International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 04 Issue: 10 | Oct -2020 ISSN: 2582-3930

Efficient Implementation of Elliptic Curve Cryptography

N. Senthilvel
Department of ECE (PG)
Sona College of Technology
Salem, INDIA.

Abstract: **Elliptic curve System as applied to cryptography was first proposed in 1985 independently by Neal koblitz and vector miller. The discrete logarithm problem on Elliptic curve group is believed more difficult than Corresponding problem in the multiplicative group of Nonzero element of underlying finite field .for elliptic Curve cryptosystem does exist many algorithm that Compute the scalar multiplication kp .some better For a software solution and other better for a Hardware solution. We compare algorithm without Pre-computation for scalar multiplication on elliptic Curve over the finite field of GF (2m).**

Keyword- Ecc, Rsa, Cryptography, fpga

I INTRODUCTION

The idea of information security lead to the evolution of Cryptography. In other words, Cryptography is the science of keeping information secure. It involves encryption and decryption of messages. Encryption is the process of converting a plain text into cipher text and decryption is the process of getting back the original message from the encrypted text. Cryptography, in addition to providing confidentiality, also provides Authentication, Integrity and Non-repudiation.

There have been many known cryptographic algorithms. The crux of any cryptographic algorithm is the “seed” or the “key” used for encrypting/decrypting the information. Many of the cryptographic algorithms are available publicly, though some organizations believe in having the algorithm a secret. The general method is in using a publicly known algorithm while maintaining the key a secret.

Based on the key, cryptosystems can be classified into two categories: S*ymmetric* and A*symmetric*. In Symmetric Key Cryptosystems, we use the same key for both Encryption as well as the corresponding decryption. i.e. if K was the key and M was the message, then, we have $D_K(E_K(M)) = M$
Asymmetric or Public key or shared key cryptosystems use two different keys. One is used for encryption while the other key is used for decryption. The two keys can be used interchangeably. One of the keys is made public (shared) while the other key is kept a secret. i.e. let k1 and k2 be public and private keys respectively. Let M be the message, then $D_{k2}(E_{k1}(M)) = D_{k1}(E_{k2}(M)) = M$

In general, symmetric key cryptosystems are preferred over public key systems due to the following factors: Ease for computation. Smaller key length providing the same amount of security as compared to a larger key in Public key systems. Smaller key length providing the same amount of security as compared to a larger key in Public key systems. Hence the common method adopted is to use a public key system to securely transmit a “secret key”. Once we have securely exchanged the Key, we then use this key for encryption and decryption using a Symmetric Key algorithm.

The idea of using Elliptic curves in cryptography was introduced by Victor Miller and Neal Koblitz as an alternative to established public-key systems. This project has been a study project, where we have studied and learnt the various concepts of elliptic curves. All the 3 team members have been actively involved in the full length of this project and the contribution from all of us is equal. Since this project involved a lot of study, discussions and analysis we cannot quantify the percentage of work done by each member as each one was equally involved in the study of various individual aspects and the entire learning involved discussions among us where each of us explained our learning to the other.

Types of operations used for transforming plaintext to cipher text: Most encryption algorithms are based on 2 general principles,

1. ***Substitution***, in which each element in plain text is mapped to some other element to form the cipher text
2. ***Transposition,*** in which elements in plaintext are rearranged to form cipher text.
3. **Number of keys used**: If both the sender and the receiver use a same key then such a system is referred

to as Symmetric, single-key, secret-key or conventional encryption. If the sender and receiver use different keys, then such a system is called Asymmetric, Two-key, or public-key encryption.

4. **Processing of Plain text**: A Block cipher processes the input one block at a time, producing an output block for each input block. A Stream cipher processes the input elements continuously producing output elements on the fly.

Most of the cryptographic algorithms are either symmetric or asymmetric key algorithms.

Secret Key Cryptography**:** This type of cryptosystem uses the same key for both encryption and decryption. Some of the advantages of such a system are

- Very fast relative to public key cryptography
- Considered secure, as long as the key is strong such as DSA and RSA.

