A Comparison among Fast Point Multiplication Algorithms in Elliptic Curve Cryptosystem

Sasmita Padhy
Koneru Lakshmaiah Education Foundation https://orcid.org/0000-0001-8345-9834
T. N. Shankar
Koneru Lakshmaiah Education Foundation
Sachikanta Dash (dash.sachikanta@gmail.com)
DRIEMS https://orcid.org/0000-0002-0807-4624

Research Article

Keywords: Elliptic Curve, scalar point multiplication, Double and Add, MOF, Complementary Recoding, Addition-Subtraction

Posted Date: February 21st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-862241/v1

A Comparison among Fast Point Multiplication Algorithms in Elliptic Curve Cryptosystem

Sasmita Padhy[ψ], T.N.Shankar(IEEE Member)[ψ], Sachikanta Dash[¥]

Dept. of CSEψ
Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, AP, INDIA
Dept. of CSE[¥]
DRIEMS, Cuttack, Odisha, INDIA
pinky.sasmita@gmail.com [ψ] , tnshankar2004@kluniversity.in [ψ], dash.sachikanta@gmail.com [¥]

***Abstract*: In the Elliptic Curve Cryptosystem, the multiplication of points is essential in the successful computation of any operation. Reduction of time complexity for the mathematical operations in Elliptic Curve Cryptosystems with minimum hardware resources, the methods: Addition and Subtraction, Mutual Opposite Form, and Complementary Recoding techniques proposed as fast scalar multiplication schemes. The fast-point multiplication method is always necessary for any mathematical operation on an Elliptic Curve Cryptosystem with a restricted system. This study compares the performance of fast-point multiplication algorithms in terms of computational and execution time to determine the quickest one. For any elliptic curve-related arithmetic operations, point multiplication has a vital role in reducing the idle time of hardware utilization. Rapid point multiplication is essential to minimize enhanced time complexity to determine the most suitable algorithm for mobile devices.**

***Keywords*: Elliptic Curve, scalar point multiplication, Double and Add, MOF, Complementary Recoding, Addition-Subtraction**

I. Introduction

Across the whole of history, civilization has felt the need to safeguard information. Several cryptographic keys of varying complexity have been developed. Fortunately, as computing power has increased over the last 20 years, now many techniques depending upon the complexity of computation are becoming more sensitive to cryptographic attack. So, to build cryptosystems it is important to implement different methods with more advanced protocols. In recent days a cryptosystem based on elliptic curves is now become one of the highly secure system is known as (Elliptic Curve Cryptosystems -ECC). To solve the discrete logarithm problems, ECC is very much secure due to unavailability of sub exponential algorithms [10, 12]. Scalar point multiplication is very costlier and are very common operations used in ECC to generate keys, encrypt/decrypt of data, and verifying and signing of digital signature.

The scalar point multiplication is a very important operation in the elliptic curve cryptosystem. With the features and comparisons provided in this paper, you can do rapid point multiplication. The NAF (Non-Adjacent Form) algorithm for implementing scalar point multiplication and its performance on GF (Galois Field) at the topmost level, as well as the calculation time, are clearly specified with the corresponding algorithm. This study effort includes a comparison of the multiplication algorithm utilizing complimentary recoding approach to other fast point

multiplication algorithms in order to determine which is the quickest. The point multiplication algorithm is defined as follows:

$$jM = \sum_{i=1}^{j} M \qquad \ldots(1.1)$$

In elliptic curve security operations, scalar point multiplication on the Elliptic curve is critical for performing efficient and quick calculations. An attempt is made to give a useful concept for the quickest algorithm point multiplication. The scalar point multiplication process plays an important and time-consuming part in ECC calculations, where the point addition procedure is repeated over the GF operation as shown in equation (1.1). On finite field GF(2^n), a random integer j chosen by the sender, and on Elliptic Curve (EC) M is the point. Elliptic Curve Cryptosystem (ECC) operations are known as elliptic curve group rules and regulations over the binary expansion fields GF(BF) (2^n) or prime fields GF(p). Based on their features, elliptic curves are classified into super and non-super singular curves. In general, the points on the curve can be created by addition of a point M on the elliptic curve, which is known as scalar multiplication with the same point added again. Because in ECC the point-multiplication (PM) is a prolonged process, the computational complexity of elliptic curve cryptographic methods is entirely dependent on the speed of point multiplication.

A classic approach for completing point multiplication, is Double-and-Add algorithm, which is also categorized as a multi-doubling and a window method. In general, window approaches such as discrete form adoption and altered Booth's encoding procedures designed to minimize the number of point additions (PA) in the PM procedure [3], where the required number of dual points stays as constant. On the other hand, the multi-doubling technique, is required to calculate $2^n M$ straight from M without intermediary points, where n is an integer. The goal of this study is to show a comparison of four ECC point multiplication techniques. To achieve this aim, first evaluate the data flow diagram to determine data dependence before doing the point multiplication jM. This paper investigated a concurrent, interpolated method in order to high-speed point multiplication which takes the benefits of the rest time caused by the dependent data in the earliest Double-and-Add technique.

