NEW REPRESENTATION METHOD FOR INTEGERS AND ITS APPLICATION ON ELLIPTIC CURVE CRYPTOGRAPHY

ARASH EGHDAMIAN

UNIVERSITI SAINS MALAYSIA

2020

NEW REPRESENTATION METHOD FOR INTEGERS AND ITS APPLICATION ON ELLIPTIC CURVE CRYPTOGRAPHY

by

ARASH EGHDAMIAN

Thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy

January 2020

ACKNOWLEDGEMENT

First and foremost, I would like to express my deepest gratitude to my supervisor Professor Dr Azman Samsudin for his invaluable support, guidance and patience during all my PhD studies. I never gave up because he never gave up on me. My thanks also to the Dean and all staff members of the School of Computer Sciences, USM, for making all these possible. A very special thank you is required to all my friends and colleagues in Information Security Lab and USM, which includes (but not limited to) Dr Shahram Jahani, Dr Fatemeh Shafaie, Dr Gelareh Abooali, Dr Amir Akhavan, Dr Farhad Fakhrian, Dr Behzad Foroughi and Dr Ali vafaei-Zadeh. I also need to thank my parents Iraj and Farideh and my only brother Ashkan for their endless support over the years of my journey. Without their support, this was not possible. Finally, I want to dedicate this work to my wife Sahar for her unconditional love, support and encouragement.

TABLE OF CONTENTS

ACKNOWLEDGEMENT ii

TABLE OF CONTENTS iii

LIST OF TABLES viii

LIST OF FIGURES x

LIST OF ABBREVIATIONS xii

ABSTRAK xiv

ABSTRACT xvi

CHAPTER ONE: INTRODUCTION 1

1.1 General Overview 1

1.2 Problem Statement 6

1.3 Research Motivation 7

1.4 Research Objectives 8

1.5 Research Scope 9

1.6 Research Methodology 10

1.7 Research Contributions 13

1.8 Thesis Outline 14

CHAPTER TWO: LITERATURE REVIEW 15

2.1 Introduction 15

2.2 Number Systems 16

2.2.1 Positional Number Systems 17

2.2.1(a) Binary Number System 19

2.2.1(b) Signed-Binary Number System 20

2.2.2 Double Base Number System (DBNS) 25

2.2.3 Multi-base Number System (MBNS) 27

2.2.4 Window Sliding Method 28

2.3 Public-key Cryptography 30

2.3.1 Diffie-Hellman Key Exchange 30

2.3.2 RSA 32

2.3.3 ElGamal Encryption Algorithm 34

2.4 Elliptic Curve Cryptography (ECC) 35

2.4.1 Arithmetic Underlying Field Operations of Elliptic Curves .. 38

2.4.2 Elliptic Curve Operations 38

2.4.2(a) Basic Elliptic Curve Operations 39

2.4.2(b) A Line Through Two Distinct Points 40

2.4.2(c) A Tangent Line of Elliptic Curve 42

2.4.2(d) Point Addition on Elliptic Curve 44

2.4.2(e) Point Doubling on Elliptic Curve 46

2.4.3 Composite Elliptic Curve Operations 48

2.5 Point Representation 49

2.6 Point Multiplication 50

2.6.1 On-the-Fly Elliptic Curve Multiplication 52

2.6.1(a) Unsigned Standard Binary Method 53

2.6.1(b) Signed Binary Schemes 54

2.6.2 Window-Based Elliptic Curve Multiplication 61

2.6.2(a) The m-ary Method 63

2.6.2(b) Sliding Window Non-Adjacent Form 64

2.6.2(c) Non-Sparse Signed Window Method 64

2.6.2(d) Window Non-Adjacent Form 66

2.6.2(e) Fractional Window Method 68

2.6.2(f) Mutual Opposite Form Window Method.......... 68

2.6.2(g) Avanzi Method... 69

2.6.2(h) Khabbazian Method 69

2.6.2(i) Muir Method 70

2.7 Summary 70

CHAPTER THREE: METHODS AND IMPLEMENTATION.................. 73

3.1 Introduction 73

3.2 Methods.. 73

3.2.1 NAF (Non-Adjacent Form) Method 75

3.2.2 GNAF (Generalized NAF) Method 76

3.2.3 MGNAF (Modified GNAF) Method 78

3.2.4 AGNAF (Alternative GNAF) Method................................... 80

3.2.5 Left-to-Right MGNAF Method 82

3.2.6 MGSDNAF (Modified Generalized Signed Digit NAF) Method 84

3.2.7 ECC Multiplication Based on Right-to-Left MGSDNAF 86

3.2.8 Left-to-Right MGSDNAF Method 87

3.2.9 ECC Multiplication Based on Left-to-Right MGSDNAF 88

3.3 Implementation 89

3.3.1 Proving Correctness of the Proposed Algorithms.................. 91

3.3.1(a) New Recoding Methods.................................... 92

3.3.1(b) New Single Scalar Multiplication Methods...... 93

3.3.2 Analysing an Algorithm.. 94

3.3.2(a) Efficiency of an Algorithm 94

3.3.2(b) Generality of an Algorithm............................... 95

3.3.3 Evaluation Metrics 95

3.3.3(a) Hamming Weight... 95

3.3.3(b) Recoding Direction 96

3.3.3(c) Number of Operations....................................... 96

3.3.4 Coding an Algorithm 97

3.3.5 Implementation Environment and Parameters....................... 98

3.4 Summary 101

CHAPTER FOUR: RESULTS AND ANALYSIS...................................... 103

4.1 Introduction.. 103

4.2 MGNAF 103

4.3 AGNAF... 108

4.4 Left-to-Right MGNAF 111

4.5 MGSDNAF 113

4.6 Left-to-Right MGSDNAF 117

4.7 The Application of Representation Methods on ECC 120

4.7.1 Frequency of Each Digit 121

4.7.2 Calculating the Number of Operations 122

4.7.2(a) Scalar Multiplication Based on MGSDNAF .. 123

4.7.2(b) Scalar Multiplication Based on Left-to-Right MGSDNAF 125

4.8 Summary 126

CHAPTER FIVE: CONCLUSION........ 128

5.1 Introduction 128

5.2 Research Contribution Summary 128

5.2.1 Proposing Right-to-Left Recoding Methods 129

5.2.2 Proposing Left-to-Right Recoding Methods 130

5.2.3 Proposing Single Scalar EC Multiplication Methods 131

5.2.4 Research Contributions Summary 131

5.3 Research Limitations 133

5.4 Recommendations for Future Research 134

REFERENCES 135

LIST OF PUBLICATIONS

LIST OF TABLES

Page

Table 1-1 Types of EC Operations 4

Table 2-1 The Number of Different DBNS Representation of Integers 25

Table 2-2 {2,3}-Integers Terms of Numbers in CDBNS Representation 26

Table 2-3 Number of Multi-Base Representation of Small Numbers Using Various Bases 27

Table 2-4 Key Size for ECC and RSA 36

Table 2-5 Types of EC Operations 38

Table 2-6 Types for EC Point Multiplication Based on Knowledge of the Key and the Exponent 51

