ECC Decryption Based On GF (2m) Irreducable All One Polynomials

Duggirala Sowjanya[1], Dr. Pushpa Kotipalli[2]
[12]*Shri Vishnu College for Women. Vishnupur, Bhimavaram, West godavari district, Andhra pradesh,*

Abstract- Elliptic Curve Cryptography has been a recent research region within the subject of Cryptography. It gives higher level of safety with lesser key length as compared to other Cryptographic strategies. A new method has been proposed in this paper where the classic technique of mapping the characters to affine points in the elliptic curve has been removed. The corresponding ASCII values of the simple text are paired up. The paired values serve as input for the Elliptic curve cryptography. This new technique avoids the costly operation of mapping and the want to proportion the common research desk among the sender and the receiver. The algorithm is designed in any such manner that it is able to be used to encrypt or decrypt any type of script with described ASCII values.

IndexTerms - Galios Field, ECC, PSFG.

I. INTRODUCTION

Cryptography is the science of hiding information which can be found out most effective by means of legitimate users. It is used to make certain the secrecy of the transmitted statistics over an unsecure channel and prevent eavesdropping and data tampering. Many cryptography schemes had been proposed and used for securing statistics, a few makes use of the shared key cryptography and a few uses the public key cryptography (PKC). Shared key cryptography is a system that is uses best one key by each sender and receiver for reason of encrypting and decrypting the message. On the other hand, public key cryptography uses keys, private-key and public-key. To encrypt a message in Public key scheme, public-key may be used and to decrypt it back a private-key's used.

As in comparison to the shared key cryptography, public key cryptography is sluggish. However, public-key cryptography can be used with shared key cryptography to get the excellent of each. Public key cryptography have many blessings over the shared key, it will increase the safety and comfort where there's no want to distribute the non-public key to everybody. Most of nowadays's utility of cryptography asks for authentication and secrecy of the records. Secret transmission of statistics is an crucial assignment to maintain the data from the proof against assaults, threats and misuse. The encrypted text or facts is much less cozy due to the fact that it could be without problems decrypted. But an picture cannot be without difficulty decrypted by using attackers. Even facts may be transmitted greater securely by means of changing it into an photograph.

The maximum of hardware and software program products and standards that use public key approach for encryption and decryption, authentication and so on. Are based totally on RSA cryptosystem with the aid of using non Conventional algorithms amongst RSA and ECC. The major enchantment of ECC is that it could provide higher overall performance and protection for small key size, in comparison of RSA cryptosystem. ECC isn't always clean to apprehend by using attackers. So it presents higher protection via insecure channels.

In 1985, Neal Koblitz and Victor Miller independently proposed public key cryptosystems using elliptic curve. Since then, many researchers have spent years reading the strength of ECC and improving strategies for its implementation. Elliptic curve cryptosystem (ECC) gives a smaller and quicker public key cryptosystem. ECC has been commercially regular, and has additionally been followed by way of many standardizing bodies inclusive of ANSI, IEEE, ISO and NIST. The operation of every of the general public-key cryptographic schemes described in this document includes arithmetic operations on an elliptic curve over a finite area decided through some elliptic curve area parameters.

The predominant attraction of ECC is that it could provide better performance and safety for small key size, in contrast of RSA cryptosystem. In ECC a one hundred sixty-bit key affords the identical security as compared to the conventional crypto gadget RSA with a 1024-bit key, accordingly inthis manner it may reduce computational fee or processing cost. The protection of ECC depends on the problem of finding the multiplicand for the given product and multiplier. ECC is not smooth to apprehend by way of attackers. So affords higher protection thru insecure channels.

II. LITERATURE REVIEW

Elliptic curves were studied for over hundred years and were used to remedy a numerous range of problems. For instance, elliptic curves are used in proving Fermat's last theorem, which states that xn+yn = zn has non 0 integer answers for x, y, and z while n > 2 [1,8].