II. Related works

Miller [32] and Koblitz [21] put forward the implementation of elliptical curves in cryptography. The work later with awesome follow-up extended in [14,22,20,1,2,3] and was approved by The National Institute of Standards and Technology (NIST). Among the early implementations are in mathematical analysis, data management system, etc. narrated in [9]. Later it has been used for other related security issues which is discussed in [16]. Smart mobiles, smart cards, firewalls, and e-cash systems are some more examples of devices that run ECC applications. m-BAT, a mobile app tool based on advanced ECC cryptosystem proposed for banking that run in a client–server environment, which connect users with the bank server via a customer portal able to run on the user's mobile phone in [25,5]. Many elliptic curve cryptographies (ECC)-based communication scheme has been developed for establishing a secure session key between IoT devices and a remote server in [18,31,24]. A protocol for key agreement based on hexadecimal enhanced ASCII Elliptic Curve Cryptography (ECC), authenticated by both parties is proposed which is pairing-free. The suggested protocol is designed very appropriate manner, which enhanced the security strength and is less expensive. The enhanced ASCII code representation of the user's identity

improves the protocol's security [17]. To prevent attacks from clone sensors in WSN and MWSN, the author examined an ECC property outlined into a fully earmarked IECC protocol. IECC mechanism is an efficient surveillance in terms of risk assessment, which emphasis on securing an industrial estate [31]. In [11] the author focused deficiencies in security there in previous algorithms and proposed an efficient RFID authentication method using Elliptic Curve Cryptography (ECC).

Many talks have focused on different ECC point multiplication fast multiplication methods without any comparison. In most situations, the true quickest and most efficient technique for point multiplication is emphasized in this work. Booth [7] introduces the fast point signed binary multiplication method. Except for the complementary recoding technique mentioned by Hankerson et al. [13], the fast point multiplication algorithms have examples. Yen et al. [15] created binary signed-digit recoding from left-to-right with a reference to the paper "Binary arithmetic, Advances in computers" [26]. Except for the complementary recording technique, Morain and Olivos[8] have described most of the rapid multiplication methods with some examples. Shankar et al. [29] compare Karatsuba multiplication and Complementary Recoding methods, as well as the evaluation procedure of two rapid algorithms Doubling and Addition and Addition-Subtraction approaches.

Following Balasubramaniam and Karthikeyan [4] and Chang et al.[8], our study affirms the facts given in these articles and justifies the quickest algorithm appropriate for point multiplication in elliptic curve cryptosystem methods. A fast parallel architecture put forward by author which implemented elliptic curve scalar multiplication on binary fields. To compute ECC, Authors suggested parallel algorithms for Elliptic Curve (EC) point multiplication, on Graphics Processing Units. The proposed method makes use of the Residue Number System (RNS) to retrieve parallelism from high accuracy integer arithmetic [28]. In [4] authors formulated an algorithm known as Integer Sub-Decomposition (ISD) which used GLV concept to calculate any number of multiplication jM of a point M of n resting on an elliptical curve.

III. The derivation of the elliptic curve

Curves that are made out of ellipses and formed by quadratic curves are commonly referred to as. Elliptic curves formation are generally cubic and had link with elliptic integrals, which may be used to calculate an ellipse's arc length. The "canonical form" of an elliptic curve is defined as

$$y^2 = x^3 + ax + b \qquad \ldots (3.1)$$

The discriminant D For $x^3 + ax + b$ can be as follows

$$D = 4a^3 + 27b^2 \neq 0 \qquad \ldots (3.2)$$

The polynomial equation for elliptic curve is $x^3 + ax + b$, the discriminant $D \neq 0$ has three different roots. More than two roots may be merged, when it founds zero valued D (discriminant), resulting in singular form curves. Because singular curves are simple to crack, they are not suitable for cryptography. As a result, non-singular curves are commonly assumed for data encryption.

Almost all elliptic curve cryptosystems use the Discrete Logarithm (DL) Problems for Elliptic Curve. The two points let be M and N on an elliptic curve such that $jM = N$, where j is a

scalar of appropriate size and represents the DL of N with the base M. Obtaining j from M and N is computationally impossible.

IV. Algorithms for Point Multiplication (PM)

The scalar multiplication in the Elliptic curve crypto system is defined as $N = jM$, where M and N are elliptic curve points and j is an integer. Repeated elliptic curve point addition and doubling procedures are used to accomplish this. Point negation may also be performed as a general operation, which can be modelled as a jM fast implementation technique.