Table 3-1 Systems Information 98

Table 4-1 Comparison of GNAF and the Proposed MGNAF 106

Table 4-2 Digit Set Size Comparison of GNAF and the Proposed MGNAF 108

Table 4-3 Extra Digits in Digit Sets of AGNAF and MGNAF 110

Table 4-4 Hamming Weight Comparison of GNAF and the Proposed AGNAF 110

Table 4-5 Hamming Weight Comparison of GNAF and the Proposed method 116

Table 4-6 The Frequency (Percent of Appearance) of Each Digit Related to the Total Number of Digits 121

Table 4-7 Number of Inversions [*i*], Squarings [*s*] and Multiplications [*m*], for Different Curve Operations Over F_2^m Using Affine Coordinates 122

Table 4-8 Number of Curve and Field Operations for 100-bit Example in Binary Field (Normal vs R2L MGSDNAF) 124

Table 4-9 Number of Curve and Field Operations for 100-bit Example in Binary Field (GNAF vs R2L MGSDNAF) 124

Table 4-10 Number of Curve and Field Operations for 100-bit Example in Binary Field (Normal vs L2R MGSDNAF) 125

Table 4-11 Number of Curve and Field Operations for 100-bit Example in Binary Field (GNAF vs L2R MGSDNAF) 126

Table 5-1 Research Contributions Summary 132

LIST OF FIGURES

		Page
Figure 1-1	Categorisation of Cryptographic Primitives	2
Figure 1-2	Parameters of EC Multiplication Algorithms	11
Figure 1-3	Research Methodology Steps	12
Figure 2-1	Diffie-Hellman Key Exchange Algorithm	31
Figure 2-2	ElGamal Cryptosystem	35
Figure 2-3	An Example of an Elliptic Curve in Real Numbers System	39
Figure 2-4	Special Point on Elliptic Curve	40
Figure 2-5	Intersection Points on Elliptic Curve	41
Figure 2-6	Tangent Line on Elliptic Curve	43
Figure 2-7	Point Addition on Elliptic Curve	44
Figure 2-8	Point Doubling on Elliptic Curve	47
Figure 2-9	Types of Elliptic Curve Point Multiplication	50
Figure 2-10	On-the-fly EC Schemes	52
Figure 2-11	Binary and Window Representation of an Integer ***k***	62
Figure 2-12	Binary Number Divided to Blocks of 4	63
Figure 2-13	Sliding Window Technique Over NAF	64
Figure 2-14	wNAF Examples	67
Figure 2-15	Number Systems Covered in Chapter 2	71
Figure 3-1	Process of Introducing New Methods	74
Figure 3-2	All 5-bit Signed Digit Representations of 13	77
Figure 3-3	Checking the Correctness of New Recoding Methods	92
Figure 3-4	Checking the Correctness of New Single Scalar Multiplication Methods	94
Figure 4-1	MGNAF Over GNAF Improvement	107

Figure 4-2 Hamming Weight Percentage Comparison of MGNAF and MGSDNAF 117

Figure 5-1 Hamming Weight of Recoding Methods 130

LIST OF ABBREVIATIONS

AGNAF	Alternative for Generalized Non-Adjacent Form
BIN	Binary
CR	Complementary Recoding
DA	Double-and-Add
DBNS	Double-Base Number System
DES	Data Encryption Standard
DH	Diffie-Hellman
DLP	Discrete Logarithm Problem
DS	Digital Signature
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECADD	Elliptic Curve Addition
ECC	Elliptic Curve Cryptography
ECDBL	Elliptic Curve Doubling
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
GNAF	Generalized Non-Adjacent Form
KH	Khabbazian
KTNS	Koyama and Tsuruoka Non-Sparse
L2R	Left-to-Right
MGNAF	Modified Generalized Non-Adjacent Form
MGSDNAF	Modified Generalized Signed-Digit Non-Adjacent Form

MOF	Mutual Opposite Form
NAF	Non-Adjacent Form
PNR	Position Number System
QA	Quadruple-and-Add
QPL	Quadrupling
R2L	Right-to-Left
RSA	Rivest-Shamir-Aldeman
TA	Triple-and-Add
TPL	Tripling

KAEDAH PERWAKILAN BAHARU UNTUK INTEGER DAN APLIKASINYA PADA KRIPTOGRAFI KELUK ELIPTIK

ABSTRAK

Sistem kriptografi kunci awam digunakan secara berleluasa di dalam protokol keselamatan sepert perjanjian, pengesahan, penyulitan dan lain-lain lagi. Dua operasi penting pada kebanyakan algoritma kunci awam adalah pendaraban and exponensi nombor majmuk/besar. Prestasi dan kecekapan yang dipamerkan oleh primitif kriptografi ini sangat bergantung kepada kecekapan operasi-operasi yang terlibat. Peningkatan pada kecekapan operasi pendaraban dan exponensi dengan menggunakan methodologi pengekodan-semula ataupun dengan menggunakan sistem nombor yang khusus di mana ia boleh mengurangkan "Hamming weight" adalah sesuatu yang lazim dilakukan. Kaji selidik ini mencadangkan satu perwakilan baru untuk integer-integer di dalam Radix-x yang dikenali sebagai "Modified Generalized Non-Adjacent Form". Perwakilan ini adalah versi yang dipertingkatkan daripada "Generalized Non-Adjacent Form". Kaedah yang dicadangkan telah diperbaiki dan menghasilkan dua kaedah baru di dalam pengekodan-semula "Right-To-Left" iaitu "AGNAF" dan "MGSDNAF", dan dua kaedah baru di dalam pengekodan-semula "Left-To-Right" iaitu "Left-To-Right MGNAF" dan "Left-To-Right MGSDNAF". Kedua-dua kaedah "Left-to-Right" mempunyai satu atribut yang baik di mana pengiraan dibuat dari arah kiri ke kanan (contohnya dari nilai paling bererti ke nilai yang kurang bererti) berbanding dengan "GNAF". Hala pemprosesan ini adalah sangat penting kerana untuk sebilangan sistem kripto, pengiraan hanya boleh dibuat dari arah kiri ke kanan. Kelebihan seterusnya adalah pengekodan-semula eksponen tidak perlu dilakukan terlebih dahulu. Maka, kaedah "Left-to-Right" menghasilkan prestasi yang lebih baik di dalam aspek penggunaan memori dan masa pelaksanaan. Akhirnya, kedua-dua kaedah ini ("MGSDNAF" dan "Left-to-Right MGSDNAF") telah dipilih untuk digunakan pada pendaraban skalar tunggal Kriptografi Keluk Eliptik untuk memperbaiki kecekapan pelaksanaan. Versi terakhir di dalam kaedah perwakilan yang dicadangkan boleh mengurangkan "Hamming Weight" integer dari 21.6% untuk Radix 3 ke 15.1% untuk Radix 9. Untuk pengekodan-semula "GNAF", angka-angka ini adalah 16.7% dan 8.9% masing-masing. Selain itu, perbandingan di antara dua kaedah pendaraban Radix 3 skalar tunggal Kriptografi Keluk Eliptik (di mana ia berdasarkan kepada "GNAF" dan "Left-to-Right MGSDNAF") menunjukkan bahawa "GNAF" boleh mengurangkan bilangan operasi sebanyak 11.5%, di mana ini adalah 14.1% untuk kaedah yang telah dicadangkan.

