Efficient Algorithms for Elliptic Curve Cryptosystems using Endomorphisms

伯田，恵輔

https://doi.org/10.15017/1398315

出版情報：九州大学，2013，博士（機能数理学），課程博士
バージョン：published
権利関係：全文ファイル公表済
Efficient Algorithms for Elliptic Curve Cryptosystems using Endomorphisms

Keisuke Hakuta
Graduate School of Mathematics
Kyushu University

A thesis submitted for the degree of

Doctor of Functional Mathematics
This thesis is dedicated to my wife and my sons for their support.
Acknowledgements

It would not have been possible to write this dissertation without the assistance, patience, and support of many individuals. I would like to express my deepest appreciation to all those who provided me the possibility to complete this dissertation.

I would like to extend my gratitude first and foremost to Professor Tsuyoshi Takagi for his supervision, guidance and the continuous support during my research at Kyushu University.

Besides my advisor, deepest gratitude are also due to the members of my committee, Professor Hiroyuki Ochiai, Professor Kouichi Sakurai, Associate Professor Yasuo Gon, without whose insightful comments, hard questions, and especially the revision process that has lead to this document, this study would not have been successful.

I would also like to extend my appreciation to Hisayoshi Sato for his immense and continued support, valuable advice, technical and non-technical assistance, and encouragements in a number of ways.

I would like express my gratitude to all the current and former Hitachi crypto team: Eriko Ando, Soichi Furrya, Yasuo Hatano, Kota Ideguchi, Shugo Mikami, Kunihiko Miyazaki, Ken Naganuma, Katsuyuki Okeya, Toru Owada, Hisao Sakazaki, Dai Watanabe, and Masayuki Yoshino, for their support and many technical and non-technical discussions. I also would like express my gratitude to Hirotaka Yoshida for not only his support and many technical and non-technical discussions, but also for the English correction of this thesis.

I am grateful for the Hitachi directors who have supported me, including Kazuo Takaragi, Yasuko Fukuzawa, Seichi Susaki, Takahiro Fujishiro, Masahiro Mimura, and Tadashi Kaji.
Last but not the least, I would like to thank my family members Tatsumi, Megumi, and Yusuke for encouragement, and my wife Aki, and my sons Sosuke and Rensuke for their endless support, patience and extreme understanding while I have pursued this degree.
Contents

Contents

List of Tables   vii

List of Figures  ix

1 Introduction and Motivation  1
   1.1 Introduction and Motivation  1
   1.2 Contribution and Organization of the thesis  2

2 Background  4
   2.1 Notation  4
   2.2 Elliptic curves, Frobenius endomorphism  5
   2.3 Public Key Cryptography  6
   2.4 Cryptographic Applications of Endomorphisms  7

3 Efficient Arithmetic on Subfield Elliptic Curves over Small Finite
Fields of Odd Characteristic  9
   3.1 Introduction  9
      3.1.1 Contribution of this chapter  11
   3.2 Preliminaries  12
      3.2.1 GNAF, rNAF  12
      3.2.2 Subfield elliptic curves, Frobenius expansion, and Scalar
           multiplication  13
      3.2.3 \(\tau\)-NAF on binary Koblitz curves and supersingular Koblitz
           curves  15
3.3 Proposed Methods (Two classes of $\phi$-NAF) ........................................ 17
3.3.1 $\phi$-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ .......................... 18
3.3.2 $\phi$-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ .......................... 28
3.3.3 $\psi$-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ .......................... 38
3.4 Scalar Multiplication Costs ............................................................... 40
3.4.1 Estimates ................................................................. 40
3.4.2 Timings ................................................................. 44
3.5 Sample Parameters ................................................................. 46
3.6 Summary of this chapter ........................................................... 49

4 Explicit lower bound for the length of minimal weight $\tau$-adic
expansions on Koblitz curves ......................................................... 51
4.1 Introduction ................................................................. 51
4.2 Key Lemma ................................................................. 55
4.3 Lower Bound for the Length .................................................. 62
4.4 Another Proof of the Minimality ........................................... 64
4.4.1 The Main Idea of Our New Proof ..................................... 64
4.4.2 Our New Proof .......................................................... 66
4.5 $\tau$-adic Minimal Length Form and Its Cryptographic Application . 69
4.5.1 $\tau$-adic Minimal Length Form ...................................... 69
4.5.2 Cryptographic Application of $\tau$-MLF ................................. 72
4.6 Summary of this chapter ........................................................ 72

5 Batch verification suitable for efficiently verifying a limited num-
ber of signatures ................................................................. 74
5.1 Introduction ................................................................. 74
5.1.1 Motivation ................................................................. 75
5.1.2 Contribution of this chapter ........................................... 76
5.2 Preliminaries ................................................................. 77
5.2.1 Batch Verification ........................................................ 77
5.2.2 Small Exponents Test .................................................... 78
5.2.3 Complex Exponent Test .................................................. 78
5.2.4 Improved Complex Exponent Test .................................... 80
## CONTENTS

5.3 Our Proposed Method ................................................. 80
  5.3.1 Properties of Certain Types of Elliptic Curves ........... 81
  5.3.2 Our Proposed Test using the Curve $E_{7,b}$ ............... 82
  5.3.3 Our Proposed Test using the Curve $E_{5,a}$ ............... 88
5.4 Sample Parameters .................................................. 94
5.5 Comparison ......................................................... 97
  5.5.1 Timings ....................................................... 97
  5.5.2 Estimate .................................................... 98
5.6 Summary of this chapter ........................................... 100

6 Conclusion and Open Problems ........................................... 101
  6.1 Conclusion ...................................................... 101
  6.2 Open Problems ................................................ 102

Bibliography ............................................................... 107

Index ................................................................. 115
List of Tables

3.1 Forms of $\ell$ digits $\phi$-GNAF representations ........................................ 25
3.2 Forms of $\ell$ digits $\phi$-rNAF representations ........................................ 36
3.3 The number of curve operations for each recoding method ($q = 3$) 41
3.4 The number of curve operations for each recoding method ($q \sim 5$) 41
3.5 The number of $F_{q^m}$-field arithmetic operations (bit length of scalar $= 192, 224, 256, 384$) and total number of multiplications ($S = 0.8M$, $I/M = 8$) for each recoding method ................................. 42
3.6 Timings of the Finite Field Operations .................................................. 45
3.7 Timings of Recoding Methods .......................... .......................... 45
3.8 Timings of Scalar Multiplications on $F_3$-Koblitz Curve $K_{(3,1)}$-173 (online precomputation case) .................................. 46
3.9 Sample Parameters for $F_3$-Koblitz Curves of $t = 1$ .................. 47
3.10 Sample Parameters for $F_3$-Koblitz Curves of $t = 1$ ............... 48

4.1 Conversion of the $\tau$-NAF into the $\tau$-MLF ($\ell < 6$) .............. 70
4.2 Conversion of the $\tau$-NAF into the $\tau$-MLF ($\ell \sim 6$) .............. 71

5.1 Classification of elliptic curves of the form $E_{7,b}$ .......................... 83
5.2 Classification of elliptic curves of the form $E_{5,a}$ .......................... 89
5.3 Sample Parameters for the elliptic curves $E_{5,a}/F_5$ .............. 94
5.4 Sample Parameters for the elliptic curves $E_{7,b}/F_7$ .............. 96
5.5 Timings of the Finite Field Operations .......................... .......................... 98
5.6 Estimate the computational costs of CE test and our test .................. 99
5.7 Values $k_j$ and $k_\omega$ such that $#S_j \subset 2^{k_{nw}}, #S_{\omega} \subset 2^{k_{nw}}$ ............... 99
5.8 the number of signatures \( N \) for which our method is faster than CE test ........................................ 100
List of Figures

5.1 A typical IP camera surveillance system 76
Chapter 1

Introduction and Motivation

1.1 Introduction and Motivation

Nowadays, Information and communication technology (ICT) assists us in many areas of our lives (e.g. business, education, our personal lives, and so on). The role of ICT is becoming more and more important. The number and variety of ICT devices is growing rapidly and different types of devices are widely communicated with each other through different types of networks. ICT systems, however, may pose security risks such as spoofing, tampering, repudiation, information disclosure, etc.

Public key schemes provide encryption, digital signature, and key exchange capabilities. Moreover, some schemes have homomorphic properties, namely, these schemes allow anyone in possession of the public key to perform computations on encrypted data without decrypting it. Many public key schemes are based on the difficulties of number-theoretic problems such as integer factoring problem, discrete logarithm problem in finite fields or elliptic curves, problem of solving a multivariate polynomial system over a finite field. In public key schemes, number-theoretic computations are the dominant operations. For instance, computing scalar multiplication (or point multiplication) is the most time consuming operation in elliptic curve cryptosystems (ECC for short). Therefore it is an important task to accelerate number-theoretic computations.

On the other hand, endomorphisms of algebraic varieties play an important
role in modern mathematics. For example, the endomorphisms on an elliptic curve play a central role in the arithmetic study of elliptic curves. Polynomial automorphisms play a key role in several open problems in affine algebraic geometry such as Jacobian conjecture, tame generators problem.

This thesis deals with endomorphisms of algebraic varieties. In the cryptographic community, endomorphisms of algebraic varieties are often used not only to improve the efficiency of public key schemes, but also to evaluate the security of public key schemes. More precisely, in ECC the Frobenius endomorphism of an elliptic curve is especially attractive because of the fact that the Frobenius endomorphism can be computed very efficiently. Roughly speaking, for a given subfield elliptic curve (i.e. an elliptic curve over a finite field which are actually defined over some subfield), Frobenius expansions are special representations of element in the endomorphism ring as polynomials in the Frobenius endomorphism with cryptographically appropriate integer coefficients. For the purpose of improve the performance, it is desirable to construct Frobenius expansions with low Hamming weight (i.e. the the number of nonzero coefficients). In addition, theoretical analysis of efficiency of Frobenius expansions with low Hamming weight may be important to explore the Hamming weight of other representations. Frobenius expansions are also attractive for signature verification, especially, in batch verification (a method to verify multiple signatures simultaneously). Frobenius expansions are used in order to randomize the verification process and to improve the efficiency. To use Frobenius expansions for batch verification, it is necessary to investigate some properties of Frobenius expansions.

The motivation for the results in this thesis comes from the importance to explore some properties of Frobenius expansions.

1.2 Contribution and Organization of the thesis

The rest of this thesis is organized as follows. Section 2 reviews some background. In Section 3, we propose efficient scalar multiplication algorithms for subfield elliptic curves of trace ±1. The results are extensions of Solinas’s τ-adic non adjacent form (τ-NAF for short) on Koblitz curves. We also present implementation results of our new algorithms and previous methods.
In Section 4, we derive an explicit lower bound for the length of minimal Hamming weight $\tau$-adic expansions on Koblitz curves. We also give a new proof of the minimality of the Hamming weight of the $\tau$-NAF on Koblitz curves. Furthermore, we classify a minimal length $\tau$-adic expansion with minimal Hamming weight except for two special cases.

Finally, in Section 5, we present two efficient batch verification techniques suitable for verifying a limited number of signatures in real-time. Our method can only be applied to elliptic curve based signatures, and uses one of the two special families of elliptic curves.
Chapter 2

Background

This section provides necessary background on elliptic curves and public key cryptography.

2.1 Notation

Throughout this thesis, we use the symbols \( \mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{C}, \mathbb{F}_q \) to represent the natural numbers, the integers, the rational numbers, complex numbers, and a finite field with \( q \) elements respectively, where \( q = p^r (r \sim 1), p = \text{char}(\mathbb{F}_q) \). For any field \( k \), we denote by \( p = \text{char}(k) \) the characteristic of the field \( k \). For a field \( k \), the \( n \)-dimensional projective space over \( k \) is denoted by \( \mathbb{P}_k^n \). For a positive integer \( n \), we denote the residue class ring modulo \( n \) by \( \mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z} \). For a finite set \( S \), we denote the cardinality of \( S \) by \( \# S \). Denote by \( \mathbb{Z}_{>0} \) the set of positive integers. For \( x \in \mathbb{C} \), we denote by \( |x| \) the absolute value of \( x \). We denote \( a \) by \( \bar{a} \) for any natural number \( a \).

For any non-zero complex number \( \psi \in \mathbb{C} \setminus \{ 0 \} \), we denote \( \psi \)-adic expansion \( \prod_{i=0}^{\ell-1} c_i \psi^i \) with \( c_i \in \mathbb{Z} \) by \( (c_{\ell-1}, \ldots, c_0)_\psi \), and denote the number of non-zero \( c_i \)'s by \( \text{wt}((c_{\ell-1}, \ldots, c_0)_\psi) \). The symbol ‘\( \pm \)' means a non-zero digit of \( \psi \)-adic expansions. We call \( \ell \) the length of the \( \psi \)-adic expansion. Let \( E \) be an elliptic curve defined over a finite field \( \mathbb{F}_q \), \( \phi : E \rightarrow E, \quad (x, y) \mapsto (x^q, y^q) \) be the \( q \)-th-power Frobenius map on \( E \). We put \( t_m := q^m + 1 \quad \# E(\mathbb{F}_{q^m}), \quad t := t_1 \), and call \( t_m \) the trace of \( E(\mathbb{F}_{q^m}) \). We can regard \( \phi \) as a complex number which satisfies the characteristic
equation $\varphi^2 t \varphi + q = 0$. We denote the cardinality of the set of the $\mathbb{F}_{q^m}$-rational points of $E$ by $\#E(\mathbb{F}_{q^m}) = hn$ ($h$: cofactor, $n$: prime). We assume that the order of a point $P / E(\mathbb{F}_{q^m})$ is $n$. $A$, $\mathcal{F}$ stand for the computational cost of point addition on $E(\mathbb{F}_{q^m})$, $q^m$-power Frobenius map on $E(\mathbb{F}_{q^m})$, respectively.

We denote by $E_a$ the Koblitz curve defined by Equation (4.1) and by $\tau$ the Frobenius map on $E_a$ defined by (4.2). Let $\mathcal{D} := \mathcal{D}_2 = \{0, \pm 1\}$. Remark that for a fixed coefficient $a / \mathbb{F}_2$ in Equation (4.1), it satisfies that $\mathcal{D} = \{0, \pm \mu\}$.

For any element $\alpha$ in $\mathbb{Z}[\tau]$, we denote by $\alpha = \prod_{i=0}^{\ell-1} b_i \tau^i (b_i / \mathcal{D})$ the $\tau$-NAF of $\alpha$, and by $\alpha = \prod_{i=0}^{\ell-1} c_i \tau^i (c_i / \mathcal{D})$ be any $\tau$-adic expansion of $\alpha$, respectively. The length of $\tau$-NAF of $\alpha$ is denoted by $\ell_{\text{NAF}}(\alpha)$. We denote by $\ell_{\text{min}}(\alpha)$ the length of $\tau$-adic expansion of minimal length among all $\tau$-adic expansions of minimal Hamming weight with digit set $\mathcal{D}$.

Additionally, we use the following notation in Section 4.2 and Section 4.4. If $\ell > \ell'$ then put $c_{\ell'} = c_{\ell'+1} = \ldots = c_{\ell-1} = 0$. Otherwise, put $b_\ell = b_{\ell+1} = \ldots = b_{\ell'-1} = 0$. Furthermore, replace $\max\{\ell, \ell'\}$ by $\ell$ if necessary. We put $S_\alpha := \{i / \{0,1,\ldots,\ell \mid b_i = 0\}, \tau$ and $T_\alpha := \{i / \{0,1,\ldots,\ell \mid c_i = 0\}$.

### 2.2 Elliptic curves, Frobenius endomorphism

**Definition 2.1 (elliptic curve [55, Definition 6.1.25])** Let $k$ be a field. We define an elliptic curve over $k$ to be a smooth projective curve $E$ over $k$, isomorphic to a closed subvariety of $\mathbb{P}^2_k$ defined by a homogeneous polynomial $F(x, y, z)$ of the form $F(x, y, z) = y^2z + (a_1x + a_3z)y(z(x^3 + a_2x^2z + a_4xz^2 + a_6z^3)$ with the privileged rational point $\mathcal{O} = (0,1,0)$.

**Corollary 2.1 (endomorphism ring of an elliptic curve [69, Corollary 9.4])** The endomorphism ring of an elliptic curve $E/K$ is either $\mathbb{Z}$, an order in an imaginary quadratic field, or an order in a quaternion algebra. If char$(K) = 0$, then only the first two are possible.

**Definition 2.2 (Frobenius endomorphism [55, Definition 3.2.21])** Let $p$ be a prime number. Let $X$ be a scheme over $\mathbb{F}_p$. We call the morphism $F_X : X \leftarrow X$ induced by the ring homomorphism $\mathcal{O}_X \leftarrow \mathcal{O}_X$, $a \mapsto a^p$ the absolute Frobenius of $X$. 
Lemma 2.1 ([55, Lemma 3.2.22])

1. Let $f: X \to Y$ be a morphism of schemes over $\mathbb{F}_p$. Then $F_X \circ f = f \circ F_Y$.

2. For any $x \in X$, we have $F_X(x) = x$.

2.3 Public Key Cryptography

We only review the definitions of a public key encryption scheme and a digital signature scheme. We denote the set of all finite-length bit strings —i.e., strings of 0’s and 1’s—by $\{0, 1\}^*$. 

Definition 2.3 (Public Key Encryption Scheme [30, Definition 5.1.1])

A public key encryption scheme is a triple, $(G, E, D)$, of probabilistic polynomial-time algorithms satisfying the following two conditions:

1. On input $1^n$, algorithm $G$ (called the key-generator) outputs a pair of bit strings.

2. For every pair $(e, d)$ in the range of $G(1^n)$, and for every $\alpha \in \{0, 1\}^*$, algorithms $E$ (encryption) and $D$ (decryption) satisfy

$$\Pr[D(d, E(e, \alpha)) = \alpha] = 1$$

where the probability is taken over the internal coin tosses of algorithms $E$ and $D$.

The integer $n$ serves as the security parameter of the scheme. Each $(e, d)$ in the range of $G(1^n)$ constitutes a pair of corresponding encryption/decryption keys. The string $E(e, \alpha)$ is the encryption of the plaintext $\alpha$ using the encryption-key $e$, whereas $D(d, \beta)$ is the decryption of the ciphertext $\beta$ using the decryption-key $d$.

Definition 2.4 (Digital Signature Scheme [30, Definition 6.1.1])

A digital signature scheme is a triple, $(G, S, V)$, of probabilistic polynomial-time algorithms satisfying the following two conditions:
1. **On input** $1^n$, algorithm $G$ (called the key-generator) outputs a pair of bit strings.

2. **For every pair** $(s, v)$ in the range of $G(1^n)$, and for every $\alpha \in \{0, 1\}^\ast$, algorithms $S$ (signing) and $V$ (verification) satisfy

$$\Pr[V(v, \alpha, S(s, \alpha)) = 1] = 1$$

where the probability is taken over the internal coin tosses of algorithms $S$ and $V$.

The integer $n$ serves as the security parameter of the scheme. Each $(s, v)$ in the range of $G(1^n)$ constitutes a pair of corresponding signing/verification keys.

We sometimes call $S(s, \alpha)$ a signature to the document $\alpha$ produced using the signing-key $s$. Likewise, when $V(v, \alpha, \beta) = 1$, we say that $\beta$ is a valid signature to $\alpha$ with respect to the verification-key $v$. This definition asserts that any signature to $\alpha$ produced using the signing-key $s$ is a valid signature to $\alpha$ with respect to the corresponding verification-key $v$. Note that there may be valid signatures (with respect to $v$) that are not produced by the signing process (using the corresponding $s$).

We do not explain the security of public key schemes, namely, provable security. For complete explanation of provable security, we refer to [29], [30].

### 2.4 Cryptographic Applications of Endomorphisms

Endomorphisms on (hyper-)elliptic curves have several cryptographic applications. Here, we review some of them.

1. **Scalar multiplication**: One of the major applications of endomorphisms in ECC is fast computation of scalar multiplication (or point multiplication). Since scalar multiplication is the fundamental operation in ECC, many efficient methods have been proposed to improve the efficiency. In particular, scalar multiplication algorithms using endomorphisms have the focus of much attention. Frobenius expansions (cf. [46], [47], [50], [70]), GLV method [28] are such examples. For more details, see Chapter 3.
2. **Bilinear pairing:** Weil pairing and Tate pairing are well known in mathematics. In addition to these pairings, other pairings such as Ate pairing have been proposed to accelerate pairing computation and/or to construct new cryptographic primitives. Endomorphisms on (hyper-)elliptic curves are also used to construct bilinear pairings. For more details, see for instance [27].

3. **Map to point hash function:** Several public key schemes need a map to point hash function. The function maps a bit string of arbitrary length to a point on an elliptic curve defined over a finite field. Efficiently computable endomorphisms on elliptic curves play a central role in the construction of map to point hash functions.

4. **Random point generation:** In many public key schemes based on elliptic curves, it requires generation of a random point on an elliptic curve. For a given point $P$ on an elliptic curve $E(\mathbb{F}_q)$, the standard method to generate a random point is to choose a random number $r$ and to compute scalar multiplication $rP$. Some methods have been proposed in order to avoid the direct computation of scalar multiplication. For more details, see for instance [51].
Chapter 3

Efficient Arithmetic on Subfield
Elliptic Curves over Small Finite
Fields of Odd Characteristic

3.1 Introduction

Elliptic curve cryptosystems (ECC) were proposed in 1985 independently by Victor Miller [56] and by Neal Koblitz [44]. Since ECC provide many advantages, for example, shorter key length and faster computation speed than those of RSA cryptosystems, ECC have been the focus of much attention. In ECC, each protocol such as ECDH, EC-ElGamal, and ECDSA involves scalar multiplications for given points on an elliptic curve by positive integers. These multiplications have much effect on the efficiency of the schemes, and many efficient methods have been proposed.

As one such method, the use of subfield elliptic curves (i.e. elliptic curves over finite fields which are actually defined over some subfield [12]) is especially attractive because by using the Frobenius maps, which is efficiently computed, scalar multiplication on subfield elliptic curves can be performed much faster than that on curves over prime fields. Koblitz [46] suggested anomalous binary curves, and Müller [60] extended Koblitz’s idea to achieve the Frobenius expansions over small fields of characteristic two. Smart [70] generalized Müller’s result to elliptic
curves over small fields of odd characteristic. Indeed, Smart shows that every element $d \in \mathbb{Z}[\phi]$ can be written as $d = \prod_{i=0}^{\ell-1} d_i \phi^i$, where $q$ is the order of the defining field of an $\mathbb{F}_q$-subfield elliptic curve $E$, $\phi$ is the $q$-th power Frobenius map on $E$, $d_i \in \{0, \pm 1, \ldots, \pm (q - 1)/2\}$, $t$ is the trace of $\phi$, $(g, t) = (5, \pm 4), (7, \pm 5)$, and $\ell$ is a non-negative integer. Therefore, scalar multiplication method using $\phi$ in place of doublings can be deployed on subfield elliptic curves. Note that neither of these methods can be applied in the case of curves over prime fields (the case in which the group of prime field rational points is used for the cryptosystem). Gallant, Lambert, and Menezes [28] proposed efficiently computable endomorphisms other than Frobenius endomorphisms that can be used for fast scalar multiplication. Moreover, Park, Lee, and Park [64] proposed two kinds of endomorphisms from [28] that can be used together for a certain class of curves, and they also presented a new expansion method.

In addition, the recoding method of the scalars also plays an important role in the performance. In general, smaller non-zero densities in the representations of scalars improve the efficiency. The non-adjacent form (NAF) [66] and its generalizations such as the generalized non-adjacent form (GNAF) [20] and the radix-$r$ non-adjacent form ($\tau$NAF) [75], are methods used for minimizing the non-zero densities. So as to achieve further improvement, it has been tried to combine the subfield curve method with the recoding methods. Solinas [72] proposed an efficient method of scalar multiplication on binary Koblitz curves, namely $\tau$-adic NAF ($\tau$-NAF), and Koblitz [47] proposed $\tau$-adic NAF on some supersingular elliptic curves defined over the prime field of characteristic three using the Frobenius endomorphism of the curves. In addition, Günther, Lange, and Stein [32] proposed a generalization of $\tau$-adic NAF on hyperelliptic Koblitz curves. Recently, Blake, Murty, and Xu [11] proposed the radix-$\tau$ width-$w$ NAF for every integer in all Euclidean quadratic imaginary fields. But, only a few curves are available for the above methods so far. Since the choice of curves can seriously affect the security and efficiency of ECC, it is highly unlikely that only binary Koblitz curves will be used as subfield elliptic curves for cryptographic usage. There might be demand for subfield elliptic curves other than binary Koblitz curves. From this reason, to find suitable subfield elliptic curves and to develop efficient scalar multiplication algorithms on those curves is a very important matter. However, in
[2, p.367], the authors say that “The study [60], [70] is not as detailed as Solinas’. This indicates that no method to combine Frobenius expansions and NAF on the curves in [60], [70] is known.