Here the integer j is can be characterize as

$$j = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \ldots + x_1 + x_0$$

Where $x_i \in \{1,0\}$ and $n = 0,1,2, \ldots, m-1$

For adding both the points on a curve over F^{2n} is described by:

Let On the curves, there are two distinct points M (a_1, b_1), and N (a_2, b_2). When $M \neq Q$, the operation $M + N = (a_3, b_3)$ may be deduced in the Fig 1. (i).

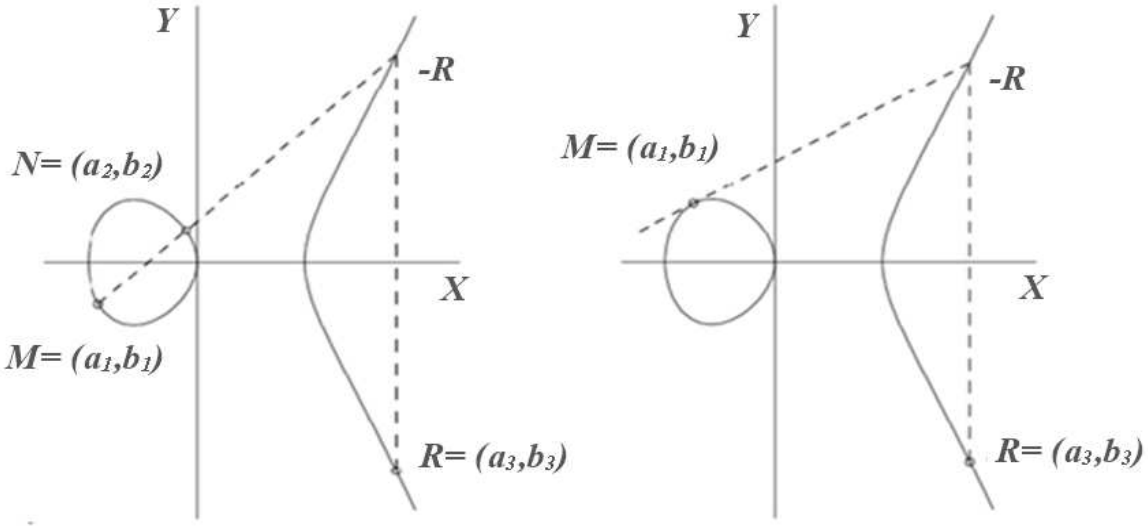

Fig 1 (*i*). *Point Addition M+N=R* ***(ii) Point Addition M+M=R***

Algorithm 1:

Function Point addition (M, N)

1. If $(M \neq N)$
2. Then $((a_1 \neq a_2)$ and $(b_1 \neq b_2))$
3. $\mu \leftarrow (b_1 + b_2) / (a_1 + a_2)$
4. $(a_3, b_3) \leftarrow (\mu^2 + \mu + a_1 + a_2 + c, \mu (a_1 + a_3) + a_3 + b_1)$
5. $M+N \leftarrow (a_3, b_3)$

Function Point double (M, N)

1. If $(M = N)$
2. then $((a_1 = a_2)$ and $(b_1 = b_2))$
3. $\mu \leftarrow a_1 + (b_1 / a_1)$

4. $(a_3, b_3) \leftarrow (\mu^2 + \mu + c, \mu\,(a_1+a_3) + a_3 + b_1)$
5. $2M \leftarrow (a_3, b_3)$

Observation From the above algorithm 1,

i. Function Point Addition, it is examined that PA require one Inverse(I) operation, two multiplication(P) operation, one square(S) operation with μ otherwise $(a_1 + a_2)$ never be zero. (S + 2P + I)

ii. Function Point double, it is noticed that PA requires 1 Inverse(I) operation, 2 multiplication(P) operation, 1 square(S) operation with μ otherwise a_1 never be zero. (S + 2P + I)

Field multiplication and division method is little complex in calculation as compared with field addition method. So, to ignore such type of complexity an algorithm known as Double and Add algorithm is introduced

---

Algorithm 2: Double and Add algorithm for computing *jM*

$j = \sum_{i=0}^{n-1} j_i 2^i$ $\qquad\qquad$ $j_i \in \{1,0\}$

1. $M \leftarrow M\,(a_1, b_1)$
2. $N \leftarrow \infty$
3. for $i \leftarrow n$-1 down to 0
   Begin
4. $\quad N \leftarrow N + N$
5. $\quad$ If $k_i \leftarrow 1$ then
6. $\quad\quad N \leftarrow M + N$
   End
7. return Q

In this representation, the multiplication cost is determined by the size of the binary form of j with the number of 1s. As an instance, if the depiction $(j_{n-1..}j_m\ , j_0)_2$ has $j_{n-1} \neq 0$ then it performs (n-1) number of doubling operations and the addition operations are the number of digits with non-zero values in $(j_{n-1..}j_m\ , j_0)_2$ minus 1. The number of non-zero digits is known as the Hamming weight of scalar representation. Hence it is clear that for an average point of view, the binary method involves with (n-1) /2 no of additions and (n-1) no of doubling operations.