NEW REPRESENTATION METHOD FOR INTEGERS AND ITS APPLICATION ON ELLIPTIC CURVE CRYPTOGRAPHY

ABSTRACT

Public-key cryptosystems are broadly used in security protocols such as key agreement, authentication, encryption and others. The two main operations in many public-key algorithms are multiplication and exponentiation of large numbers. The performance and efficiency of these cryptographic primitives are highly reliant on the efficiency of these operations. Improving the efficiency of multiplication and exponentiation by applying a recoding method or using a specific number system which can reduction the Hamming Weight of numbers is very common. This study proposes a new Radix-*r* representation for integers which is known as Modified Generalized Non-Adjacent Form (MGNAF). This representation is the enhanced version of Generalized Non-Adjacent Form (GNAF). The proposed method has been improved and resulted in two Right-to-Left recoding methods, AGNAF and MGSDNAF, and two Left-to-Right recoding methods: Left-to-Right MGNAF and Left-to-Right MGSDNAF. The two Left-to-Right methods, unlike the GNAF, has a nice property in which the calculation is performed from the left to the right (i.e., from the most significant digit to the least significant one). This processing direction is of great importance since for some cryptosystems the calculation can only be performed in Left-to-Right manner. A subsequent advantage is that the exponent does not need to be re-coded in advance. Hence, the Left-to-Right method resulted in better performances in both the running time and memory utilization. Finally, two of these methods (MGSDNAF and Left-to-Right MGSDNAF) were chosen based on their features to improve the efficiency of single scalar multiplication for Elliptic Curve

Cryptography. The final version of the proposed representation method can reduce the Hamming Weight of integers from nearly 21.6% for Radix 3 to 15.1% for Radix 9. For GNAF recoding, these numbers are 16.7% and 8.9% respectively. Moreover, a comparison between two Radix-3 single scalar multiplication methods for Elliptic Curve Cryptography (which are based on GNAF and Left-to-Right MGSDNAF) shows that the GNAF can reduce the number of operations by 11.5% where it is 14.1% for the proposed method.

CHAPTER ONE: INTRODUCTION

1.1 General Overview

Computer science has unceasingly developed very fast over the past few years. This advance has inspired human to look for greater data efficiency and convenience. As the result of this tendency, cryptography, which is a branch of computer science providing security over data, has become a part of people’s daily life. Cryptography deals with many aspects of humans’ modern life such as secure communications, financial transactions, education, healthcare, etc. This marvellous advance of information technology has led to great attention to information security more than before (Menezes et al., 1997; Mogollon, 2008). According to (Menezes et al., 1997; Mogollon, 2008), the major functions of cryptography in the information security are related to achievement of Confidentiality, Data integrity, Authentication, and Non-repudiation. In order to reach these aims, it was attempted to design some cryptographic primitives which are listed as follows and detailed out in Figure 1-1.

Unkeyed primitives: These primitives are not based on any keys. There are two main primitives in this class, namely, Hash functions and random sequence generators.

Symmetric-key primitives: In this class, two parties share a single key which is called secret key. The functions of these primitives are encrypting a message, authenticating sender and data integrity. Some of the most famous symmetric ciphers are DES (Mogollon, 2008), AES (Stallings, 2016) and RC5 (Tilborg & Jajodia, 2011).

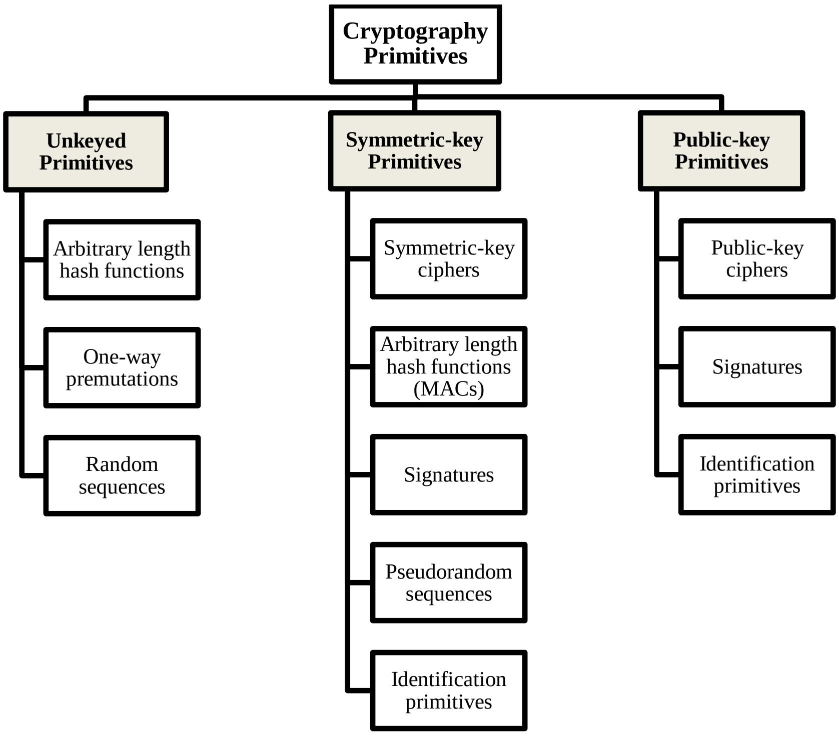

Figure 1-1: Categorisation of Cryptographic Primitives (Menezes et al., 1997)

Asymmetric-key (Public-key) primitives: Asymmetric-key primitive is also identified as public-key, uses two mathematically related keys, namely, public key and private key. Extracting one key with just knowing the other key is basically impracticable. However, public-key primitives assist the users to transfer information over insecure channels whereas, in the symmetric key schemes, the key must be only transferred through a secure channel.

All the four cryptographic goals mentioned above (confidentiality, data integrity, authentication, non-repudiation) could be achieved by using Public-key cryptosystems but the symmetric key schemes do not include the non-repudiation characteristics. It happens because in public-key cryptosystems, there is exclusive private key for each user that is not shared with other users.

Public-key cryptosystems are used for applications which deal with small messages. In addition, digital signature and key exchange schemes are examples of the mentioned applications. The reason behind it is that Public-key cryptosystems are slower than symmetric key cryptosystems (Henri Cohen et al., 2016). The operations used in the current defector Public-key cryptosystems are algebraic operations whereas the operations in the symmetric key are logical operations which perform more quickly on computer (Mollin, 2002). Actually, symmetric and asymmetric cryptosystems perform as each other's complement so that a combination of them can meet all of the cryptographic goals (Stallings, 2016).

