ECC Encryption and Decryption based on GF(2m) Irreducable all one Polynomials

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

Abstract- Redundant basis (RB) multipliers over Galois Field (GF (2^m)) have gained huge popularity in elliptic curve cryptography (ECC) mainly because of their negligible hardware cost for squaring and modular reduction. In this paper, we have proposed a novel recursive decomposition algorithm for RB multiplication to obtain high-throughput digit-serial implementation. Through efficient projection of signal- flow graph (SFG) of the proposed algorithm, a highly regular processor-space flow-graph (PSFG) is derived. By identifying suitable cut-sets, we have modified the PSFG suitably and performed efficient feed-forward cut-set retiming to derive three novel multipliers which not only involve significantly less time-complexity than the existing ones but also require less area and less power consumption compared with the others. Both theoretical analysis and synthesis results con firm the efficiency of proposed multipliers over the existing ones. The synthesis results for field programmable gate array (FPGA) and application specific integrated circuit (ASIC) realization of the proposed designs and competing existing designs are compared. It is shown that the proposed high-throughput structures are the best among the corresponding designs, for FPGA and ASIC implementation.

IndexTerms **-** Galios Field, ECC, PSFG.

I. INTRODUCTION

Cryptography is the science of hiding information which can be revealed only by legitimate users. It is used to ensure the secrecy of the transmitted data over an unsecure channel and prevent eavesdropping and data tampering.Many cryptography schemes were proposed and used for securing data, some uses the shared key cryptography and some uses the public key cryptography (PKC). Shared key cryptography is a system that is uses only one key by both sender and receiver for purpose of encrypting and decrypting the message. On the other hand, public key cryptography uses two keys, private-key and public-key. To encrypt a message in Public key scheme, public-key will be used and to decrypt it back a private-key is used.

As compared to the shared key cryptography, public key cryptography is slow. However, public-key cryptography can be used with shared key cryptography to get the best of both. Public key cryptography have many advantages over the shared key, it increases the security and convenience where there is no need to distribute the private key to anyone. Most of today's application of cryptography asks for authentication and secrecy of the data. Secret transmission of data is an important task to preserve the data from the immune to attacks, threats and misuse. The encrypted text or data is less secure since it can be easily decrypted. But an image cannot be easily decrypted by attackers. Even data can be transmitted more securely by converting it into an image.

The most of hardware and software products and standards that use public key technique for encryption and decryption, authentication etc. are based on RSA cryptosystem by using non Conventional algorithms among RSA and ECC. The main attraction of ECC is that it can provide better performance and security for small key size, in comparison of RSA cryptosystem. ECC is not easy to understand by attackers. So it provides better security through insecure channels.

In 1985, Neal Koblitz and Victor Miller independently proposed public key cryptosystems using elliptic curve. Since then, many researchers have spent years studyingthe strength of ECC and improving techniques for its implementation. Elliptic curve cryptosystem (ECC) provides a smaller and faster public key cryptosystem. ECC has been commercially accepted, and has also been adopted by many standardizing bodies such as ANSI, IEEE, ISO and NIST. The operation of each of the public-key cryptographic schemes described in this document involvesarithmetic operations on an elliptic curve over a finite field determined by some elliptic curve domainparameters.

The main attraction of ECC is that it can provide better performance and security for small key size, in comparison of RSA cryptosystem. In ECC a 160-bit key provides the same security as compared to the traditional crypto system RSA with a 1024-bit key, thus inthis way it can reduce computational cost or processing cost. The security of ECC depends on the difficulty of finding the multiplicand for the given product and multiplier. ECC is not easy to understand by attackers. So provides better security through insecure channels.

II. LITERATURE REVIEW

Elliptic curves have been studied for over hundred years and have been used to solve a diverse range of problems. For example, elliptic curves are used in proving Fermat's last theorem, which states that $x^n+y^n= z^n$has non zero integer solutions for x, y, and z when n >2 [1,8].

The use of elliptic curves in public key cryptography was first proposed independently by Koblitz [1,9] and Miller [10] in the 1980s. Since then, there has been an abundance of research on the security of ECC. In the 1990's ECC began to get accepted by several accredited organizations, and several security protocols based on ECC [14, 20, 21] were standardized. The main advantage of ECC over conventional asymmetric crypto systems [2] is the increased security offered with smaller key sizes. For example, a 256 bit key in ECC produces the same level of security as a 3072 bit RSA key1. The smaller key sizes leads to compact implementations and increased performance.