A. The Addition and Subtraction Operation

Booth [7] developed a novel signed binary technique with scalar representation in 1951, and subsequently Rietweisner [26] demonstrated representation of each and every integer unique representation. Any two consecutive numbers, at most one, are non-zero, according to the representation. If $d \epsilon Z$, then there is one signed binary expansion of d, $j \epsilon \{1, 0, -1\}$ which is as follows:

$d = \sum_{i=0}^{n-1} j_i 2^i$, where each $x_i \in \{0,1\}$

In terms of $j_i\ j_{i+1} = 0$ is the representation of d in Non-Adjacent Form (NAF). j is one digit longer in the NAF form than in the radix 2 representation {0,1}. The Algorithm 3 is for

converting a right-to-left representation of k (an integer) of the same using three digits radix 2 represntation {0-1-1} into the NAF.

Algorithm 3: NAF Computation for Positive Integer

Input: +VE integer j
Output: NAF representation of j

1. $i \leftarrow 0$
2. While ($j > 0$)
3. *If* ($j/2 \neq 0$) then
4. $j_i \leftarrow 2 - j \% 2^2$
5. $j \leftarrow j - j_i$
6. Else $j_i \leftarrow 0$
7. *Endif*
8. $j \leftarrow j/2$
9. $i \leftarrow i + 1$
10. End While
11. Return j

The representation of 255 in binary = $(11111111)_2$
NAF of (255), as per the above mentioned algorithm = $(10000000\text{-}1)_{NFA}$

The computation of inverse of a point may be almost free in the elliptic curve group, which is a remarkable characteristic. This is why it's crucial to have a signed binary representation of the scalar. The binary technique is updated as a result, and the new algorithm is referred to as the addition and subtraction operation [7] in Algorithm 4.

Algorithm 4: Addition and Subtraction Operation

Input: j and M as +VE integer
Output: $N \leftarrow jM$

1. Calculate NAF(j) $\leftarrow \sum_{i=0}^{n-1} j_i 2^i$
2. $N \leftarrow \infty$
3. For $i \leftarrow n$ -1 down to 0
4. Begin
5. $N \leftarrow 2N$
6. *If* ($j_i = 1$)
7. $N \leftarrow N + M$
8. *If* ($j_i = -1$)
9. $N \leftarrow N - M$
10. End
11. Return Q

In binary form, the Hamming distance of 255 is 8, therefore 6 addition operations are required. In NAF representation, the Hamming distance of 255 is 2. It indicates that addition operations are 6 times faster than using binary representation. On average, this method produces (n-1)

doublings and (n – 1)/3 additions. The left-to-right evaluation step is a logical choice for elliptic curve scalar multiplication. The downside of the addition–subtraction technique is that the recoding and storage must be completed before the left-to-right assessment step can begin. As a result, the right-to-left exponent recoding necessitates the use of extra bit memory.

B. Mutual Opposite Form (MOF)

Okeya [23] In 2004, had introduced a novel efficient left-to-right recoding technique called as Mutual Opposite Form (MOF), and the property of n-bit mutual opposite form (MOF) with n-bit signed binary string fulfils the following properties:

1. The signs of adjacent non-zero bits (excluding zero bits) are the polar opposites.
2. Unless all bits are 0, the most non-zero bit and the least non-zero bit are 1 and 1, respectively. Between non-zero bits with mutually opposing signs, some zero bits are added. 0100101000100110 is an example of MOF. MOF may uniquely represent each positive integer, which is a crucial finding. The following theorem can help you convert a binary string to a MOF. Below is an example of a simple and flexible translation from an n-bit binary text to a (n + 1)-bit MOF.

The following remark is really important. The following is an example of converting an n-bit binary string x to a signed binary string using the formula $z = 2x \ominus x$, where '⊖' stands for a bitwise subtraction.