3.1.1 Contribution of this chapter

The contribution of this chapter is to propose two generalizations of \( \tau \)-NAF, that is, two classes of \( \phi \)-adic NAF (\( \phi \)-GNAF and \( \phi \)-rNAF) using the techniques of GNAF and \( r \)-NAF, respectively, which can be applied to a family of subfield elliptic curves defined over finite fields of odd characteristic. The digit set of NAF is \( \{0, \pm 1\} \) and the digit set of the Frobenius expansion is \( \{0, \pm 1, \ldots, \pm (q-1)/2\} \).

We may well not be able to directly apply the technique of NAF to the Frobenius expansions except for \( \tau \)-NAF on binary Koblitz curves because of the narrowness of the digit set of NAF. Thus as a natural development, we apply the GNAF and \( r \)-NAF techniques, which are the generalizations of the ordinary NAF, to apply \( \tau \)-NAF to elliptic curves in odd characteristic. For the resulting recoding methods, \( \phi \)-GNAF and \( \phi \)-rNAF, if the radix is small (e.g., 3, 5), the difference between the computational costs for the precomputation tables of \( \phi \)-GNAF and \( \phi \)-rNAF is relatively small (a few elliptic additions). But, if the radix is significantly large, the computational cost for the precomputation table of \( \phi \)-rNAF is quite large compared to that for \( \phi \)-GNAF. However the non-zero density for \( \phi \)-rNAF is significantly smaller than that for \( \phi \)-GNAF. Thus, these two generalizations are complementary. The speed of the proposed methods improve between 8% and 50% over that for the Frobenius expansion method. In this chapter, as the first step in the generalizations of \( \phi \)-NAF, we concentrate on investigating only \( \phi \)-GNAF and \( \phi \)-rNAF, and we do not deal with the width-\( w \) versions of these.

The family of subfield elliptic curves is a natural generalization of binary Koblitz curves and some examples of the curves with a large prime divisor in the group order are listed in [70]. These curves are considered to be very useful for ECC.

This chapter is organized as follows. Section 3.2 reviews the ordinary GNAF, \( r \)-NAF, and \( \tau \)-adic NAF on binary Koblitz curves. Section 3.3 shows how to generalize \( \tau \)-NAF on binary Koblitz curves to two classes of \( \phi \)-adic NAF on a family of subfield elliptic curves and proves some properties of \( \phi \)-GNAF and
\(\phi\)-rNAF. Section 3.4 estimates and compares the total computational costs of several previous methods and the proposed methods. In addition, we implement scalar multiplication on an subfield elliptic curve belonging to the above family, for several recoding methods.

### 3.2 Preliminaries

#### 3.2.1 GNAF, rNAF

In this section, we review the ordinary GNAF and rNAF. Let \( r, \alpha \) be relatively prime positive integers. We denote \( D_{r,\alpha} \) a set defined as follows.

\[
D_{r,\alpha} := \begin{cases} 
\{0, \pm 1, \ldots, \pm \alpha\} & \text{if } \alpha < r, \\
\{0, \pm 1, \ldots, \pm \alpha\} \setminus \{\pm r, \pm 2r, \ldots, \pm \lfloor \alpha / r \rfloor r\} & \text{otherwise.}
\end{cases}
\]

For an integer radix \( r \sim 2 \), GNAF and rNAF have been proposed for minimizing the numbers of non-zero densities in the representations of integer scalars. In [20] and [75], the authors calculate the non-zero densities using Markov chains. In this chapter, we regard non-zero densities of some representations as average densities of non-zero digits of the representations (See Section 3.3 for precise definitions).

**Definition 3.1 (GNAF [20])** A radix-r generalized non-adjacent form (GNAF) of a positive integer \( d \) is a representation \( d = \prod_{i=0}^{\ell-1} e_i r^i \) where \( e_i / D_{r-1, r}, e_{\ell-1} = 0 \) and for each \( i \), one of the following holds: (1) \( e_{i+1} e_i = 0 \), (2) if \( e_{i+1} e_i > 0 \), then \( \psi_{i+1} e_i + e_i \psi_i < r \), (3) if \( e_{i+1} e_i < 0 \), then \( \psi_{i+1} e_i - e_i \psi_i > 0 \). The length of the GNAF is \( \ell \).

For \( a, b \in D_{r-1, r} \), if \( a, b \) satisfy one of the followings : (1) \( ab = 0 \), (2) if \( ab > 0 \), then \( |a + b| < r \), (3) if \( ab < 0 \), then \( |\psi a| > \psi \), then we call a pair \((a, b)\) radix-r admissible pair, and otherwise, we call \((a, b)\) radix-r non-admissible pair.

**Definition 3.2 (rNAF [75])** A radix-r non-adjacent form (rNAF) of a positive integer \( d \) is a representation \( d = \prod_{i=0}^{\ell-1} e_i r^i \) where \( e_i / D_{(r^2-1)/2, \ell-1} = 0 \) and for each \( i \), it satisfies \( e_{i+1} e_i = 0 \) where we define \( e_{\ell} = 0 \). The length of the rNAF is \( \ell \).

For \( a, b \in D_{(r^2-1)/2, \ell} \), if \( ab = 0 \), then we call a pair \((a, b)\) radix-r non-adjacent pair, and otherwise, we call \((a, b)\) radix-r adjacent pair.
In the above definitions, note that for the radix \( r = 2 \), GNAF and rNAF coincide, and in this case, we call these recoding method “NAF” ([75]). It can be seen that GNAF and rNAF have some interesting properties. For details, consult [20] for GNAF, and [75] for rNAF.

**Proposition 3.1 (Properties of GNAF (resp. rNAF) [20, 75])**

1. Every positive integer \( d \) has a unique GNAF (resp. rNAF).
2. GNAF (resp. rNAF) of \( d \) has the smallest Hamming weight among all signed representations of \( d \) with digit set \( D_{r,r-1} \) (resp. \( D_{r,(r^2-1)/2} \)).
3. The average non-zero density of GNAF (resp. rNAF) is asymptotically \( (r-1)/(r+1) \) (resp. \( (r-1)/(2r-1) \)).

### 3.2.2 Subfield elliptic curves, Frobenius expansion, and Scalar multiplication

We briefly review subfield elliptic curves, the Frobenius expansion on subfield elliptic curves, and scalar multiplication on subfield elliptic curves. For detail, refer to [70], [12] and [69].

**Definition 3.3 (\( \mathbb{F}_q \)-subfield elliptic curves [70])** Let \( p \) be a prime, \( q = p^e \) a power of \( p \), and \( \mathbb{F}_q \) the finite field with \( q \)-elements. An elliptic curve defined over \( \mathbb{F}_q \) is called an “\( \mathbb{F}_q \)-subfield elliptic curve” if for some cryptographic usage, we focus on the group of \( \mathbb{F}_q^m \)-rational points \( E(\mathbb{F}_q^m) \) for some \( m \sim 2 \). An \( \mathbb{F}_q \)-subfield elliptic curve \( E \) is given by a Weierstrass equation

\[
y^2 + a_1 xy + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6,
\]

where \( a_i / \mathbb{F}_q \), and if \( q \sim 5 \), then an \( \mathbb{F}_q \)-subfield elliptic curve \( E \) is given by a short Weierstrass equation

\[
y^2 = x^3 + ax + b,
\]

where \( a, b / \mathbb{F}_q \). Let us denote \( \phi \) the \( q^k \)-th-power Frobenius map on \( E \).

\[
\phi : E \leftarrow E, \quad (x, y) \mapsto (x^q, y^q),
\]
and set $t_m := q^m + 1$ where $E(F_{q^m})$ means the set of $F_{q^m}$-rational points on $E$. We can regard $\phi$ as a complex number which satisfies the following characteristic equation: $\phi^2 + \phi + q = 0$.

Scalar multiplication is the operation of computing $dP$ for given $P$ on an elliptic curve by positive integer $d$. We describe a scalar multiplication algorithm using window $w$-NAF method in [37, Algorithm 3.6]. The computational cost of Algorithm 1 is approximately

$$D_\phi + (2^{w-2} + 1)A_\phi\{ + \frac{\ell}{w+1}A_\delta + \ell D_\delta\},$$

where $A_\phi, D_\phi$ (resp. $A_\delta, D_\delta$) stand for the computational cost of the point addition, point doubling in the precomputational step (resp. scalar multiplication step), respectively.

<table>
<thead>
<tr>
<th>Algorithm 1 Scalar multiplication (window $w$-NAF) [37]</th>
<th>Algorithm 2 Scalar multiplication (Frobenius expansion)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $d = (d_{\ell-1}, \ldots, d_0)<em>{w\text{-NAF}} / \mathbb{Z}</em>{&gt;0}$, $d_i / D_{w\text{-NAF}} = {0, \mp 1, \mp 3, \ldots, \mp(2^{w-1} - 1)}$, $\ell \geq \lceil \log_2(d) \rceil \sqrt{4}$, $P / E(F_{q^m})$</td>
<td><strong>Input:</strong> $d = (d_{\ell-1}, \ldots, d_0)<em>{\phi} / \mathbb{Z}</em>{&gt;0}$, $d_i / D_\phi = {0, \mp 1, \mp 2, \ldots, \mp \alpha}$, $\ell \geq \left\lceil 2\log_q 2 \right\rceil \sqrt{N_{Z[\phi]/Z(d)}} \sqrt{4}$, $P / E(F_{q^m})$</td>
</tr>
<tr>
<td><strong>Output:</strong> $dP$</td>
<td><strong>Output:</strong> $dP$</td>
</tr>
<tr>
<td>1: Compute $P_1 = iP$ for $i / {1, 3, 5, \ldots, 2^{w-1}}$</td>
<td></td>
</tr>
<tr>
<td>2: $Q \rightarrow 0$</td>
<td></td>
</tr>
<tr>
<td>3: for $i$ from $\ell$ downto $0$ do</td>
<td></td>
</tr>
<tr>
<td>4: $Q \rightarrow 2Q$</td>
<td></td>
</tr>
<tr>
<td>5: $Q \rightarrow Q + P_{d_i}$</td>
<td></td>
</tr>
<tr>
<td>6: end for</td>
<td></td>
</tr>
<tr>
<td>7: return $Q$</td>
<td></td>
</tr>
<tr>
<td>1: Compute $P_1 = iP$ for $i / {1, 2, \ldots, \alpha}$</td>
<td></td>
</tr>
<tr>
<td>2: $Q \rightarrow 0$</td>
<td></td>
</tr>
<tr>
<td>3: for $i$ from $\ell$ downto $0$ do</td>
<td></td>
</tr>
<tr>
<td>4: $Q \rightarrow \phi(Q)$</td>
<td></td>
</tr>
<tr>
<td>5: $Q \rightarrow Q + P_{d_i}$</td>
<td></td>
</tr>
<tr>
<td>6: end for</td>
<td></td>
</tr>
<tr>
<td>7: return $Q$</td>
<td></td>
</tr>
</tbody>
</table>

It is well-known that the cost of the Frobenius map $\phi$ is almost free in normal basis representation and a few finite field multiplication (if $q$ is small) even if in polynomial basis representation. Smart [70] shows that every element $d / \mathbb{Z}[\phi]$
has a $\phi$-adic representation with some digit set. More precisely, they show the followings.

**Theorem 3.1 (Frobenius expansion on subfield elliptic curves [70])** Let $E$ be an elliptic curve over $\mathbb{F}_q$, $\phi$ be its $q^{th}$-power Frobenius map of $E$, $t$ is the trace of $\phi$. We assume that $(q, t) = (5, \mp 4), (7, \mp 5)$. Let $d / \mathbb{Z}[\phi]$, then we can write $d = \prod_{i=0}^{\ell-1} c_i \phi^i$, where $c_i \not\in \{0, \mp 1, \ldots, \mp (q-1)/2\}$ and $\ell \geq \left\lceil 2 \log_q 2 \frac{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}{4} \right\rceil$. We denote $\ell_{\phi, \text{EXP}}(d)$ the length of $\phi$-adic expansion of $d / \mathbb{Z}[\phi]$.

From Theorem 3.1, we can compute $dP$ efficiently using a precomputation table $iP \mod (q-1)/2$ and Horner’s method:

$$dP = \left( c_0, c_1, \ldots, c_{\ell-1} \right) \phi \left( c_{\ell-1} \phi(P) + c_{\ell-2}P \right) + \cdots + c_1P + c_0 P.$$

We describe a scalar multiplication algorithm using Horner’s method based on Frobenius expansion with digit set $D_{\phi} = \{0, \mp 1, \mp 2, \ldots, \mp \alpha\}$. The computational cost of Horner’s method using Smart’s Frobenius expansion method (Algorithm 2 with $\alpha = (q-1)/2$, digit set $D_{\phi} = D_{q,(q-1)/2}$) is approximately

$$\frac{q}{2} \ell A_S + \ell T_S \left\{ \begin{array}{l} q = 3, \end{array} \right. \left\{ \begin{array}{l} q = 5, \end{array} \right.$$

where $T_S$ stand for the computational cost of the Frobenius map in the scalar multiplication step. In step 5 of Algorithm 1 and Algorithm 2, if $d_i < 0$, we compute $Q \rightarrow Q \ P_{-d_i}(=Q+P_{d_i})$ and if $d_i = 0$, we do not need to compute $Q \rightarrow Q + P_{d_i}$. This is the reason that smaller non-zero densities in the representations of the scalars improve the efficiency.

### 3.2.3 $\tau$-NAF on binary Koblitz curves and supersingular Koblitz curves

Next, we review $\tau$-NAF on binary Koblitz curves and supersingular Koblitz curves.
Definition 3.4 (τ-NAF on binary Koblitz curves [72]) Let $E_a$ be an elliptic curve defined over $\mathbb{F}_2$ with defining equation as follows.

$$E_a : y^2 + xy = x^3 + ax^2 + 1, \quad a = 0 \text{ or } 1.$$  

It can be proven that these curves are ordinary, and we call these “binary Koblitz curves”. Let us denote $\tau$ the Frobenius map on $E_a$:

$$\tau : E_a \rightarrow E_a, \quad (x, y) \mapsto (x^2, y^2).$$

We can regard $\tau$ as a complex number which satisfies the following characteristic equation: $\tau^2 + \tau + 1 = 0$, $\mu = (1)^{-a}$. A $\tau$-adic NAF representation (τ-NAF) on binary Koblitz curves of an element $d / \mathbb{Z}[\tau]$ is a representation $d = \prod_{i=0}^{\ell-1} e_i \tau^i$ where $e_i / D_{2,1}$, $e_{\ell-1} = 0$ and no two consecutive digits $e_i$ are nonzero. The length of the $\tau$-NAF is $\ell$.

Theorem 3.2 (Properties of τ-NAF [72], [4])

1. Every $d / \mathbb{Z}[\tau]$ has a unique τ-NAF.
2. τ-NAF of $d$ has the smallest Hamming weight with digit set $D_{2,1}$.
3. The average non-zero density of τ-NAF is asymptotically $1/3$ and it has the same non-zero density of ordinary NAF.

Koblitz [47] also proposed another possibility of NAF-like recoding method for Frobenius representations on supersingular elliptic curves $E_a : y^2 = x^3 + ax + x (1)^a / \mathbb{F}_3$, where $a = 0$ or $1$. Similarly to the case of binary Koblitz curves, the uniqueness of τ-NAF on these curves can be proven. Moreover, it is also proven that the non-zero density for this method is $2/5$. It is unknown whether any analogue of τ-NAF exists on another subfield elliptic curves except for binary Koblitz curves and elliptic curves in [11]. But, only a few curves are available for the above methods so far. Our goal is to develop efficient scalar multiplication algorithms on a more general family of subfield elliptic curves. However, because of the narrowness of the digit set of NAF, we may well not be able to directly apply the technique of NAF to the Frobenius expansions except for τ-NAF on binary Koblitz curves. So as to reduce the non-zero densities of the Frobenius expansion, we use redundant digit sets instead of the digit set of NAF. In these
cases, it is necessary to know the non-zero densities and the maximum lengths of the proposed methods. In the next section, we will propose two classes of $\phi$-NAF on a family of subfield elliptic curves. In the following, we call these two classes of $\phi$-NAF $\phi$-GNAF and $\phi$-rNAF, respectively.

3.3 Proposed Methods (Two classes of $\phi$-NAF)

In this section, we investigate how to expand two classes of $\phi$-NAF on a family of subfield elliptic curves. We call the family of subfield elliptic curves “$\mathbb{F}_q$-Koblitz curves”.

**Definition 3.5 ($\mathbb{F}_q$-Koblitz curves)** Let $p$ be a prime, $q = p^r$ a power of $p$, and $\mathbb{F}_q$ the finite field with $q$-elements. Let $E$ be an elliptic curve defined over $\mathbb{F}_q$ which is given by a Weierstrass equation

$$
\equiv p = 2: y^2 + xy = x^3 + ax^2 + b \quad (a, b / \mathbb{F}_q),
$$

$$
\equiv p = 3: y^2 = x^3 + ax^2 + b \quad (a, b / \mathbb{F}_q),
$$

$$
\equiv p \sim 5: y^2 = x^3 + ax + b \quad (a, b / \mathbb{F}_q).
$$

If the trace of the $q^{th}$-power Frobenius map $t = \mp 1$, we call $E$ “$\mathbb{F}_q$-Koblitz curve”.

In the above definitions, note that for $q = 2$, $\mathbb{F}_2$-Koblitz curves are the above binary Koblitz curves $E_2$. Since the trace of the Frobenius map $\tau$ of $E_{2}$ is $\mu = (1)^{(1-\alpha)} = \mp 1$, $\mathbb{F}_q$-Koblitz curve is a natural generalization of binary Koblitz curve.

In the following, we consider a scalar multiplication for a given integer $d$ and for a given point $P / E(\mathbb{F}_q^m)$, where $q \sim 5$ and $E$ is a $\mathbb{F}_q$-Koblitz curve. In this case, it satisfies that

$$
\phi^2 \circ \phi + q = 0 \quad (1, \emptyset, q) = 0^1,
$$

and by easy calculation, we have

$$
\pm \phi^3 + (q \quad 1)\phi^2 + q^2 = 0 \quad (\mp 1, q \quad 1, 0, q^2)_\phi = 0^1.
$$
The property of being anomalous depends on the base field. If $E$ is anomalous over $\mathbb{F}_q$, it is not necessarily anomalous over any $\mathbb{F}_{q^m}$ for $m \sim 2$. For details, refer to [76]. Weil descent attack (or GHS attack) in the case of odd characteristic [13] is presented in [22], [23], etc. In order to keep the security, we should avoid the extension degree in [22], [23], [2], etc. For details, refer to [2].

For all $P / E(\mathbb{F}_q^m), (\phi^m 1)P = \emptyset$ is satisfied. Hence $dP = (d \mod (\phi^m 1))P$ for any integer scalar $d$. From [70], there exist $Q, d' \in \mathbb{Z}[\phi]$ such that $d = Q(\phi^m 1) + d'$ with $d' = 0$ or $\Psi(d') < \lambda \Psi(\phi^m 1)$, where $\Psi$ is a multiplicative function. Note that this provides a 50% improvement in the performance thanks to a shorter length of the Frobenius expansions of $d$. In this chapter, “the Frobenius expansion” means the expansion in [70].

Here, we review the following Lemma to prove the uniqueness of $\phi$-GNAF and $\phi$-rNAF. For details, refer to [2].

**Lemma 3.1 (Divisibility by $\phi$ [2])** Let $\alpha = a + b\phi / \mathbb{Z}[\phi]$ $(a, b / \mathbb{Z})$, then we have

$$\phi \mid \alpha \Rightarrow a \mid \phi \mid a.$$  

In particular, for rational integer $a / \mathbb{Z}$, we have $\phi \mid a \Rightarrow a \mid \phi \mid a$.

### 3.3.1 $\phi$-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$

At first, we show how to expand the multiplication by $d$ map on $E(\mathbb{F}_{q^m})$ in terms of $\phi$-GNAF and prove some properties of this. In this subsection, we assume that $t = 1$. We begin with the definition of $\phi$-GNAF on every subfield elliptic curves, and give an algorithm which computes $\phi$-GNAF for a given $d / \mathbb{Z}[\phi]$, where $q \sim 7$.

**Definition 3.6 ($\phi$-GNAF)** Let $E$ be a $\mathbb{F}_q$-Koblitz curve and $d / \mathbb{Z}[\phi]$. A $\phi$-adic GNAF ($\phi$-GNAF) of $d$ on $E$ is a representation $d = \prod_{i=0}^{\ell-1} e_i \phi^i$ where $e_i / D_{q, \alpha-1}$ for each $i$, $e_{\ell-1} = 0$, and one of the followings holds

1. $e_i e_{i+1} = 0$,
2. if $e_i e_{i+1} > 0$, then $e_{i+1} + e_i \mid q$,
3. if $e_i e_{i+1} < 0$, then $e_{i+1} \mid e_i$.
Let \( a, b \in D_{q-1} \). If a pair \((a, b)_\phi\) satisfies one of the followings: (1) \( ab = 0 \), (2) if \( ab > 0 \), then \( |a + b| < q \), (3) if \( ab < 0 \), then \( |a| > |b| \) then we call the pair \((a, b)_\phi\) a \( \phi \)-admissible pair. \(^{\ast 1}\) Otherwise, we call the pair \((a, b)_\phi\) a \( \phi \)-non-admissible pair. We denote \( \ell_{\phi \text{-GNAF}}(d) \) the length of \( \phi \text{-GNAF} \) of \( d / \mathbb{Z}[\phi] \).

Note that the condition that \( d = \prod_{i=0}^{\ell_{\phi \text{-GNAF}}-1} e_i \phi^i / \mathbb{Z}[\phi] \) is a \( \phi \text{-GNAF} \) is equivalent to the condition that any adjacent digits \((e_i, e_{i+1})_\phi\) are \( \phi \)-admissible.

**Example 3.1 (Example of Frobenius expansion and \( \phi \text{-GNAF} \))** Let \( q = 7 \), \( d = 314 / \mathbb{Z}[\phi] \). By easy calculation, we have the Frobenius expansion \( d = (1, 3, 3, 2, 3, 3, 1)_\phi \). Using Algorithm 3, we compute the \( \phi \text{-GNAF} \) of \( d = 314 \). We scan consecutive two digits of the Frobenius expansion of \( d \) from right to left.

In the first step of the main loop of Algorithm 3, we scan \((3, 1)_\phi\). This pair is a \( \phi \)-admissible pair. We set \( e_0 = 1 \).

In the second step, we scan \((3, 3)_\phi\). This pair is also a \( \phi \)-admissible pair. We set \( e_1 = 3 \).

In the third step, we scan \((2, 3)_\phi\). This pair is also a \( \phi \)-admissible pair. We set \( e_2 = 3 \).

In the fourth step, we scan \((3, 2)_\phi\). This pair is also a \( \phi \)-admissible pair. We set \( e_3 = 2 \).

In the fifth step, we scan \((3, 3)_\phi\). This pair is a \( \phi \)-non-admissible pair and \( b_4 = 3 < 0 \). We set \( b_5 = 1 + 1 = 0, \ b_5 = 3, \ e_4 = 3 + 7 = 4 \).

In the sixth step, we scan \((0, 2)_\phi\). This pair is a \( \phi \)-admissible pair. We set \( e_5 = 2 \).

Hence, we obtain the \( \phi \text{-GNAF} \) of \( d = (2, 4, 2, 3, 3, 1)_\phi \) (We omit the leading 0’s). We can see that in the above example the Hamming weight of the \( \phi \text{-GNAF} \) of \( d \) is smaller than that of the Frobenius expansion of \( d \).

\(^{\ast 1}\) In the case that \( \phi \) is a positive integer, a similar notation can be found in [54].
Algorithm 3 $\phi$-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ ($q \sim 7$)

**Input:** $d / \mathbb{Z}[\phi]$

**Output:** $\phi$-GNAF of $d$

1. $d'^{'} \rightarrow d \mod (\phi^m - 1)$
2. $\ell \rightarrow \left\lceil 2 \log_q 2 \cdot \frac{N_{\mathbb{Z}[\phi] / \mathbb{Z}(d)}}{\sqrt{N}} \right\rceil + 4$
3. Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of $d'$
4. $b_0 \rightarrow c_0, b_1 \rightarrow c_1, b_\ell \rightarrow 0, b_{\ell+1} \rightarrow 0$
5. $i \rightarrow 0$
6. **while** $i \geq \ell$ **do**
   7. **if** $(b_{i+1}, b_i) : \phi$-admissible pair **then**
   8. $b_{i+2} \rightarrow c_{i+2}, e_i \rightarrow b_i$
   9. **else** if $b_i > 0$ **then**
   10. $b_{i+2} \rightarrow c_{i+2} - 1, b_{i+1} \rightarrow b_{i+1} + 1, e_i \rightarrow b_i + q$
   11. **else**
   12. $b_{i+2} \rightarrow c_{i+2} + 1, b_{i+1} \rightarrow b_{i+1} + 1, e_i \rightarrow b_i + q$
   13. **end if**
   14. $i \rightarrow i + 1$
7. **end while**
16. **return** $(c_{\ell+1}, e_\ell, \ldots, c_1, c_0)_\phi$

The following lemma and theorem show the correctness of Algorithm 3, thus the existence of $\phi$-GNAF. From the lemma, for any given Frobenius expansion, we can have a sequence with digits in $D_{q^2-1}$ such that any adjacent digits are $\phi$-admissible. Moreover, the theorem below gives the finiteness of the sequence (hence $\phi$-GNAF) and evaluates the upper bound of the length of $\phi$-GNAF. For the proof of the lemma, it is easily seen that the proof of [54, Theorem 12.2.3] can be applied. For details, refer to [54].

