point at infinity (f) is also defined. With a suitable definition of addition and doubling of points, this enables the points of an elliptic curve to form a group with addition and doubling of points being the group operation, and the point at infinity being the identity element. We then further define scalar point multiplication of a point P by a scalar k as being the result of adding the point P to itself k times (i.e. kP = P + P + P + · · · + P (k- times)). The elliptic curve discrete logarithm problem is then defined as to compute scalar k such that Q = kP; given the prime modulus p, the curve constants a and b, and two points P and Q. This problem is infeasible for secure elliptic curves, and thus point multiplication is the basic cryptographic operation of an elliptic curve. Point multiplication involves mainly three modular operations: addition, multiplication and inversion, where the modular addition operation is the simplest and least to be worried about [9].

## 3. Operations required by ECC:

The Point multiplication, or repeated addition, of EC points is the main operation required by ECC schemes, although other operations such as division may also be needed. For exact long integer wordlength mathematical arithmetic operations implied by such encryption/decryption systems are slow and possibly unique to this application area, since all rounding schemes are automatically excluded. The addition of two EC points and doubling of a point is illustrated in figure.1 below.



(a) Addition: $P+Q=R$.

(b) Doubling: $P+P=R$.

Figure 1: Illustration of point addition and point doubling.

As it is the point multiplication operation that dominates the actual execution timing of ECC schemes, its efficient implementation is crucial. The actual mathematics depends on the chosen curve and underlying field, however there is a clear hierarchy of underlying mathematical operations, as shown in figure. 2 below.

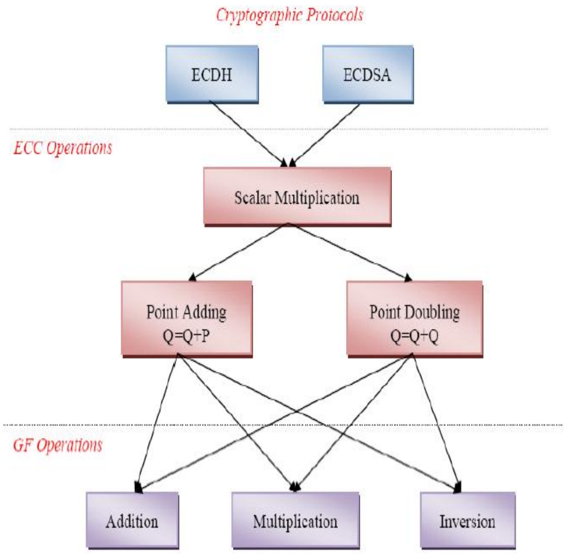


Figure 2: Hierarchy of ECC mathematical operations

## 4. Elliptic Curve on Prime field:

The equation of the elliptic curve on a prime field Fp is $\mathbf{y^2 mod\ p= x^3 + ax + b\ mod\ p}$, where $\mathbf{4a^3 + 27b^2 mod\ p \neq 0}$. Here the elements of the finite field are integers between 0 and p – 1. All the operations such as addition, substation, division, multiplication involves integers between 0 and p – 1. The prime number p is chosen such that there is finitely large number of points on the elliptic curve to make the cryptosystem secure.

Point addition operation is computed as shown below:

$(x1 , y1) + (x2 , y2) = (x3 , y3)$ ; where $x1 \neq x2$

$\lambda = (y2 - y1)/(x2 - x1)$

$x3 = \lambda 2 - x1 - x2$

$y3 = \lambda(x1 - x3) - y1$

However, the addition of a point to itself (doubling a point) on the elliptic curve is computed as show below:

*(x1 , y1) + (x1 , y1) = (x3 , y3)*; where *x1 ≠ 0*
*λ = (3(x1)2 + a) /(2y1)*
*x3 = λ2 – 2x1*
*y3 = λ(x1 – x3) – y1*

In both point addition and point doubling, we need an inversion step to calculate λ. The inversion is the most expensive operation.

Let us consider the elliptic curve over Fp where a = 1, b = 6, p = 11 with the equation $y^2 \equiv x^3 + x + 6$ (mod 11). The set of solutions are E = {(2,4),(2,7), (3,5), (3,6), (5,2), (5,9), (7,2), (7,9), (8,3), (8,8),(10,2), (10,9), O}, including the point infinity O.

Choose P = (2, 4) and Q = (10, 9) and theelliptic curve point addition is performed as follows.