$2x = x_{n-1} \mid x_{n-2} \mid \ldots \mid x_{i-1} \mid \ldots \mid x_1 \mid x_0 \mid$
$x = \mid x_{n-1} \mid \ldots \mid x_i \mid \ldots \mid x_2 \mid x_1 \mid x_0$
$z = x_{n-1} \mid x_{n-2} - x_{n-1} \mid \ldots \mid x_{i-1} - x_i \mid \ldots \mid x_1 - x_2 \mid x_0 - x_1 \mid -x_0.$

Algorithm 5: Left to Right Generation from Binary to MOF

Input: a non-zero binary string $x \leftarrow \mid x_{n-1} \mid \ldots \mid x_i \mid \ldots \mid x_2 \mid x_1 \mid x_0$
Output: MOF $\mu_n \mid \ldots \mid \mu_1 \mid \mu_0$ of x

1. $\mu_n \leftarrow x_{n-1}$
2. for i ← n-1 down to 1
3. *begin*
4. $\mu_n \leftarrow 1, \ldots, n-1$ and
5. $\mu_n \leftarrow x_{i-1} - x_{i-1}$
6. $\mu_n \leftarrow - x_{i-1}$.
7. *end*
8. return $(\mu_n, \mu_{n-1}, \ldots, \mu_1, \mu_0)$

This technique is designed to speed up multiplication when the multiplier contains consecutive ones, and to give a multiplication approach that works for both signed and unsigned integers.

C. Complementary Recoding Method or 2's Complement Method

An effective technique Chang et al. [6] suggested complementary recoding to compute universal multiplication by conducting complement operations in particular instances.

Assuming that the scalar k has binary representation $(x_{n-1}, \ldots x_m, x_0)$ then
$j = \sum_{i=0}^{n-1} x_i 2^i = (1000\ldots00)_{(n+1)}$ bits $-\bar{j}-1$ ($\bar{j}$ is complement of j)
where $\bar{j} = \bar{x}_{n-1}, \ldots \bar{x}_m, \bar{x}_0$ ($\bar{x}$ is complement of x)

Algorithm 6: Use of Complementary recoding algorithm for Scalar Multiplication

Input: j and M
Output: $N \leftarrow jM$

1. $p \leftarrow (100\ldots00)_{n+1}$bits
2. $q \leftarrow \bar{j}$
3. $r \leftarrow p - q - 1$
4. $N \leftarrow$ O
5. For i ← n-1down to 0
6. Begin
7. $N \leftarrow 2N$
8. if (r_i =1)
9. $N \leftarrow N + M$
10. Else if (r_i = -1)
11. $N \leftarrow N - M$
12. Endif
13. End

Example

If $j = 6839 = (1101010110111)_2$
$= (10000000000000)_2 - (0010101001000)_2 - 1$
$= (100–10–10–100–100–1)_2 = (100–10–10–100–100–1)_{NAF}$

The binary represented Hamming weight of the above number = 9
The Hamming weight of signed binary complementary recoding operation = 6.
The addition operation has been saved for the case of three elliptic curves.
By this method, it will save around 3S+6P+3I as it saves S + 2P+I for one addition operation.

In an average, this algorithm performs $(n - 1)/3$ no of Additions and $(n\text{-}1)$ no of Doublings operations. The left-to-right assessment phase is a natural choice for the case of Elliptic Curve scalar multiplication.

Table 1 NAF Representation

Number of cases	Numbers(j)	Binary representation$(j)_2$	$_{NAF}$
1	29	11101	1000-101
2	131	10000011	1000010-1
3	515	1000000011	100000010-1
4	687	1010101111	10-10-10-1000-1
5	2927	101101101111	10-100-100-1000-1
6	4027	111110111011	100000-1000-10-1
7	4195	1000001100011	1000010-10010-1
8	48079	1011101111001111	10-1000-1000-101000-1

Table 2 Comparison of MOF with Complementary Recoding representation

Number of Cases	MOF	Complementary Recoding representation
1	100-11-1	1000-1-1
2	1-1000010-1	10-1-1-1-1-10-1
3	1-100000010-1	10-1-1-1-1-1-1-10-1
4	1-11-11-11000-1	10-10-10-1000-1
5	1-110-110-11000-1	10-100-100-1000-1
6	10000-1100-110-1	100000-1000-10-1
7	1-1000010-1001-1	10-1-1-1-1-100-1-1-10-1
8	1-1100-11000-101000-1	10-1000-10000-1-1000-1

From the above observations shown in table 1 and table 2, it is clear that the hamming weights in NAF method are less in comparison to others. In most of the cases NAF can be applied for addition-subtraction method in most efficient form.

Table 3 Run Time Analysis of Different Algorithms

Algorithms	**Worst case**	**Average case**	**Best case**
Double and Add algorithm	n(add)+n(double) (If all $(j)_2$ are 1s)	n/2(add)+n(double)	n/2(add)+n(double)
MOF	n/2(add)+n(double)	n/2(add)+n(double)	n/3(add)+n(double)
Complementary Recoding representation	n/2(add)+n(double)	n/2(add)+n(double)	n/3(add)+n(double)
Addition Subtraction	n/2(add)+n(double)	n/3(add)+n(double)	n/3(add)+n(double)

Notes: where n = $(j)_2$ or binary representation of j.

By considering Table 3 it is observed that Addition Subtraction method is the most efficient one.