During the recent decades, number theory and Public-key cryptosystem have become intertwined more and more. Diffie and Hellman introduced the first key exchange scheme in 1967 based on modular exponentiation (1976). A few years later, in 1978 one the most used Public-key cryptosystem, called RSA, was introduced by Rivest, Shamir, and Adleman (1978). The main operation in RSA cryptosystem is also based on the modular exponentiation. ElGamal key exchange (Elgamal, 1985) is another example of Public-key cryptosystems, which has been developed based on the modular exponentiation. Consequently, more efforts in Public-key cryptography have been dedicated to find algebraic operations or one-way functions that meet the specifications of Public-key cryptography (Menezes et al., 1997).

Elliptic Curve Cryptography (ECC), proposed by Koblitz and Miller in 1985, is also another Public-key cryptosystem which is based on the Elliptic Curve discrete logarithm problem (ECDLP) (Hankerson et al., 2013). ECDLP complexity is considered exponential whereas the complexity of factorization and discrete logarithm problems are considered sub-exponential (Eisentrager et al., 2003). ECC seems attractive because it has many features such as performance and security. Therefore, these features should be given attention once designing an encryption system (Hankerson et al., 2013). The security offered by ECC is equivalent with the security offered by RSA but it has smaller key size and lower processing power. For example, in EC cryptosystems a 160-bit key size is equal to RSA with 1024-bit key size (Stallings, 2016). Portable devices, such as personal digital assistants (PDA), mobile phones, and smart cards, are provided with limited memory. Therefore, amongst other Public-key cryptosystems, ECC is more appropriate for such devices (Tsaur & Chou, 2005).

Table 1-1: Types of EC Operations

EC Operation type	**Operation name**	**Math representation**	**Denoted by**
Basic	Point Addition	P + Q	A
	Doubling	2P	D
Composite	Double-and-add	2P + Q	DA
	Tripling	3P	TPL
	Triple-and-add	3P + Q	TA
	Quadrupling	4P	QPL
	Quadruple-and-add	4P + Q	QA

In ECC, curve operations are performed over a finite field. The effectiveness of underlying field operations plays an essential role in designing ECC algorithms. ECC is usually defined over binary, prime and extension fields (Hankerson et al., 2013) while in case of hardware implementations, binary fields have privilege over prime fields (Stallings, 2016). In order to compute kP, where $P \in E(F_{2^m})$ and $k \in I$, many operations can be implemented. There are two types of Elliptic Curve (EC) operations namely, basic and composite (or extended). Some of them are listed in Table 1-1.

There are several methods which contribute in speeding up EC scalar multiplication. The first method is by inventing recoding methods that decrease the number of additions by reducing the Hamming Weight of the re-coded key $w(k)$. Window recoding methods are known as a generalized technique for the recoding methods, for instance, NAF and w-NAF (window NAF) (Blake et al., 2005; Koblitz, 1991; Koyama & Tsuruoka, 1992; Solinas, 2000), MOF and w-MOF (Okeya et al., 2004) and w-Fractional window methods (Moller, 2002; Schmidt-Samoa et al., 2006). The second method is by reducing the complexity of Elliptic Curve operations by:

I. Enhancing the basic operations and inventing new fast composite methods (Ciet et al., 2006; Eisentrager et al., 2003),

II. Enhancing or inventing Elliptic Curve operations that use coordinate systems other than Affine coordinates or using mixed coordinates (P Mishra & Dimitrov, 2007; Sakai & Sakurai, 2001).

Using the enhanced pre-computation techniques is the third method for increasing the speed of EC computations (Chen et al., 1996; Dahmen et al., 2007; Lim & Hwang, 1999; Lim & Lee, 1994; Longa & Miri, 2008b) and the fourth method is by using customized hardware (Jiahong et al., 2009; MuthuKumar & Jeevananthan, 2010; Šimka et al., 2005). EC operation also can be speeded up by software optimizations and combining solutions together. The efficiency of EC single scalar multiplication can also be enhanced by customizing software implementation and combining the optimal methods to solve EC and EC related problems.

1.2 Problem Statement

Every second, huge amounts of data can be transmitted through computer networks. Since most of the data are private, they should be kept safe and concealed. Using cryptography provides the cheapest solution for securing these data while being either transferred or stored. According to (Tilborg & Jajodia, 2011), private key cryptosystems are used to encrypt/decrypt large amounts of data due to their high speed compared to Public-key cryptosystems, but Public-key cryptosystems are still commonly used for key exchange and digital signatures. Since Public-key cryptosystems depend on large number operation, they need to use complex mathematical computations. These kinds of systems are normally used for key exchange and digital signatures. Consequently, there is always high tendency toward finding cheaper, faster and more effective solutions for public keys.

One of these solutions is using signed binary representations. The main reason of using signed binary representation is to increase the speed of the Classical multiplication method on computers (Booth, 1951). Base on the reviews of literature, there are a number of signed representations of integer k in Radix-2 and higher. These representations are suitable for Elliptic Curve scalar multiplication as well, since the Hamming Weight (HW) of some of these signed binary representations is less than the HW of the unsigned binary.

During recent years, a lot of researchers have investigated on cryptography projects by inventing new single scalar EC multiplication algorithms or by enhancing the efficiency of EC operations. In the current study, the concentration is on designing and introducing new representation methods for integers and apply them on ECC at the scalar arithmetic level, specifically for the case of standard curves (NIST) over finite fields.

1.3 Research Motivation

In most public-key cryptosystems there is at least one operation that dominates the execution time. For instance, this operation in EC cryptosystems is the point multiplication. Therefore, various methods have been introduced to enhance the performance of this operation (Blake et al., 2005; Henri Cohen et al., 2016; Jurišic & Menezes, 1997). Specially, the integer representation of the multiplier performs an important role in the performance of these methods (Gordon, 1998). Integer representations with minimal average number of nonzero digits (Hamming Weight), are more appealing among the existing ones. This is because of the fact that they decrease the required number of point additions or subtractions. Therefore, introducing a new integer representation that can reduce the hamming weight of integer numbers more than the existing methods, will improve this enhancement of the efficiency.

Additionally it is of interest to have a representation that can be obtained by scanning the bits from left to right (from the most significant bit to the least significant one) (Müller, 1998; Solinas, 2000). This property removes the need for recoding and storing the multiplier beforehand, which leads to enhancing the performance of Left-to-Right point multiplication methods in terms of memory and running time. For this reason, introducing a Left-to-Right version of the proposed representation will potentially increase the usage of the new method for improving the performance of the public-key cryptosystems.

Research Questions: Based on this motivation, the following research questions are retrieved from problem statement:

I. How to improve the efficiency of current signed digit representations?

II. Is it possible to perform the proposed signed digit representation from Left to Right?

III. Are the proposed signed digit representations applicable to ECC?

IV. Are the proposed EC multiplication methods more efficient than existing ones?

1.4 Research Objectives

The main goal of this study is defined based on the answer to the first research question. After reviewing the literature, the main goal for this research is to introduce a new representation for integers which can be used to enhance the efficiency of some cryptosystems. For example, by proposing efficient single scalar EC multiplication algorithm based on the proposed representation method, the efficiency of ECC can be improved.