The use of elliptic curves in public key cryptography was first proposed independently by using Koblitz [1,9] and Miller [10] within the Nineteen Eighties. Since then, there has been an abundance of research on the security of ECC. In the 1990's ECC commenced to get common by several accepted groups, and several protection protocols based on ECC [14, 20, 21] were standardized. The foremost gain of ECC over conventional uneven crypto systems [2] is the expanded safety presented with smaller key sizes. For example, a 256 bit key in ECC produces the identical level of security as a 3072 bit RSA key1. The smaller key sizes leads to compact implementations and elevated overall performance.

This makes ECC desirable for low strength aid constrained gadgets. An elliptic curve is the set of solutions (x, y) to

Equation 2.1 together with the point at infinity (O). This equation is referred to as the Weierstraß equation [1,8].

$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$(1)

For cryptography, the points at the elliptic curve are selected from a massive finite subject. The set of factors at the elliptic curve shape a group underneath the addition rule. The factor O is the identity detail of the organization. The operations on the elliptic curve, i.E. The institution operations are factor addition, factor doubling and point inverse. Given a factor P = (x, y) on the elliptic curve, and a high quality integer n, scalar multiplication is defined as

nP = P + P + P + ···P(n times) (2)

The order of the factor P is the smallest advantageous integer n such that nP = O. The points O, P, 2P, 3P, • • • (n − 1)P shape a group generated via P. The organization is denoted as < P >.

The safety of ECC is furnished through the elliptic curve discrete logarithm trouble (ECDLP), that's described as follows : Given a point P on the elliptic curve and any other factor Q ∈< P >, determine an integer okay (zero ≤ k ≤ n) such that Q = kP. The problem of ECDLP is to calculate the price of the scalar okay given the factors P and Q. K is referred to as the discrete logarithm of Q to the base P. P is the generator of the elliptic curve and is referred to as the base point.

There have been several pronounced excessive overall performance FPGA processors for elliptic curve cryptography. Various acceleration techniques were used ranging from efficient implementations to parallel and pipelined architectures. In [2] the 1st viscount montgomery of alamein multiplier [3] is used for scalar multiplication. The finite discipline multiplication is completed the usage of a digit-serial multiplier proposed in [3]. The Itoh-Tsujii algorithm is used for finite area inversion. A factor multiplication over the sphere GF(2167) is achieved in 0.21ms.

In [3] a totally parameterizable ABC processor is brought, which may be used with any subject and irreducible polynomial without need for reconfiguration. This implementation despite the fact that quite bendy is slow and does not attain required speeds for high bandwidth packages. A 239 bit point multiplication calls for 12.8ms, really this is extremely excessive compared to different reported implementations.

In [3], the ECC processor designed has squarers, adders ,and multipliers within the statistics path. The authors have used a hybrid coordinate illustration in affine, Jacobian, and López-Dahab shape. In [3,4] an cease-to-quit system for ECC is developed, which has a hardware implementation for ECC on an FPGA. The excessive overall performance is received with an optimized field multiplier. A digit-serial shift-and-upload multiplier is used for the motive. Inversion is carried out with a committed division circuit.

The processor supplied in [5] achieves point multiplication in 0.074ms over the field GF(2163). However, the implementation is for a specific form of elliptic curves called Koblitz curves. On these curves, numerous acceleration techniques based on precomputation [6] are viable. However our work makes a speciality of popular curves in which such accelerations do no longer paintings.

In [7] a excessive velocity elliptic curve processor is presented for the sector GF(2191), in which factor multiplication is finished in 0.056ms. A binary Karatsuba multiplier is used for the field multiplication. However, no inverse set of rules appears to be designated in the paper, making the implementation incomplete.

In [8] a microcoded technique is observed for ECC making it clean to regulate, alternate, and optimize. The microcode is stored within the block RAM [9] and does no longer require additional assets.

In [4], the finite area multiplier in the processor is avoided from becoming idle. The finite area multiplier is the bottle neck of the layout therefore stopping it from becoming idle improves the general overall performance. Our design of the ECCP is on similar lines in which the operations required for factor addition and point doubling are scheduled

IV. METHODOLOGY

We received several curves $y^2 = x^3 + ax + b$ over $\mathbb{Z}_p$ where a, b, p are given, and the order of $E(\mathbb{Z}_p)$ is a prime number which is also given.