**Lemma 3.2** Let $b / D_{q,(q+1)/2}, b' / D_{q,(q-1)/2}$, and $e / D_{q,q-1}$. We assume that $(b', e)_\phi$ is a $\phi$-admissible pair and $(b, b')_\phi$ is a $\phi$-non-admissible pair. If we convert

$$(b, b', e)_\phi \uplus \begin{cases} (1, c, c', c)_\phi := (1, b + 1, b', q, e)_\phi & \text{if } b' > 0, \\
(1, c, c', e)_\phi := (1, b, 1, b', q, e)_\phi & \text{otherwise}, \end{cases}$$

then...
then \((c, c')_\phi, (c', e)_\phi\) are \(\phi\)-admissible pairs.

**Theorem 3.3 (Maximum length of \(\phi\)-GNAF)** Let \(d \in \mathbb{Z}[\phi]\) and \(\ell = \ell_{\phi-\text{EXP}}(d)\) be the length of Frobenius expansion of \(d\). Then \(d\) has a \(\phi\)-GNAF with digit set \(D_{q,q-1}\) such that the length is at most \(\ell + 2\).

**Proof** Let \(\ell' = \ell_{\phi-\text{GNAF}}(d)\) and \(\ell = \ell_{\phi-\text{EXP}}(d)\). In the process of Algorithm 3, we will have a representation \(d = (c, b, b', e_{\ell-4}, \ldots, e_0)_\phi\), where \(c_i / D_{q,q-1}, c / D_{q,(q-1)/2}, b / D_{q,(q+1)/2}, b' / D_{q,(q+3)/2}\), and \((b', e_{\ell-4}, \ldots, e_1, e_0)_\phi\) is a \(\phi\)-GNAF. Now we scan the two digits \((b, b')_\phi\). If the pairs \((c, b)_\phi\) and \((b, b')_\phi\) are \(\phi\)-admissible, there is nothing to do and \(\ell' = \ell\). So we can assume that \((b, b')_\phi\) is \(\phi\)-non-admissible. Then we convert

\[
(c, b, b')_\phi \leftarrow (\tilde{c}, c', e_{\ell-3})_\phi := \begin{cases} 
(c - 1, b + 1, b' - q)_\phi & \text{if } b' > 0, \\
(c + 1, b - 1, b' + q)_\phi & \text{otherwise}.
\end{cases}
\]

If \((\tilde{c}, c')_\phi\) is \(\phi\)-admissible, then \(\ell' = \ell\). Otherwise, we convert

\[
(\tilde{c}, c')_\phi \leftarrow (a, a', e_{\ell-2})_\phi := \begin{cases} 
(1, \tilde{c} + 1, c' - q)_\phi & \text{if } c' > 0, \\
(1, \tilde{c} - 1, c' + q)_\phi & \text{otherwise},
\end{cases}
\]

then \(a = 1\) or \(\bar{a} / D_{q,(q+3)/2}\). If \((a, a')_\phi\) is \(\phi\)-admissible, then \(\ell' = \ell + 1\). If \((a, a')_\phi\) is \(\phi\)-non-admissible, then by the definition of \(\phi\)-GNAF, it is obvious that \(a = 1, a' < 0\) or \(a = \bar{a}, a' > 0\). In this case, we convert

\[
(a, a')_\phi \leftarrow \begin{cases} 
(1, 0, a' - q)_\phi & a' > 0, \\
(1, 0, a' + q)_\phi & a' < 0.
\end{cases}
\]

Then \(\ell' = \ell + 2\). Therefore \(\ell_{\phi-\text{GNAF}}(d)\) is at most 2 more than \(\ell_{\phi-\text{EXP}}(d)\). \(\square\)

We can also extend Algorithm 3 in the case of \(q = 3\) or 5. However, in this case, there is a possibility that \(b_i\) is a non-zero multiple of \(q\). If \(b_i\) is a non-zero multiple of \(q\), we convert \((b_{i+1}, b_i)_\phi \leftarrow (b_i/q, b_{i+1} + b_i/q, 0)_\phi\). It is easy to show that if \(b_i\) is a non-zero multiple of \(q\), then it satisfies that \(b_i = \pm q\). Thus for all \(i\), we always have \(|b_{i+1}| \geq (q + 1)/2, |b_i| \geq (q + 3)/2\), as Algorithm 3 (or Theorem 3.3). Remark that it does not occur that \(b_{i+1}\) is a non-zero multiple of \(q\). This shows the correctness of Algorithm 4.

21
Algorithm 4 $φ$-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ ($q = 3$ or $q = 5$)

Input: $d / \mathbb{Z}[φ]$

Output: $φ$-GNAF of $d$

1: $d'$ $→$ $d$ mod ($φ^n = 1$)
2: $ℓ$ $→$ $\left\lfloor 2 \log_q 2 \frac{N_{\mathbb{Z}[φ]/\mathbb{Z}}(d)}{φ} + 4 \right\rfloor$
3: Compute the Frobenius expansion $(c_{ℓ−1}, c_{ℓ−2}, \ldots, c_1, c_0)_{φ}$ of $d'$
4: $b_0 \rightarrow c_0, b_1 \rightarrow c_1, b_ℓ \rightarrow 0, b_{ℓ+1} \rightarrow 0$
5: $i \rightarrow 0$
6: while $i ≥ ℓ$ do
7:  if $b_i$ mod $q = 0$ then
8:     $b_{i+1} \rightarrow b_{i+1} + b_i/q, b_{i+2} \rightarrow c_{i+2} + b_i/q, e_i \rightarrow 0$
9:  else if $(b_{i+1}, b_i) : φ$-admissible pair then
10:     $b_{i+2} \rightarrow c_{i+2}, e_i \rightarrow b_i$
11:  else if $b_i > 0$ then
12:     $b_{i+2} \rightarrow c_{i+2} + 1, b_{i+1} \rightarrow b_{i+1} + 1, e_i \rightarrow b_i + q$
13:  else
14:     $b_{i+2} \rightarrow c_{i+2} + 1, b_{i+1} \rightarrow b_{i+1} + 1, e_i \rightarrow b_i + q$
15:     end if
16:     $i \rightarrow i + 1$
17:  end while
18: return $(e_{ℓ+1}, e_ℓ, \ldots, e_1, e_0)_{φ}$

Let $φ$-GNAF$_ℓ$ be the set of $φ$-GNAF of length $ℓ$. We put $A_ℓ = \#φ$-GNAF$_ℓ$, $S_ℓ = \prod_{d ∈ φ$-GNAF$_ℓ}(ℓ \cdot w(d))$, and $C_ℓ = \# \{d / φ$-GNAF$_ℓ \setminus w(d) = ℓ \}$, where $w(d)$ means the Hamming weight of $d$. In other words, $C_ℓ$ is the number of $φ$-GNAF with length $ℓ$ such that all digits are non-zero. Then the non-zero density of $φ$-GNAF is defined by

$$\lim_{ℓ → ∞} \frac{S_ℓ}{(ℓA_ℓ)}.$$

$φ$-GNAF has properties same as GNAF except for the minimality of the Hamming weight. Although $φ$-GNAF does not have minimal Hamming weight, asymptotic average non-zero density of $φ$-GNAF is same as GNAF. This indicates that the average number of point additions required by the $φ$-GNAF is smaller than
that of point additions required by the Frobenius expansion.

**Proposition 3.2 (Properties of \(\phi\)-GNAF)**

1. Every \(d \in \mathbb{Z}[\phi]\) has a unique \(\phi\)-GNAF.
2. The average density of non-zero digits among \(\phi\)-GNAF’s of length \(\ell\) is equal to \((q - 1)/(q+1) + (3q - 1)/(q+1)^2\ell + O(q^{-\ell})\). In particular, for a fixed \(q\), the average non-zero density of \(\phi\)-GNAF is asymptotically \((q - 1)/(q+1)\).
3. \(\phi\)-GNAF does not have minimal Hamming weight among all Frobenius expansion with digit set \(D_{q,q-1}\).

**Proof.** (1) We suppose that \(d \in \mathbb{Z}[\phi]\) has two such representations

\[
d = \sum_{i=0}^{\ell-1} e_i \phi^i = \sum_{i=0}^{\ell-1} e'_i \phi^i.
\]

If \(i_0\) is the smallest number such that \(e_i = e'_i\) for \(0 \leq i \leq i_0\) and \(e_{i_0} = e'_{i_0}\), then we can replace \(d\) by \((d \prod_{i=0}^{i_0} e_i \phi^i)/\phi^{i_0} = \prod_{i=0}^{\ell-1-i_0} e_i \phi^{i-i_0}\), and we have the representations

\[
d = \sum_{i=0}^{\ell-1} e_i \phi^i = \sum_{i=0}^{\ell-1} e'_i \phi^i, \quad e_0 = e'_0,
\]

where \(\ell\) is the maximum of length of the two representations. Then from Lemma 3.1, we have \(q \not| (e_0 - e'_0)\), and by the assumption, \(\not| (e_0 - e'_0) \geq q - 1\), hence the inequality \(\not| (e_0 - e'_0) \geq q - 1\) is satisfied. Therefore we have \(e_0 = e'_0 = \pm q\). In the following, we will give a proof for the case \(e_0 = e'_0 = q\). For another case, we can prove similarly, thus we omit it.

Since \(e_0 = e'_0 = q\) and \(e_0, e'_0 \in D_{q,q-1}\), we have \(1 \geq e_0 \geq q\) and \((q - 1) \geq e'_0 \geq 1\). We also have \(q = (e_0 - e'_0) = \prod_{i=1}^{\ell-1} (e'_i - e_i) \phi^i\) and by the characteristic polynomial of \(\phi\), we have \(\phi \phi^2 = \prod_{i=1}^{\ell-1} (e'_i - e_i) \phi^i\). Thus we can see

\[
(e_1 - e'_1 + 1) = (e'_2 - e_2 + 1) \phi + \sum_{i=3}^{\ell-1} (e'_i - e_i) \phi^{i-1}.
\]

Hence, from Lemma 3.1, we also have \(q \not| (e_1 - e'_1 + 1)\). It follows that \(e'_1 \not| (e_1 - e'_1 + 1)\). We shall see that in each case, there is a contradiction. (For details, refer to [54, p.177]).
Case 1. Suppose that $e'_1 = e_1 + 1$. We have $(q \ 1) \geq e'_1 \geq 0$ and $0 \geq e_1 \geq (q \ 1)$. We assume that $(q \ 1) \geq e'_1 \geq 1$. Then from $e'_0 e'_1 > 0$, we have $e'_0 \ e'_1 \geq q \ 1$. By substituting $e_0$ for $e'_0$ and $e_1$ for $e'_1$, we obtain $e_0 + e_1 \sim g$. On the other hand, since $e_0 > 0$ and $e_1 \sim 0$, we have $e_0 + e_1 \geq q \ 1$. This is a contradiction. If $e'_1 = 0$, then $e_1 = q \ 1$. From $e_0 \sim 1$, we have $e_0 + e_1 \sim q$. This is a contradiction.

Case 2. Suppose that $e'_1 = e_1 + 1$. We have $(q \ 2) \geq e'_1 \geq q \ 1$ and $(q \ 1) \geq e_1 \geq (q \ 2)$. We assume that $2 \geq e'_1 \geq q \ 1$. Then $1 \geq e_1 \geq (q \ 2)$. Since $e'_0 e'_1 < 0$, we have $e_0 + e'_1 > 0$. By substituting $e_0$ for $e'_0$ and $e_1 + 1$ for $e'_1$, we obtain $e'_0 + e'_1 = (e_0 \ q) + e_1 + 1 > 0$. Thus $e_0 + e_1 \sim q$. On the other hand, by $e_0 > 0$, $e_1 > 0$, we have $e_0 + e_1 \geq q \ 1$. This is a contradiction. Next, suppose that $e'_1 = 1$. In this case, $e_1 = 0$. Since $e'_0 e'_1 < 0$, we must have $e'_1 = 1 > e'_0 > 0$. However there does not exist $e'_0$ which satisfies the above inequality. This is a contradiction. Suppose that $e'_1 = 0$. In this case, $e_1 = 1$. By a similar argument as in the case of $e'_1 = 1$, one can obtain that $e'_1 = 1 > e'_0 > 0$. However there does not exist $e'_0$ which satisfies the above inequality. This is a contradiction.

Finally, we assume that $(q \ 2) \geq e'_1 \geq 1$. Then $(q \ 1) \geq e_1 \geq 2$. Since $e'_0 e'_1 > 0$, we have $e'_0 \ e'_1 \geq q \ 1$. By substituting $e_0$ for $e'_0$ and $e_1 + 1$ for $e'_1$, we obtain $e'_0 \ e'_1 = (e_0 \ q) \ (e_1 + 1) \geq q \ 1$. Thus $e_0 + e_1 \sim 0$. On the other hand, by $e_0 > 0$, $e_1 < 0$, we have $e_0 + e_1 < 0$. This is a contradiction.

Case 3. Suppose that $e'_1 = e_1 + 1 + q$. We have $2 \geq e'_1 \geq q \ 1$ and $(q \ 1) \geq e_1 \geq 2$. Since $e'_0 e'_1 < 0$, we have $e'_0 + e'_1 > 0$. By substituting $e_0$ for $e'_0$ and $e_1 + 1 + q$ for $e'_1$, we obtain $e_0 \ q + e_1 + 1 + q > 0$. Hence $e_0 + e_1 \geq 0$. On the other hand, since $e_0 > 0$ and $e_1 < 0$, we have $e_0 + e_1 < 0$. This is a contradiction.

This concludes the proof.

(2) At first, we show that $C_{\ell + 1} = (q \ 2) C_\ell$, $C_1 = 2(q \ 1)$. We fix an element $d = (e_{\ell-1}, \ldots, e_0)_\phi$ of $\mathbb{Z}[\phi]$ which is $\phi$-GNAF such that each $e_i = 0$. For the fixed $d$, we count the number of $e$ which satisfy $1 \geq e \geq q \ 1$, and $(e_{\ell-1}, \ldots, e_0, e)_\phi$ is $\phi$-GNAF. We can assume that $e_0 > 0$. If $e_0 e > 0$, $e$ satisfy $1 \geq e \geq q \ 1$, $e_0$ and otherwise $1 \geq e < e_0$. Thus we have that the number of $e$ is $q \ 2$, hence the recurrence equation for $C_\ell$. It is trivial that $C_1 = 2(q \ 1)$.
Next, we show that $A_\ell$ and $S_\ell$ satisfy the recurrence equations $A_1 = 2(q \ell + 1)$, $A_2 = 2(q\ell + 1)^2$, $A_{\ell+2} = (q\ell + 1)A_{\ell+1} + qA_\ell = 0 \ (\ell \sim 1)$, and $S_1 = 0, S_2 = 2(q\ell + 1)$, $S_{\ell+2} = (q\ell + 1)S_{\ell+1} + qS_\ell = 2(q\ell + 1)
1 + \frac{2(q\ell + 1)}{(q + 1)}q(\frac{q}{q + 1})^1 \frac{(1)^\ell}{2} 1$

Table 3.1: Forms of $\ell$ digits $\phi$-GNAF representations

<table>
<thead>
<tr>
<th>Forms</th>
<th># of representations</th>
<th># of 0 digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>($\pm0 \pm \cdots$)</td>
<td>$C_1A_{\ell-2}$</td>
<td>$C_1A_{\ell-2} + C_1S_{\ell-2}$</td>
</tr>
<tr>
<td>($\pm00 \pm \cdots$)</td>
<td>$C_1A_{\ell-3}$</td>
<td>$2C_1A_{\ell-3} + C_1S_{\ell-3}$</td>
</tr>
<tr>
<td>($\pm000 \pm \cdots$)</td>
<td>$C_1A_{\ell-4}$</td>
<td>$3C_1A_{\ell-4} + C_1S_{\ell-4}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>($\pm0000 \pm \cdots$)</td>
<td>$C_2A_{\ell-3}$</td>
<td>$C_2A_{\ell-3} + C_2S_{\ell-3}$</td>
</tr>
<tr>
<td>($\pm0000 \pm \cdots$)</td>
<td>$C_2A_{\ell-4}$</td>
<td>$2C_2A_{\ell-4} + C_2S_{\ell-4}$</td>
</tr>
<tr>
<td>($\pm0000 \pm \cdots$)</td>
<td>$C_2A_{\ell-5}$</td>
<td>$3C_2A_{\ell-5} + C_2S_{\ell-5}$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>($\pm0000 \pm \cdots$)</td>
<td>$C_2A_{\ell-4}$</td>
<td>$\ell(4)C_2A_{\ell-4} + C_2S_{\ell-4}$</td>
</tr>
<tr>
<td>($\pm0000 \pm \cdots$)</td>
<td>$C_2A_{\ell-5}$</td>
<td>$\ell(3)C_2A_{\ell-5} + C_2S_{\ell-5}$</td>
</tr>
<tr>
<td>($\pm0000 \pm \cdots$)</td>
<td>$C_2A_{\ell-6}$</td>
<td>$\ell(2)C_2A_{\ell-6} + C_2S_{\ell-6}$</td>
</tr>
</tbody>
</table>

In Table 3.1, the symbol ‘$\pm$’ means a non-zero digit. We explain how to interpret Table 3.1. The left column stands for forms of $\ell$ digits $\phi$-GNAF, the central column stands for the number of each $\phi$-GNAF corresponding to each form of the left column, and the right column stands for each number of 0 digits corresponding to each form of the left column. From Table 3.1, we can derive
\[
A_\ell = \left( \sum_{j=1}^{\tau-2} C_j \right)^{\ell=1-j} A_i + 1 + C_{\ell-1} + C_\ell,
\]
\[
S_\ell = \left( \sum_{j=1}^{\tau-2} C_j \right)^{\ell=1-j} S_i + \frac{(\ell \ i \ j)A_i}{i=1} + \frac{\tau-1}{i=1} (\ell \ i)C_i.
\]

From these equations, we have the desired recurrence equations, and by elementary calculations, we have \( A_\ell = 2(q - 1) q^\ell \ (1)^{\ell+1}/(q + 1) \), and
\[
S_\ell = \left( \frac{2(q - 1)}{(q + 1)^2} \right) (2q^\ell - (q - 1)(1)^\ell) \frac{3q - 1}{q + 1}(q^\ell - (1)^\ell).
\]

From the equations of \( A_\ell, S_\ell \), the average number of non-zero digits for \( \ell \) digits numbers in \( \mathbb{Z}[\phi] \) is equal to \( \ell \ S_\ell/A_\ell \). We put
\[
f(\ell) := \left( \frac{S_\ell}{\ell A_\ell} \right) \left( \frac{q}{q + 1} \right) = \frac{3}{q + 1} - \frac{1}{(q + 1)^2} \ell.
\]

Then
\[
f(\ell) = \frac{q}{q + 1} \times \frac{3}{q + 1} - \frac{1}{(q + 1)^2} \ell.
\]

Let us take \( g(\ell) := q^{-\ell} \). Since it satisfies that
\[
\frac{f(\ell)}{g(\ell)} = \frac{q}{q + 1} \times \frac{3}{1} - \frac{1}{(q - 1)} \ell
\]
we obtain
\[
\frac{f(\ell)}{g(\ell)} = \frac{q}{q + 1} \times \frac{3}{1} \frac{1}{(q - 1) \ell} \leq \frac{q}{q + 1} \frac{3}{1} \ell \ (\ell \leq \in).
\]

Hence \( f(\ell) = O(g(\ell)) \). Thus \( \lim_{\ell \to \infty} S_\ell/(\ell A_\ell) = 1 \ 2/(q+1) = (q - 1)/(q+1) \).
Hence we have the desired result.

(3) We give a counter example that is an element \( \alpha / \mathbb{Z}[\phi] \) that does not have minimal Hamming weight. Let \( q \) is a power of odd prime \( p \), and \( \alpha = \phi \ (q - 1)/\mathbb{Z}[\phi] \). An Frobenius expansion of \( \alpha \) with digit set \( D_{q,q-1} \) is \( (1,1,q)_\phi \) and its Hamming weight is 2. But \( \phi \)-GNAF of \( \alpha \) is \( (1,0,q,2,1)_\phi \) and its Hamming weight is 3. Therefore \( \phi \)-GNAF does not have minimal Hamming weight among all Frobenius expansion with digit set \( D_{q,q-1} \). \( \Box \)
Algorithm 5 computes the $\phi$-GNAF for a given $d / \mathbb{Z}[\phi]$ without the calculation of the Frobenius expansion of $d$ and reduces the memory consumption to calculates the $\phi$-GNAF compared to Algorithm 3 and Algorithm 4. From Theorem 3.3 (especially the finiteness of the length of $\phi$-GNAF), it is easy to show the correctness of Algorithm 5.

**Algorithm 5 $\phi$-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$**

**Input:** $d / \mathbb{Z}[\phi]$

**Output:** $\phi$-GNAF of $d$

1: $d' := d_0 + d_1 \phi \rightarrow d \mod (\phi^n - 1)(d_0, d_1 / \mathbb{Z})$

2: $\ell \rightarrow \lceil 2 \log_2 \sqrt{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)} \rceil + 4$

3: Compute $Q, b / \mathbb{Z}$ such that $d_0 = Qq + b (b / D_q(q-1)/2)$ (see [70])

4: $d_0 \rightarrow Q + d_1, d_1 \rightarrow Q$

5: $i \rightarrow 1$

6: **while** $i \geq \ell + 1$ **do**

7: Compute $Q, a / \mathbb{Z}$ such that $d_0 = Qq + a (a / D_q(q-1)/2)$

8: $d_0 \rightarrow Q + d_1, d_1 \rightarrow Q$

9: if $(a, b) : \phi$-admissible pair then

10: $e_{i-1} \rightarrow b, b \rightarrow a$

11: else if $b > 0$ then

12: $e_{i-1} \rightarrow b, q, b \rightarrow a + 1, d_0 \rightarrow d_0 + 1$

13: else

14: $e_{i-1} \rightarrow b + q, b \rightarrow a + 1, d_0 \rightarrow d_0 + q$

15: **end if**

16: **end while**

17: **return** $(e_{\ell+1}, e_\ell, \ldots, e_1, e_0)_\phi$

If we compute $(i + 1)P = P + iP$ for $1 \geq i \geq q - 2$, the computational cost of Horner’s method using $\phi$-GNAF (Algorithm 2 based on $\phi$-GNAF with $\alpha = q - 1$, digit set $D_\phi = D_{q,q-1}$) is approximately

$$\left[ D_\phi + \frac{q}{q+1} \ell A_8 + \ell F_8 \left\{ (q = 3), D_\phi + (q - 3) A_3 + \frac{q}{q+1} \ell A_8 + \ell F_8 \left\{ (q \sim 5). \right. \right. \right.$$
3.3.2 \( \phi \text{-rNAF on } \mathbb{F}_q \)-Koblitz curves of \( t = 1 \)

Next, we show how to expand the multiplication by \( d \) map on \( E(\mathbb{F}_q^m) \) in terms of \( \phi \)-adic \( \text{rNAF} \) and prove some properties of this. As with the previous subsection, we assume that \( t = 1 \) and we begin with the definition of \( \phi \)-\( \text{rNAF} \) on every subfield elliptic curves, and give an algorithm which computes the \( \phi \)-\( \text{rNAF} \) for a given \( d \) / \( \mathbb{Z}[\phi] \), where \( q \sim 7 \).

**Definition 3.7 (\( \phi \)-rNAF)** Let \( E \) be a \( \mathbb{F}_q \)-Koblitz curve and \( d \) / \( \mathbb{Z}[\phi] \). A \( \phi \)-adic \( \text{rNAF} \) (\( \phi \)-\( \text{rNAF} \)) of \( d \) on \( E \) is a representation \( d = \prod_{i=0}^{\ell-1} e_i \phi^i \) such that \( e_i \) / \( D_{q,((q^2-1)/2)} \), \( e_{\ell-1} = 0 \) and \( e_{i+1}e_i = 0 \) for each \( i \). Let \( a, b \) / \( D_{q,((q^2-1)/2)} \). If \( ab = 0 \), we call \((a, b)\phi \) \( \phi \)-non-adjacent pair. Otherwise, we call \((a, b)\phi \) \( \phi \)-adjacent pair. We denote \( \ell_{\phi, \text{rNAF}}(d) \) the length of \( \phi \)-\( \text{rNAF} \) of \( d \) / \( \mathbb{Z}[\phi] \).