Since ECC uses a shorter key compared to the other Public-key cryptosystems such as RSA, Diffie Hellman key exchange, El-Gamal, which means fewer computations, it is becoming more important for certain real-world operations, particularly for devices having limited resources. Thus, Elliptic Curve Cryptography is chosen as the case study for this research and two single scalar EC multiplications based on proposed recoding methods are presented.

In order to achieve the goal of this research, the main objectives of this research are set as:

I. To propose a new recoding method to reduce the Hamming Weight of numbers with radixes higher than two.

II. To propose the Left-to-Right approach of the proposed recoding method.

III. To propose a single scalar Elliptic Curve multiplication method based on the proposed method.

The last research question is answered in Chapter 4 as part of the analysis.

1.5 Research Scope

Modular exponentiation is considered as one of the most time-consuming (expensive) operations in most of cryptosystems. Accordingly, not only an effective algorithm which can implement this operation is very important, but also has direct effect on the performance of the resulting protocol of cryptography. In order to compute g^m, two main types of exponentiation might be essentially distinguished.

In the first type such as ElGamal cryptosystems, the base (g) is fixed but the exponent (m) varies. In such cases, good performances would be attained by the basic 'square and multiply' technique. In the second type such as RSA cryptosystems, there is a fixed exponent (m) and a variable base (g). Thus, the purpose of this type would be an efficient computing of g^m.

The current thesis is mainly concerned with the first type of exponentiation. Further use of the methods proposed in this research can be demonstrated when it is possible to compute inverses (virtually) for free (for instance Elliptic Curves). The main idea is to enhance the efficiency of multiplication by decreasing the Hamming Weight of exponent. This study emphasizes on Elliptic Curves defined over binary fields in order to show the application of the proposed recoding methods in ECC.

1.6 Research Methodology

This section explains the research methodology which is followed in this study. This methodology comprises five different steps starting with a comprehensive review of previous works. All the next steps are designed to address each of the research questions and fulfil the objectives of this research.

Based on the review of the previous related works, the efficiency of many public-key cryptosystems depends on the presentation of the integers which are used. To give an example, the computation of kP over Elliptic Curves is the main operation in Elliptic Curve Cryptography (ECC) (Stallings, 2016). The efficiency of the kP calculation depends on the representation of the scalar k. The double-and-add algorithm (also called the binary algorithm) is the standard unsigned scheme that is used to compute the EC point $Q = kP$.

Scholars have reached to the conclusion that the binary EC multiplication algorithm is not the most effective scheme for applying EC computations. Consequently, there is a need for other representations or recoding methods such as Complementary Recoding (CR), Non-Adjacent Form (NAF), and Mutual Opposite Form (MOF) in order to enhance the efficiency of EC computations. Therefore, the second step of the research methodology is to propose new Radix-*r* representations in order to accelerate the single scalar multiplication for EC computations. Meanwhile, the comparison metrics for the performance analysis are defined.

In the second step, new Right-to-Left recoding methods are designed and introduced. The purpose of presenting these methods is reducing the Hamming Weight of integers with radixes higher than two compared to the previous recoding methods so that they can be used to enhance the performance of some cryptosystems like ECC. As formerly mentioned, reduction of the Hamming Weight of numbers in a multiplication, decreases the number of sub-operations in multiplication calculation. Therefore, a number representation with lower Hamming Weight can improve multiplication calculation in terms of mentioned aspect.

Left-to-Right recoding is considered a natural choice in ECC because window methods can be used more efficiently. Furthermore, multiplication method and recoding method can be combined so that storing the re-coded scalar k is not necessary. Therefore, this method seems more appropriate for limited hardware devices (Okeya et al., 2004). Accordingly, the third step is allocated to find a suitable existing left to right approach for the method proposed in step two and apply on that.

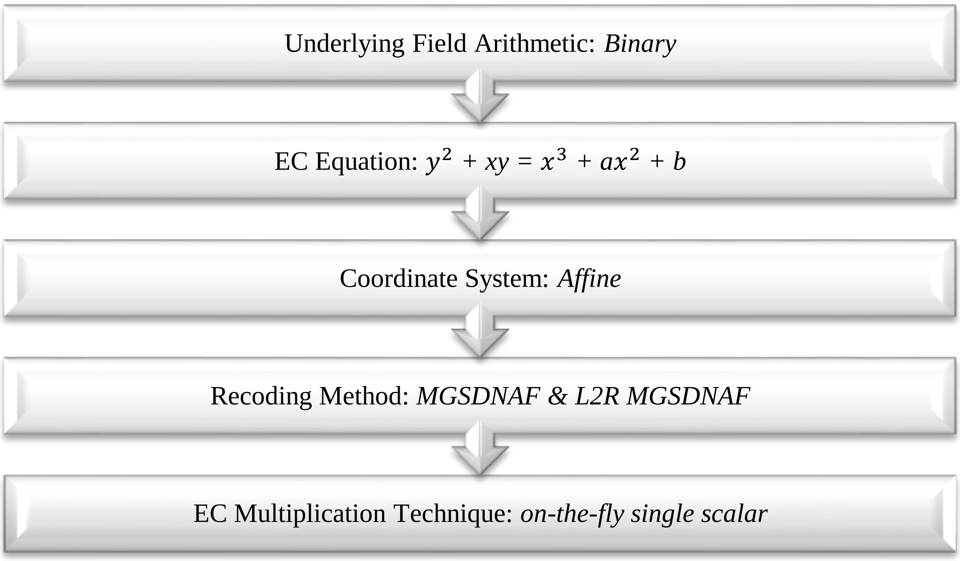
Underlying Field Arithmetic: *Binary*
EC Equation: $y^2 + xy = x^3 + ax^2 + b$
Coordinate System: *Affine*
Recoding Method: *MGSDNAF & L2R MGSDNAF*
EC Multiplication Technique: *on-the-fly single scalar*

Figure 1-2: Parameters of EC Multiplication Algorithms

Therefore, in the third step, the Left-to-Right version of proposed recoding methods which can reduce the Hamming Weight of numbers with radixes higher than two are designed and introduced.

In the fourth step of this research's methodology, to examine the application of the proposed methods, two EC multiplication algorithms are proposed. There are some parameters which are necessary to be considered when designing an EC multiplication algorithm. One of these parameters is the underlying field arithmetic (prime or binary) that should be initially determined in order to choose the EC equation that will be used over the field. The other parameter is the coordinate system such as Affine coordinate or Projective coordinate which should be selected.

In fact, the EC point arithmetic operations can be defined with these parameters. Then, after the more appropriate recoding method is chosen from the proposed methods in order to be applied on k in kP, the EC multiplication method can be then introduced. Figure 1-2 illustrates the relation between the parameters that have been discussed earlier, (i.e. the components of an EC cryptosystem).