This makes ECC suited for low power resource constrained devices. An elliptic curve is the set of solutions (x, y) to Equation 2.1 together with the point at infinity (O). This equation is known 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 on the elliptic curve are chosen from a large finite field. The set of points on the elliptic curve form a *group* under the addition rule. The point O is the identity element of the group. The operations on the elliptic curve, i.e. the group operations are *point addition*, *point doubling* and *point inverse*. Given a point P = (x, y) on the elliptic curve, and a positive integer n, *scalar multiplication* is defined as

nP = P + P + P + ⋯P(n times) (2)

The *order* of the point P is the smallest positive integer n such that nP = O. The points {O, P, 2P, 3P, ⋯(n −1)P} form a group generated by P. The group is denoted as < P >.

The security of ECC is provided by the elliptic curve discrete logarithm problem (ECDLP), which is defined as follows : Given a point P on the elliptic curve and another point Q ∈ < P >, determine an integer k (0 ≤k ≤n) such that Q = kP. The difficulty of ECDLP is to calculate the value of the scalar k given the points P and Q. k is called the discrete logarithm of Q to the base P. P is the generator of the elliptic curve and is called the base point.

There have been several reported high performance FPGA processors for elliptic curve cryptography. Various acceleration techniques have been used ranging from efficient implementations to parallel and pipelined architectures. In [2] the Montgomery multiplier [3] is used for scalar multiplication. The finite field multiplication is performed using a digit-serial multiplier proposed in [3]. The Itoh-Tsujii algorithm is used for finite field inversion. A point multiplication over the field GF(2^{167}) is performed in 0.21ms.

In [3] a fully parameterizable ABC processor is introduced, which can be used with any field and irreducible polynomial without need for reconfiguration. This implementation although highly flexible is slow and does not reach required speeds for high bandwidth applications. A 239 bit point multiplication requires 12.8ms, clearly this is extremely high compared to other reported implementations.

In [3], the ECC processor designed has squarers, adders ,and multipliers in the data path. The authors have used a hybrid coordinate representation in affine, Jacobian, and López-Dahab form. In [3,4] an end-to-end system for ECC is developed, which has a hardware implementation for ECC on an FPGA. The high performance is obtained with an optimized field multiplier. A digit-serial shift-and-add multiplier is used for the purpose. Inversion is done with a dedicated division circuit.

The average results from our runs:

The processor presented 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, several acceleration techniques based on precomputation [6] are possible. However our work focuses on generic curves where such accelerations do not work.

In [7] a high speed elliptic curve processor is presented for the field GF(2191), where point multiplication is done in 0.056ms. A binary Karatsuba multiplier is used for the field multiplication. However, no inverse algorithm seems to be specified in the paper, making the implementation incomplete.

In [8] a microcoded approach is followed for ECC making it easy to modify, change, and optimize. The microcode is stored in the block RAM [9] and does not require additional resources.

In [4], the finite field multiplier in the processor is prevented from becoming idle. The finite field multiplier is the bottle neck of the design therefore preventing it from becoming idle improves the overall performance. Our design of the ECCP is on similar lines where the operations required for point addition and point doubling are scheduled

III. 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, $\left| \#E(\mathbb{Z}_p) - (p+1) \right| \leq 2\sqrt{\mathrm{p}}$, 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.

	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 worth pointing out that the average iteration count for n bits is approximately $\frac{n}{2}$-bits, as we would expect from PR.

The average runtime per iteration is lower for 60 bit than it is for 40 and 50 bit. This could result from the fact that the average here reflects only 2 60-bit runs, and that even for 40 and 40 bit there was some distribution of this parameter- all runs were done on two different laptops with other active processes, and perhaps the changes in workload had an effect on the average runtime per iteration.

These runs were done on a different computer than the one used for the first challenge where we got 0.22ms per iteration, which is much better- even if we consider the fact that q in the first challenge was only 28-bits.

All-one polynomials (AOP) or 1-equally spaced polynomials form a special class which can be used for simpler and more efficient implementation compared to trinomials and penanomial-based multipliers. The AOP-based representation of elements, thus, expected to have potential application in efficient hardware implementation of elliptic curve cryptosystems and error control coding. Irreducible AOPs are not as abundant as irreducible trinomials or pentanomials, but it is also not difficult to find the AOP bases for generating the finite fields.