Note that the condition that \( d = \prod_{i=0}^{\ell-1} e_i \phi^i \) / \( \mathbb{Z}[\phi] \) is a \( \phi \)-\( \text{rNAF} \) is equivalent to the condition that any adjacent digits \((e_{i+1}, e_i)\phi \) are \( \phi \)-non-adjacent.

**Example 3.2 (Example of Frobenius expansion and \( \phi \)-rNAF)** As in Example 3.1, let us consider \( q = 7 \) and \( d = 314 \) / \( \mathbb{Z}[\phi] \). The Frobenius expansion of \( d \) is \((\bar{1}, 3, \bar{3}, 2, 3, 3, \bar{1})\phi \). Using Algorithm 6, we compute the \( \phi \)-\( \text{rNAF} \) of \( d = 314 \). We scan consecutive two digits of the Frobenius expansion of \( d \) from right to left.

In the first step of the main loop of Algorithm 6, we scan \((3, \bar{1})\phi \). This pair is a \( \phi \)-adjacent pair and \( \begin{array}{l} 3 \times 7 \; \bar{1} \end{array} = 20 \ < 2^4 \). We set \( b_3 = 2, b_2 = 3 + 3 = 6, e_1 = 0, e_0 = 3 \times 7 \; 1 \) = 20.

In the second step, we scan \((2, 6)\phi \). This pair is also a \( \phi \)-adjacent pair and \( \begin{array}{l} 2 \times 7 + 6 \end{array} = 20 \ < 2^4 \). We set \( b_5 = 3, b_4 = 3 + 2 = 1, e_3 = 0, e_2 = 2 \times 7 + 6 = 20 \).

In the third step, we scan \((3, \bar{1})\phi \). This pair is also a \( \phi \)-adjacent pair and \( \begin{array}{l} 3 \times 7 \; \bar{1} \end{array} = 20 \ < 2^4 \). We set \( b_7 = 0, b_6 = 1 + 3 = 2, e_5 = 0, e_4 = 3 \times 7 \; 1 \) = 20.

In the fourth step, we scan \((0, 2)\phi \). This pair is a \( \phi \)-non-adjacent pair. We set \( e_6 = 2 \).

Hence, we obtain the \( \phi \)-\( \text{rNAF} \) of \( d = (2, 0, 20, 0, 20, 0, 0) \) (we omit the leading 0’s). We can see that in the above example the Hamming weight of the \( \phi \)-\( \text{rNAF} \) of \( d \) is smaller than that of the Frobenius expansion of \( d \). Moreover, the Hamming
weight of the $\phi$-rNAF of $d$ is also smaller than that of the $\phi$-GNAF of $d$ (See Example 3.1).

**Algorithm 6** $\phi$-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ ($q \sim 7$)

**Input:** $d / \mathbb{Z}[\phi]$

**Output:** $\phi$-rNAF of $d$

1: $d' \rightarrow d \mod (\phi^m - 1)$

2: $\ell \rightarrow \left \lceil 2 \log_q 2 \frac{N_{\mathbb{Z}[\phi]/\mathbb{Z}(d)} \sqrt{q + 4}}{q^2} \right \rceil$

3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of $d'$

4: $b_0 \rightarrow c_0, b_1 \rightarrow c_1, b_\ell \rightarrow 0, b_{\ell+1} \rightarrow 0, b_{\ell+2} \rightarrow 0, b_{\ell+3} \rightarrow 0$

5: $i \rightarrow 0$

6: while $i \geq \ell + 2$ do

7: if $b_i = 0$ then

8: $b_{i+2} \rightarrow c_{i+2}, e_i \rightarrow 0, i \rightarrow i + 1$

9: else if $b_{i+1}q + b_i \geq 0$ then

10: $b_{i+3} \rightarrow c_{i+3}, b_{i+2} \rightarrow c_{i+2} + b_{i+1}, e_i \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i, i \rightarrow i + 2$

11: else if $b_{i+1}q + b_i < (q^2 - 1)/2$ then

12: $b_{i+3} \rightarrow c_{i+3} + 1, b_{i+2} \rightarrow c_{i+2} + b_{i+1} + (q - 1), e_i \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i + q^2, i \rightarrow i + 2$

13: else

14: $b_{i+3} \rightarrow c_{i+3} + 1, b_{i+2} \rightarrow c_{i+2} + b_{i+1} + (q - 1), e_i \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i + q^2, i \rightarrow i + 2$

15: end if

16: end while

17: return $(c_{\ell+3}, c_{\ell+2}, \ldots, c_1, c_0)_\phi$

At first sight, it seems that Algorithm 6 is a straightforward combination of the Frobenius expansions and rNAF. But the proof of the correctness of Algorithm 6 is complicated. To prove this, we focus on the fact that in Algorithm 6, the following conversion does not occur:

$$(b_{i+1}, b_i)_\phi \leftrightarrow \begin{cases} \left( \frac{b_i' q}{\mathbb{Z}}, 0 \right)_\phi & \text{for some } b_i' / \mathbb{Z} \\ \left( \frac{b_i + b_{i+1} q}{\mathbb{Z}}, b_i \right)_\phi & \text{for some } b_{i+1} / \mathbb{Z} \end{cases}.$$

29
In other words, there is no possibility that \( b_i \) or \( b_{i+1} \) is a non-zero multiple of \( q \) when we scan \((b_{i+1}, b_i)\). More precisely, the following lemma is satisfied. For the proof of the lemma, consult [75].

**Lemma 3.3** Let \( c, c', c'' / D_{q,(q-1)/2}, b / D_{q,(q+1)/2}, b' / D_{q,2q-1} \). We convert \((c, c', c'', b, b')\) from right-to-left according to the following rule and we denote the result of the conversion \((a, a', e, e', e''')\).

**The rule:** We assume that we scan consecutive two digits \((a, b)\), then

**Rule 1.** If \( a = 0, b = 0 \), then convert \((a, b)\)

\[
(a, b) \begin{cases} 
(a, 0, aq + b) & \text{if } |aq + b| \leq \frac{(q^2 - 1)}{2}, \\
(1, a + (q - 1), 0, (aq + b) + q^2) & \text{else if } \frac{(q^2 - 1)}{2} < |aq + b|, \\
(1, a + (q - 1), 0, |aq + b| - q^2) & \text{otherwise}.
\end{cases}
\]

**Rule 2.** If \( a = 0, b = 0 \), then skip the 1-digit \( b \). We scan the next consecutive two digits which include \( a \).

**Rule 3.** if \( a = 0, b = 0 \), then skip the 2-digits \( a \) and \( b \). We scan the next consecutive two digits which do not include \( a \).

Then, it always satisfies that \((c, c', c'', e''')\) is a \( r \)-NAF, \( a / D_{q,(q+1)/2}, a' / D_{q,2q-1} \). In particular, \( a \) and \( a' \) are not divisible by \( q \).

**Proof** We apply the rule in the statement of Lemma 3.3 to \((c, c', c'', b, b')\). We denote the conversion of \( i \)-th digit and \((i+1)\)-st digit by \( b \) and the conversion of \((i+2)\)-nd digit and \((i+3)\)-rd digit by \( \# \). We assume \( a' \) is a non-zero multiple of \( q \). From \( |a'| \geq 2q \), we have \( a' = q \) or \( a' = q \). It suffices to consider the case of \( a' = q \) (In the same way as the case of \( a' = q \), we can also prove the case of \( a' = q \)). It is easy to understand the following : the situations which have the possibility of \( a' \) is a non-zero multiple of \( q \), is \((\# b) = ((1), (2)), ((2), (1)), ((2), (2)), ((3), (2))\), where \((1)\) means that we apply rule 1 in the statement of Lemma 3.3 to 2 digits that we scan and so on. From now on, we investigate each situation.

**Case 1.** \((\# b) = ((1), (2))\). It is easy to show that \( a' = c + c' + (q - 1) \), i.e. \( c' = c + 1 \). We apply the rule to \((c, c', c'', b, b')\). We have \( e' = c + q + e'' + b + q^2 \) and it is easy to show that \( e' > (q^2 - 1)/2 \). This is contrary to \( |e'| \geq (q^2 - 1)/2 \).
Case 2. $(\sharp, b) = ((2), (1))$. It is easy to show that $a' = c + c' + 1$, i.e. $c = c' = (q 1)/2$. We apply the rule to $(c, c', c'' ,b, b')_\phi$, We have $c'' = bq + b' + q^2$ and $c' = c'' + b + (q^2 + 3q 2)/2$. It is easy to show that $b = (q 3)/2, (q 1)/2, (q + 1)/2$. It is also easy to see that $b \geq q$. This is contrary to $\forall \psi \geq (q + 1)/2$.

Case 3. $(\sharp, b) = ((2), (2))$. It is easy to show that $a' = c + c' + q$, i.e. $c' = c$. We apply the rule to $(c, c', c'' ,b, b')_\phi$, We have $c'' = bq + b' + q^2$ and $c' = (c + 1)q + c'' + b + (q 1) + q^2$. It is easy to show that $b = (q 3)/2, (q 1)/2, (q + 1)/2$. It is also easy to see that $cq + c'' < (q^2 1)/2$. This is contrary to the ranges of $c', c$.

Case 4. $(\sharp, b) = ((3), (2))$. It is easy to show that $a' = c + c' + q$ 2, i.e. $c' = c + 2$. We apply the rule to $(c, c', c'' ,b, b')_\phi$, We have $c'' = bq + b' + q^2$ and $c' = (c + 1)q + c'' + b + (q 1) + q^2$. It is easy to show that $b = (q 3)/2, (q 1)/2, (q + 1)/2$. It is also easy to see that $cq + c'' < (q^2 1)/2$. This is contrary to the ranges of $c', c$.

Therefore the fact that $a'$ is not a non-zero multiple of $q$ is always satisfied. □

Theorem 3.4 (Maximum length of $\phi$-rNAF) Let $d \in \mathbb{Z}[\phi]$ and $\ell_\phi, \text{exp}(d)$. Then $d$ has a $\phi$-rNAF with digit set $D_{\phi, (q^2 - 1)/2}$ such that $\ell_\phi, \text{rNAF}(d)$ is at most $\ell + 4$.

Proof Let $\ell' = \ell_\phi, \text{rNAF}(d)$. As in the proof of Theorem 3.3, in the process of Algorithm 6, it is easily seen that we will have two cases as follows.

Case 1. $d = (b, b', e_{\ell - 3}, e_{\ell - 4}, \ldots ,e_1, e_0)_\phi$, where $(b', e_{\ell - 3}, \ldots, e_0)_\phi$ is a $\phi$-rNAF. Now we scan the two digits $(b, b')_\phi$. We can assume that $e_{\ell - 3} = 0$ because if $e_{\ell - 3} = 0$, we can reduce this case to the case 2 (See below for details). If $(b, b')_\phi$ is the $\phi$-non-adjacent pair, then $\ell' = \ell$. Otherwise, we convert the pair $(b, b')_\phi$ as follows :

\[
(b, b')_\phi \overset{\ell}{\rightarrow} \begin{cases} 
(b, 0, bq + b')_\phi & bq + b' \geq (q^2 1)/2, \\
(1, b + (q 1), 0, bq + b' + q^2)_\phi & bq + b' < (q^2 1)/2, \\
(\bar{1}, b + (q 1), 0, bq + b' + q^2)_\phi & bq + b' > (q^2 1)/2, 
\end{cases}
\]
and we put the result \((c, c', e_{\ell-1}, e_{\ell-2})_\phi\). If \(\forall bq + b' \geq (q^2 \quad 1)/2\), then \(\ell' \geq \ell + 1\). Otherwise, since \(\forall b \geq (q + 1)/2\),

\[\forall c + c' = \forall q \neq (q \quad 1) + b \geq \forall (2q \quad 1) + b \geq (q^2 \quad 1)/2,\]

when \(q \sim 5\). We can convert \((1, b + (q \quad 1))_\phi \equiv (1, 0, b + (2q \quad 1))_\phi\) or \((1, b \quad (q \quad 1))_\phi \equiv (\bar{1}, 0, b \quad (2q \quad 1))_\phi\), thus we have \(\ell' = \ell + 2\).

**Case 2.** \(d = (c, b, b', e_{\ell-4}, \ldots, e_1, e_0)_\phi\), where \((b', e_{\ell-4}, \ldots, e_1, e_0)_\phi\) is a \(\phi\)-rNAF. Now we scan the two digits \((b, b')_\phi\). We can assume that \(b' = 0\) because if \(b' = 0\), we can reduce this case to the case 1. Note that \(e_{\ell-4} = 0\). If \(b = 0\), there is nothing to do and \(\ell' = \ell\). So we can also assume \(b = 0\). Then we convert

\[
(c, b, b')_\phi \equiv \begin{cases} (c + b, 0, bq + b')_\phi & \forall bq + b' \geq (q^2 \quad 1)/2, \\ (1, c + b + (q \quad 1), 0, bq + b' + q^2)_\phi & bq + b' < (q^2 \quad 1)/2, \\ (\bar{1}, c + b \quad (q \quad 1), 0, bq + b' \quad q^2)_\phi & bq + b' > (q^2 \quad 1)/2, \end{cases}
\]

and we put the result \((a, a', e_{\ell-2}, e_{\ell-3})_\phi\). If \(\forall bq + b' \geq (q^2 \quad 1)/2\), there is nothing to do and \(\ell' \geq \ell\). Otherwise, the most significant two digits are \((1, (q \quad 1) + b + c)_\phi\) or \((1, \quad (q \quad 1) + b + c)_\phi\), we convert

\[\forall aq + a' = \forall q \neq (q \quad 1) + b + c \geq \forall (2q \quad 1) + b + c \geq (q^2 \quad 1)/2,\]

when \(q \sim 9\). We can convert

\[
(\mp 1, \mp (q \quad 1) + b + c)_\phi \equiv (\mp 1, 0, \mp (2q \quad 1) + b + c)_\phi.
\]

Then we have \(\ell' = \ell + 2\) when \(q \sim 9\). If \(q = 7\) and \(\forall bq + b' > (q^2 \quad 1)/2\), then we convert

\[
(\mp 1, b + c \mp (q \quad 1))_\phi \equiv (\mp (q \quad 2), 0, b + c \mp (q \quad 1)_\phi.
\]

Note that because of Lemma 3.3, the following situations do not occur:

\[
(\mp 1, b + c \mp (q \quad 1)_\phi \equiv (\mp 1, \mp q, 0, b + c \mp (q \quad 1)_\phi.
\]
Here, $\mp q = \pm q^2 = \mp 2(q - 1)$, so $\mp 2(q - 1) \geq (q^2 - 1)/2$. Then $\ell' = \ell + 4$. Therefore $\ell_{\Phi-rNAF}(d)$ is at most 4 more than $\ell_{\Phi-EXP}(d)$.

We can also extend Algorithm 6 to the case of $q = 3$ or 5. However, Lemma 3.3 is not satisfied in this case, namely there is the possibility that $b_i$ is a non-zero multiple of $q$. If $b_i$ is a non-zero multiple of $q$, we convert

$$(b_{i+1}, b_i)_{\Phi} \leftarrow (b_i/q, b_{i+1} + b_i/q, 0)_{\Phi}.$$ 

It is easy to show that if $b_i$ is a non-zero multiple of $q$, then it satisfies $b_i = \mp q$. Thus for all $i$, we always have

$$\psi_{i+1} \geq (q - 1)/2, \psi_i \geq q - 1 \text{ or } \psi_{i+1} \geq (q + 1)/2, \psi_i \geq 2q - 1 (\psi_i \leq q),$$

as Algorithm 6 (or Theorem 3.4). Note that it does not occur that $b_{i+1}$ is a non-zero multiple of $q$. This shows the correctness of Algorithm 7.

Let $\Phi-\text{rNAF}_\ell$ be the set of $\Phi-\text{rNAF}$ of the length $\ell$. We put $B_\ell = \#\Phi-\text{rNAF}_\ell$, $T_\ell = \prod_{d \in \Phi-\text{rNAF}_\ell} w(d)$, where $w(d)$ means the Hamming weight of $d$. Then as is the case with $\Phi-GNAF$, the non-zero density of $\Phi-\text{rNAF}$ is defined by

$$\lim_{\ell \to \infty} T_\ell/\ell B_\ell.$$ 

For $\Phi-\text{rNAF}$, we also have similar properties except for the minimality of the Hamming weight. As is the case of $\Phi-GNAF$, although $\Phi-\text{rNAF}$ does not have minimal Hamming weight, asymptotic average non-zero density of $\Phi-\text{rNAF}$ is the same as $r\text{NAF}$. This indicates that it can be reduced the average number of point additions required by the $\Phi-\text{rNAF}$ than the average number of point additions required by the Frobenius expansion.
Algorithm 7 $\phi$-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ ($q = 3$ or $5$)

Input: $d / \mathbb{Z}[\phi]$

Output: $\phi$-rNAF of $d$

1: $d' \rightarrow d$ mod $(\phi^n - 1)$
2: $\ell \rightarrow \lceil 2 \log_q 2 \ N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d) \rceil + 4$
3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_{\phi}$ of $d'$
4: $b_0 \rightarrow c_0$, $b_1 \rightarrow c_1$, $b_\ell \rightarrow 0$, $b_{\ell+1} \rightarrow 0$, $b_{\ell+2} \rightarrow 0$, $b_{\ell+3} \rightarrow 0$
5: $i \rightarrow 0$
6: while $i \geq \ell + 2$ do
7: if $b_i \text{ mod } q = 0$ then
8: $b_{i+1} \rightarrow b_{i+1}$, $b_i / q, b_{i+2} \rightarrow c_{i+2} + b_i / q, i \rightarrow i + 1$
9: end if
10: if $b_i = 0$ then
11: $b_{i+2} \rightarrow c_{i+2}, e_i \rightarrow 0, i \rightarrow i + 1$
12: else if $\lfloor b_{i+1}q + b_i \rfloor \geq (q^2 - 1)/2$ then
13: $b_{i+3} \rightarrow c_{i+3}, b_{i+2} \rightarrow c_{i+2} + b_{i+1}, e_{i+1} \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i, i \rightarrow i + 2$
14: else if $b_{i+1}q + b_i < (q^2 - 1)/2$ then
15: $b_{i+3} \rightarrow c_{i+3} + 1, b_{i+2} \rightarrow c_{i+2} + b_{i+1} + (q - 1), e_{i+1} \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i + q^2, i \rightarrow i + 2$
16: else
17: $b_{i+3} \rightarrow c_{i+3} + 1, b_{i+2} \rightarrow c_{i+2} + b_{i+1} + (q - 1), e_{i+1} \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i + q^2, i \rightarrow i + 2$
18: end if
19: end while
20: return $(e_{\ell+3}, e_{\ell+2}, \ldots, e_1, e_0)_{\phi}$

Proposition 3.3 (Properties of $\phi$-rNAF)

1. Every $d / \mathbb{Z}[\phi]$ has a unique $\phi$-rNAF.
2. The average density of non-zero digits among $\phi$-rNAF's of length $\ell$ is equal to $(q - 1)/(2q - 1) + (2q^2 - 2q + 1)/((2q - 1)^2\ell) + O(((q - 1)/q)^\ell)$. In particular, for a fixed $q$, the average non-zero density of $\phi$-rNAF is asymptotically $(q - 1)/(2q - 1)$.
3. $\phi$-rNAF does not have minimal Hamming weight among all Frobenius expansion with digit set $D_{q,(q^2-1)/2}$.  

34
Proof

(1) By the same way as Proposition 3.2, we suppose \( d / \mathbb{Z}[\phi] \) has two such representations
\[
d = \sum_{i=0}^{\ell-1} e_i \phi^i = \sum_{i=0}^{\ell-1} e'_i \phi^i, \quad e_0 = e'_0.
\]
Then from Lemma 3.1, we have \( q \mid (e_0 - e'_0) \), and by the assumption, \( \varepsilon_0 - \varepsilon'_0 \geq (q^2 - 1)/2 \), hence \( \varepsilon_0 \leq \varepsilon'_0 \leq \varepsilon_0 + \varepsilon'_0 \leq q^2 - 1 \). Therefore we have
\[
e_0 = e'_0 = \pm q, \pm 2q, \ldots, \mp (q - 1)q.
\]
We put \( e_0 = e'_0 = aq \), where \( a \in \{ \mp 1, \mp 2, \ldots, \mp (q - 1) \} \).
We must have \( e_0, e'_0 = 0 \). Because if \( e_0 = 0, e'_0 = (e_0 e'_0) \) is divisible by \( q \) but \( e'_0 / q^{(q^2 - 1)/2} \), i.e. \( q \mid e'_0 \). This is contrary to \( e_0 = 0 \). Similarly, we can show that \( e'_0 = 0 \). By the definition of \( \phi \)-rNAF, we must have \( e_1, e'_1 = 0 \). We have
\[
aq = (e_0 e'_0) = \prod_{i=1}^{\ell-1} (e'_i e_i) \phi^i
\]
and by the characteristic polynomial of \( \phi \), we have \( a \phi a \phi^2 = \prod_{i=1}^{\ell-1} (e'_i e_i) \phi^i \). Thus we can see
\[
(e_1 e'_1 + a) = (e'_2 e_2 + a) \phi + \sum_{i=3}^{\ell-1} (e'_i e_i) \phi^{i-1}.
\]
Hence, from Lemma 3.3, we also have \( q \mid (e_1 e'_1 + a) \). It follows that \( q \nmid a \).
This is contrary to \( a \in \{ \mp 1, \mp 2, \ldots, \mp (q - 1) \} \).

(2) We will show that \( B_\ell \) and \( T_\ell \) satisfy the recurrence equations \( B_1 = B_2 = q^2 - q, B_{\ell+2} B_{\ell+1} = (q^2 - q) B_\ell = 0 \) and \( T_1 = 0, T_2 = q^2 - q, \)
\[
T_{\ell+2} T_{\ell+1} = (q^2 - q) T_\ell = \prod_{i=1}^{\ell} B_i.
\]

In Table 3.2, the symbol \( ' \mp \) means a non-zero digit. We explain how to interpret Table 3.2. The left column stands for forms of \( \ell \) digits \( \phi \)-rNAF, the central column stands for the number of each \( \phi \)-rNAF corresponding to each form of the left column, and the right column stands for each number of 0 digits corresponding to each form of the left column. From Table 3.2, we can derive
\[ B_\ell = B_1 \left( 1 + \prod_{i=1}^{\ell-2} B_i \right), \text{ and} \]

\[ T_\ell = B_1 \left( 1 + \prod_{i=1}^{\tau-2} T_i \right) \left[ + B_1 \left( \prod_{i=1}^{\tau-2} (\ell \ 1 \ i)B_i + (\ell \ 1) \right) \right]. \]

Table 3.2: Forms of \( \ell \) digits \( \phi \)-rNAF representations

<table>
<thead>
<tr>
<th>forms</th>
<th># of representations</th>
<th># of 0 digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pm 0 \pm \cdots \pm 0 )</td>
<td>( B_1B_{\ell-2} )</td>
<td>( B_1B_{\ell-2} + B_1T_{\ell-2} )</td>
</tr>
<tr>
<td>( \pm 00 \pm \cdots \pm 00 )</td>
<td>( B_1B_{\ell-3} )</td>
<td>( 2B_1B_{\ell-3} + B_1T_{\ell-3} )</td>
</tr>
<tr>
<td>( \pm 000 \pm \cdots \pm 000 )</td>
<td>( B_1B_{\ell-4} )</td>
<td>( 3B_1B_{\ell-4} + B_1T_{\ell-4} )</td>
</tr>
<tr>
<td>( \cdots )</td>
<td>( \cdots )</td>
<td>( \cdots )</td>
</tr>
<tr>
<td>( \pm 0000 \pm 0 \pm 0 \pm 0 )</td>
<td>( B_1B_2 )</td>
<td>( (\ell \ 3)B_1B_2 + B_1T_2 )</td>
</tr>
<tr>
<td>( \pm 0000 \pm 0 \pm 0 )</td>
<td>( B_1B_1 )</td>
<td>( (\ell \ 2)B_1B_1 + B_1T_1 )</td>
</tr>
<tr>
<td>( \pm 0000 \pm 0 )</td>
<td>( B_1 )</td>
<td>( (\ell \ 1)B_1 )</td>
</tr>
</tbody>
</table>

From these equations, we have the desired recurrence equations, and by elementary calculations, we have \( B_\ell = q(q \ 1)(q^\ell \ (1 \ q^\ell))/2q, \) and

\[ T_\ell = \frac{q(q \ 1)}{2q} \left( q^{\ell+1} + (1 \ q^{\ell+1})\ell \right) - \frac{2q^2}{2q} \frac{2q + 1}{1} - \frac{(q^\ell \ (1 \ q^\ell))}{1}. \]

From the equations of \( B_\ell, T_\ell, \) The average number of non-zero digits for \( \ell \) digits numbers in \( \mathbb{Z}[\phi] \) is equal to \( \ell \cdot T_\ell/B_\ell. \) We put

\[ f(\ell) := \left( 1 \frac{T_\ell}{\ell B_\ell} \right) \left[ \frac{q}{2q} \frac{1}{1} + \frac{2q^2}{2q} \frac{2q + 1}{1} \right]. \]

Then

\[ f(\ell) = \frac{1}{2q} \times \left[ \frac{1}{1} \left( \frac{1}{1} \right)^\ell \right]. \]

Let us take \( g(\ell) = ((q \ 1)/q^\ell). \) Since it satisfies that

\[ \frac{f(\ell)}{g(\ell)} = \frac{1}{2q} \times \left( \frac{1}{1} \left( \frac{1}{1} \right)^{\ell-1} \right). \]
we obtain

\[
\left( \frac{f(\ell)}{g(\ell)} \right) = \frac{1}{2q} \times \frac{1}{(q - 1)/q)} \ell \leq \frac{1}{2q} \quad (\ell \ll \epsilon).
\]

Hence \( f(\ell) = O(g(\ell)) \). Thus we have \( \lim_{\ell \to \infty} T_\ell/(\ell B_\ell) = 1 \) \( q/(2q - 1) = (q - 1)/(2q - 1) \) as desired.