According to Hasse's theorem,$\left| \#E(\mathbb{Z}_p) - (p+1) \right| \leq 2\sqrt{\text{p}}$, where $\#E(\mathbb{Z}_p)$ is the size of $E(\mathbb{Z}_p)$, and thus $\#E(\mathbb{Z}_p) \approx p$. In the curves we got, p and $\#E(\mathbb{Z}_p)$ were of 40-60 bits.

When the group's order is prime, any $P \neq \infty$ is of the same order as the group, since $Order(P) | \#E(\mathbb{Z}_p)$ and $Order(P) \neq 1$.

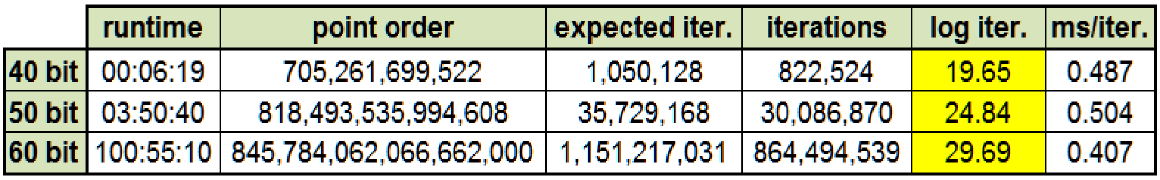
We got 7 40-bit curves, 10 50-bit curves and 9 60-bit curves, out of which we have attacked all 40-50 bit curves and 2 of the 60 bit curves, some several times with the isomorphism enhancement enabled or disabled, or for different values of L.

For each curve we have attacked, we found a point P on the curve in the following manner: starting from $x = 0$ we checked if $x^3 + ax + b$ is a quadratic residue modulo p, if so- we found a square root modulo p of $x^3 + ax + b$ and set it to be y, and then $P = (x, y)$, otherwise we advance to the next value of x.

The most attempts it took us to find a point was 6, and usually either $x = 0$ or $x = 1$ worked.

After finding the first point, we randomly chose an integer $0 \leq k < \#E(\mathbb{Z}_p)$, computed $Q = kP$ and solved ECDLP$(P, \#E(\mathbb{Z}_p), Q)$ using Pollard's Rho algorithm, since the order is a prime.

The average results from our runs:



	runtime	point order	expected iter.	iterations	log iter.	ms/iter.
40 bit	00:06:19	705,261,699,522	1,050,128	822,524	19.65	0.487
50 bit	03:50:40	818,493,535,994,608	35,729,168	30,086,870	24.84	0.504
60 bit	100:55:10	845,784,062,066,662,000	1,151,217,031	864,494,539	29.69	0.407

Where the expected iteration number is estimated as $\sqrt{\frac{\pi \cdot \#E(\mathbb{Z}_p)}{2}}$.

It is well worth mentioning that the common new release matter for n bits is approximately n/2-bits, as we would anticipate from PR.

The average runtime per generation is lower for 60 bit than it's far for 40 and 50 bit.

This should end result from the fact that the common here displays most effective 2 60-bit runs, and that even for forty and 40 bit there has been some distribution of this parameter- all runs have been done on unique laptops with different energetic processes, and perhaps the modifications in workload had an impact on the common runtime in keeping with new release.

These runs were completed on a unique pc than the one used for the first assignment in which we were given zero.22ms in keeping with new release, which is plenty better- despite the fact that we recollect the reality that q in the first mission changed into simplest 28-bits.

All-one polynomials (AOP) or 1-equally spaced polynomials form a unique magnificence which may be used for less difficult and more efficient implementation as compared to trinomials and penanomial-based multipliers. The AOP-primarily based illustration of factors, for this reason, predicted to have potential application in green hardware implementation of elliptic curve cryptosystems and errors manipulate coding. Irreducible AOPs are not as abundant as irreducible trinomials or pentanomials, however it's also now not tough to find the AOP bases for generating the finite fields.