The Elliptical curve Discrete Log Problem (ECDLP) makes it difficult to break an ECC as compared to RSA and DSA where the problems of factorization or the discrete log problem can be solved in sub-exponential time.

II ALGORITHMS FOR ELLIPTIC

i. Scalar Multiplications

In all the protocols that were discussed (ECDH, ECDSA, ECAES), the most time consuming part of the computations are scalar multiplications. That is the calculations of the form

$$Q= k P = P + P + P\ldots \text{ k times}$$

Here P is a curve point, k is an integer in the range of order of P (i.e. n). P is a fixed point that generates a large, prime subgroup of $E(F_q)$, or P is an arbitrary point in such a subgroup. Elliptic curves have some properties that allow optimization of scalar multiplications. The following sections describe some efficient algorithms for computing kP.

ii. Non adjacent form

This is a much efficient method used in the computation of kP. Here, the integer k is represented as $k = \sum_{j=0}^{l-1} k_j 2^j$, where each kj ∊ {–1, 0, 1}. The weight of NAF representation of a number of length l is l/3. Given below is an algorithm for finding NAF of a number

iii.Complexity Analysis of Elliptic Scalar Multiplications Algorithm Binary Method:

The simplest formula for calculating kP is based on the binary representation of k, i.e.,

$k = \sum_{j=0}^{l-1} k_j 2^j$, where $k_j \in \{1,0\}$, the value kP can be computed by kP given as

$$\sum_{j=0}^{l-1} k_j 2^j .P = 2(\ldots 2(2k_{l-1.} P + k_{l-2} P) + \ldots) + k_{0.} P$$

This method requires l doublings and w_k-1 additions, where w_k (the weight) is the number of 1s in the binary representation of k.

For k = 7 = $(111)_2$, the value of kP would be

$$kP = \sum_{j=0}^{l-1} k_j 2^j P = 2(2.P + P) + 1P$$

iv.Addition-Subtraction method:

Here the number k is represented in NAF form. The algorithm performs addition or subtraction depending on the sign of each digit, scanned from left to right. The algorithm performs l doublings and l/3 additions on an average. For k = 7, the binary method would require 3 doublings and 3 additions. In case of Addition-Subtraction method (the value of 7 in NAF form is 1 0 0 –1), it would require 4 doublings and 2 additions.

v.Repeated doubling method:

A point on the elliptic curve over F_{2m} is represented inn the form of (x, λ) rather than in the form of (x, y) when using the repeated doubling method for scalar multiplication. Every point P = (x, y) ∊ $E(F_{2m})$, where $x \neq 0$, P can be represented as the pair (x, λ), where $\lambda = x + y/x$.

III RSA

RSA is a public-key cryptosystem that gets its name from its inventors – Rivest, Shamir and Adleman and was developed in 1977. It has since withstood years of extensive cryptanalysis. It is used for electronic commerce and many other secure communications over the Internet. RSA is a Block cipher in which the plain text and cipher text are integers between 0 and n – 1 for some integer n. RSA gets its security from the difficulty of factoring large numbers

i.Working of RSA:

Select 2 random large prime numbers p and q of almost equal length. Compute their product $n = pq$. The Euler's Totient function $\phi(n)$ is computed, i.e. $\phi(n) = (p - 1)(q - 1)$. We then choose two keys a and b such that, $a.b \equiv 1 \pmod{\phi(n)}$. One of the keys say a is made public while the other key b is kept a secret. At this point, we no more require p, q and $\phi(n)$. We can discard these values. If we have a message M, encryption of M is $C = Ma \bmod n$, C is the resultant cipher text. Decryption of C is achieved by $M' = Cb \bmod n$. Consider $M' = Mab \bmod n = Mk\phi(n) + 1 \bmod n$(Since $a.b \equiv 1 \pmod{\phi(n)}$) $\Rightarrow M' = M . Mk\phi(n) \bmod n = M \bmod n$ (It can be proved that $x\phi(n) \equiv 1 \pmod{n}$). Hence we see that $M = M'$. Thus we have achieved efficient encryption and decryption using RSA.

ii. Security of RSA

Three possible approaches to attacking the RSA algorithm are as follows:

Brute Force: This involves trying out all the possible private keys.