(3) We give a counter example that is an element \( \alpha / \mathbb{Z}[\phi] \) that does not have minimal Hamming weight. Let \( q = 5, \alpha = 3\phi + 8 / \mathbb{Z}[\phi] \). An Frobenius expansion of \( \alpha \) with digit set \( D_q(q^2 - 1)/2 \) is \((3, 8)\) and its Hamming weight is 2. But \( \phi\text{-rNAF} \)
of \( \alpha \) is \((1, 0, 6, 0, 2)\) and its Hamming weight is 3. Therefore \( \phi\text{-GNAF} \) does not have minimal Hamming weight among all Frobenius expansion with digit set \( D_q(q^2 - 1)/2 \).

Algorithm 8 computes the \( \phi\text{-rNAF} \) for a given \( d / \mathbb{Z}[\phi] \) without the computation of the Frobenius expansion of \( d \) and reduces the memory consumption to compute the \( \phi\text{-rNAF} \) compared to Algorithm 6 and 7. From Theorem 3.4, it is easy to show the correctness of Algorithm 8.

If we compute \( 2P = P + P, 3P = P + 2P \quad (3P = \phi(P) \quad \phi^2(P) \) when \( q = 3 \),

\( 4P = P + 3P, (q - 1)P = P + (q - 2)P, (q + 1)P = P + (q - 1)P \), and so on,

the computational cost of Horner’s method using \( \phi\text{-rNAF} \) (Algorithm 2 based on \( \phi\text{-rNAF} \) with \( \alpha = (q^2 - 1)/2 \), digit set \( D_\phi = D_q(q^2 - 1)/2 \) is approximately

\[
\begin{align*}
D_\phi + 2A_\phi + 2F_\phi + \frac{q}{2q} \ell A_\phi + \ell F_\phi \quad (q = 3),
& \\
D_\phi + q^2 \frac{q}{4} A_\phi + \frac{q}{2q} \ell A_\phi + \ell F_\phi \quad (q \sim 5),
\end{align*}
\]

where \( F_\phi \) stands for the computational cost of the Frobenius map in the precomputation step.
Algorithm 8 $\phi$-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$

Input: $d / \mathbb{Z}[\phi]$

Output: $\phi$-rNAF of $d$

1: $d' := d_0 + d_1 \phi \mod (\phi^n - 1)(d_0, d_1 / \mathbb{Z})$
2: $\ell \rightarrow \left\lceil \log_q 2 \right\rceil + 4$
3: $i \rightarrow 0$
4: while $i \geq \ell + 2$ do
5: Compute $Q, b / \mathbb{Z}$ such that $d_0 = Qq + b$ (see [70])
6: $d_0 \rightarrow Q + d_1, d_1 \rightarrow Q$
7: if $b = 0$ then
8: $e_i \rightarrow 0, i \rightarrow i + 1$
9: else
10: Compute $Q, a / \mathbb{Z}$ such that $d_0 = Qq + a$ (see [70])
11: if $\langle aq + b \rangle \geq (q^2 - 1)/2$ then
12: $e_i \rightarrow aq + b, e_{i+1} \rightarrow 0, d_0 \rightarrow a, d_1 \rightarrow Q$
13: else if $aq + b < (q^2 - 1)/2$ then
14: $e_i \rightarrow aq + b + q^2, e_{i+1} \rightarrow 0, d_0 \rightarrow a + (q - 1), d_1 \rightarrow Q + 1$
15: else
16: $e_i \rightarrow aq + b q^2, e_{i+1} \rightarrow 0, d_0 \rightarrow a (q - 1), d_1 \rightarrow Q + 1$
17: end if
18: $i \rightarrow i + 2$
19: end if
20: end while
21: return $(e_{\ell+3}, e_{\ell+2}, \ldots, e_1, e_0)_\phi$

3.3.3 $\phi$-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$

In this subsection, we assume that $t = 1$. We consider how to expand the multiplication by $d$ map on $E(\mathbb{F}_q^m)$ in terms of $\phi$-adic GNAF and $\phi$-adic rNAF.

Contrary to the $\phi$-GNAF of $t = 1$, it seems that we may well not be able to directly use the technique of Section 3.3.1 to construct the $\phi$-GNAF of $t = 1$. For example, let $a, b / D_{q^2 - 1}$ and we assume that $a, b > 0$. If the pair $(a, b)_\phi$ is
ϕ-non-admissible pair, we have \( a + b \equiv a + b \sim q \). Even if we convert \((a, b)_\phi\) to 
\((1, a, 1, b - q)_\phi\), the pair \((a, 1, b - q)_\phi\) is also ϕ-non-admissible pair when it satisfies that \(a + b \mid q = 0\) or \(a + b \mid q = 1\). Besides, if we convert \((1, a, 1, b - q)_\phi\) to \((2, a, 1, q, b - q)_\phi\), the pair \((a, 1, q, b - q)_\phi\) is also ϕ-non-admissible pair when it satisfies that \(a + b \mid q = 0\) or \(a + b \mid q = 1\).

However, similarly to the case of the \(\phi\)-rNAF of \(t = 1\), we can construct the \(\phi\)-rNAF of \(t = 1\). Let \(a, b / D_{q, (q^2 - 1)/2}\). If the pair \((a, b)_\phi\) is ϕ-adjacent pair, we have \(ab = 0\). We convert \((a, b)_\phi\)

\[
(a, b)_\phi \left\{ \begin{array}{ll}
(1, q, a + (q - 1), 0, (aq + b + (q^2 - 1)/2)_\phi) & \text{if } aq + b \geq (q^2 - 1)/2,

(1, a, 0, aq + b + (q^2 - 1)/2)_\phi) & \text{else if } aq + b < (q^2 - 1)/2,

\end{array} \right.
\]

Then for each case, two most significant digits of the resulting expansion is ϕ-
non-adjacent pair. How to construct an algorithm which computes the \(\phi\)-rNAF of \(t = 1\) is similar to Algorithm 6. Moreover, Theorem 3.4 and Proposition 3.3
is also true for the \(\phi\)-rNAF of \(t = 1\). In the case of \(t = 1\), the proofs of
Theorem 3.4 and Proposition 3.3 are very similar to the case of \(t = 1\). So we
omit them. Furthermore, we can construct algorithms similar to Algorithm 6 and
Algorithm 7. We also omit to describe them.

Here, we only give an algorithm which computes the \(\phi\)-rNAF of \(t = 1\) for a
given \( d / \mathbb{Z}[\phi] \), where \(q \sim 7\).
Algorithm 9 \( \phi \)-rNAF on \( \mathbb{F}_q \)-Koblitz curves of \( t = 1 \) \( (q \sim 7) \)

**Input:** \( d / \mathbb{Z}[\phi] \)

**Output:** \( \phi \)-rNAF of \( d \)

1: \( d' \rightarrow d \mod (\phi^m - 1) \)
2: \( \ell \rightarrow \left\lceil 2 \log_q 2 \frac{N_{\mathbb{Z}[\phi]/\mathbb{Z}(d)}}{\sqrt{\mathbb{Z}[\phi]/\mathbb{Z}(d)}} \right\rceil + 4 \)
3: Compute the Frobenius expansion \((c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_{\phi}\) of \( d' \)
4: \( b_0 \rightarrow c_0, b_1 \rightarrow c_1, b_\ell \rightarrow 0, b_{\ell+1} \rightarrow 0, b_{\ell+2} \rightarrow 0, b_{\ell+3} \rightarrow 0 \)
5: \( i \rightarrow 0 \)
6: while \( i \geq \ell + 2 \) do
7: if \( b_i = 0 \) then
8: \( b_{i+2} \rightarrow c_{i+2}, e_i \rightarrow 0, i \rightarrow i + 1 \)
else if \( \left\{ \begin{array}{l} b_{i+1}q + b_i \geq (q^2 - 1)/2 \end{array} \right\) then
9: \( b_{i+3} \rightarrow c_{i+3}, b_{i+2} \rightarrow c_{i+2}, b_{i+1}, c_{i+1} \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i, i \rightarrow i + 2 \)
else if \( b_{i+1}q + b_i < (q^2 - 1)/2 \) then
10: \( b_{i+3} \rightarrow c_{i+3}, 1, b_{i+2} \rightarrow c_{i+2}, b_{i+1} + (q - 1), e_{i+1} \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i + q^2, i \rightarrow i + 2 \)
else
14: \( b_{i+3} \rightarrow c_{i+3} + 1, b_{i+2} \rightarrow c_{i+2}, b_{i+1}, (q - 1), e_{i+1} \rightarrow 0, e_i \rightarrow b_{i+1}q + b_i + q^2, i \rightarrow i + 2 \)
end if
16: end while
17: return \((e_{\ell+3}, e_{\ell+2}, \ldots, e_1, e_0)_{\phi}\)

### 3.4 Scalar Multiplication Costs

#### 3.4.1 Estimates

We estimate scalar multiplication costs on \( \mathbb{F}_q \)-Koblitz curves in terms of elliptic curve operations and finite field operations using several recoding methods and standard left-to-right method (for details, refer to [37]). Let \( d \) be a large positive integer and we focus on the group of \( \mathbb{F}_{q^m} \)-rational points \( E(\mathbb{F}_{q^m}) \) for sufficiently large \( m \) which satisfy \( d \subset q^m \). Let \( m_q \) be the length of the unsigned \( q \)-adic expansion of \( d \) with digit set \( D = \{0, \mp 1, \ldots, \mp (q - 1)\} \) and \( d_0 = d \mod (\phi^m - 1) \),
respectively.

As \( d \subset q^m \), the norm of \( d \) will be equal to \( d^2 \subset q^{2m} \) and \( d_0 \subset q^{m+1} \) (for detail, refer to [70]). So

\[
\log_q d = \log_q q^{m} \quad \text{for } \log_q q^{m} \quad \text{and } \ell \geq \left\lceil \log_q 2 \sqrt{\frac{N_{d_0}}{|Z|}} \right\rceil + 4 \geq m + 6.
\]

To simplify the evaluation of the computational cost, we assume that \( m_q \), \( \ell_{\phi, \text{EXP}}(d_0) \), \( \ell_{\phi, \text{GNAF}}(d_0) \), and \( \ell_{\phi, \text{rNAF}}(d_0) \) are equal to each other (Strictly speaking, we should analyze each average of \( \ell_{\phi, \text{EXP}}(\phi) \), \( \ell_{\phi, \text{GNAF}}(\phi) \) and \( \ell_{\phi, \text{rNAF}}(\phi) \) among positive integers in the range \([1, \#E(F_{q^m})] \) to evaluate the exact computational costs. However, we do not deal with this analysis). In practical meaning, the shift operations are essentially free, thus the cost of Frobenius map on subfield elliptic curves is almost free in normal basis representation. Note that when we compute point multiplication by \( q \) for ordinary GNAF or rNAF, we use \((\phi^2 \phi)\), i.e., the computational cost of a point multiplication by \( q \) is \( A_8 + 2 \ast T_8 \). Table 3.3 and Table 3.4 provide the number of curve operations for each recoding method.

**Table 3.3:** The number of curve operations for each recoding method (\( q = 3 \))

<table>
<thead>
<tr>
<th>Method</th>
<th>#Table</th>
<th>( A_p )</th>
<th>( D_p )</th>
<th>( T_p )</th>
<th>( A_s )</th>
<th>( D_s )</th>
<th>( T_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>w-NAF [72]</td>
<td>2(^{w-2})</td>
<td>(2(^{w-2}) - 1)</td>
<td>1</td>
<td>0</td>
<td>( \frac{m_2}{w+1} )</td>
<td>( \ell )</td>
<td>0</td>
</tr>
<tr>
<td>Frobenius expansion [70]</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.67( \ell )</td>
<td>0</td>
<td>( \ell )</td>
</tr>
<tr>
<td>( \phi )-GNAF</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.5( \ell )</td>
<td>0</td>
<td>( \ell )</td>
</tr>
<tr>
<td>( \phi )-rNAF</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0.4( \ell )</td>
<td>0</td>
<td>( \ell )</td>
</tr>
</tbody>
</table>

**Table 3.4:** The number of curve operations for each recoding method (\( q \sim 5 \))

<table>
<thead>
<tr>
<th>Method</th>
<th>#Table</th>
<th>( A_p )</th>
<th>( D_p )</th>
<th>( T_p )</th>
<th>( A_s )</th>
<th>( D_s )</th>
<th>( T_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>w-NAF [72]</td>
<td>2(^{w-2})</td>
<td>(2(^{w-2}) - 1)</td>
<td>1</td>
<td>0</td>
<td>( \frac{m_2}{w+1} )</td>
<td>( \ell )</td>
<td>0</td>
</tr>
<tr>
<td>Frobenius expansion [70]</td>
<td>( \frac{q-1}{q} )</td>
<td>( \frac{q-5}{q} )</td>
<td>1</td>
<td>0</td>
<td>( \frac{g-1}{q} )</td>
<td>( \ell )</td>
<td>0</td>
</tr>
<tr>
<td>( \phi )-GNAF</td>
<td>( q )</td>
<td>1</td>
<td>( q )</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>( \phi )-rNAF</td>
<td>( \frac{q^2-1}{q} )</td>
<td>( \frac{q^2-q-1}{q} )</td>
<td>1</td>
<td>0</td>
<td>( \frac{q-1}{2q-1} )</td>
<td>( \ell )</td>
<td>0</td>
</tr>
</tbody>
</table>

In Table 3.3 and Table 3.4, the value \#Table (in the second column) indicates the number of elements, that have to be precomputed and stored.
An elementary multiplication in \( \mathbb{F}_{q^m} \) (resp. a squaring and an inversion) will be abbreviated by \( M \) (resp. \( S \) and \( I \)), and affine coordinates (resp. Jacobian coordinates) will be abbreviated by \( A \) (resp. \( J \)). If we choose

\[
\equiv A_\phi : A + A \leftarrow A (2M + S + I), \quad D_\phi : 2A \leftarrow A (2M + 2S + I),
\]

\[
\equiv A_\delta : J + A \leftarrow J (8M + 3S), \quad D_\delta : 2J \leftarrow J (4M + 4S),
\]

the total number of \( M \) (resp. \( S \) and \( I \)) to compute scalar multiplication for each method is as follows.

Table 3.5: The number of \( \mathbb{F}_{q^m} \)-field arithmetic operations (bit length of scalar = 192, 224, 256, 384) and total number of multiplications (\( S = 0.8M, I/M = 8 \)) for each recoding method

<table>
<thead>
<tr>
<th>Method</th>
<th>#Table</th>
<th>M</th>
<th>S</th>
<th>I</th>
<th># of M</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textbf{192bit}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w )-NAF ( (w = 6) )</td>
<td>16</td>
<td>1019.4</td>
<td>850.3</td>
<td>1</td>
<td>1721.3</td>
</tr>
<tr>
<td>Frobenius expansion ( (q = 3) )</td>
<td>1</td>
<td>650.7</td>
<td>244.0</td>
<td>0</td>
<td>845.9</td>
</tr>
<tr>
<td>( \phi )-GNAF ( (q = 3) )</td>
<td>2</td>
<td>490.0</td>
<td>185.0</td>
<td>1</td>
<td>646.0</td>
</tr>
<tr>
<td>( \phi )-rNAF ( (q = 3) )</td>
<td>3</td>
<td>396.4</td>
<td>150.4</td>
<td>3</td>
<td>424.6</td>
</tr>
<tr>
<td>Frobenius expansion ( (q = 5) )</td>
<td>2</td>
<td>533.2</td>
<td>201.2</td>
<td>1</td>
<td>629.0</td>
</tr>
<tr>
<td>( \phi )-GNAF ( (q = 5) )</td>
<td>4</td>
<td>448.7</td>
<td>170.0</td>
<td>3</td>
<td>608.7</td>
</tr>
<tr>
<td>( \phi )-rNAF ( (q = 5) )</td>
<td>12</td>
<td>295.1</td>
<td>120.7</td>
<td>9</td>
<td>481.5</td>
</tr>
<tr>
<td>Frobenius expansion ( (q = 7) )</td>
<td>3</td>
<td>477.3</td>
<td>180.3</td>
<td>2</td>
<td>592.6</td>
</tr>
<tr>
<td>( \phi )-GNAF ( (q = 7) )</td>
<td>6</td>
<td>424.0</td>
<td>161.3</td>
<td>5</td>
<td>593.0</td>
</tr>
<tr>
<td>( \phi )-rNAF ( (q = 7) )</td>
<td>24</td>
<td>294.8</td>
<td>119.5</td>
<td>20</td>
<td>548.0</td>
</tr>
<tr>
<td>\textbf{224bit}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( w )-NAF ( (w = 6) )</td>
<td>16</td>
<td>1184.0</td>
<td>1009.0</td>
<td>1</td>
<td>1999.2</td>
</tr>
<tr>
<td>Frobenius expansion ( (q = 3) )</td>
<td>1</td>
<td>757.3</td>
<td>284.0</td>
<td>0</td>
<td>984.5</td>
</tr>
<tr>
<td>( \phi )-GNAF ( (q = 3) )</td>
<td>2</td>
<td>570.0</td>
<td>215.0</td>
<td>1</td>
<td>742.0</td>
</tr>
<tr>
<td>( \phi )-rNAF ( (q = 3) )</td>
<td>3</td>
<td>460.4</td>
<td>174.4</td>
<td>3</td>
<td>599.9</td>
</tr>
<tr>
<td>Frobenius expansion ( (q = 5) )</td>
<td>2</td>
<td>622.8</td>
<td>234.8</td>
<td>1</td>
<td>818.6</td>
</tr>
<tr>
<td>Method</td>
<td>#Table</td>
<td>M</td>
<td>S</td>
<td>I</td>
<td># of M</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>----</td>
<td>--------</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 5))</td>
<td>4</td>
<td>523.3</td>
<td>198.0</td>
<td>3</td>
<td>705.7</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 5))</td>
<td>12</td>
<td>362.9</td>
<td>139.3</td>
<td>9</td>
<td>546.4</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 7))</td>
<td>3</td>
<td>552.6</td>
<td>208.7</td>
<td>2</td>
<td>735.5</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 7))</td>
<td>6</td>
<td>490.0</td>
<td>186.0</td>
<td>5</td>
<td>678.8</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 7))</td>
<td>24</td>
<td>335.4</td>
<td>131.8</td>
<td>20</td>
<td>600.8</td>
</tr>
<tr>
<td>256bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(w)-NAF ((w = 6))</td>
<td>16</td>
<td>1348.6</td>
<td>1150.7</td>
<td>1</td>
<td>2277.1</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 3))</td>
<td>1</td>
<td>864.0</td>
<td>324.0</td>
<td>0</td>
<td>1123.2</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 3))</td>
<td>2</td>
<td>650.0</td>
<td>245.0</td>
<td>1</td>
<td>846.0</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 3))</td>
<td>3</td>
<td>524.4</td>
<td>198.4</td>
<td>3</td>
<td>683.1</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 5))</td>
<td>2</td>
<td>712.4</td>
<td>268.4</td>
<td>1</td>
<td>935.1</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 5))</td>
<td>4</td>
<td>598.0</td>
<td>226.0</td>
<td>3</td>
<td>802.8</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 5))</td>
<td>12</td>
<td>412.7</td>
<td>158.0</td>
<td>9</td>
<td>611.1</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 7))</td>
<td>3</td>
<td>634.9</td>
<td>239.6</td>
<td>2</td>
<td>842.5</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 7))</td>
<td>6</td>
<td>562.0</td>
<td>213.0</td>
<td>5</td>
<td>772.4</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 7))</td>
<td>24</td>
<td>379.7</td>
<td>148.4</td>
<td>20</td>
<td>658.4</td>
</tr>
<tr>
<td>384bit</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(w)-NAF ((w = 6))</td>
<td>16</td>
<td>2006.9</td>
<td>1717.6</td>
<td>1</td>
<td>3389.0</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 3))</td>
<td>1</td>
<td>1296.0</td>
<td>486.0</td>
<td>0</td>
<td>1684.8</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 3))</td>
<td>2</td>
<td>974.0</td>
<td>366.5</td>
<td>1</td>
<td>1267.2</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 3))</td>
<td>3</td>
<td>783.6</td>
<td>295.6</td>
<td>3</td>
<td>1020.1</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 5))</td>
<td>2</td>
<td>1064.4</td>
<td>400.4</td>
<td>1</td>
<td>1392.7</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 5))</td>
<td>4</td>
<td>891.3</td>
<td>336.0</td>
<td>3</td>
<td>1184.1</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 5))</td>
<td>12</td>
<td>608.2</td>
<td>231.0</td>
<td>9</td>
<td>865.3</td>
</tr>
<tr>
<td>Frobenius expansion ((q = 7))</td>
<td>3</td>
<td>878.8</td>
<td>330.8</td>
<td>2</td>
<td>1151.4</td>
</tr>
<tr>
<td>(\phi)-GNAF ((q = 7))</td>
<td>6</td>
<td>832.0</td>
<td>314.3</td>
<td>5</td>
<td>1123.4</td>
</tr>
<tr>
<td>(\phi)-rNAF ((q = 7))</td>
<td>24</td>
<td>545.8</td>
<td>210.7</td>
<td>20</td>
<td>874.4</td>
</tr>
</tbody>
</table>

The above table shows that the speed of the proposed methods improves between 8% and 50% over that for the Frobenius expansion method.
3.4.2 Timings

In order to evaluate the efficiency of several previous methods and a proposed method on a general purpose computer, we implement scalar multiplication on $\mathbb{F}_2$-Koblitz Curve $K_{(3,1)}$-173 (See Table 3.9 in Section 3.5) for binary method, $w$-NAF, $r$NAF, the Frobenius expansion, and $\phi$-$r$NAF. The bit length of the prime order $n$ of this curve is 257. The base point $P$ in Table 3.9 is represented using a polynomial basis.

For the finite field implementation, we use polynomial basis. In our implementation, the field inversion cost to field multiplication cost ratio $I/M$ is 17.26 (See Table 3.6).

In the recoding step, we implement [37, Algorithm 3.35] for $w$-NAF, [75, Algorithm 1] for $r$NAF, [70, Theorem 3] and modular reduction in $\mathbb{Z}[\phi]$ ([70, pp.146–147]) for the Frobenius expansion, Algorithm 7 for $\phi$-$r$NAF, respectively.

In the precomputation step, we use Weierstrass affine coordinates. Precomputation points are stored in Weierstrass affine coordinates for $r$NAF, the Frobenius expansion, and $\phi$-$r$NAF. For binary method and $w$-NAF, we transform Weierstrass affine coordinates of all precomputation points to Hessian affine coordinates and stored them. The computational cost of the above transformation is only two finite field multiplications [71].

In the scalar multiplication step, we use mixed coordinates. We use Hessian affine-projective point addition and Hessian projective point doubling for binary method and $w$-NAF. We use Weierstrass affine-projective point addition and Weierstrass point doubling, point tripling, and the Frobenius map for $r$NAF, the Frobenius expansion, and $\phi$-$r$NAF.

In the scalar multiplication step, we use mixed coordinates. We implement Hessian affine-projective point addition and Hessian projective point doubling for binary method and $w$-NAF. We use Weierstrass affine-projective point addition and Weierstrass point doubling, point tripling, and the Frobenius map for $r$NAF, the Frobenius expansion, and $\phi$-$r$NAF. Additionally, we use in [37, Algorithm 3.27] for binary method, Algorithm 1 for $w$-NAF, Triple and Add Algorithm in [71] for $r$NAF, Algorithm 2 with $\alpha = (q - 1)/2$ for the Frobenius expansion, Algorithm 2 with $\alpha = (q^2 - 1)/2$, $\ell = \ell + 4$ for $\phi$-$r$NAF, respectively.
The platform is an Intel® *1) Core™ *1) (1.86 GHz) with 1 GB RAM computer, Windows® *2) XP Home Edition. Programs are all written in ANSI C language with gcc 3.4.4 compiler using the flags “-O3 -fomit-frame-pointer -funroll-loops” for optimizing the speed.