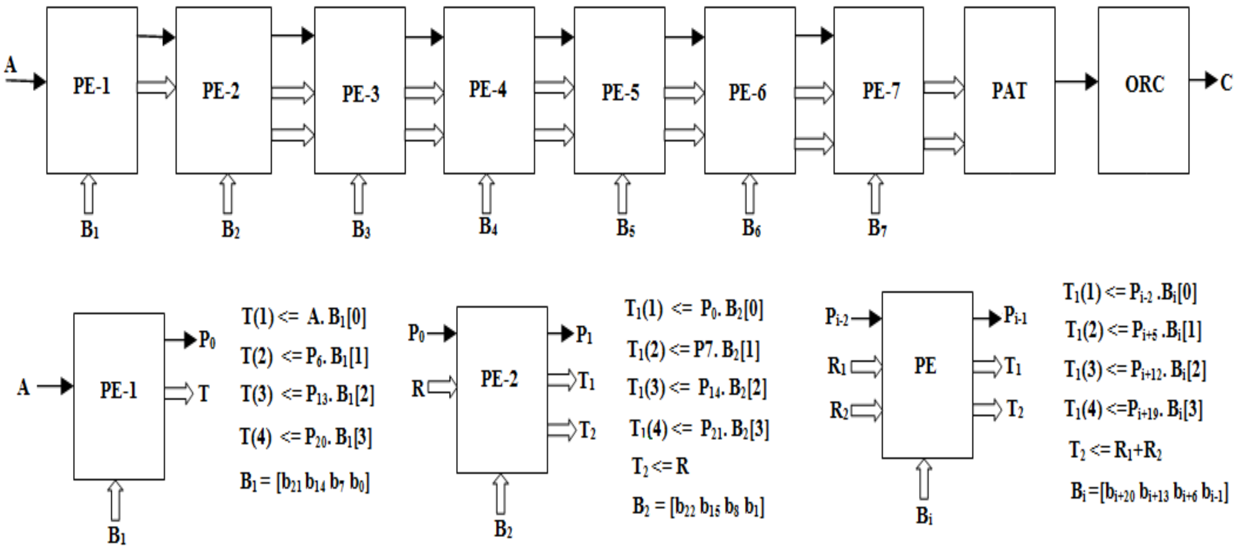


Fig.1: Systolic Array all irreducible polynomials

It is known that for m < 2000, there exists 108 possible AOP bases, e.G., m = 2,4, 10, 12, 18, 28, 36, 52, 58, 60, 82, 100, 106, 148, 172, 178, 226, 268, 292, 316, 346, 388, 466, 508, 556, 562 etc, and infinitely many more for m > 2000 [2]. Efficient architectures for the field multiplication and the computation of power-sum of the form (A+B2) for discipline generated by AOPs.

For both fixed point and GF multiplications, step one is producing a matrix of partial products. These are calculated by ANDing the corresponding term in X and Y as: Partial products are arranged in rows, with each row shifted positions to the left as in Figure 2. Each dot represents the output of an AND gate. The fixed point product is received by way of including the ensuing partial products.

The partial product matrix is composed of 4 sub-matrices . The upper-right and lower-left sub-matrices correspond to the partial merchandise to be added for GF multiplications. These partial merchandise are indicated by hole dots in Figure 2. The partial products within the different two sub-matrices, indicated through black dots, are set to 0 whilst calculating a GF product, by ANDing the to those sub-matrices with the manipulate sign. This greater hardware most effective represents 28 AND gates. It adds most effective one AND gate postpone to the essential path. The XORing represents the GF sum, for an iterative key era unit

The layout supplied on this paper uses the pre-calculated canonical illustration of the seven GF-factors of the form. Each of these seven values is an 28-bit vector. The discount is accomplished through including the corresponding GF element to alternative for each bit. The seven 28-bit values to be introduced are computed as quickly as the prolonged result is prepared. The modulo2 addition of every to the 28 least massive bits of the prolonged end result is achieved in parallel, in a binary tree configuration, which has logarithmic delay. An implementation is shown in Figure 1, in which the AND blocks according to-form and the XOR blocks carry out GF addition.