V. Comparison of different methods

Double and Add, Addition Subtraction, MOF, and Complementary Recording Methods are compared. Various point-multiplication methods cause different encryption and decryption times (ms). Using an Intel Core 2 Duo CPU with 3 GB RAM and a 1.80 GHz processing speed, the run time of the evaluated algorithms was measured in milliseconds (ms) as shown in table 4 and table 5.

Table- 4 Encryption Timing for various scalar multiplication algorithms (in ms)

Block Size	Double and Add	MOF	Complementary Recoding	Addition-subtraction
160	5.74359	5.05645	4.59849	4.31213
200	6.51164	5.78066	5.13774	4.76585
250	8.82456	7.87893	6.19058	5.44693
300	12.79876	11.3953	7.83431	6.5742

350	16.60246	15.15613	11.76631	10.79074

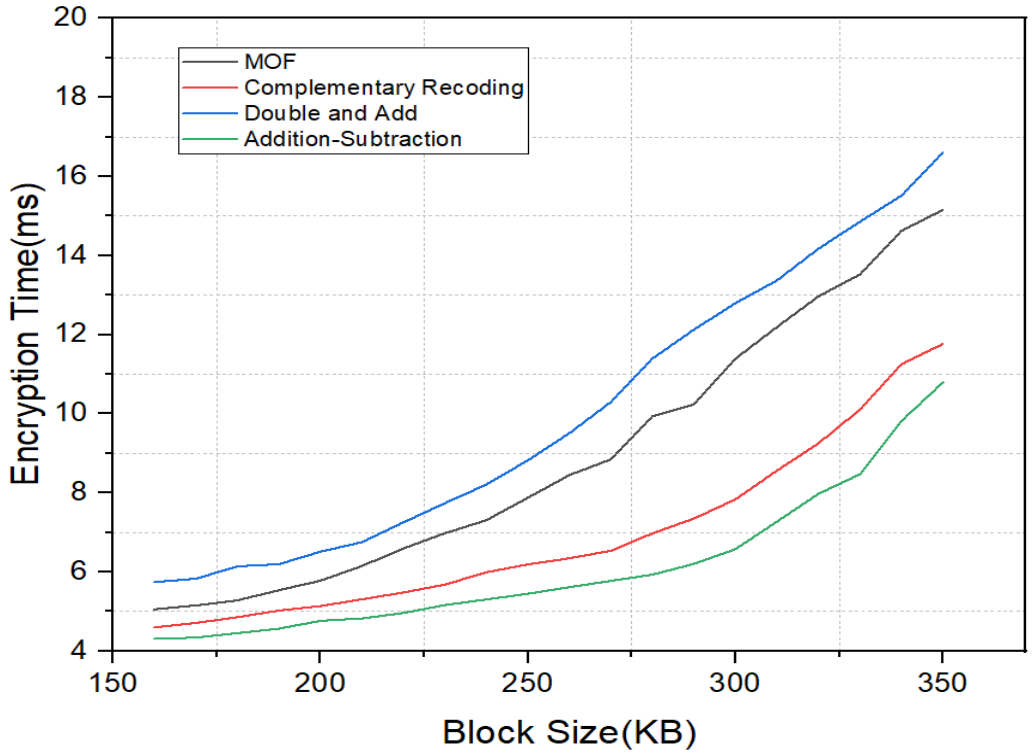

Fig 2 Block Size Vs. Encryption Time

Table- 5 Decryption Timing for various scalar multiplication algorithms (in ms)

Block Size	Double and Add	MOF	Complementary Recoding	Addition-subtraction
160	5.52743	5.3521	5.25452	5.17647
200	6.85208	6.41964	5.71505	5.4999
250	8.95709	8.24713	6.95967	6.51633
300	10.73028	10.02104	8.38191	7.77404
350	13.25459	12.87886	10.29114	9.33531

Figure 2 and 3 shows the comparison of different elliptic curve algorithm based on time taken to encrypt and decrypt files with different block size. Experiment result proves that Addition – Subtraction algorithm in elliptic Curve cryptosystem runs faster Double and Add and MOF algorithm. Complementary Recoding runs closer with Addition – Subtraction algorithm. Thus, Addition-Subtraction algorithm is an efficient algorithm compared to other.