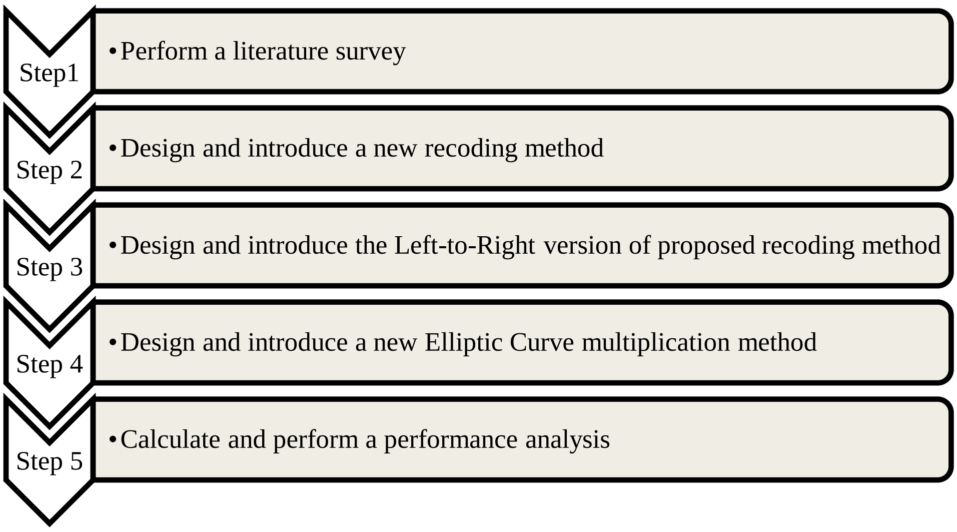
Step1
•Perform a literature survey
Step 2
•Design and introduce a new recoding method
Step 3
•Design and introduce the Left-to-Right version of proposed recoding method
Step 4
•Design and introduce a new Elliptic Curve multiplication method
Step 5
•Calculate and perform a performance analysis

Figure 1-3: Research Methodology Steps

The final step of this research's methodology is calculating and carrying out a performance analysis of the proposed EC multiplication method to determine its effectiveness compared to the current state of art methods. The research methodology proceeded in this study is graphically shown in Figure 1-3.

1.7 Research Contributions

Since a valuable research should add to knowledge and have great contribution to the related field, this study also takes step forward to have significant contributions to the available knowledge in the related area. These contributions involve introducing the following methods:

I. New Right-to-Left recoding methods (Presenting an efficient class of the Radix-*r* representation can increase the performance of some mathematical operations like scalar multiplication, for example, by reducing the number of non-zero digits):

    a. Modified Generalized Non-Adjacent Form (MGNAF)
    b. Alternative Generalized Non-Adjacent Form (AGNAF)
    c. Modified Generalized Signed Digit Non-Adjacent Form (MGSDNAF)

II. New Left-to-Right recoding methods (In many operations the algorithm which performs from Left-to-Right, is generally preferred because if pre-computation is required, Left to Right approach can increase the efficiency. For example, in single scalar multiplication, the values of Pi, which will be set in the first loop, can be pre-computed and stored in advance):

    a. Left-to-Right MGNAF

    b. Left-to-Right MGSDNAF

III. New single scalar multiplications for Elliptic Curve Cryptography (To check the application of proposed method in cryptography, a single scalar multiplication based on proposed algorithm should be introduced and studied):

    a. Based on MGSDNAF recoding method

    b. Based on Left-to-Right MGSDNAF recoding method

1.8 Thesis Outline

This thesis is organized into five chapters. The first chapter provides an overview of the research content and its procedure. Chapter 2 carries on with the literature review on the number systems, Public-key cryptography, Elliptic curves and most frequently used methods in single scalar multiplications. Chapter 3 proposes some new recoding methods such as MGNAF, L2R MGNAF, AGNAF, MGSDNAF, L2R MGSDNAF and the enhanced single scalar multiplication algorithms for Elliptic Curve cryptography based on the proposed recoding methods. Furthermore, this chapter provides details on the implementation of the conversion methods as well as all the enhanced algorithms. Chapter 4 presents the results and discussion about the proposed methods and finally, the research ends up by Chapter 5 which provides conclusion.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

Analysis of arithmetic functions can be simplified by choosing a proper number representation. Digit set or radix can be chosen to match the characteristics of the algorithm or implementation technology. Such changes can achieve lots of benefits. For instance, the frequency of useful digits (like zero) can be increased and the total amount of digits necessary to represent a number can be reduced. The cardinality of the digit set can be decreased, this change may decrease the number of pre-computed middle results to store. Decreasing the cardinality of the digit set also simplifies digit encoding for hardware execution and escalates the frequency of a given digit. Therefore, the benefits of using available pre-computations will increase.

It is usually essential to trade these benefits one against the other. For instance, increasing the radix usually leads to reducing the number of digits required to represent a number. On the other hand, this change will also increase the digit set cardinality. Therefore, as it was mentioned before the number representation should be chosen based on the characteristics of the algorithm.

To give an example, point multiplication is considered as the main operation in EC cryptography, so the efficiency of the EC cryptosystem is highly depended on efficiency of this multiplication. The methods of EC multiplication can be classified into two types: single scalar multiplication (kP) and multi-scalar multiplication ($kP + lQ$). These methods are also heavily depending on the representation of big-integer k and its Hamming Weight.

The Hamming Weight, for instance $w(k)$, is the number of nonzero digits in the binary representation of any integer number. This number affects the total cost of EC point computation. Reducing the number of nonzero digits will reduce the number of additions needed to compute kP. For binary representation of the key k, the $w(k) = \frac{1}{2}n$, where n is the length of k in bits. Other recoding methods such as signed methods were invented in order to reduce the amount of Hamming Weight, for example the Non-Adjacent Form (NAF) signed binary method were used to accelerate the EC multiplications (Solinas, 2001).

This chapter continues with the review of the works related to number systems specially the related works in Radix 2. Section 2.3 is dedicated to the number theory-based public-key cryptosystems and review some of these algorithms. The Elliptic Curve Cryptography is looked at and reviewed in depth in Section 2.4. Sections 2.5 and 2.6 are about point representation and point multiplication respectively.

2.2 Number Systems

Even though there is no clear background regarding the creation of numbers, it can be claimed that civilizations have been developed with respect to the presence of numbers. The critical role of numbering systems in everyone’s life (especially in academic life) is now being more highlighted compared to previous eras when numbers were only used for calculation and comparison. In fact, the old numeral systems were not useful for calculation. Therefore, the early Egyptians and Greeks were inspired to discover new numbering system with computational abilities in order to enhance their trading, seasonal-agriculture, and astronomy.

Positional number system was one of the most important transformations in the history of numbering systems. The advantage of positional number systems was the easy representation of the large numbers which was considered as the disadvantage of non-positional number systems. In fact, positional number systems are so important that without them, complex calculations are not possible. Each of the number systems will be discussed as follows.