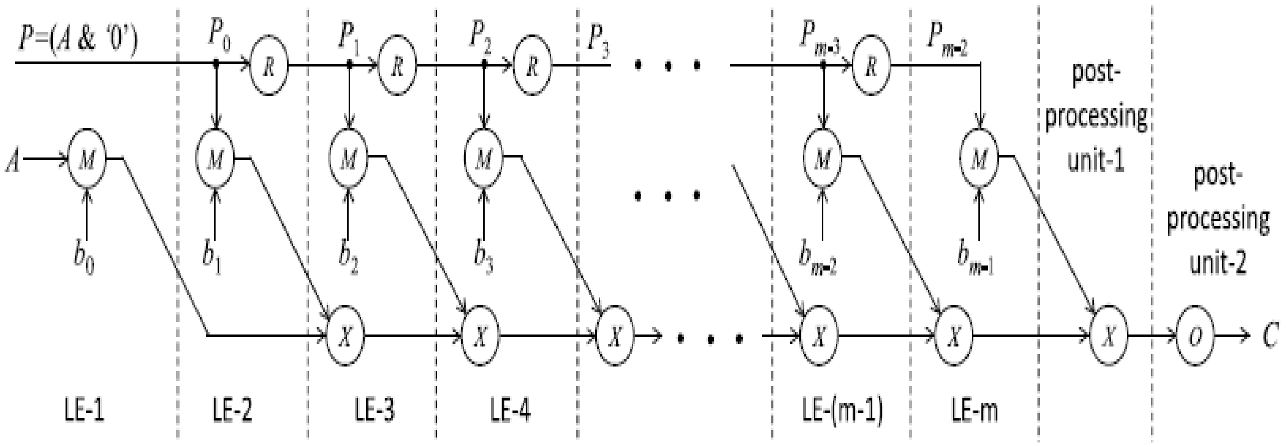


Fig.2: Galios field multiplier

The proposed scheme for implementation of finite field multiplication over *GF(2m)* generated by AOP can be outlined as follows.

Algorithm for Multiplication:

STEP-1: Perform multiplication of bit *b*0 with the input operand *A*, to obtain *b*0 ・ *A*; and initialize the first *(m*−1*)*-bits of a finite field accumulator (FFA) by *(b*0*.ai)*, for 0 ≤*i* ≤*m* −1 according to (13d). The *m*-th location (i.e., the MSB) of the FFA is initialized to zero.

STEP-2: For *i* = 1 to *m* −1 Perform cyclic left-shift operation of the polynomial *Pi*−2*α* of degree *(m* + 1*)* to reduce its degree by one to obtain the operand *Pi*−1 of degree *m*. Perform bit-level multiplication of *bi* with *Pi*−1 to obtain *Yi*. Add *Yi* to the content of the FFA to obtain the partial result of degree *m*.

STEP-3: Perform modular reduction of *Y* to reduce its degree from *m* to *(m* −1) to obtain the desired product value. Recursive operations of the proposed algorithm are in STEP-2, while STEP-1 may be considered as pre-processing step and STEP-3 may be considered as a post-processing step.