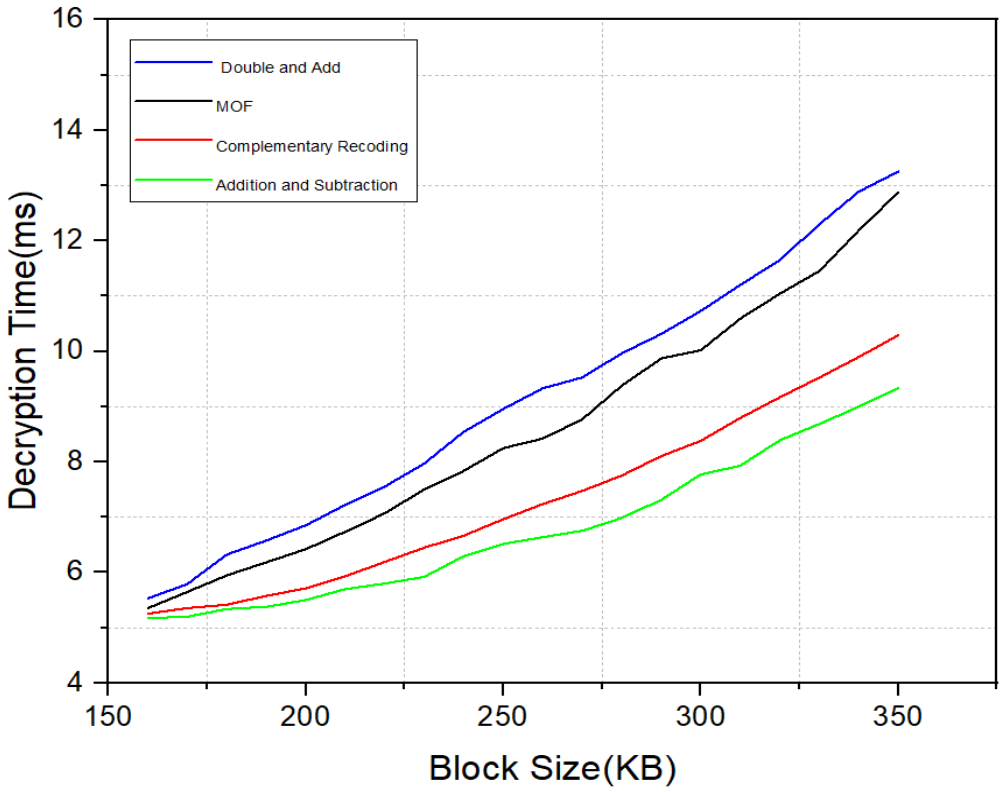

Fig 3 Block Size Vs. Decryption Time

VI. Conclusion

Many techniques introduced computational methods of the general multiplication operation on the Elliptic curve cryptosystem for common-multiplicand multiplication. Learning the best approach for point multiplication is challenging. This paper presents a brief overview of elliptic curve cryptosystems with point multiplication using methods as Double and Add, Addition Subtraction, MOF, and complementary recoding, as well as a comparative study to determine which of these algorithms is the fastest and most suitable for point multiplication implementation. Even in a single example of a presentation, the Addition - Subtraction technique outperforms complimentary recoding.

Declarations

Funding: Not applicable

Conflicts of interest/ Competing interests: The authors declare that they have no conflicts of interest in this work.

Availability of data and material: Not applicable

Code availability: Not applicable

Authors' contributions: The study's idea, design, and analysis were all contributed by all authors, and the final article was reviewed and approved by them all.

REFERENCES

[1] A. Lenstra, and E. Verheul, ``Selecting cryptographic key sizes", Third International Workshop on Practice and Theory in Public Key Cryptography-PKC, LNCS 1751, 2000.

[2] A.J. Menezes, ``Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers, 1993.

[3] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, ``Handbook of Applied Cryptography", CRC Press, 1997.

[4] Ajeena, Ruma Kareem K., and Hailiza Kamarulhaili. "Point multiplication using integer sub-decomposition for elliptic curve cryptography." *Applied Mathematics & Information Sciences* 8.2 (2014): 517.

[5] Amol Dabholkar and Kin Choong Yow. 2004. Efficient Implementation of Elliptic Curve Cryptography (ECC) for Personal Digital Assistants (PDAs). Wirel. Pers. Commun. 29, 3–4 (June 2004), 233–246. DOI:https://doi.org/10.1023/B:WIRE.0000047066.74117.86

[6] Balasubramaniam, P. and Karthikeyan, E. (2007) 'Elliptic curve scalar multiplication algorithm using complementary recoding', Applied Mathematics and Computation, Vol.190, No.1, pp.51-56.

[7] Booth A.D., A signed binary multiplication technique, Journal of Applied Mathematics 4 (2) (1951) 236–240.

[8] Chang C. C., Kuo Y.T., and Lin C.H., Fast algorithms for common multiplicand multiplication and exponentiation by performing complements, in: Proceeding of 17th International Conference on Advanced Information Networking and Applications, March, 2003, pp. 807–811.

[9] G.B. Agnew, R.C. Mullin, and S.A. Vanstone, ``An Implementation of elliptic curve cryptosystem over , IEEE journal on selected areas in communication, Vol. 11, No. 5, 1993, pp. 804-813.