λ= (9-4)/(10-2) mod 11 = 2
P+Q = (2,4) + (10,9)
x3 = $2^2$– 2- 10 = -8 = 3
y3 = 2 ( 2-3) - 4 = -2- 4 = -6 = 5
P+Q = (2,4) + (10,9) = (3, 5)

Select the point P = (8,8) and the doubling operation is done as follows

λ__= (3 * $8^2$ +1)/(2 * 8) mod 11 = (6 /5) mod 11= (50 / 5) mod 11 = 10
x3 = $10^2$- 2 * 8 = 84 mod 11= 7
y3 = 10(8-7) -8 = 10-. 8 = 2
2P = (8,8)+ (8,8) = (7,2)

Note that the result of addition and doubling is (3,5) and (7,2)

## 5. Point multiplication algorithm:

The algorithm used for point multiplication is based on the binary method [10], since it is efficient for hardware implementation. The binary method algorithm is shown below:

```
Algorithm : Binary method
Input: Binary representation of k and point P
k = (kn-1….k1k0)2
Output: kP
1. R ← P
2. For i = n-2 to 0 do
        2.1 R ←2R (Point Doubling)
        2.2 If ki =1 then
                R = R + P (Point Addition)
        2.3 i ←i −1
3. Return R
```

The cost of multiplication depends on the length of the binary representation of k and the number of 1s in this representation. If the representation $(kn\text{-}1….k1k0)_2$ has kn-1 # 0 then the number of doubling operation is (n - 1) and the number of addition operations is one less than the number of non-zero digits in $(kn1….k1k0)_2$. The number of non-zero digits is called the Hamming weight of scalar representation. In an average, binary method requires n -1 doublings and n -1/ 2 additions. For example, the integer k = 729 and the binary representation is $(1011011001)_2$, computation of 729P requires 9 doublings and additions. Whenever the bit is 1, two elliptic curve arithmetic operations such as Point double and Point addition will be made and if it is 0, only one operation, Point double is required. So if we reduce the number of 1s in the scalar representation or hamming weight, we could speed up the above computation.

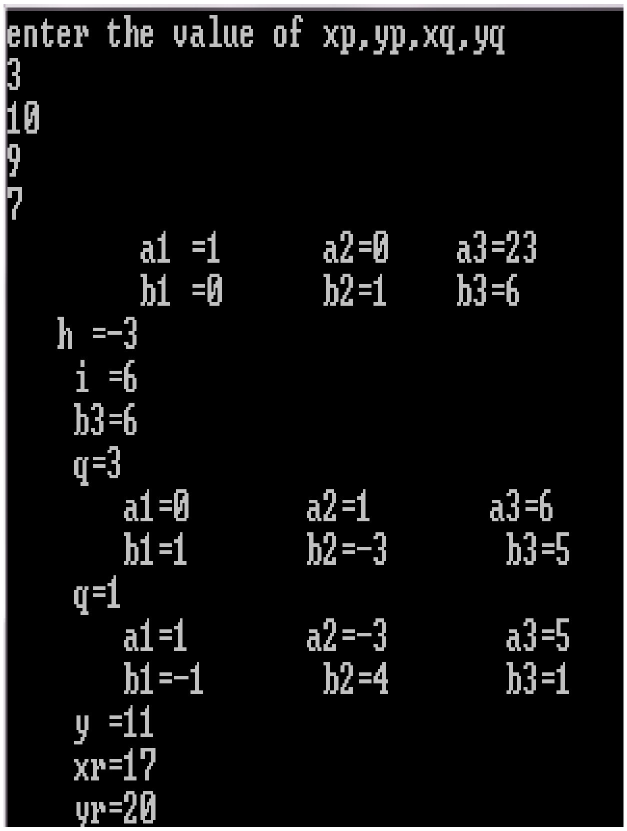
## 6. Experimental Results:

Points on the elliptic curve $E_{23}(1,1)$

| | | | | |
|---|---|---|---|---|
| **(0 , 1**) | ( 4, 0) | (7 ,12) | (12, 19) | (18, 20) |
| (0, 22) | ( 5, 4) | (9 , 7) | (13, 7) | (19, 5) |
| (1 , 7) | **(5, 19)** | **(9, 16)** | **(13, 16)** | **(19, 18)** |
| **(1 ,16)** | **( 6, 4)** | **(11, 3)** | **(17, 3)** | |
| **(3, 10)** | **(6, 19)** | **(11, 20)** | **(17, 20)** | |
| **(3 ,13)** | **( 7,11)** | **(12, 4)** | **(18, 3)** | |