DECRYPTION PROCESS

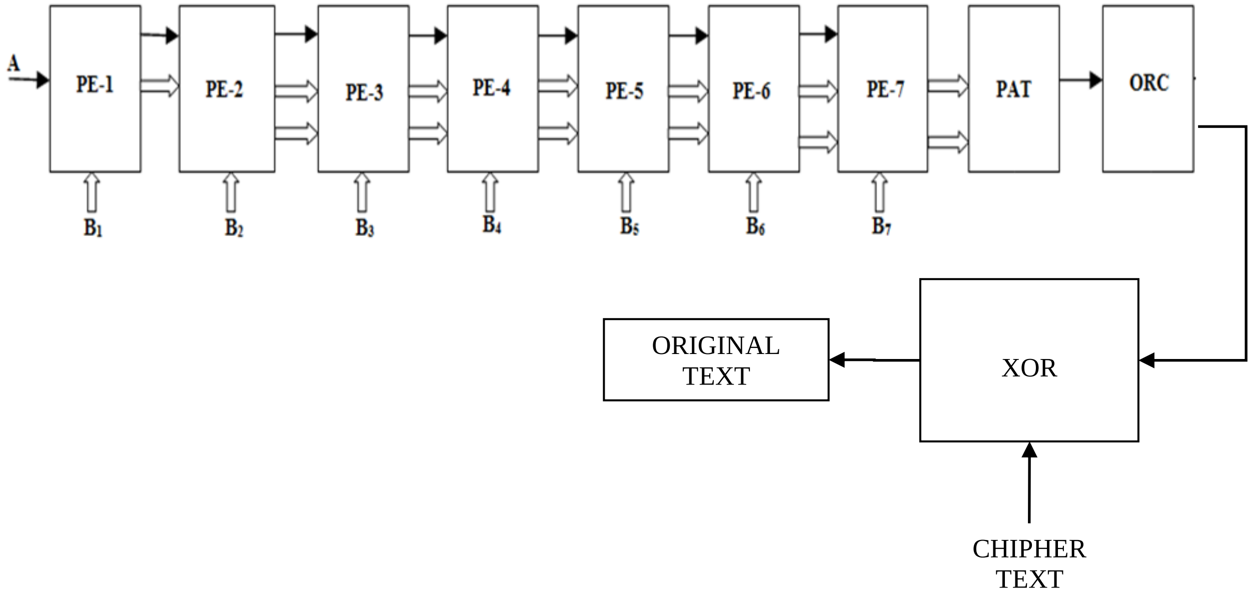


Fig.3: Decryption for parallel systolic

In encryption and decryption process we do find AND and XOR gates. During the process of key generation B is the public key and A is the private key. D stands for original text and Chipher text is denoted by ET in our programming implementation.

So as per the limitation of encryption and decryption, decryption is the reciprocal process for encryption. The addition process was shifted to subtraction process after key generation unit. B is a public key will be generated by using system.

DECRYPTION ALGORITHM:-

To decrypt the cipher text, following steps are performed:-

Step 1. The receiver computes the product of B1 and its private key

Step 2. Then the receiver subtracts this product from the second point B2

M = B2- (B1)