Mathematical attacks: There are several approaches, all equivalent in effect to factoring the product of 2 primes.

Timing attacks: These depend on the running time of the decryption algorithm.

Choosing large p and q values can prevent such attacks. Security of RSA thus lies in choosing the value n, which makes such attacks extremely difficult

iii. Elliptic Curve Diffie-Helman Protocol

ECDH is elliptic curve version of Diffie-Hellman key agreement protocol (refer section 2.1.1). The protocol for generation of the shared secret using ECC is as described below.

Alice takes a point Q and generates a random number $k_{a.}$ Alice computes the point $P = k_a Q$ and sends it to Bob (It should be noted that Q, P are public). Bob generates a random number k_b and computes point $M = k_b.Q$ and sends it to Alice. Alice now computes $P_1 = k_aM$ and Bob computes $P_2 = k_bP$

$P1 = P2 = k_b\ k_bQ$, this is used as the shared secret key.

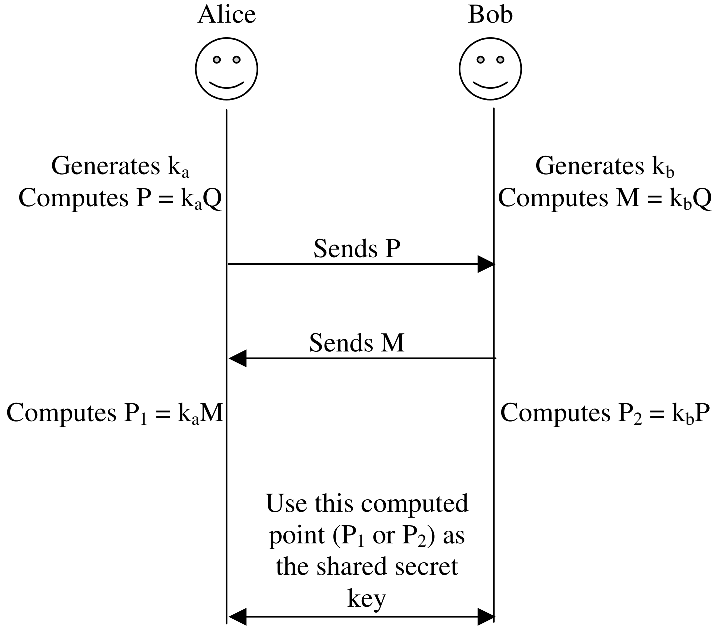


Figure1. Illustration of Elliptic Curve Diffie-Hellman Protocol

IV PROPOSED METHOD

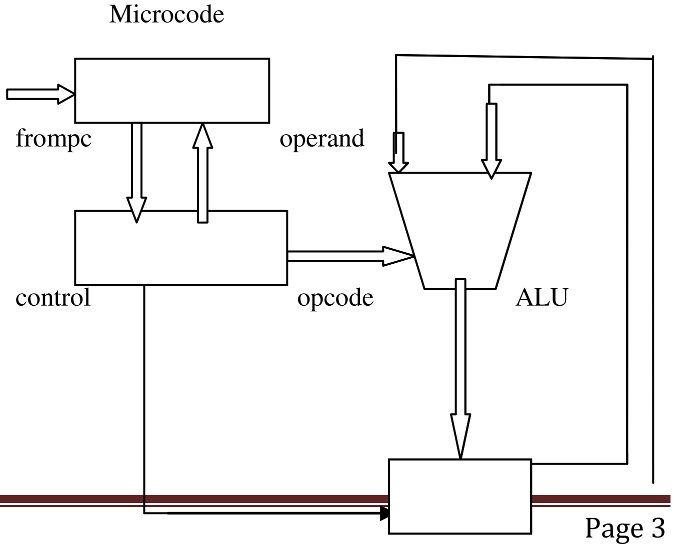
i.Microcode:

Algorithmic optimizations to the processor can be performed entirely in microcode. The size of the microcode is currently less than 512 16-bit words. A microcode description is a higher level abstraction. Easier to develop and debug.

ii.Galois field:

A field that contains a finite number of elements Elements of Galois field may be represented as polynomials.