Results for Point addition

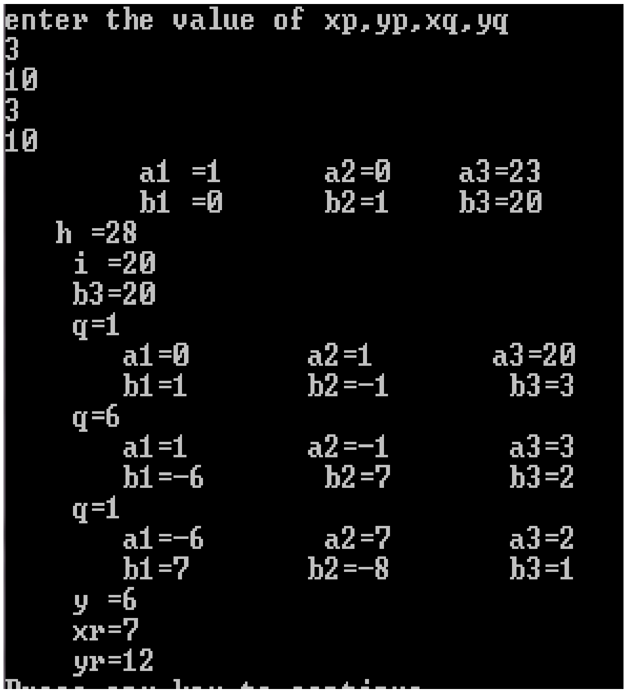
For $E_{23}(1, 1)$ POINT ADDING (R = P + Q); P=(3,10), Q=(9,7)
In this case P≠Q; R=P+Q = (17,20)



```
enter the value of xp,yp,xq,yq
3
10
9
7
        a1 =1       a2=0     a3=23
        b1 =0       b2=1     b3=6
   h =-3
    i =6
    b3=6
    q=3
       a1=0        a2=1        a3=6
       b1=1        b2=-3        b3=5
    q=1
       a1=1        a2=-3        a3=5
       b1=-1        b2=4        b3=1
    y =11
    xr=17
    yr=20
```

Results for Point doubling

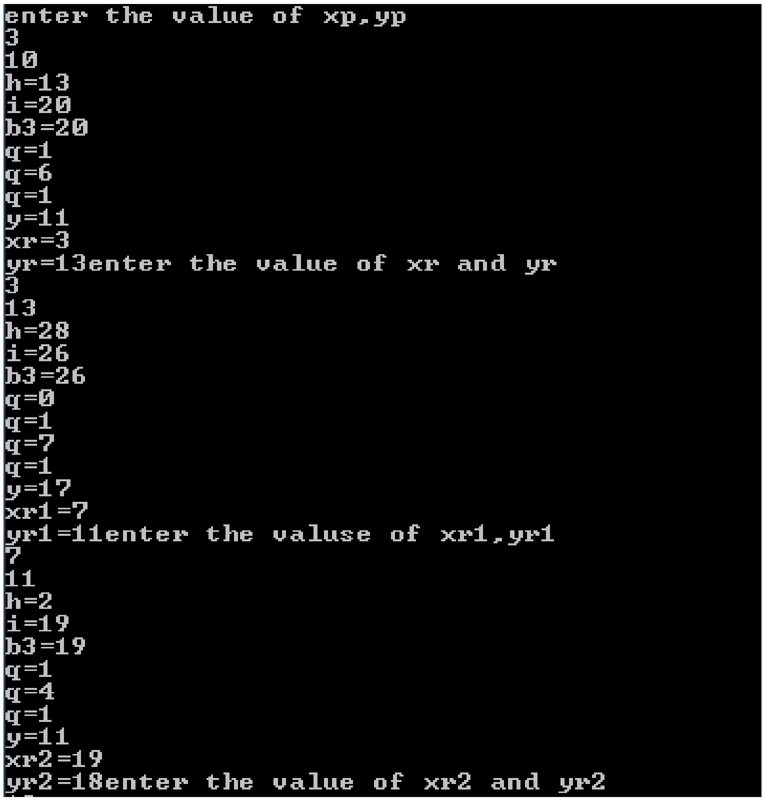
IF P=Q then (POINT DOUBLING) i.e.. R = 2P = (7,12)



```
enter the value of xp,yp,xq,yq
3
10
3
10
        a1 =1       a2=0     a3=23
        b1 =0       b2=1     b3=20
   h =28
    i =20
    b3=20
    q=1
       a1=0        a2=1        a3=20
       b1=1        b2=-1        b3=3
    q=6
       a1=1        a2=-1        a3=3
       b1=-6        b2=7        b3=2
    q=1
       a1=-6        a2=7        a3=2
       b1=7        b2=-8        b3=1
    y =6
    xr=7
    yr=12
Press any key to continue
```