Table 3.6 presents timings of the finite field operations, Table 3.7 presents timings of the recoding methods, and Table 3.8 presents timings of scalar multiplications on $\mathbb{F}_3$-Koblitz Curve $K_{(3,1)}$-173 for each method (online precomputation case). All results in Table 3.8 are averages from 100 tests with fixed scalar $d = n - 2$. For each method, the representation of $n - 2$ has average Hamming weight. Note that the timing of each method in Table 3.8 does not constrain the timing of the recoding method.

Table 3.6: Timings of the Finite Field Operations

<table>
<thead>
<tr>
<th>Finite Field Operations</th>
<th>timing (in $\mu$s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>0.02</td>
</tr>
<tr>
<td>Cube</td>
<td>0.51</td>
</tr>
<tr>
<td>Multiplication</td>
<td>3.81</td>
</tr>
<tr>
<td>Inversion</td>
<td>65.77</td>
</tr>
</tbody>
</table>

Table 3.7: Timings of Recoding Methods

<table>
<thead>
<tr>
<th>Recoding Methods</th>
<th>timing (in $\mu$s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>binary</td>
<td>n/a</td>
</tr>
<tr>
<td>$w$-NAF ($w = 6$)</td>
<td>53.42</td>
</tr>
<tr>
<td>$r$NAF ($r = 3$)</td>
<td>159.54</td>
</tr>
<tr>
<td>Frobenius expansion</td>
<td>169.92</td>
</tr>
<tr>
<td>$\phi$-$r$NAF</td>
<td>177.75</td>
</tr>
</tbody>
</table>

In Table 3.8, $A_W$, $P_W$, $A_H$, $P_H$ stand for the Weierstrass Affine coordinates, the Weierstrass Projective coordinates, the Hessian Affine coordinates, and the Hessian Projective coordinates, respectively. From Table 3.8 we see that the

*1) Intel, Core are trademarks or a registered trademarks of Intel Corporation in the United States and other countries.

*2) Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
Table 3.8: Timings of Scalar Multiplications on $\mathbb{F}_3$-Koblitz Curve $K_{(3,1)}$-173 (online precomputation case)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Coordinates (Precomputation)</th>
<th>Coordinates (Scalar Multiplication)</th>
<th>timing (in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Point Addition</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$P_\infty + A_\infty \rightarrow P_\infty$</td>
<td>$6.77$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$2A_\infty \rightarrow A_\infty$</td>
<td>$7.77$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$P_\infty + A_\infty \rightarrow P_\infty$</td>
<td>$5.75$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$3P_\infty \rightarrow 3P_\infty$</td>
<td>$8.14$</td>
</tr>
<tr>
<td>$w$-NAF ($w = 6$)</td>
<td>$A_W + A_W \rightarrow A_W$</td>
<td>$2P_W \rightarrow P_W$</td>
<td>$6.29$</td>
</tr>
<tr>
<td>$r$NAF ($r = 3$)</td>
<td></td>
<td>$\phi P_W \rightarrow P_W$</td>
<td>$3.00$</td>
</tr>
<tr>
<td>Frobenius expansion</td>
<td>$A_W + A_W \rightarrow A_W$</td>
<td>$3P_W \rightarrow P_W$</td>
<td>$7.77$</td>
</tr>
<tr>
<td>$\phi$-rNAF</td>
<td></td>
<td>$\phi P_W \rightarrow P_W$</td>
<td>$5.75$</td>
</tr>
</tbody>
</table>

The timing of $\phi$-rNAF is between 1.5 and 2 times faster than those of binary method, $w$-NAF, $r$NAF, and the Frobenius expansion.

3.5 Sample Parameters

Table 3.9 and Table 3.10 list sample parameters for $\mathbb{F}_3$-Koblitz curves. The extension degree $m$ is a prime and is selected so that there exists the $\mathbb{F}_3$-Koblitz curve having relatively small cofactor. It is easy to see that the coefficients $a, b$ of $\mathbb{F}_3$-Koblitz curves satisfy $(a, b) = (1, 2)$ ($t = 1$) or $(a, b) = (2, 1)$ ($t = 1$). The prime order $n$ is presented in hexadecimal form. A backslash at the end of a line indicates that the number (hexadecimal) is continued in the next line. The $x$ and $y$ coordinates of the base point $P$ are represented in hexadecimal form in the standard way. The following parameters are given for each curve:

$m$ The extension degree of the finite field $\mathbb{F}_{3^m}$.

$f(x)$ The irreducible trinomial of degree $m$ in $\mathbb{F}_3[x]$.

$a, b$ The coefficients of the elliptic curve $E/\mathbb{F}_3: y^2 = x^3 + ax^2 + b$.

$n$ The prime order of the base point $P$.

$h$ The cofactor, $h = \#E(\mathbb{F}_{3^m})/n$.

$x, y$ The $x$ and $y$ coordinates of the base point $P / E(\mathbb{F}_{3^m})$. $P = (x, y)$.  

46
Table 3.9: Sample Parameters for $\mathbb{F}_3$-Koblitz Curves of $t = 1$

| $K_{(3,1)}$-67: $m = 67$, $f(x) = x^{67} + 2x^2 + 1$, $a = 1$, $b = 2$, $h = 3$, $|h| = 2$, $|n| = 105$ |
|---|
| $n = 0x 00000186$ 0D7A20CB 42D347DA 07AF8AED |
| $x = 0x 00000360$ E36A968A D56D0ACF C011EC9 |
| $y = 0x 000000A9$ 2EF5B4E2 8D7C4805 78640E2B |

| $K_{(3,1)}$-71: $m = 71$, $f(x) = x^{71} + 2x^{20} + 1$, $a = 1$, $b = 2$, $h = 3 \times 1279 \times 12497$, $|h| = 26$, $|n| = 88$ |
|---|
| $n = 0x 00000000$ 0818A9D 3E1DDF77 1C305C95 |
| $x = 0x 00004F6F$ 583FBB97 5F959E85 14B11CB3 |
| $y = 0x 0000B9B3$ A6F791B9 B1B45197 28BDECA7 |

| $K_{(3,1)}$-73: $m = 73$, $f(x) = x^{73} + 2x + 1$, $a = 1$, $b = 2$, $h = 3 \times 1615637$, $|h| = 23$, $|n| = 94$ |
|---|
| $n = 0x 00000000$ 2D0E2F3F 2D043DF5 5D2C98D |
| $x = 0x 0005C879$ CE7EF384 F7718F95 68942920 |
| $y = 0x 00024CDA$ B442B22E 6D48E8AD B5A81F1D |

| $K_{(3,1)}$-79: $m = 79$, $f(x) = x^{79} + 2x^{26} + 1$, $a = 1$, $b = 2$, $h = 3 \times 317$, $|h| = 10$, $|n| = 116$ |
|---|
| $n = 0x 0009FA57$ 92211E79 BD415322 C009C02 |
| $x = 0x 239FDA7B$ F71E7F65 60C8B1A7 8038D660 |
| $y = 0x 10555DF1$ 9F88B12D 46CE40D0 F39519F9 |

| $K_{(3,1)}$-89: $m = 89$, $f(x) = x^{89} + 2x^{13} + 1$, $a = 1$, $b = 2$, $h = 3 \times 179$, $|h| = 125$, $|n| = 125$ |
|---|
| $n = 0x 00000000$ 16C525D0 8B65F7F3 274A8564 81F60CA7 |
| $x = 0x 0000B321$ E2ADF87B A018D4A3 9A947D58 ED6484BC |
| $y = 0x 0000E9B9$ 57923010 7FD33B49 22F7E4FB 23D63281 |

| $K_{(3,1)}$-131: $m = 131$, $f(x) = x^{131} + 2x^{27} + 1$, $a = 1$, $b = 2$, $h = 3 \times 103322094$, $|h| = 29$, $|n| = 180$ |
|---|
| $n = 0x 00000000$ 00008BED 4C070E99 F70846E9 3C840AC1 E24A0484 |
| $x = 0x 00003D21$ EB4D1339 567FEB98 610C1BD3 664F022A A6F2B8A5 8C58EB60 |
| $y = 0x 00070772$ 8F59D82A 47E1B0F1 844209F2 628F90E0 7F22BC77 CF59F646 |

| $K_{(3,1)}$-137: $m = 137$, $f(x) = x^{137} + 2x^{27} + 1$, $a = 1$, $b = 2$, $h = 3 \times 82349398913$, $|h| = 37$, $|n| = 181$ |
|---|
| $n = 0x 00000000$ 00008BED 4C070E99 F70846E9 3C840AC1 E24A0484 |
| $x = 0x 00003D21$ EB4D1339 567FEB98 610C1BD3 664F022A A6F2B8A5 8C58EB60 |
| $y = 0x 00070772$ 8F59D82A 47E1B0F1 844209F2 628F90E0 7F22BC77 CF59F646 |

| $K_{(3,1)}$-157: $m = 157$, $f(x) = x^{157} + 2x^{27} + 1$, $a = 1$, $b = 2$, $h = 3 \times 12420395 \times 196520401$, $|h| = 53$, $|n| = 197$ |
|---|
| $n = 0x 00000000$ 00000001 998535CF E33E4F32 FEBEEF37 CB429B9A 1D0A9C1A 6300FF7A |
| $x = 0x 010444E6$ 95E2CDA1 4A93C4A3 50004F3A 49D9C085 04073A47 3A7E5B7A 7E2B468 |
| $y = 0x 005F298F$ 8C62C455 30C425F9 12E52B86 C33B8CA5 3E2C1307 177CDA3 CDF61F1A |

| $K_{(3,1)}$-173: $m = 173$, $f(x) = x^{173} + 2x^{27} + 1$, $a = 1$, $b = 2$, $h = 3 \times 13841$, $|h| = 16$, $|n| = 259$ |
|---|
| $n = 0x 00000007$ 3E98AF5C 5852BFF6 488420E2 204975C9 9D3F6EBB 7FD35200 460A5265 |
| $06C40847$ |

<table>
<thead>
<tr>
<th>$x = 0x 0001D2C7$ 010307F6 B71AD345 80795B36 566494FD 68EF690D B03A82F1 1F116BE3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$65C16782$</td>
</tr>
<tr>
<td>$y = 0x 000093C3$ 414F0E94 54E959C4 1A658879 3DAF3112 ADB60704 DD117F34 41B00AFE</td>
</tr>
<tr>
<td>$673DFAFA$</td>
</tr>
</tbody>
</table>

<p>| $K_{(3,1)}$-271: $m = 271$, $f(x) = x^{271} + 2x^{30} + 1$, $a = 1$, $b = 2$, $h = 3$, $|h| = 2$, $|n| = 428$ |
|---|
| $n = 0x 0000F58F$ CE524847 41035912 5CF1CB97 581BDA26 8B69A0ED 0D80B8B 96AEEF24 |
| $DB049A5F$ 9C9A8541 1FE21CED 5286C3D3 AA57896B 6D00E4E5 |</p>
<table>
<thead>
<tr>
<th>$x = 0x 00013D40$ 3729B9A3 49A42A26 86A335B0 75A66E71 AFF97595 B1A7C27 3F46478E</th>
</tr>
</thead>
<tbody>
<tr>
<td>$6C65B69B$ 62304FE3 389310AF 5F8413C7 9F65B0C3 5D062DA7</td>
</tr>
</tbody>
</table>

---

47
\[ y = \text{A19CDF26 C40422FD B32E6902 07A9D50F 430566BD 4B595497} \]
\[ 34C5B21B 69DD4479 65395AFE 3D6ACD9E 85D617AA 6D59DCEF \]

\[ K_{(3,1)-409}: m = 409, f(x) = x^{409} + 2x^{20} + 1, a = 1, b = 2, h = 3, |h| = 2, |n| = 647 \]
\[ n = 0 \]
\[ 00000005 748DF2AB 676F58B3 7CA0FD2 34B6DE2D E347D04D 2B67CDEE 9932BCB \]
\[ 6011F568 B0AD3499 A75DF4F9 A5D7371 C89204A4 S99C22C2 D5826376 862038AC \]
\[ 851A2E2A F958FC0B 7CE12764 E47BF2F7 C62C7331 \]
\[ x = 0 \]
\[ 00000080 C84D105 28E2A719 8B9FA2DC 972A17F6 532FDF11 F686E578 2D629980 \]
\[ D0939D10 9A1C2A93 80B8E7C0 A1A7B992 A2A30700 4207E5A4 88AB477C DB05C035 \]
\[ D817B53E FF036178 CDD6D07 82E2E1E1 DDC3C75A \]
\[ y = 0 \]
\[ 0000003C 6B83AEF3 9A9E9F8A 08E78F82 8402CE2F 7855E5B4 139C5EBA B5F07FDD \]
\[ B109772D D55BEE7 97915F24 97984554 C7E62067 91ADB5EF 2181A67 DEBBDC15 \]
\[ 2A67163F CBE1426B 78EEA3A8 2B6649BD D0C2B34F \]

\[ K_{(3,1)-449}: m = 449, f(x) = x^{449} + 2x^{24} + 1, a = 1, b = 2, h = 3, |h| = 2, |n| = 711 \]
\[ n = 0 \]
\[ 00000042 DDB063CA 56D7C3AC 6D85162F D7870343 84222089 9FAF0DE4 42B80C10 \]
\[ 8F348C06 234DB8B7 67F1D79A 9C2B2A7D 23BEE3D5 89BDAF46 0C80CEB4 F164B9B9 \]
\[ 93CEFB3F 44A495F6 BFE3C1AC BF45457E D66413F9 4824A053 EACC0B6F \]
\[ x = 0 \]
\[ 00000043 146CCB46 4879C9F5 89A49E56 3593A1EB B3439F7A 40A416E3 AF66CE48 \]
\[ 27C6DEB9 8CDB789D 7220289C 3CF13AAB 0E73BD98 B8BD3F79 ED2F528D A161C1D0 \]
\[ 85DB9016 E6F298B5 372D7A9A C30154E7 3719666B 24268D09 222B2739 \]
\[ y = 0 \]
\[ 00000098 53E0C43B 05A8E7F7 26FE5FB9 13670160 CE8437C8 F898BB78 2C6B2F41 \]
\[ A1E2B97A D52F5DE3 02C7B85F FE71E666 C992AAAD 486CD60A DE3C3C2C CBBBC5C4 \]
\[ C6B999E7 79CF484E 91A17267 5FD64795 5BBFE5C7 8ED9C0FB 622F769B \]

### Table 3.10: Sample Parameters for $F_3$-Koblitz Curves of $t = 1$

| $K_{(3,1)}-109$: | $m = 109, f(x) = x^{109} + 2x^{50} + 1, a = 1, b = 2, h = 5 \times 427 = 1036799, |h| = 35, |n| = 129$ |
|------------------|------------------------------------------------------------------------------|
| $n = 0$          | 00000000 00000001 D2B35574 33F8C490 9322D60B C5F9092D                        |
| $x = 0$          | 00000007 284B450E 869B7E1F 306F457B CF64A41D E306EE2E                       |
| $y = 0$          | 00000005 E0DE63B A0582E2D 7F1E9865 391F1A51 E1247711                      |

| $K_{(3,1)}-131$: | $m = 131, f(x) = x^{131} + 2x^{21} + 1, a = 2, b = 1, h = 5 \times 1049 + 8647, |h| = 26, |n| = 183$ |
|------------------|------------------------------------------------------------------------------|
| $n = 0$          | 00000000 0049A480 7D9AD887 76EC9979 EF332739 F1F41526 9603562D             |
| $x = 0$          | 000077D4 1FB4D2AF 46D83E2D 460D4FE4 981C058A E0B5360F D9EE7B7A              |
| $y = 0$          | 00009F9E 48ABAC88 A072100 7F17FD1C 93A7744C 98EBAC9F EC969F2                  |

| $K_{(3,1)}-139$: | $m = 139, f(x) = x^{139} + 2x^{59} + 1, a = 2, b = 1, h = 5 \times 4907, |h| = 18, |n| = 203$ |
|------------------|------------------------------------------------------------------------------|
| $n = 0$          | 00000548 60F67059 D7493A9E 8F6AE340 7E156532 1A0DD14 A672BB1               |
| $x = 0$          | 0DAB0EC2 79393D7 7346EA27 1BBC3A4E 82AB9EDE 6A6EEFDD F3701348             |
| $y = 0$          | 1016DCC0 3C12020A F02A716 088B55A4 84D65D5D 5EB1BF32 FC5D7A8              |

| $K_{(3,1)}-179$: | $m = 179, f(x) = x^{179} + 2x^{59} + 1, a = 2, b = 1, h = 5, |h| = 3, |n| = 282$ |
|------------------|------------------------------------------------------------------------------|
| $n = 0$          | 029D3B06 85310E1C 7728A466 37C3F081 06A9D59B 45AED6FC 6073BE02 2C7CDD7\ |
| $x = 0$          | 07BEC98A 2D42704F 77663B23 D1110AF5 1557E002 8A888E24 A5FD90C3 9FDB3151\ |
| $y = 0$          | 069EEBE6 4F2BD8B7 8D2E5BF9 E95E17BE 0E712CC1 9413E313 BC17E3CF 390F3987\ |
| $K_{(3,1)}-193$: | $m = 193, f(x) = x^{193} + 2x^{12} + 1, a = 2, b = 1, h = 5 \times 773, |h| = 12, |n| = 294$ |
|------------------|------------------------------------------------------------------------------|
| $n = 0$          | 00000303 2F5B6990 37A4E5F9 31A876D 332BC8C9 2640ADFD 6562517E C9D9EAA9\ |
| $A01A216B 0FA39F4D$ |                                                                               |
\[ x = 0x \quad 00006045 \quad 73E846E8 \quad 561B0972 \quad 8AD4B750 \quad 3A45AD53 \quad BD1D4E80 \quad E4752FEC \quad 54A66869 \] 
\[ y = 0x \quad 00016680 \quad 15F3C500 \quad 0B99293A \quad 152F07E2 \quad 89029FAE \quad 5EEACCEF \quad DD207F1B \quad 0F8CB4B6 \] 

**K\(_{(3, -1)}\)**

- **267:** \( m = 227, f(x) = x^{227} + 2x^{11} + 1, a = 2, b = 1, h = 5 \times 94433, |h| = 19, |n| = 341 \)
- **299:** \( m = 269, f(x) = x^{269} + 2x^2 + 1, a = 2, b = 1, h = 5, |h| = 3, |n| = 425 \)
- **349:** \( m = 349, f(x) = x^{349} + 2x^{12} + 1, a = 2, b = 1, h = 5 \times 48163 \times 1423921, |h| = 39, |n| = 515 \)
- **383:** \( m = 383, f(x) = x^{383} + 2x^{53} + 1, a = 2, b = 1, h = 5, |h| = 3, |n| = 605 \)

**K\(_{(3, -1)}\)**

- **269:** \( m = 269, f(x) = x^{269} + 2x^2 + 1, a = 2, b = 1, h = 5, |h| = 3, |n| = 425 \)
- **349:** \( m = 349, f(x) = x^{349} + 2x^{12} + 1, a = 2, b = 1, h = 5 \times 48163 \times 1423921, |h| = 39, |n| = 515 \)
- **383:** \( m = 383, f(x) = x^{383} + 2x^{53} + 1, a = 2, b = 1, h = 5, |h| = 3, |n| = 605 \)

### 3.6 Summary of this chapter

It has been an unsolved problem to generalize \(\tau\)-NAF techniques on binary Koblitz curves to a more general family of subfield elliptic curves whose endomorphism rings are not necessarily subrings of Euclidean quadratic imaginary number fields. In this chapter, we have described two generalized methods on a family of subfield elliptic curves. Those methods are two classes of \(\phi\)-NAF (\(\phi\)-GNAF and \(\phi\)-rNAF).
$\phi$-GNAF can be applied to every subfield elliptic curves where the trace of the Frobenius map equals to 1, and $\phi$-$r$NAF can be applied to every subfield elliptic curves where the trace of the Frobenius map equals to $\pm 1$, regardless of whether or not the endomorphism rings are Euclidean. We also prove that these representations have the same non-zero densities as the corresponding original GNAF and rNAF except for the minimalities of the Hamming weights. Because of the high efficiency in computing Frobenius maps, our proposed methods improve the efficiency of scalar multiplication significantly compared to previous methods. The speed of the proposed methods improves between 8% and 50% over that for the Frobenius expansion method.
Chapter 4

Explicit lower bound for the length of minimal weight $\tau$-adic expansions on Koblitz curves

4.1 Introduction

Many public key cryptosystems are based on the computational complexity of number-theoretic problems (i.e. integer factoring problem, discrete logarithm problem in finite fields or elliptic curves). In such cryptosystems, number-theoretic computations are the dominant operations. The de facto standards for public-key cryptosystems are RSA cryptosystems [67], which are based on the difficulty of integer factorization. However, due to advances in algorithms to solve integer factoring problem and improvements of computing power, at least 2048 bit RSA is recommended after 2010 [63]. On the other hand, elliptic curve cryptosystems (ECC) [44], [56] which depend on the elliptic curve discrete logarithm problem, provide shorter key length and faster computation speed than those of RSA cryptosystems. For example, 224 bit ECC provides the same security level as 2048 bit RSA [63]. In ECC, scalar multiplication (or point multiplication) is the dominant operation, namely computing $dP$ from a point $P$ on an elliptic curve and $d$ is an integer, defined as the point resulting of adding $P + P + \ldots + P$, $d$ times. However, for practical use, it is a very important matter to improve the efficiency...
of scalar multiplication.

A common way for computing scalar multiplication is known as the double-and-add method:

$$dP = 2 \cdot \cdots \cdot 2(\ell_{\ell-1}2P + \ell_{\ell-2}P) + \cdots + \ell_1P + \ell_0P,$$

where $$\prod_{i=0}^{\ell-1} \ell_i2^i = (\ell_{\ell-1}, \ell_{\ell-2}, \ldots, \ell_1, \ell_0)_2$$ is the binary representation of $$d$$. In order to improve the performance of scalar multiplication, recoding methods of scalars play an important role. Especially, number systems which have low Hamming weight and short length, are attractive to accelerate scalar multiplication, and many efficient methods have been proposed (cf. [20], [66], [72], [75]).

On the other hand, instead of integer bases, efficiently computable endomorphisms on elliptic curves (as complex numbers) bases number systems are also attractive because it can be expected that the endomorphism-and-add method is more efficient than the double-and-add method (cf. [28], [46], [60], [64], [70]).

Koblitz [46] introduced a family of elliptic curves which admit especially fast scalar multiplication. These curves are called Koblitz curves *1) (also known as anomalous binary curves). Koblitz curves are defined by

$$E_a : y^2 + xy = x^3 + ax^2 + 1, \quad a \in \mathbb{F}_2$$ \hspace{1cm} (4.1)

over a finite field $$\mathbb{F}_2$$. We identify }0,1\{ with $$\mathbb{F}_2$$ via the natural map

$$f : }0,1\{ \leftrightarrow \mathbb{F}_2, \quad a \leftrightarrow a \mod 2.$$ For some cryptographic usage, we focus on the group of $$\mathbb{F}_{2m}$$-rational points $$E_a(\mathbb{F}_{2m})$$ for some $$m \sim 2$$. In practical use, the extension degree $$m$$ is usually chosen to be a prime at least 163 (cf. [73]). Let $$\tau$$ be the Frobenius map on $$E_a$$:

$$\tau : E_a(\mathbb{F}_{2m}) \leftrightarrow E_a(\mathbb{F}_{2m}), \quad (x, y) \leftrightarrow (x^2, y^2).$$ \hspace{1cm} (4.2)

We can regard $$\tau$$ as a complex number which satisfies the following characteristic equation

$$\tau^2 + \mu \tau + 2 = 0, \quad \text{where} \quad \mu = (1)^{1-a}. \hspace{1cm} (4.3)$$

*1) The reason that Koblitz curves are so named is because Koblitz [46] firstly suggested that the curves are suitable for efficient implementation of ECC.
The roots of Equation (4.3) are \( \tau = (\mu \mp \sqrt{7})/2 \), that is, the Koblitz curve has complex multiplication by \( \tau^{*1} \). Since the cost of the Frobenius map \( \tau \) is cheaper than that of point doubling, and a scalar can be written as a \( \tau \)-adic expansion, the Frobenius map allows for scalar multiplication without the need for point doubling [46].