Polynomial: $x^6 + x^4 + x + 1$



Data register

Figure2. Architecture of EC Processor

Advantage of Project:

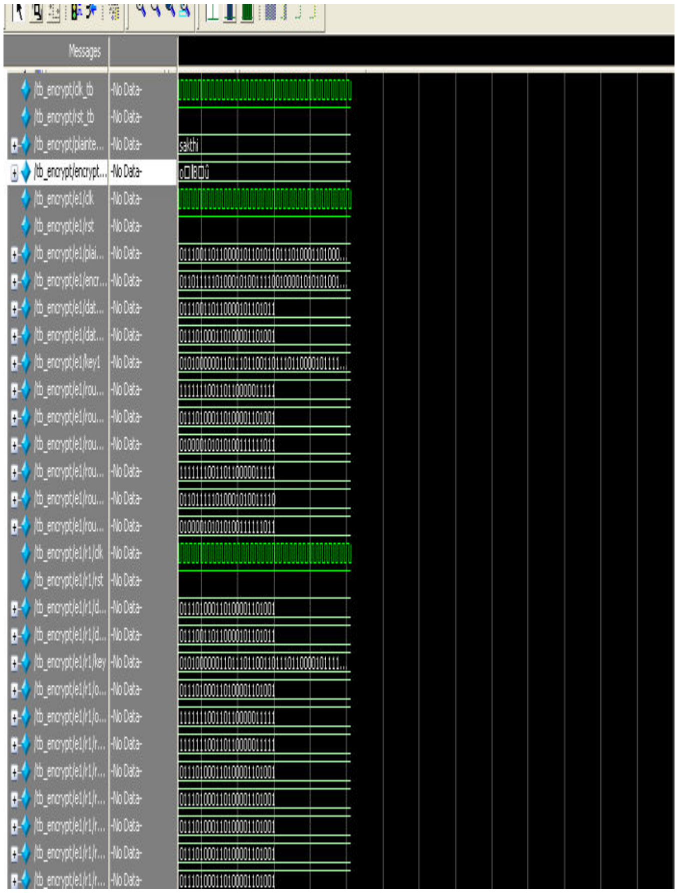
The advantage of elliptic curve over the other public key systems such as RSA, DSA etc is the key strength. The following table [3] summarizes the key strength of ECC based systems in comparison to other public key schemes.

1. Smaller key-length, less computation power security is high.
2. Same benefits of the other cryptosystems: but shorter key lengths.
3. Encryption and decryption and signature verification speed up.
4. Storage and bandwidth saving.

V RESULT AND CONCLUSION

.

In this project we perused the concept of Cryptography including the various schemes of system based on the kind of key and a few algorithms such as RSA and DSA. We studied in detail the mathematical foundations for elliptical curve based systems, basically the concepts of rings, fields, groups, Galois finite fields and elliptic curves and their properties. The various algorithms for the computation of the scalar product of a point on the elliptic curve were studied and their complexity were analyzed. Figure.3 shows the output for the proposed algorithm.



```
Messages
/tb_encrypt/clk_tb        -No Data-
/tb_encrypt/rst_tb        -No Data-
/tb_encrypt/plainte...    -No Data-  sakthi
/tb_encrypt/encrypt...    -No Data-  o□l8□û
/tb_encrypt/e1/clk        -No Data-
/tb_encrypt/e1/rst        -No Data-
/tb_encrypt/e1/plai...    -No Data-  0111001101100001011010110111010001101000...
/tb_encrypt/e1/encr...    -No Data-  011011111010001010011110010000101010100...
/tb_encrypt/e1/dat...     -No Data-  011100110110000101101011
/tb_encrypt/e1/dat...     -No Data-  011101000110100001101001
/tb_encrypt/e1/key1       -No Data-  01010000001101110110011011101100000101111...
/tb_encrypt/e1/rou...     -No Data-  111111100110110000011111
/tb_encrypt/e1/rou...     -No Data-  011101000110100001101001
/tb_encrypt/e1/rou...     -No Data-  010000101010100111111011
/tb_encrypt/e1/rou...     -No Data-  111111100110110000011111
/tb_encrypt/e1/rou...     -No Data-  011011111010001010011110
/tb_encrypt/e1/rou...     -No Data-  010000101010100111111011
/tb_encrypt/e1/r1/clk     -No Data-
/tb_encrypt/e1/r1/rst     -No Data-
/tb_encrypt/e1/r1/d...    -No Data-  011101000110100001101001
/tb_encrypt/e1/r1/d...    -No Data-  011100110110000101101011
/tb_encrypt/e1/r1/key     -No Data-  01010000001101110110011011101100000101111...
/tb_encrypt/e1/r1/o...    -No Data-  011101000110100001101001
/tb_encrypt/e1/r1/o...    -No Data-  111111100110110000011111
/tb_encrypt/e1/r1/r...    -No Data-  111111100110110000011111
/tb_encrypt/e1/r1/r...    -No Data-  011101000110100001101001
/tb_encrypt/e1/r1/r...    -No Data-  011101000110100001101001
/tb_encrypt/e1/r1/r...    -No Data-  011101000110100001101001
/tb_encrypt/e1/r1/r...    -No Data-  011101000110100001101001
/tb_encrypt/e1/r1/r...    -No Data-  011101000110100001101001
```