Results for point multiplication

Point Q=7P =(19,18)



```
enter the value of xp,yp
3
10
h=13
i=20
b3=20
q=1
q=6
q=1
y=11
xr=3
yr=13enter the value of xr and yr
3
13
h=28
i=26
b3=26
q=0
q=1
q=7
q=1
y=17
xr1=7
yr1=11enter the valuse of xr1,yr1
7
11
h=2
i=19
b3=19
q=1
q=4
q=1
y=11
xr2=19
yr2=18enter the value of xr2 and yr2
```

## 7. Conclusion:

Elliptic Curve Cryptography offers a promising approach in the areas of public-key or dedicated areas of cryptography, due to the hardness, or lack of sub-exponential time attack on the discrete-log problem in elliptic fields. Elliptic curves are based on very sound mathematical foundations, of centuries, and distribute the potential cipher text values fairly randomly to make any guess or attack difficult. The elliptic curve discrete logarithm problem makes ECC most efficient with smaller key size compared to earlier RSA algorithm. The much shorter key sizes make them suitable for lightweight computing, bandwidth, power devices as mobiles, laptops, mobile web browsers etc. There is a lot of potential of ECC in general cryptography, as well as the area of light weight cryptography. In this paper, we propose an algorithm for point multiplication on ECC. It make the Elliptic Curve cryptosystem very suitable for low-cost implementations and also feasible in the restricted computing environments.

## References:

[1] N. Kobliz, \Elliptic Curve Cryptosystems," Math. Comp, Vol. 48, pp. 203-209, 1987.

[2] V.S. Miller, \Use of Elliptic Curves in Cryptography," Advances in Cryptology-CRYPTO 85, Lecture Notes in Computer Science, No. 218, Springer-Verlag, Berlin, pp. 417-426, 1986.

[3] J.H. Cheon, H.J. Kim, S.G. Hahn, "Elliptic curve discrete logarithm and integer factorization", The Math Net Korea, Information Center for Mathematical Sciences (ICMS), February 7, 1999, http://mathnet.kaist.ac.kr/

[4] A Certicom Whitepaper, "The Elliptic Curve Cryptosystem", July 2000, http://www.certicom.com/

[5] Hitchcock, Yvonne Roslyn, "Elliptic Curve Cryptography for Lightweight Applications", *Institution Queensland University of Technology*, 2003. http://adt.library.qut.edu.au/adtqut/ public/adt-QUT20040723.150510/

[6] Naoya Torii and Kazuhiro Yokoyama, "Elliptic Curve Cryptosystem", *FUJITSU Sci. Tech. Journal*, Vol. 36, No. 2, pages 140-146, December 2000. www.fujitsu.com/downloads/MAG/vol36-2/paper05.pdf

[7] O. Al-Khaleel, C. Papachristou, F. Wolff, K. Pekmestzi, "An Elliptic Curve Cryptosystem Design Based on FPGA Pipeline Folding", *13th IEEE International On-Line Testing Symposium, IOLTS 07*, pages 71 – 78, 8-11 July 2007.

[8] A.J. Menezes, T. Okamoto, S.A. Vanstone, S, "Reducing elliptic curve logarithms to logarithms in a finite field", *IEEE Transactions on Information Theory*, Volume 39, Issue 5, pages 1639 – 1646, Sept. 1993.

[9] T. Hasegawa, J. Nakajima, M. Matsui, "A practical implementation of elliptic curve cryptosystems over GF(p) on a 16-bit microcomputer", *In Public Key Cryptography – PKC '98, Proceedings*, volume 1431 of Lecture Notes in Computer Science, pages 182–194, Springer-Verlag, 1998.

[10] Scott Vanstone, "Crypto Column: The Importance of Good Crypto and Security Standards", Code & Cipher- Certicom's Bulletin of Security and Cryptography, Volume 1, Issue 4, 2004, http://www.certicom.com/codeandcipher

[11] Adnan Abdul-Aziz Gutub 424 International Journal of Computer Science and Security (IJCSS), Volume (4) : Issue(4) Multiplication and RSA Modular Exponentiation on Reconfigurable Logic", *Proceedings of the ACM/SIGDA tenth international symposium on Field-programmable gate arrays*, pages: 40 - 49, Monterey, California, USA, 2002

[12] J. Lopez, R. Dahab (2000), "An overview of elliptic curve cryptography", Technical report, IC-00-10, May 22. Available at http://www.dcc.unicamp.br/ic-main/public-cation -e.html.