[10] Gong G and Lam C 2002 Linear recursive sequences over elliptic curves. In: Sequences and their applications. (London: Springer) pp 182–196

[11] Han Shen, Jian Shen, Muhammad Khurram Khan, and Jong-Hyouk Lee. 2017. Efficient RFID Authentication Using Elliptic Curve Cryptography for the Internet of Things. Wirel. Pers. Commun. 96, 4 (October 2017), 5253–5266. DOI:https://doi.org/10.1007/s11277-016-3739-1

[12] Hankerson D, Vanstone S and Menezes A 2004 Guide to Elliptic Curve Cryptography (Springer Verlag/New York)

[13] Hankerson D., A. Menezes, and S. Vanstone, "Guide to Elliptic Curve Cryptography", 2004 Springer-Verlag New York.

[14] I. Blake, G. Seroussi, and N. Smart, ``Elliptic Curves in Cryptography", Cambridge University Press, 1999.

[15] Joye M., Yen S., Optimal left-to-right binary signed digit recoding, IEEE Transactions on Computers 49 (2000) 740–748.

[16] K.D. Edoh, and H. Wahab, ``Multicast security: Issues and new schemes for key management", Information Resource Management Association International Conference, 2003, pp. 123-126.

[17] Kumar, V., Ray, S., Dasgupta, M. *et al.* A Pairing Free Identity Based Two Party Authenticated Key Agreement Protocol Using Hexadecimal Extended ASCII Elliptic

Curve Cryptography. *Wireless Pers Commun* 118, 3045–3061 (2021). https://doi.org/10.1007/s11277-021-08168-x

[18] Majumder, S., Ray, S., Sadhukhan, D. *et al.* ECC-CoAP: Elliptic Curve Cryptography Based Constraint Application Protocol for Internet of Things. *Wireless Pers Commun* 116, 1867–1896 (2021). https://doi.org/10.1007/s11277-020-07769-2

[19] Morain F., Olivos J., Speeding up the computations on an elliptic curve using addition–subtraction chains, RAIRO Theoretical Informatics and Applications 24 (1990) 531–543.

[20] N. Koblitz, ``CM-curves with good cryptographic properties'', Advances in Cryptology-CRYPTO '91, LNCS 576, Springer-Verlag, New York, 1992, pp. 279-287.

[21] N. Koblitz, ``Elliptic curve cryptosystem'', Mathematics of Computation, Vol. 48, 1987, pp. 203-209.

[22] N. Koblitz, A.J. Menezes, and S.A. Vanstone, ``The state of elliptic curve cryptography'', Vol. 19, Issue 2-3, 2000, pp. 173-193.

[23] Okeya, K., Signed binary representations revisited, Proceedings of CRYPTO'04 (2004) 123–139.

[24] Pradeep, S., Muthurajkumar, S., Ganapathy, S. *et al.* A Matrix Translation and Elliptic Curve Based Cryptosystem for Secured Data Communications in WSNs. *Wireless Pers Commun* 119, 489–508 (2021). https://doi.org/10.1007/s11277-021-08221-9

[25] Ray, S., Biswas, G.P. & Dasgupta, M. Secure Multi-Purpose Mobile-Banking Using Elliptic Curve Cryptography. *Wireless Pers Commun* 90, 1331–1354 (2016). https://doi.org/10.1007/s11277-016-3393-7

[26] Reitwiesner G.W., Binary arithmetic, Advances in computers 1 (1960) 231–308.

[27] Rodríguez-henríquez, Francisco & Saqib, Nazar Abbas & Díaz-Pérez, Arturo. (2004). A fast parallel implementation of elliptic curve point multiplication over GF(2m). Microprocessors and Microsystems. 28. 329-339. 10.1016/j.micpro.2004.03.003.

[28] Samuel Ant˜ao, Jean-Claude Bajard, Leonel Sousa. Elliptic Curve point multiplication on GPUs. ASAP 2010 — 21st IEEE International Conference on Application-specific Systems, Architectures and Processors, Jul 2010, Rennes, France. pp.192 - 199, 2010,

[29] Shankar T.N., G.Sahoo, S.Niranjan "Cryptography with fast point multiplication by using ASCII codes and its implementation", IJCNDS, Vol. 10, No. 3, pp.258–279.

[30] Shankar T.N., G.Sahoo, S.Niranjan "Elliptic Curve Point Multiplication by Using Complementary Recording for Image Encryption", INCOCCI 2010, ieee Xplore, pp.*546-551*.

[31] Sujihelen, L., Jayakumar, C. Inclusive Elliptical Curve Cryptography (IECC) for Wireless Sensor Network Efficient Operations. *Wireless Pers Commun* 99, 893–914 (2018). https://doi.org/10.1007/s11277-017-5157-4

[32] V. Miller, ``Use of elliptic curves in cryptography''. Advances in Cryptography-Crypto '85, Springer-Verlag New York, LNCS 218, 1986, pp. 417-426.