M is the original data sent by the sender

B. Results of Simulation

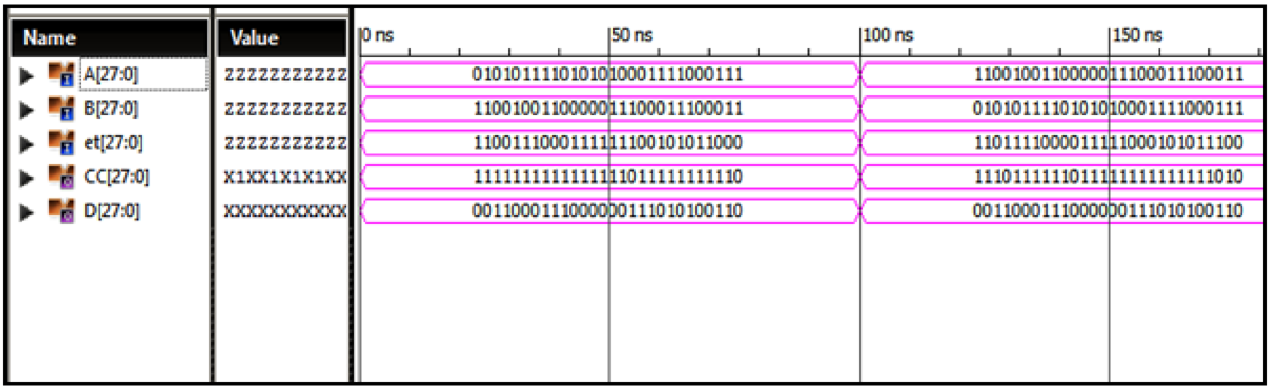


Fig.4: DECRYPTION

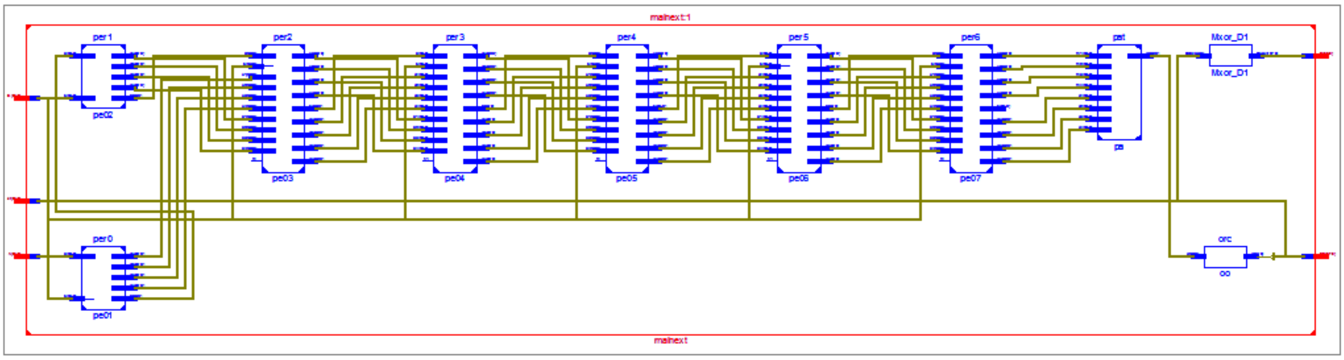


Fig.5: DECRYPTION RTL SCHEMATIC

V. RESULTS AND DISCUSSION

A. Results of Descriptive Statics of Study Variables

PARAMETERS	DECRYPTION
AREA (UM^2)	15402
CRITICAL PATH DELAY	0.94NS
POWER(MW)	2.75
AREA DELAY PRODUCT	14494
POWER DEALY PRODUCT	2.585

VI. CONLUSION

Decryption operation is carry out edvery hastily regardless of large range of phrases as enter, provides smaller length cipher text in comparison to different approach which significantly helps in saving bandwidth even as sending and we don't require mapping and common appearance up table. ECC provide a higher protection with lesser key size as compared to the very a success RSA. Elliptic curve discrete logarithm problem could be very difficult to remedy, this assets is used in ECC. As ECC offers identical security like other cryptographic system but with much less key size, it's far very suitable for devices which have power, storage and processing hindrance.

VII. REFERENCES

[1]. Deyan Chen Hong Zhao, Data Security and Privacy Protection Issues in Cloud Computing, and ICCSEE, 2012

International Conference on (Volume: 1) ePrint23-25 March 2012the IEEE website. http://www.ieee.org/

[2]. Parsi Kalpana, Sudha Singaraju, Data Security in Cloud Computing using RSA Algorithm, International Journal of Research in Computer and Communication technology, IJRCCT, ISSN 2278-5841, Vol 1, Issue 4, September 2012.

[3]. Neha Tirthani, and Ganesan. R R, Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography, International Association for Cryptologic Research Cryptology ePrint 4-9, 2014.

[4]. N. Koblitz, elliptic curve cryptosystem, mathematics of Computation, Volume 48-1987, PP-203-209.

[5]. Ms. Bhavana Sharma, Security Architecture Of Cloud Computing Based On Elliptic Curve Cryptography (ECC), International Journal of Advances in Engineering Sciences Vol.3 (3), July, 2013 e-ISSN: 2231-0347 Print-ISSN: 2231-2013.

[6]. Ms. Priyanka Sharda, Providing data security in cloud computing using elliptical curve cryptography, International Journal on Recent and innovation trends in computing and communication, vol.3 issue 2, 2015.

[7]. Elliptical curve cryptography https://en.Wikipedia.org/wiki/Elliptic_Curve_Cryptography.

[8]. "A Performance Comparison of Data Encryption Algorithms," IEEE [Information and Communication Technologies, 2005. ICICT 2005. First International Conference, 2006-02-27,

[9]. Nicholas Jansma, Brandon Arrendond, “Performance Comparison of Elliptic Curve and RSA Digital Signatures” April, 2004. [10] Veerraju Gampala, Srilakshmi Inuganti, Satish Muppidi, “Data Security in Cloud Computing with Elliptic Curve Cryptography” vol. 2 Issue 3, July 2012.