Solinas [72] proposed a low Hamming weight \( \tau \)-adic expansion on Koblitz curves, namely the width-\( w \) \( \tau \)-adic non-adjacent form (\( w \)-\( \tau \)-NAF for short). \( w \)-\( \tau \)-NAF of \( d/\mathbb{Z}[\tau] \) with digit set \( D_w \), is a \( \tau \)-adic expansion \( d = \prod_{i=0}^{\ell-1} e_i \tau^i \) such that

\[
e_i = 0 \implies e_{i+w-1} = \cdots = e_{i+1} = 0
\]

and \( e_i / D_w \) for all \( i \), where \( D_w \) is a finite subset of the rational integer ring \( \mathbb{Z} \). In this chapter, we focus on the digit set of zero and the odd integers with absolute value less than \( 2^{w-1} \), that is, \( D_w = \{ 0, \mp 1, \mp 3, \ldots, \mp (2^{w-1} - 1) \} \). Solinas proved some desired properties \( ^*3 \) of the \( \tau \)-NAF with respect to the Hamming weight, namely, the \( \tau \)-NAF has the existence and uniqueness, and the non-zero density of the \( \tau \)-NAF is asymptotically \( 1/3 \) [72]. Subsequently, Avanzi, Heuberger, and Prodinger [4] have proven the minimality of the Hamming weight of the \( 2-\tau \)-NAF (or \( \tau \)-NAF \( ^*4 \)).

The computational cost of scalar multiplication \( dP \) using the \( \tau \)-and-add method with \( \tau \)-NAF, is approximately \( (\ell/3)A + \ell F \), where \( \ell \) is the length of the \( \tau \)-NAF of \( d \), and \( A, F \) stand for the computational cost of the point addition, the Frobenius map, respectively.

In order to take advantage of the efficiency of the \( \tau \)-NAF, it is necessary that the \( \tau \)-NAF has appropriate length. The length of the \( \tau \)-NAF of \( d \) using [72, 

\( ^*1 \) For detail, refer to [69].

\( ^*2 \) A digit set which has the property (4.4), is called a width-\( w \) non-adjacent digit set (\( w \)-NADS). Otherwise it is called a \( w \)-NADS. \( w \)-NADS have already been investigated in the case of \( w \)-NAF (cf. [58], [8]). Solinas also proposed the digit set of minimal norm representatives [72]. Subsequently, Avanzi, Heuberger, and Prodinger [6], [7] proposed two other digit sets.

\( ^*3 \) For the width-\( w \) non-adjacent form (\( w \)-NAF), the desired properties are shown in [8], [57], [58], [59], [72].

\( ^*4 \) For \( w = 3 \), the minimality with digit set has also been shown by Avanzi, Heuberger, and Prodinger [6]. Unlike in the case of \( w = 2, 3 \), this is not true for \( w / \{ 4, 5, 6 \} \) [41]. Similar results for the \( \tau \)-NAF and its desired properties are proved in [11] and in [35] on some other types of elliptic curves, respectively.
Algorithm 1) is \( \log_2(N_{\mathbb{Z}[\tau]/\mathbb{Z}}(d)) = 2 \log_2 d \), which is twice the length of the NAF of \( d \). In order to circumvent the problem, Solinas [72] has developed modular reduction in \( \mathbb{Z}[\tau] \). This technique is called the reduced \( \tau \)-NAF. By using modular reduction in \( \mathbb{Z}[\tau] \), we can reduce the length \( \ell \) to a maximum of \( m + a \), where \( a \) is the coefficient in Equation (4.1), and \( m \) is the extension degree. However, a lower bound for the length of minimal Hamming weight \( \tau \)-adic expansions with digit set \( \{0, \mp 1\} \), is not known yet. If the lower bound is quite small compared to the length of the \( \tau \)-NAF, further speed up can be achieved in the case of polynomial basis representation.

In this chapter, we shall derive an explicit lower bound for the length of minimal Hamming weight \( \tau \)-adic expansions. Firstly, we give a lemma which will be needed in the proof of the lower bound and a new proof of the minimality of the Hamming weight of the \( \tau \)-NAF. Secondly, we derive an explicit lower bound for the length of minimal Hamming weight \( \tau \)-adic expansions based on the lemma. We also give a new proof of the minimality of the Hamming weight of the \( \tau \)-NAF on Koblitz curves using the lemma. Further, by using the proof of lower bound and the new proof of the minimality of the Hamming weight of the \( \tau \)-NAF, we classify a minimal length \( \tau \)-adic expansion with minimal Hamming weight except for two special cases. The classification shows the following two facts. One is that the \( \tau \)-NAF has almost minimal length among all \( \tau \)-adic expansions of minimal Hamming weight with digit set \( \{0, \mp 1\} \). The other is that we can easily convert the \( \tau \)-NAF into a minimal length \( \tau \)-adic expansion without changing the Hamming weight. These facts follow immediately from the proof of the lower bound and our new proof.

This chapter is organized as follows. Section 4.2 shows a key lemma which will be needed in the proof of the lower bound and our new proof of the minimality of the Hamming weight of the \( \tau \)-NAF. Section 4.3 derives an explicit lower bound for the length of minimal Hamming weight \( \tau \)-adic expansions. Section 4.4 gives the new proof of minimality of the Hamming weight of the \( \tau \)-NAF on Koblitz curves. Section 4.5 classifies a minimal length \( \tau \)-adic expansion except for two special cases.
4.2 Key Lemma

In this section, we show a key lemma (Lemma 4.2) which will be needed in the proof of the lower bound and the new proof. We begin with recursive formulas to convert any \( \tau \)-adic expansion into the \( \tau \)-NAF. The following lemma is useful to obtain such recursive formulas.

**Lemma 4.1** Let \( \ell / \mathbb{N} \) be a natural number. If \( \prod_{i=0}^{\ell-1} a_i \tau^i = 0 \) (\( a_i / \mathcal{D} \)), then \( a_i = 0 \) for all \( i \) (\( 0 \leq i \leq \ell - 1 \)).

**Proof** From \( \prod_{i=0}^{\ell-1} a_i \tau^i = a_0 + \prod_{i=1}^{\ell-1} a_i \tau^i \) and \( \tau \setminus a_0 \), we have \( 2 \setminus a_0 \). By \( a_0 \setminus \mathcal{D} \), we have \( a_0 = 0 \). We put \( \alpha' := \prod_{i=0}^{\ell-2} a_i \tau^i \). By the same argument as above, it satisfies \( a_1 = 0 \). Similar to the case of \( a_0 \) and \( a_1 \), we also have \( a_2 = 0, \ldots, a_{\ell-1} = 0 \). Therefore \( a_i = 0 \) for all \( i \). \( \square \)

The \( \tau \)-adic expansion \( \prod_{i=0}^{\ell-1} (b_i - c_i) \tau^i \) is not necessarily \( \tau \)-adic expansion with \( \mathcal{D} \), because \( (b_i - c_i) / \{0, \mp 1, \mp 2\} \). However, by using the following carry rules from right to left (i.e. from the least significant digit to the most significant digit), we can convert \( \prod_{i=0}^{\ell-1} (b_i - c_i) \tau^i \) into \( \tau \)-adic expansion \( \prod_{i=0}^{\ell-1} a_i \tau^i \) (\( a_i / \mathcal{D} \)). For each \( i \) (\( i = 0, 1, 2, \ldots, \ell - 1 \)), \( a_i \)'s are obtained by the following recursive formulas:

\[
 a_i = (b_i - c_i) \mu D_{i-1}^* + D_{i-2}^* + D_i, \tag{4.5}
\]

where \( D_{-1}^* = D_{-2}^* = 0 \), and for all \( i \),

\[
 D_i := \begin{cases} 
 (b_i - c_i) - \mu D_{i-1}^* + D_{i-2}^* \left\{ \begin{array}{l} \times 2 \quad \text{(* 2)} \\
 (\text{if } (b_i - c_i) \mu D_{i-1}^* + D_{i-2}^* \sim 0), \tag{4.6} \end{array} \right. \\
 - (b_i - c_i) - \mu D_{i-1}^* + D_{i-2}^* \left\{ \begin{array}{l} \times 2 \quad \text{(* 2)} \\
 \end{array} \right. \quad \text{(otherwise)},
\end{cases}
\]

and

\[
 D_i^* = D_i / 2 \ (i \sim 0). \tag{4.7}
\]

From (4.5), (4.6), and (4.7), each \( a_i \) is an element in \( \{0, \mp 1\} \). From Lemma 4.1, we have \( a_i = 0 \) for all \( i \). In other words, for any \( \alpha / \mathbb{Z}[\tau] \) and any \( \tau \)-adic expansion \( \alpha = \prod_{i=0}^{\ell-1} c_i \tau^i \) with digit set \( \mathcal{D} \), we can compute the \( \tau \)-NAF of \( \alpha \) (\( \alpha = \prod_{i=0}^{\ell-1} b_i \tau^i \))
using the recursive formulas (4.5), (4.6), and (4.7).

Example 4.1 Let $\mu = 1$, $\alpha = 3597 \ 11571\tau$.

$$(b_{29}, \ldots, b_0)_\tau = (1, 0, 1, 0, 0, 1, 0, 0, \bar{1}, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, \bar{1})_\tau,$$

$$(c_{29}, \ldots, c_0)_\tau = (0, 1, \bar{1}, 1, 0, 0, 0, 1, 0, \bar{1}, 1, 1, 0, 0, 0, \bar{1}, 0, \bar{1}, \bar{1}, \bar{1}, 0, 0, 1, 0, 1)_{\tau}$$

are the $\tau$-NAF of $\alpha$, and a $\tau$-adic expansion of $\alpha$, respectively. We demonstrate how to apply the recursive formulas (4.5), (4.6), and (4.7) to the $\tau$-adic expansion $(c_{29}, \ldots, c_0)_2$ in order to compute the $\tau$-NAF of $\alpha$.

$$(0, 1, \bar{1}, 1, 0, 0, 1, 0, \bar{1}, 1, 1, 0, 0, 0, \bar{1}, 0, \bar{1}, \bar{1}, \bar{1}, 0, 0, 1, 0, 1, 1)_{\tau}$$

$$(\bar{1}, 1, 2)_{\tau}$$

$$(\bar{1}, 1, 2, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$+ (1, \bar{1}, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)_{\tau}$$

$$(1, 0, 1, 0, 0, 1, 0, \bar{1}, 0, 1, 0, \bar{1}, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, \bar{1})_{\tau}$$
The lower bound and our new proof of the minimality of the Hamming weight of the $\tau$-NAF are based on the following lemma.

**Lemma 4.2 (Key Lemma)**

Let $S := \{0, \pm 1\} \times \{0, \pm 1\} \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$. Let $H_i := (b_i, c_i, D_{i-1}^*, D_{i-2}^*, D_i) / S$ for $i (0 \leq i \leq \ell - 1)$, where $b_i$, $c_i$, $D_{i-1}^*$, $D_{i-2}^*$, and $D_i$ satisfy Equation (4.6) and (4.7). Let $A_1 := \{(\mu, 0, 1, 0, 0), (\mu, 0, 0, 0, \mu)\}$, $A_2 := \{(\mu, 0, 1, 0, 0), (\mu, 0, 0, \mu, 0)\}$, $A_3 := \{(\mu, 0, 1, 0, 2\mu), (\mu, 0, 1, 0, 2\mu)\}$, $A_4 := \{(\mu, 0, 0, \mu, 2\mu), (\mu, 0, 0, \mu, 2\mu)\}$ be the subset of $S$, respectively.

1. $D_i = 0, \pm 2$ ($D_i^* = 0, \pm 1$) for all $i$.
2. If $c_{i+1} = 0, b_{i+1} \neq 0$ for some $i \sim 0$, then $H_{i+1} / A_1 \cap A_2 \cap A_3 \cap A_4$.
3. If $H_{i+1} / A_1 \cap A_3$, then $b_i = 0, c_i = 0$. If $H_{i+1} / A_4$, then it hold $b_{i+2} = 0$ and $c_{i+2} = 0$. In particular, if $H_i / A_4$, then $H_{i+2} / A_1 \cap A_3$.
4. For $i_0 / \{0, 1, \ldots, \ell - 1\}$, the following conditions are equivalent:
   
   (a) $\prod_{i=0}^{i_0} b_i \tau_i = \prod_{i=0}^{i_0} c_i \tau_i$;
   
   (b) $D_{i_0-1} = 0$ and $D_{i_0} = 0$.
5. Suppose that $H_{i+1} / A_2$. If $(D_{j+1}, D_j) = (0, 0)$ for all $j (1 \leq j \leq i + 1)$, then $i \sim 2$ and

$$
\begin{pmatrix}
  b_2 & b_1 & b_0 \\
  c_2 & c_1 & c_0
\end{pmatrix}
$$

satisfies one of the following two cases:

$$
\begin{pmatrix}
  b_2 & b_1 & b_0 \\
  c_2 & c_1 & c_0
\end{pmatrix} / \Gamma_1, \quad (4.8)
$$
or

$$
\begin{pmatrix}
  b_2 & b_1 & b_0 \\
  c_2 & c_1 & c_0
\end{pmatrix} / \Gamma_2, \quad (4.9)
$$

57
where
\[
\Gamma_1 = \left\{ \begin{pmatrix} 0 & 0 & 1 \\ 0 & \mu & 1 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 1 \\ 0 & \mu & 1 \end{pmatrix} \right\},
\]
\[
\Gamma_2 = \left\{ \begin{pmatrix} \pm & 0 & 1 \\ \pm & \mu & 1 \end{pmatrix}, \begin{pmatrix} \pm & 0 & 1 \\ \pm & \mu & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 1 \\ 0 & \mu & 1 \end{pmatrix} \right\}.
\]

In particular, if
\[
\begin{pmatrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{pmatrix} \in \left\{ \begin{pmatrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{pmatrix} \right\} \subset \Gamma_2
\]
holds, then \( i \sim 3, H_2 / A_4, \) and
\[
\begin{pmatrix} b_3 & b_2 & b_1 & b_0 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \Gamma_3,
\]
where
\[
\Gamma_3 = \left\{ \begin{pmatrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ \mu & 0 & \mu \end{pmatrix}, \begin{pmatrix} 1 & 0 & 1 \\ \mu & 0 & \mu \end{pmatrix} \right\}. \tag{4.10}
\]

Proof
(1) Let us assume the contrary and seek a contradiction. Suppose that there exists \( i / } 0, 1, 2, \ldots, \ell 1\} such that \( i \) does not satisfy \( D_i = 0, \mp 2, \) and \( i_0 \) be the minimal such \( i / } 0, 1, 2, \ldots, \ell 1\}. We evaluate the range of \( D_{i_0}. \) For \( i \) which satisfies \( i \geq i_0 1, \) we have \( D_i = 0, \mp 2 (D_i^* = 0, \mp 1). \) Then
\[
\begin{align*}
\|v_{i_0} \mid c_{i_0} & \mu D_{i_0-1}^* + D_{i_0-2}^* \| \\
\geq & \|v_{i_0} \| + |v_{i_0} \| + \mu |D_{i_0-1}^* \| + |D_{i_0-2}^* \| \\
\geq & 1 + 1 + 1 + 1 = 4.
\end{align*}
\]

By Equation (4.6), \( D_i \) is an even number, and \( |D_{i_0}| > 2, \) we have \( D_{i_0} = \mp 4. \) There are two cases to consider, \( D_{i_0} = 4 \) and \( D_{i_0} = 4. \) We only consider
the former because the latter may be treated similar to the former case. From
\[ \nu_{i_0} \geq 1, D_{i_0-1}^* \geq 1, \] we must have \( b_{i_0} = 1, \) \( c_{i_0} = 1, D_{i_0-1}^* = \mu, \)
and \( D_{i_0-2}^* = 1 \) in order to satisfy \( 0 = a_{i_0} = b_{i_0} \quad c_{i_0} = \mu D_{i_0-1}^* + D_{i_0-2}^* + D_{i_0}. \) So
\( D_{i_0-1}^* = 2D_{i_0}^* = 2\mu. \) On the other hand, \((b_{i-1}, \ldots, b_{i}, b_{i_0})_{\tau} \) is the \( \tau \)-NAF, so
\( b_{i_0} = 1 \) implies \( b_{i_0-1} = 0. \) Hence
\[
0 = a_{i_0-1} = b_{i_0-1} \quad c_{i_0-1} \quad \mu D_{i_0-2}^* + D_{i_0-3}^* + D_{i_0-1}^*
= c_{i_0-1} \quad \mu * 1 + D_{i_0-3}^* \quad 2\mu
= c_{i_0-1} + D_{i_0-3}^* \quad 3\mu,
\]
we obtain \( c_{i_0-1} = D_{i_0-3}^* = 3\mu. \) However, \( D_{i_0-3}^* / 0, \mp 1 \rangle = 0, \mp \mu \rangle, \) we have
\( \nu_{i_0-1} = D_{i_0-3}^* \quad 3\mu \sim \langle D_{i_0-3}^* \rangle \quad \langle 3\mu \rangle \sim \langle D_{i_0-3}^* \rangle \quad 3 \rangle \sim 2. \) This is a contradiction.

(2) We assume that \( c_{i+1} = 0, b_{i+1} = \mp \mu. \) Then we have \( 0 = a_{i+1} = b_{i+1} \quad \mu D_i^* + D_{i+1}^* + D_{i+1}. \) It is necessary to treat the cases \( D_{i+1} = 0 \) and \( D_{i+1} = 0 \) separately.

**Case 1.** \( D_{i+1} = 0. \)

It is easy to see that \( H_{i+1} / A_1 \nmid A_2. \)

**Case 2.** \( D_{i+1} = 0. \)

If the sign of \( b_{i+1} \) is same as that of \( D_{i+1}, \) we must have \( \nu_{i+1} = D_{i+1} \geq 3. \) So
it does not occur that \( b_{i+1} \quad \mu D_i^* + D_{i+1}^* + D_{i+1} = 0. \) Hence from \( b_{i+1} \) and \( D_{i+1} \)
have the opposite signs, we have \( H_{i+1} / A_3 \nmid A_4. \)

Therefore, if \( c_{i+1} = 0, b_{i+1} = 0 \) then \( H_{i+1} / A_1 \nmid A_2 \nmid A_3 \nmid A_4. \)

(3) First, we assume that \( H_{i+1} / A_1. \) Since \( (b_{i-1}, \ldots, b_1, b_0)_{\tau} \) is the \( \tau \)-NAF and
\( b_{i+1} = 0, \) we have \( b_i = 0. \) We substitute \( b_i = 0 \) into \( a_i = (b_i \quad c_i) \quad \mu D_{i-1}^* + D_{i-2}^* + D_i \) and
\( c_i = \mu D_{i-1}^* + D_{i-2}^* + D_i. \) Since \( H_{i+1} / A_1 \) and
\( c_i = \mu \pm 0 + D_{i-2}^* \mp 2 = D_{i-2}^* \mp 2 = 0, \) we have \( c_i = 0. \)

Next, suppose that \( H_{i+1} / A_3. \) Similar to the above case, since \( c_i = \mu \pm 0 + D_{i-2}^* \mp 2 = D_{i-2}^* \mp 2 = 0, \) we also have \( b_i = 0, c_i = 0. \)

We assume that \( H_{i+1} / A_4. \) Since \( (b_{i-1}, \ldots, b_1, b_0)_{\tau} \) is the \( \tau \)-NAF and \( b_{i+1} = 0, \)
we have \( b_{i+2} = 0. \) From \( a_{i+2} = b_{i+2} \quad c_{i+2} \quad \mu D_{i+1}^* + D_{i+1}^* + D_{i+1} = 0 \) and
\( c_{i+2} = \mp 1 + D_{i+2} = 0, \) we have \( c_{i+2} = 0. \) Moreover, if \( H_i / A_4, \) from \( D_i = \mp 0, \)
we have \( D_i^* = \mp \mu. \) Therefore it does not occur that \( H_{i+2} / A_1 \nmid A_3. \)
(4) \[
\begin{align*}
\sim_{i_0} b_i \tau^i &= \sim_{i_0} c_i \tau^i \\
\Rightarrow \infty &\quad (b_i \ c_i) \tau^i = 0 \\
\Rightarrow \infty &\quad (b_i \ c_i) \tau^i + (\tau^2 \ \mu \tau + 2) D_i^* \tau^i = 0 \\
\Rightarrow \infty &\quad \{ (b_i \ c_i) \mu D_{i-1}^* + D_{i-2}^* + D_i(\tau^i \\
&\quad + (D_{i-0}^* \tau + D_{i-0}^* + D_{i-1}^* \tau)^{i_0+1} = 0 \\
\Rightarrow \infty &\quad a_i \tau^i + (D_{i-0}^* \tau + D_{i-0}^* + D_{i-1}^* \tau)^{i_0+1} = 0 \\
\Rightarrow \infty &\quad D_{i-0}^* \tau + (\mu D_{i-0}^* + D_{i-0}^* + D_{i-1}^* \tau)^{i_0+1} = 0 \\
\Rightarrow \infty &\quad D_{i-0}^* \tau = 0, D_{i-0} = 0
\end{align*}
\]
(5) Since \((D_0, D_{-1}) = (0, 0)\) and \(D_{-1} = 0\), we must have \(D_0 = 0\). There are two cases to consider, \(D_0 = 2\) and \(D_0 = 2\). We only consider the former because the latter may be treated similar to the former case. From \(D_0 = 2\), we must have \(b_0 = 1\) and \(c_0 = 1\). Since \((b_{-1}, \ldots, b_1, b_0)\) is the \(\tau\)-NAF and \(b_0 = 0\), we have \(b_1 = 0\). Hence from \(b_i \ c_i \mu D_0 + D_{i-1} + D_1 = 0, D_1 = 0\), we have \((c_1, D_1) = (\mu, 2\mu)\) or \((\mu, 0)\).

**Case 1.** \((c_1, D_1) = (\mu, 2\mu)\).

By \(b_2 \ c_2 \mu D_1^* + D_0^* + D_2 = b_2 \ c_2 + D_2 = 0\), we have \((b_2, c_2, D_2) = (0, 0, 0), (\mp 1, \mp 1, 0), \) or \((\mp \mu, \mu, \mp 2\mu),\) where double signs are taken in the same order. Thus
\[
\begin{pmatrix}
  b_2 & b_1 & b_0 \\
  c_2 & c_1 & c_0
\end{pmatrix} =
\begin{pmatrix}
  0 & 0 & 1 \\
  0 & \mu & 1 \\
  \mu & 1 & 1
\end{pmatrix}
\]
\[
\begin{pmatrix}
  1 & 0 & 1 \\
  1 & \mu & 1 \\
  \mu & \mu & 1
\end{pmatrix}
\]
Case 2. \((c_1, D_1) = (\mu, 0)\).

By \(b_2 c_2 \mu D_1^* + D_1^* + D_2 = b_2 c_2 + D_2 = 0\), we have \((b_2, c_2, D_2) = (0, 1, 2)\), \((0, 1, 0)\), \((1, 0, 2)\), or \((1, 0, 0)\). Thus, in the cases of \((b_2, c_2, D_2) = (0, 1, 2)\) or \((0, 1, 0)\), we have

\[
\begin{pmatrix}
 b_2 & b_1 & b_0 \\
 c_2 & c_1 & c_0
\end{pmatrix}
= \begin{pmatrix}
 0 & 0 & 1 \\
 1 & \mu & 1
\end{pmatrix}
\text{ or }
\begin{pmatrix}
 0 & 0 & 1 \\
 1 & \mu & 1
\end{pmatrix}.
\]

In the case of \((b_2, c_2, D_2) = (1, 0, 2)\), from \(b_3 c_3 \mu D_2^* + D_1^* + D_3 = c_2 \mu + D_3 = 0\), we have \((c_2, D_3) = (\mu, 2\mu)\) or \((\mu, 0)\). Thus

\[
\begin{pmatrix}
 b_3 & b_2 & b_1 & b_0 \\
 c_3 & c_2 & c_1 & c_0
\end{pmatrix}
= \begin{pmatrix}
 0 & 1 & 0 & 1 \\
 \mu & 0 & \mu & 1
\end{pmatrix}
\text{ or }
\begin{pmatrix}
 0 & 1 & 0 & 1 \\
 \mu & 0 & \mu & 1
\end{pmatrix}.
\]

Moreover, if \((c_2, D_3) = (\mu, 2\mu)\), then \(H_2 = (1, 0, 0, 1, 2) / A_4\), and if \((c_2, D_3) = (\mu, 0)\), then \(H_2 = (\mu, 0, 0, \mu, 2\mu) / A_4\). In the case of \((b_2, c_2, D_2) = (1, 0, 0)\), we have \((D_1, D_2) = (0, 0)\). This is a contradiction.