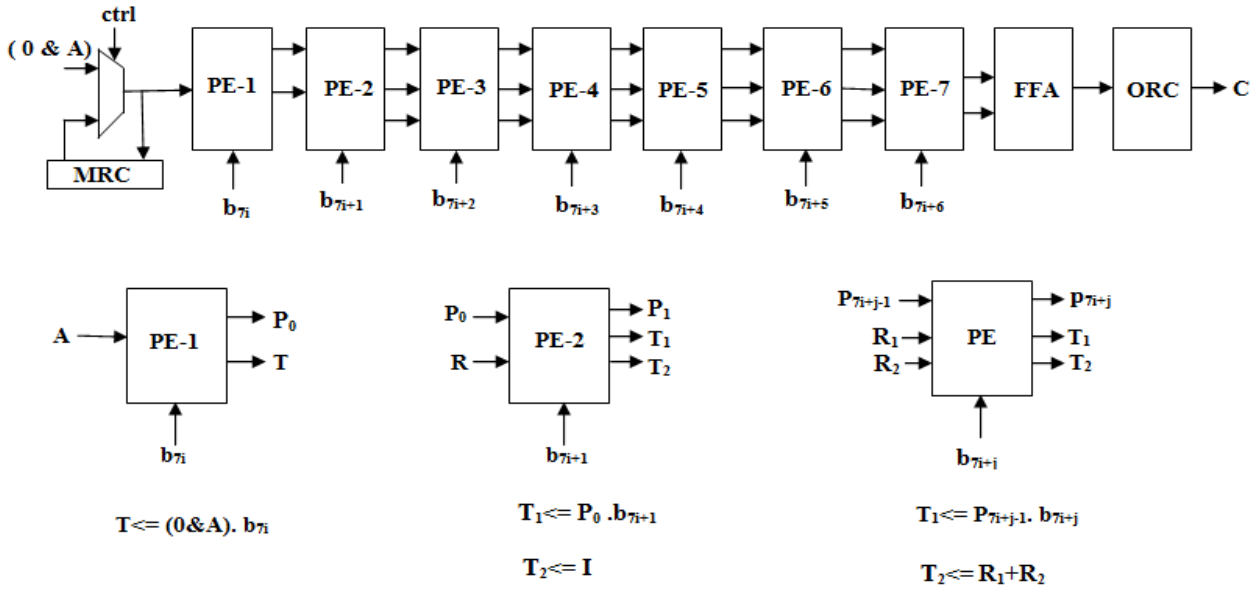


Fig.1: Systolic Array all irreducible polynomials

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 field generated by AOPs.

For both fixed point and GF multiplications, the first stepis generating a matrix of partial products. These are calculated by ANDing the corresponding term in X and Y as: Partial products are arranged in rows, witheach row shiftedpositions to the left as in Figure 2. Eachdot represents the output of an AND gate. The fixed pointproductis obtained by adding the resulting partialproducts.

The partial product matrix is composed of four sub-matrices . The upper-right and lower-left sub-matrices correspond to the partial products to be added forGF multiplications. These partial products are indicated byhollow dots in Figure 2. The partial products in the othertwo sub-matrices, indicated by black dots, are set to zerowhen calculating a GF product, by ANDing theto thesesub-matrices with the control signal. This extra hardware only represents 28 AND gates. It adds only one AND gate delay tothe critical path. The XORing represents the GF sum, for an iterative key generation unit

The design presented in this paper uses the pre-calculatedcanonical representation of the seven GF-elements of theform. Each of these seven values is an 28-bit vector. The reduction is performed by adding the corresponding GF elementto substitute for each bit. The seven 28-bit values to be added are computedas soon as the extended result is ready. The modulo2 addition of eachto the 28 least significant bits of theextended result is done in parallel, in a binary tree configuration, which has logarithmic delay. An implementation is shown in Figure 1, where the AND blocks per-form and the XOR blocks perform GF addition.