Figure.3 Output for ECC algorithm

The Table.1. given below shows very clearly that elliptic curves offer a comparable amount of security offered by the other popular public key for a much smaller key strength. This property of ECC has made the scheme quite popular of late.

Over the years, there have been software implementations of ECDSA over finite fields such as $F_{2^{155}}$, $F_{2^{167}}$, $F_{2^{176}}$, $F_{2^{191}}$ and F_p (p: 160 and 192 bit prime numbers). Schroppel et. Al [13] mentions an implementation of an elliptic curve analogue of the Diffie-Hellman key exchange algorithm over $F_{2^{155}}$ with a trinomial basis representation. The elliptic curve based public key cryptography schemes has been standardized by the Institute of Electrical and Electronic Engineers (IEEE) and the standard is available as IEEE P1363.

RSA/DSA Key length	ECC Key Length for Equivalent Security
1024	160
2048	224
3072	256
7680	384
15360	512

Table1. Comparison of the key strengths

VI REFERENCES

[1] B.Schneier. *Applied Cryptography*. John Wiley and Sons, second edition, 1996

[2] www.tcs.hut.fi/helger/crypto/link/public/elliptic

[3] Julio Lopez and Ricardo Dahab, "An overview of elliptic curve cryptography", May 2000.

[4] V. Miller, "Uses of elliptic curves in cryptography", Advances in Cryptology - CRYPTO'85, LNCS 218, pp.417-426, 1986.

[5] Jeffrey L. Vagle, "A Gentle Introduction to Elliptic Curve Cryptography", BBN Technologies
Mugino Saeki, "Elliptic curve cryptosystems", M.Sc. thesis, School of Computer Science, McGill University, 1996.

[5] http://citeseer.nj.nec.com/saeki97elliptic.html

[6] J. Borst, "Public key cryptosystems using elliptic curves", Master's thesis, Eindhoven University of Technology, Feb. 1997.

[7] http://citeseer.nj.nec.com/borst97public.html

[8] http://world.std.com/~franl/crypto.html

[9] Aleksandar Jurisic and Alfred Menezes, "Elliptic Curves and Cryptography", Dr. Dobb's Journal, April 1997, pp 26ff

[10] Robert Milson, "Introduction to Public Key Cryptography and Modular Arithmetic"

[11] Aleksandar Jurisic and Alfred J. Menezes, Elliptic Curves and Cryptography

[12] William Stallings, Cryptography and Network Security-Principles and Practice second edition, Prentice Hall publications.

[13] R. Schroppel, H. Orman, S. O'Malley and O. Spatscheck, "Fast key exchange with elliptic key systems", Advances in Cryptography, Proc. rypto'95, LNCS 963, pp. 43-56, Springer-Verlag, 1995.