2.2.1 Positional Number Systems

The following equation represents an integer of Radix-*r* in *Positional Number Systems* (PNR):

$$\begin{aligned}(a_n \dots a_2 a_1 a_0)_r &= a_n r^n + \cdots + a_2 r^2 + a_1 r^1 + a_0 \\ &= \sum_{i=0}^{n} a_i\, r^i.\end{aligned} \tag{2-1}$$

If $a_i \in S = \{0, \dots, r-1\}$; then this representation is unique (Bilal et al., 2009; Gossett, 2009). a_i and *S* are called as *digit*, and *digit set* respectively. The symbols a_n and a_0 are accordingly called the *Most Significant Digit* (MSD) and the *Least Significant Digit* (LSD).

Positional number systems can be categorized into two significant types (Knuth, 1997) which are namely, Decimal numbers ($S = \{0, \dots, 9\}$ and $r = 10$) and Binary numbers ($S = \{0,1\}$ and $r = 2$).

Equation (2-1) does not represent all the *Position Number Systems* (PNR); It only represents PNRs which are called *Fixed-base number system* (FBNS). Each term, $a_i r^i$, in the summation in this equation comprises of two parts, namely, digit a_i and $w_i = r^i$ that is called *weight* or *place value* of a_i. Each digit's weight is achieved by multiplying the weight of previous digit by base.

$$w_{i+1} = r \times w_i. \tag{2-2}$$

FBNS and *mixed-based number system* (MBNS) are differentiated by the value of r in Equation (2-2). If r is constant similar to what is seen in Equation (2-1), it can be said that this is a FBNS but if this value is a variable, it can be called that it is a MBNS. Equation (2-3) illustrates this feature.

$$w_{i+1} = r_i \times w_i, \tag{2-3}$$

where there are integers i and j such as $r_i \neq r_j$.

If the representation of Equation (2-4) is used to show a number by its digits and their corresponding weights:

$$A = \left\{\genfrac{}{}{0pt}{}{digit}{weight}\right\} = \left\{\genfrac{}{}{0pt}{}{a_n}{w_n}\genfrac{}{}{0pt}{}{a_{n-1}}{w_{n-1}}, \ldots, \genfrac{}{}{0pt}{}{a_0}{w_0}\right\}, \tag{2-4}$$

then, FBNSs can also be represented by:

$$\left\{\genfrac{}{}{0pt}{}{digit}{weight}\right\} = \left\{\genfrac{}{}{0pt}{}{a_n}{rw_{n-1}}\genfrac{}{}{0pt}{}{a_{n-1}}{rw_{n-2}}, \ldots, \genfrac{}{}{0pt}{}{a_0}{(w_0 = 1)}\right\}, \tag{2-5}$$

and MBNS can be represented as follow:

$$\left\{\genfrac{}{}{0pt}{}{digit}{weight}\right\} = \left\{\genfrac{}{}{0pt}{}{a_n}{r_{n-1}w_{n-1}}\genfrac{}{}{0pt}{}{a_{n-1}}{r_{n-2}w_{n-2}}, \ldots, \genfrac{}{}{0pt}{}{a_0}{(w_0 = 1)}\right\}. \tag{2-6}$$

The following representation has been used by Knuth (Knuth, 1997) for positional number systems:

$$(a_n \ldots a_2a_1a_0)_{\{r_n, r_{n-1}, \ldots, 1\}} = \left\{\genfrac{}{}{0pt}{}{digits}{radices}\right\} = \left\{\genfrac{}{}{0pt}{}{a_n}{r_n}\genfrac{}{}{0pt}{}{a_{n-1}}{r_{n-1}}, \ldots, \genfrac{}{}{0pt}{}{a_0}{(r_0 = 1)}\right\}, \tag{2-7}$$

where the weight of digit can be achieved by Equation (2-3).

Moreover, there are some non-standard number systems which has non-regular bases such as negative bases (Masáková et al., 2011), fractional bases, real bases

(Frougny, 2003; Frougny & Surarerks, 2003), complex bases (Frougny & Surarerks, 2003) and quadratic bases (Masáková et al., 2011). As the focus of this research is on Binary Number Systems, this numbering system is more explained.

2.2.1(a) Binary Number System

Some scholars address the invention of binary number to Pingala (Van Nooten, 1993) in the fifth century B.C. However, others believe that Gottfried Leibniz (1703) (a mathematician) invented and reported this modern system (Glaser, 1971). The disadvantage of binary numbers was their long representations which would make their frequent daily practices difficult. However, invention of computer put an end to this problem and paved the way for binary number system to become an essential part of human life. Later on in 1854, George Boole (a British mathematician) presented a new algebra called *Boolean algebra* (Goodstein, 2012) which was based upon this number representation. In fact, development of computer science and digital systems root in the foundation of this algebra. There are other subjects, for example number theory, statistics and set theory which have been established based upon Boolean algebra.

As it was mentioned in Section 2.2.1 , binary number system is a positional number system, with the digit set of $S = \{0,1\}$ and base $r = 2$. Comparing number representation in binary number system and decimal numbers system, it can be derived that the former is lengthier compared to the latter. For example:

$$1000000_{10} = 11110100001001000000_{2}.$$

Each digit in binary system is called *bit* and the length of a number A in *bits*, given by the formula

$$L(A) = L(A,2) = \lfloor \log_2 A \rfloor + 1. \tag{2-8}$$

2.2.1(b) Signed-Binary Number System

There are some algorithms, for instance multiplication and exponentiation in number theory, which their effectiveness depends on the Hamming Weight of the binary numbers. Scholars have suggested new number systems which are derived from binary number system in order to drop the Hamming Weight of binary numbers (Booth, 1951). *Signed-binary* (SB) system is one of these number systems which started by using '-1' in symbolization of a binary number. These types of number systems are derived from the following series expansion in number theory:

$$\left(\overbrace{1 \ldots 1}^{n+1}\right)_2 = \sum_{i=0}^{n} 2^i = 2^{n+1} - 1 = \left(1\,\overbrace{0 \ldots 0}^{n}\,\bar{1}\right)_2, \tag{2-9}$$

where $\bar{1} = (-1)$.

The Hamming Weight (HW) of each n-bit sequence of symbol '1' reduces from n to 2. For example:

$$(1111111)_2 = (1000000\bar{1})_2,$$

where

$$HW(1111111)_2 = 7 \quad \text{and} \quad HW(1000000\bar{1})_2 = 2.$$

The following equation illustrates digit set and base in signed-binary number systems:

$$S = \{0,1,-1\} \quad \text{and} \quad r = 2. \tag{2-10}$$

The representation of numbers by signed-binary is not unique. There are other methods with this type of representations. The most famous ones are Booth algorithm

(Booth, 1951), NAF (Non-Adjacent Form Recoding) (Reitwiesner, 1960)**,** CR (Complementary recoding) (C. Chang, Kuo, Lin, & Engineering, 2003) and MOF (Mutual Opposite Form) (Okeya et al., 2004) representations, however, they have different Hamming Weight.