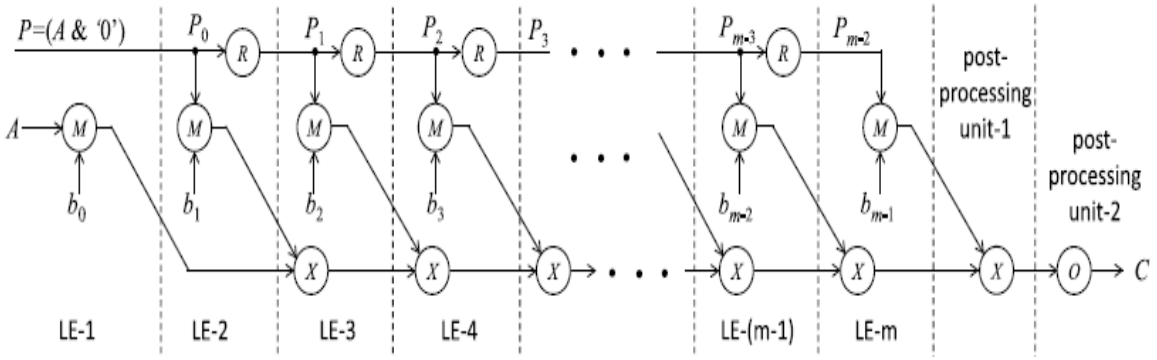


Fig.2: Galios field multiplier

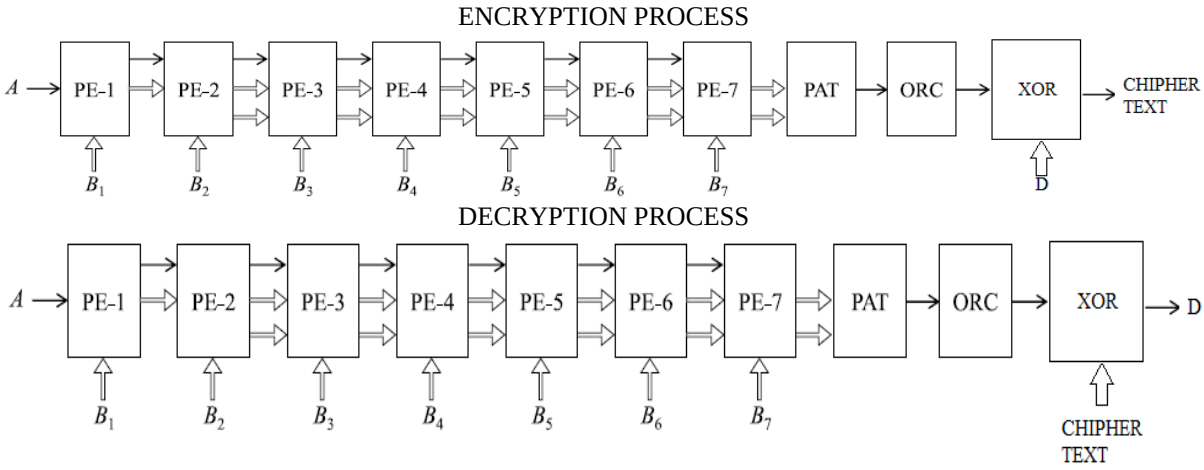
The proposed scheme for implementation of finite field multiplication over *GF(2m)* generated by AOP can be outlined as follows.

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.



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.

ENCRYPTION ALGORITHM:-

Suppose sender wants to send a message m to the receiver

Step 1. Let m has any point M on the elliptic curve

Step 2. The sender selects a random number k from [1,n-1]

Step 3. The cipher texts generated will be the pair of points (A,B) where

B1= A⊕ *B**G

B2= M ⊕ B1

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

IV. RESULTS AND DISCUSSION

A. Results of Descriptive Statics of Study Variables

PARAMETERS	ENCRYPTION	DECRYPTION
AREA (UM^2)	15402	15402
CRITICAL PATH DELAY	0.72NS	0.94NS
POWER(MW)	2.75	2.75
AREA DELAY PRODUCT	11089	14494
POWER DEALY PRODUCT	1.98	2.585

V. CONLUSION

In this project, we have presented an application of ECC with Generator g data encryption. ECC points convert into cipher text at sender side and Decryption algorithm is used to get original text within a very short time with a high level of security at the receiver side. Elliptic curves cryptography are believed to provide good security with smaller key sizes, something that is very useful in many applications. Smaller key sizes may result in faster execution timings for the image encryption, which are beneficial to systems where real time performance is a critical factor ECC can be used into a security system such as Video Compression, Face recognition, Voice recognition, thumb impression, Sensor network, Industry and Institutions.

Nowadays, RSA generally uses public key cryptosystem in most applications that use PKC. However, recently ECC has a trend which makes it become the convenient cryptography system. ECC is also becomes substitute for RSA in efficacious applications caused by its efficiency in software as well as in hardware realizations. ECC provides a better security with shorter bit sizes than in RSA. Shorter key length saves bandwidth, power, and it enhances the performance. In contrast with the past, pairing in ECC attracts more attention of experts because it can be used to build a number of cryptographic schemes that cannot be constructed in any other way.

VI. 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.