Booth algorithm**:** Booth (1951) presented an elegant algorithm to accelerate the multiplication algorithm on digital processors. This method is considered as the foundation of the signed-binary representation of a number. Improving the common method of *add and shift* in the multiplication algorithm, Booth (1951) suggested a method that scans the multiplicand and then decides on doing addition, subtraction or nothing then shifting the result (Bewick & Flynn, 1992). In fact, Booth's method embraces the signed-binary idea, even though the numbers were not directly represented in signed-binary form in his method.

The process of Booth recoding is illustrated in Algorithm (2.1). Let $A = (a_{n-1}, \dots a_0)$ and $B = (b_{n-1}, \dots b_0)$. Let $a_{-1} = 0$. A is scanned from right to left for finding two adjacent bits $a_i a_{i-1}$ in the form of "01" or "10". If $a_i a_{i-1} =$ "01", then, b_i is set to "1". Where $a_i a_{i-1} =$ "10", then, b_i is set to "-1". The rest of b_i's will be remained zero. ($A = (a_{n-1}, \dots a_0)$ is a binary number and $B = (b_{n-1}, \dots b_0)$ is its signed-binary representation.)

Algorithm (2.1) : Booth Recoding
Input : $A = (a_{n-1}, \dots a_0)$
Output : $B = (b_{n-1}, \dots b_0)$

```
1.  For $i = -1$ up to $n - 1$ do
2.      If $a_i = 0$ and $a_{i-1} = 1$ then $b_i = 1$
        If $a_i = 1$ and $a_{i-1} = 0$ then $b_i = -1$
3.  Return B
```

The following example as shown by Equation (2-11) illustrates how Booth algorithm works.

$$A = \overbrace{111}^{100\bar{1}}\,000\,\overbrace{01111}^{1000\bar{1}} \;\rightarrow\; B = 100\bar{1}00010001\bar{1}. \qquad (2\text{-}11)$$

In the above example, the number of non-zero digits reduced from $HW(A) = 7$ to $HW(B) = 4$. However, it may differ in some cases like having pairs of "01" or "10". This issue is more clarified in the following example:

$$A = 10101 : HW(A) = 3 \;\rightarrow\; B = 1\bar{1}1\bar{1}1\bar{1} : HW(B) = 7. \qquad (2\text{-}12)$$

Since in Booth algorithm two bits are scanned, it is sometimes called Booth 2. Booth 3, Booth 4 and higher (Ghosh & Basuray, 2010; Madrid, Millar, & Swartzlander, 1992, 1993) have been also proposed in order to decrease the Hamming Weight of binary numbers which are shown in Example (2-12).

```
Algorithm (2.2)    : NAF Recoding
Input              : $A = (a_{n-1}, \dots a_0)$
Output             : $B = (b_n, \dots b_0)$

While $A > 0$
   For $i = 0$ up to $n - 1$ do
      If $a$ is odd then do
         $b_i = 2 - (A \ mod\ 4)$
         $A = A - a_i$
      Else
         $b_i = 0$
         $A = \frac{A}{2}$
         $i = i + 1$
Return B
```

Non-Adjacent Form Recoding (NAF): Reitwiesner suggested *Non-Adjacent Form* (NAF) in 1960 (1960). As Algorithm (2.2) illustrates the NAF representation of a number which can be gained by scanning the integer from right to left. ($A =$

$(a_{n-1}, \dots a_0)$ is a binary number and $B = (b_n, \dots b_0)$ is its NAF representation.)A binary number and its NAF representation are illustrated in the following example:

$$\begin{aligned} A &= 1\,\overbrace{01111}^{1000\bar{1}}\,0000\,\overbrace{01111}^{1000\bar{1}} \;\rightarrow\; NAF(A). \\ &= 10\bar{1}000\bar{1}000010001\bar{1} \end{aligned} \tag{2-13}$$

One of the advantages of NAF representation is that it guarantees that there is at least a zero between two non-zero digit. Therefore, this new algorithm has solved one of the biggest problems of Booth algorithm.

Solinas (2000) generalized the NAF recoding that is very useful for enhancing the performance of EC computations while Joye and Sung-Ming (2000) put forward a Left-to-right NAF recoding algorithm. Darrel et al. (2013) confirmed that every integer can be uniquely symbolized by NAF recoding. On the other hand, Morain et.al (1990) evidenced that the average of Hamming Weight of an integer after NAF recoding would be minimal. With *n* as length of integer A, $HW\ (A)\ \cong\ \frac{n}{3}$.

Mutual Opposite Form (MOF): As it is seen in Algorithm (2.3), algorithm NAF demonstrates that the recoding method in NAF is done with Right-to-Left method, however, Left-to-Right methods are given more priority for calculating exponentiation and EC multiplication (Blake et al., 2005). *Mutual Opposite Form* (MOF) was the first Left-to-Right recoding algorithm which was proposed by Okeya et al. (2004).

According to its inventers (Okeya et al., 2004), MOF representation of a number is unique as it is bidirectional. Similar to the binary representation, the average Hamming Weight of a number in MOF representation is about 50% (Okeya et al., 2004).

The process of MOF recoding is described in Algorithm (2.3). It should be assumed as $A = (a_{n-1}, \dots a_0)$ is a binary number and $B = (b_n, \dots b_0)$ is its MOF representation.

```
Algorithm (2.3)   : MOF recoding (Left-to-Right)
Input             : $A = (a_{n-1}, \dots a_0)$
Output            : $B = (b_n, \dots b_0)$

1.   $b_n = a_{n-1}$
2.   For $i = n - 1$ downto 1 do
             $b_i = a_{i-1} - a_i$
3.   $b_0 = -a_0$
4.   Return B
```

Equation (2-14) in the following example exemplifies the procedure of MOF.

$$\begin{array}{rcccccccc} A & & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 2A & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ B = MOF(A) = 2A - A & 1 & 0 & \bar{1} & 0 & 1 & 0 & 0 & \bar{1} \end{array} \tag{2-14}$$

Complementary Recoding method (CR): Complementary recoding (CR) is another recoding method which was presented to increase the speed of *common-multiplicand* multiplications (Chang et al., 2003). The advantage of this method can be noticed in large integer multiplication. In this method, a new signed-binary form of $A = (a_{n-1}, \dots a_0)$ is achieved by its first complement of $\hat{A} = (\hat{a}_{n-1}, \dots \hat{a}_0)$ where $\hat{a}_i = 1$ if $a_i = 0$ and $\hat{a}_i = 0$ if $a_i = 1$. On the other hand, if $\hat{a}_i = 1 - a_i$ and let $B = CR(A) = (b_{n-1}, \dots b_0)$. Then, the following equation will be obtained:

$$B = 2^n - \hat{A} - 1. \tag{2-15}$$

Algorithm (2.4) displays this recoding. Inexpensive operations such as addition, subtraction, bitwise operations are used by CR and MOF while operations such as division are used by NAF (Chang et al., 2003). ($A = (a_{n-1}, \dots a_0)$ is a binary number and $B = (b_n, \dots b_0)$ is its CR representation.)