Hence we obtain

\[
\begin{pmatrix}
 b_2 & b_1 & b_0 \\
 c_2 & c_1 & c_0
\end{pmatrix} / \Gamma_1 | \Gamma_2,
\]

or

\[
\begin{pmatrix}
 b_3 & b_2 & b_1 & b_0 \\
 c_3 & c_2 & c_1 & c_0
\end{pmatrix} / \Gamma_3.
\]

In particular,

\[
\begin{pmatrix}
 b_2 & b_1 & b_0 \\
 c_2 & c_1 & c_0
\end{pmatrix}
= \begin{pmatrix}
 0 & 0 & \pm \\
 0 & \pm & \pm
\end{pmatrix}
\text{ or }
\begin{pmatrix}
 0 & 0 & \pm \\
 0 & \pm & \pm
\end{pmatrix}
\]

or

\[
\begin{pmatrix}
 b_3 & b_2 & b_1 & b_0 \\
 c_3 & c_2 & c_1 & c_0
\end{pmatrix}
= \begin{pmatrix}
 0 & \pm & \pm \\
 0 & \pm & \pm
\end{pmatrix}.
\]

It is easy to see that \(i \sim 2\) when

\[
\begin{pmatrix}
 b_2 & b_1 & b_0 \\
 c_2 & c_1 & c_0
\end{pmatrix} / \Gamma_1 | \Gamma_2,
\]

61
and \( i \sim 3 \) when
\[
\begin{pmatrix}
  b_3 & b_2 & b_1 & b_0 \\
  c_3 & c_2 & c_1 & c_0
\end{pmatrix} / \Gamma_3.
\]

\[\square\]

4.3 Lower Bound for the Length

This section derives an explicit lower bound for the length of minimal Hamming weight \( \tau \)-adic expansions. From the definition of \( \ell_{\text{min}} \), the following upper bound for \( \ell_{\text{min}} \) is trivially true for all \( \alpha / \mathbb{Z}[\tau] \):

\[\ell_{\text{min}}(\alpha) \geq \ell_{\text{NAF}}(\alpha).\]

An lower bound \( \ell_{\text{min}} \) can also be derived in terms of the length of the \( \tau \)-NAF. The following lower bound for \( \ell_{\text{min}} \) is based on Lemma 4.2.

**Theorem 4.1 (Lower Bound for \( \ell_{\text{min}}(\alpha) \))**

Suppose that \( \ell' < \ell \). Then for any \( \alpha / \mathbb{Z}[\tau] \),

\[\ell_{\text{NAF}}(\alpha) \quad 3 \leq \ell' \quad \text{for any } \alpha \in \mathbb{Z}[\tau].\]  

In particular,

\[\ell_{\text{NAF}}(\alpha) \quad 3 \geq \ell_{\text{min}}(\alpha).\]  

**Proof**  The latter part follows immediately from the former part. We show the former part. We assume that \( c_\ell = 0, c_{\ell+1} = 0, \ldots, c_{\ell-1} = 0 \). Note that \( b_{\ell-1} = 0 \) and \( c_{\ell-1} = 0 \). From Lemma 4.2 (2), it satisfies that \( H_{\ell-1} / A_1 \mid A_2 \mid A_3 \mid A_4 \). Since \( \prod_{i=0}^{\ell-1} b_i \tau^i = \prod_{i=0}^{\ell-1} c_i \tau^i \), we have \( D_i = 0 \) for all \( i \sim \ell \). It follows that \( H_{\ell-1} / A_1 \mid A_3 \mid A_4 \). We only deal with the case of \( H_{\ell-1} / A_2 \). There are two cases to consider, \( H_{\ell-1} = (\mu, 0, 0, \mu, 0) \) and \( H_{\ell-1} = (\mu, 0, 0, \mu, 0) \). Without loss of generality, we may assume that \( H_{\ell-1} = (\mu, 0, 0, \mu, 0) \) because the latter may
be treated in exactly the same way. By $b_{\ell - 1} = 0$, it satisfies $b_{\ell - 2} = 0$. From

$$a_{\ell - 2} = (b_{\ell - 2} \ c_{\ell - 2}) \mu D_{\ell - 3}^* + D_{\ell - 4}^* + D_{\ell - 2}
= c_{\ell - 2} + 1 + D_{\ell - 4}^*
= 0,$$

we have $c_{\ell - 2} = D_{\ell - 4}^* + 1$. Hence we obtain $(c_{\ell - 2}, D_{\ell - 4}^*) = (1, 0)$ or $(0, 1)$.

**Case 1.** $(c_{\ell - 2}, D_{\ell - 4}^*) = (1, 0)$.

It is easily to see that $\ell' = \ell = 1$.

**Case 2.** $(c_{\ell - 2}, D_{\ell - 4}^*) = (0, 1)$.

It holds that

$$a_{\ell - 3} = (b_{\ell - 3} \ c_{\ell - 3}) \mu D_{\ell - 4}^* + D_{\ell - 5}^* + D_{\ell - 3}
= (b_{\ell - 3} \ c_{\ell - 3}) + \mu + D_{\ell - 5}^* 2\mu
= (b_{\ell - 3} \ c_{\ell - 3}) + D_{\ell - 5}^* \mu
= 0.$$

So

$$(b_{\ell - 3}, c_{\ell - 3}, D_{\ell - 5}^*) = (0, 0, \mu), (0, \mu, 0), (\mu, 0, 0), (\mu, \mu, \mu), \text{ or } (\mu, \mu, \mu).$$

However, if $(b_{\ell - 3}, c_{\ell - 3}, D_{\ell - 5}^*) = (\mu, \mu, \mu)$ or $(\mu, \mu, \mu)$, then $b_{\ell - 3} = 0$ and $b_{\ell - 4} = 0$. This contradicts the fact that $\alpha = \prod_{i=0}^{\ell - 1} b_i \tau^i$ is the $\tau$-NAF of $\alpha$. Therefore it does not occur that $(b_{\ell - 3}, c_{\ell - 3}, D_{\ell - 5}^*) = (\mu, \mu, \mu)$ and $(\mu, \mu, \mu)$. If $(b_{\ell - 3}, c_{\ell - 3}, D_{\ell - 5}^*) = (0, \mu, 0)$ or $(\mu, \mu, \mu)$, then $\ell' = \ell = 2$.

It remains to consider the case that $(b_{\ell - 3}, c_{\ell - 3}, D_{\ell - 5}^*) = (0, 0, \mu)$ and $(\mu, 0, 0)$. If $(b_{\ell - 3}, c_{\ell - 3}, D_{\ell - 5}^*) = (0, 0, \mu)$, then from

$$a_{\ell - 4} = (b_{\ell - 4} \ c_{\ell - 4}) \mu D_{\ell - 5}^* + D_{\ell - 6}^* + D_{\ell - 4}
= (b_{\ell - 4} \ c_{\ell - 4}) + D_{\ell - 6}^* 3
= 0,$$
we have \((b_{\ell-4}, c_{\ell-4}, D_{\ell-6}^*) = (1, 1, 1)\). This indicates that \(\ell' = \ell\). If \((b_{\ell-3}, c_{\ell-3}, D_{\ell-5}^*) = (\mu, 0, 0)\), we must have \(b_{\ell-4} = 0\). Then

\[
\begin{align*}
a_{\ell-4} &= (b_{\ell-4} \ c_{\ell-4}) \ \mu D_{\ell-5}^* + D_{\ell-6}^* + D_{\ell-4}^* \\
&= c_{\ell-4} + D_{\ell-6}^* 2 \\
&= 0.
\end{align*}
\]

Hence we obtain \((c_{\ell-4}, D_{\ell-6}^*) = (1, 1, 1)\). Thus \(\ell' = \ell\). 

As already mentioned, \(\tau\)-NAF has the smallest Hamming weight with digit set \(\{0, \mp 1\}\). Further, Theorem 4.1 tells us that \(\tau\)-NAF also has **almost minimal length** with digit set \(\{0, \mp 1\}\).

### 4.4 Another Proof of the Minimality

In this section, we give a new proof of minimality of the Hamming weight of the \(\tau\)-NAF on Kobitz curves.

#### 4.4.1 The Main Idea of Our New Proof

The minimality of the Hamming weight of the \(\tau\)-NAF on Kobitz curves was first proved by Avanzi, Heuberger, and Prodingter [4], [5]. They have presented two proofs for the minimality. One is referred as the Direct proof, which is induction on the Hamming weight. The other is referred as the Automatic proof, which is based on a weighted digraph induced by the transducer to compute the \(\tau\)-NAF from any \(\tau\)-adic expansion from right to left (see [4], [5] for proofs).

The strategy of our new proof of the minimality is as follows. For any \(\alpha / \mathbb{Z}[\tau]\), we directly construct an injection map from \(S_\alpha\) into \(T_\alpha\). Notice that if it is possible to construct an injection map from \(S_\alpha\) to \(T_\alpha\) for any \(\alpha\) and any \(\tau\)-adic expansion of \(\alpha\), then the Hamming weight of the \(\tau\)-NAF of \(\alpha\) is always smaller than that of the \(\tau\)-adic expansion, that is, the \(\tau\)-NAF minimizes the Hamming weight with digit set \(\{0, \mp 1\}\).

A similar strategy is already used for the proof of minimality of the Hamming

---

64
weight of the generalized non-adjacent form (GNAF) \[20\]. We briefly review
the strategy to prove the minimality of the Hamming weight of the GNAF. Let
\( r \sim 2 \) be a positive integer, \( \beta \) be any element of \( \mathbb{Z}_{\geq 0} \). We denote by \( \beta = \prod_{i=0}^{r-1} g_i r^i \) (\( g_i / \mathcal{D}_G \)) the GNAF of \( \beta \), where \( \mathcal{D}_G = \{0, \mp 1, \ldots, \mp (r - 1)\} \). Let
\( \beta = \prod_{i=0}^{\ell-1} h_i r^i \) (\( h_i / \mathcal{D}_G \)) be any \( r \)-adic expansion of \( \beta \). If \( \ell > \ell' \), then put \( h_{\ell+1} = h_{\ell-1} = 0 \). Otherwise, put \( g_{\ell} = g_{\ell+1} = \cdots = g_{\ell-1} = 0 \). Furthermore, replace \( \max \{\ell, \ell'\} \) by \( \ell \) if necessary. We put \( S_\beta := \{i / \{0,1,\ldots,\ell \} \setminus g_i = 0\}, \) and \( T_\beta := \{i / \{0,1,\ldots,\ell \} \setminus h_i = 0\} \).

Then the following claim holds.

**Claim 4.1 (Key Point of the Minimality [20])** If \( h_{i+1} = 0 \) for some \( i \sim 0 \),
then either \( g_{i+1} = 0 \) or \( h_i = 0 \) and \( g_i = 0 \).

Thus from Claim 4.1, we can construct the following simple injection map.

\[ \varphi_\beta : S_\beta \hookrightarrow T_\beta \]

\[ i \not\in i \quad (g_i = 0, \ h_i = 0), \]

\[ i \not\in i \quad 1 \quad (g_i = 0, \ h_i = 0). \]

We can see that Lemma 4.2 is analogous result for \( \tau \)-adic expansion.

**Example 4.2** We put \( \beta = 686204583 \).

\[ (b_{29}, \ldots, b_0)_2 = (1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, \bar{1}), \]

\[ (c_{29}, \ldots, c_0)_2 = (1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, \bar{1}, \bar{1}). \]

are the NAF of \( \beta \) and the binary representation of \( \beta \), respectively. With the above
notation, \( S_\beta = \{0, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 24, 27, 29\} \) and

\[ T_\beta = \{0, 1, 2, 5, 7, 9, 10, 13, 15, 17, 18, 21, 22, 23, 27, 29\}. \]
\( \varphi_\beta : S_\beta \leftrightarrow T_\beta \) using (4.14):

\[
\begin{array}{c|c}
\varphi_\beta : & S_\beta & T_\beta \\
\hline
0 & 0 & 0 \\
3 & 2 & 2 \\
5 & 5 & 5 \\
7 & 7 & 7 \\
9 & 9 & 9 \\
11 & 10 & 10 \\
13 & 13 & 13 \\
15 & 15 & 15 \\
17 & 17 & 17 \\
19 & 18 & 18 \\
21 & 21 & 21 \\
24 & 23 & 23 \\
27 & 27 & 27 \\
29 & 29 & 29 \\
\end{array}
\]

Thus the map \( \varphi_\beta \) is injective.

### 4.4.2 Our New Proof

We are now in a position to give our new proof of the minimality of the Hamming weight of the \( \tau \)-NAF on Koblitz curves.

**Our New Proof of the Minimality.**

With notation as above, we directly construct an injection map \( \varphi_\alpha : S_\alpha \leftrightarrow T_\alpha \) for each case.
Case 1. $H_i / A_2$ for all $i$ ($0 \geq i \geq \ell_1$).

We define a map $\varphi_\alpha : S_\alpha \leftrightarrow T_\alpha$ as follows.

$$
\varphi_\alpha : \begin{array}{c}
S_\alpha \\
\upharpoonright
\end{array} \leftrightarrow \begin{array}{c}
T_\alpha \\
\upharpoonright
\end{array}
$$

\begin{align}
& i \not\in i \quad (b_i = 0, c_i = 0), \\
& i \not\in i \quad (H_i / A_1 | A_3), \\
& i \not\in i + 1 \quad (H_i / A_4).
\end{align}

(4.15)

From the recursive formula (4.5) and $D^*_{-1} = D^*_{-2} = 0$, it does not occur that $b_0 = 0$ and $c_0 = 0$. This implies that if $0 / S_\alpha$, then $\varphi_\alpha(0) / T_\alpha$. From Lemma 4.2 (3), the map $\varphi_\alpha$ does not satisfy $\varphi_\alpha(i) = i + 1$ and $\varphi_\alpha(i + 2) = i + 1$ for any $i$. Thus, the map $\varphi_\alpha : S_\alpha \leftrightarrow T_\alpha$ is injective.

Case 2. $H_i / A_2$ for some $i$ ($0 \geq i \geq \ell_1$).

Let $\{i_1, i_2, \ldots, i_k\}$ be the set so that $H_{i_j} / A_2$ for $1 \geq j \geq k$ and $k < k'$ implies $i_k < i_k'$. We denote $i_0 := 1$ for convenient. Since $H_{i_j} / A_2$ and $(D_{-1}, D_{-2}) = (0, 0)$, we have $(D_{i_j}, D_{i_j-1}) = (0, 0)$ for $0 \geq j \geq k$. From Lemma 4.2 (4), we have $\prod_{i=1}^{b_j+1} b_i\tau i = \prod_{i=1}^{b_j+1} c_i\tau i$ for $0 \geq j \geq k$.

For each $i_j$ ($1 \geq j \geq k$), we denote

$$
M_j := \{n \in \mathbb{Z} | i_{j-1} \leq n \leq i_j, (D_n, D_{n-1}) = (0, 0)\}.
$$

(4.16)

Note that for $1 \geq j \geq k$, the set $M_j$ is not empty, because $i_{j-1} / M_j$. We put $m_j = \max M_j$. Then, we have $(D_n, D_{n-1}) = (0, 0)$ for $m_j + 1 \geq n \geq i_j$. By Lemma 4.2 (5), we have $(b_{m_j+2}, c_{m_j+2}) = (0, \pm)$. Moreover, from Lemma 4.2 (5), if

$$
\begin{bmatrix}
b_{m_j+4} & b_{m_j+3} & b_{m_j+2} & b_{m_j+1} \\
c_{m_j+4} & c_{m_j+3} & c_{m_j+2} & c_{m_j+1}
\end{bmatrix} / \Gamma_3,
$$

then $H_{m_j+2} / A_4$. Therefore we obtain

$$
i_0 \geq m_1 \geq i_1 \geq \cdots \geq m_j \geq i_j \geq \cdots \geq m_k \geq i_k.
$$

(4.17)
Furthermore, by Lemma 4.2 (5), if
\[
\begin{bmatrix}
\beta_{m_j+3} & \beta_{m_j+2} & \beta_{m_j+1} \\
\alpha_{m_j+3} & \alpha_{m_j+2} & \alpha_{m_j+1}
\end{bmatrix} / \Gamma_1 \cup \Gamma_2,
\]
then \(i_j \sim m_j + 3\), and if
\[
\begin{bmatrix}
\beta_{m_j+4} & \beta_{m_j+3} & \beta_{m_j+2} & \beta_{m_j+1} \\
\alpha_{m_j+4} & \alpha_{m_j+3} & \alpha_{m_j+2} & \alpha_{m_j+1}
\end{bmatrix} / \Gamma_3,
\]
then \(i_j \sim m_j + 4\). We define a map \(\varphi_\alpha : S_\alpha \leftarrow T_\alpha\) as follows.

\[
\varphi_\alpha : S_\alpha \leftarrow T_\alpha
\]

\[
\begin{align*}
& \begin{array}{c}
i \nleq i & (b_i = 0, c_i = 0), \\
i \nleq i + 1 & (H_i / A_1 \cup A_3), \\
i_j \nleq m_j + 2 & (H_{ij} / A_2).
\end{array}
\end{align*}
\]

By the same argument as case 1, if \(0 / S_\alpha\), then \(\varphi_\alpha(0) / T_\alpha\). By Lemma 4.2 (5), for all \(i\) which satisfy \(i \not\in \{i_1, i_2, \ldots, i_k\}\), it does not occur \(\varphi_\alpha(i) = m_j + 2\). Thus, the map \(\varphi_\alpha : S_\alpha \leftarrow T_\alpha\) is injective. This completes the proof.

Example 4.3 Let us consider \(\mu = 1, \beta = 3597\ 11571\tau\).

\[
(b_{29}, \ldots, b_0)_\tau = (1, 0, 1, 0, 0, 1, 0, 0, \overline{1}, 0, 1, 0, 1, 0, 0, \overline{1}, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, \overline{1})_\tau,
\]

\[
(c_{29}, \ldots, c_0)_\tau = (0, 1, \overline{1}, 1, 0, 0, 0, 0, 1, 0, \overline{1}, 1, 1, 0, 0, 0, \overline{1}, 0, 1, \overline{1}, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1)_\tau
\]

are the \(\tau\)-NAF of \(\alpha\), and a \(\tau\)-adic expansion of \(\alpha\), respectively.

With the above notation, \(S_\alpha = \{0, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 24, 27, 29\}\) and \(T_\alpha = \{0, 1, 4, 7, 8, 9, 11, 13, 17, 18, 19, 21, 25, 26, 27, 28\}\). We construct a map
\( \varphi_\alpha : S_\alpha \leq T_\alpha \) using (4.18):

\[
\begin{array}{c|c}
\varphi_\alpha : & S_\alpha & \leq T_\alpha \\
\hline
0 & \equiv & 0 \\
3 & \equiv & 4 \\
5 & \equiv & 1 \\
7 & \equiv & 7 \\
9 & \equiv & 9 \\
11 & \equiv & 11 \\
13 & \equiv & 13 \\
15 & \equiv & 8 \\
17 & \equiv & 17 \\
19 & \equiv & 19 \\
21 & \equiv & 21 \\
24 & \equiv & 25 \\
27 & \equiv & 27 \\
29 & \equiv & 28
\end{array}
\]

Thus the map \( \varphi_\alpha \) is injective.

4.5 \( \tau \)-adic Minimal Length Form and Its Cryptographic Application

4.5.1 \( \tau \)-adic Minimal Length Form

This section classifies a minimal length \( \tau \)-adic expansion with minimal Hamming weight except for two special cases. In the case of the ordinary NAF, minimal length binary representation with minimal Hamming weight is shown in [14, Corollary 3]. From Theorem 4.1 and our new proof, we now obtain analogous result for \( \tau \)-adic expansion. Corollary 4.1 shows that we can convert \( \tau \)-NAF into a minimal length \( \tau \)-adic expansion without changing the Hamming weight. This fact follows immediately from the proof of the lower bound and our new proof of
the minimality of the Hamming weight of the $\tau$-NAF.

**Corollary 4.1 (\(\tau\)-adic Minimal Length Expansion)**

Let \(d\) be an element of \(\mathbb{Z}[\tau]\), and \(\prod_{i=0}^{\ell-1} e_i \tau^i\ (e_i / D, e_{\ell-1} \equiv 0)\) be the \(\tau\)-NAF of \(d\). We convert the \(\tau\)-NAF \(d = \prod_{i=0}^{\ell-1} e_i \tau^i\) into \(d = \prod_{i=0}^{\ell'-1} e'_i \tau^i\ (e'_i / D, e'_{\ell'-1} \equiv 0)\) as follows.

**Case 1. \(\ell < 6\).**

If \((e_{\ell-1}, \ldots, e_0)_\tau\) is equal to one of the \(\tau\)-NAF in Table 4.1 (double signs are taken in the same order), then we convert \((e_{\ell-1}, \ldots, e_0)_\tau\) into \((e'_{\ell-1}, \ldots, e'_0)_\tau\) using Table 4.1. Otherwise, \(\ell = \ell'\) and \(e_i = e'_i\) for all \(i\).

**Table 4.1**: Conversion of the \(\tau\)-NAF into the \(\tau\)-MLF (\(\ell < 6\))

<table>
<thead>
<tr>
<th>((e_{\ell-1}, \ldots, e_0)_\tau)</th>
<th>((e'_{\ell-1}, \ldots, e'<em>0)</em>\tau)</th>
<th>(\ell)</th>
<th>(\ell')</th>
</tr>
</thead>
<tbody>
<tr>
<td>((\mp \mu, 0, 0, \pm 1)_\tau)</td>
<td>((\mp \mu, 0, 0, \pm 1)_\tau)</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>((\mp \mu, 0, 0, \pm 1, 0)_\tau)</td>
<td>((\mp \mu, 0, 0, \pm 1, 0)_\tau)</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>((\mp \mu, 0, 0, \mp \mu)_\tau)</td>
<td>((\mp \mu, 0, 0, \mp \mu)_\tau)</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>((\mp \mu, 0, \pm \mu, 0)_\tau)</td>
<td>((\pm 1, \mp \mu, 0)_\tau)</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>((\mp \mu, 0, \mp \mu, 0, \pm \mu)_\tau)</td>
<td>((\mp 1, \pm \mu, 0, 0)_\tau)</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>((\mp \mu, 0, \mp \mu, 0, \pm \mu)_\tau)</td>
<td>((\pm 1, 0, \pm \mu)_\tau)</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>((\mp \mu, 0, 0, \mp \mu, 0)_\tau)</td>
<td>((\pm 1, 0, 0, \mp \mu)_\tau)</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

**Case 2. \(\ell = 6\).**

We convert \((e_{\ell-1}, \ldots, e_0)_\tau\) into \((e'_{\ell-1}, \ldots, e'_0)_\tau\) using Table 4.2 (double signs are taken in the same order).

**Case 3. \(\ell \sim 7\).**

(i) \((e_{\ell-1}, \ldots, e_0)_\tau = (\mp \mu, 0, \mp \mu, 0, 0, 0, 0, \mu)_\tau \) or \((\mp \mu, 0, \mp \mu, 0, \mu, 0, 0, 0, \mu)_\tau\).

\(e_i = e'_i\) for all \(i \geq \ell\). \(8, \ell' = \ell\). 3, and we convert \((e_{\ell-1}, \ldots, e_{\ell-7})_\tau = (\mp \mu, 0, \mp \mu, 0, e_{\ell-5}, 0, 0, 0, \mu)_\tau\)

into \((e'_{\ell-1}, \ldots, e'_0)_\tau\).\(=(\mp \mu, 0, 0, e_{\ell-5}, 0, 0, 0, 0, \mp \mu)_\tau\).

(ii) \((e_{\ell-1}, \ldots, e_0)_\tau = (\mp \mu, 0, \mp \mu, 0, 0, 0, 0, 0, \mu)_\tau\) and \((\mp \mu, 0, \mp \mu, 0, \mu, 0, 0, 0, \mu)_\tau\).

\(e_i = e'_i\) for all \(i \geq \ell\). \(7\) and we convert \((e_{\ell-1}, \ldots, e_{\ell-6})_\tau = (e'_{\ell-1}, \ldots, e'_0)_\tau\)

using Table 4.2.

Then, except for the cases that \((e_{\ell-1}, \ldots, e_0)_\tau = (\mu, 0, 0, 0, 0, 0, 0, 0, 0, 0, \mu)_\tau\) and \((e_{\ell-1}, \ldots, e_0)_\tau = (\mu, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, \mu)_\tau\), the \(\tau\)-adic expansion \(d = \prod_{i=0}^{\ell'-1} e_i \tau^i\) is a minimal
Table 4.2: Conversion of the $\tau$-NAF into the $\tau$-MLF ($\ell \sim 6$)

<table>
<thead>
<tr>
<th>$(e_{\ell-1}, \ldots, e_{\ell-6})_\tau$</th>
<th>$(e'<em>{\ell-1}, \ldots, e'</em>{\ell-6})_\tau$</th>
<th>$\ell'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mp \mu, 0, 0, 0, 0, 0)_\tau$</td>
<td>$(\mp \mu, 0, 0, 0, 0, 0)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, 0, 0, \mp \mu)_\tau$</td>
<td>$(\mp \mu, 0, 0, 0, 0, \mp \mu)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, 0, 0, \mu)_\tau$</td>
<td>$(\mp \mu, 0, 0, 0, 0, \mu)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, 0, \mp \mu, 0)_\tau$</td>
<td>$(\pm \mu, 0, 0, 0, \mp \mu, 0)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, 0, \mu, 0)_\tau$</td>
<td>$(\mp \mu, 0, 0, 0, \mu, 0)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \mp 1, 0, 0)_\tau$</td>
<td>$(\mu, \mu, 0, 0, 0)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \mp 1, 0, 0)_\tau$</td>
<td>$(\mu, \mu, 0, 0, 0)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \mp 1, 0, \mp 1)_\tau</td>
<td>$</td>
<td>$(\pm \mu, 0, 0, \mp 1, 0, \mp 1)_\tau</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, 0)_\tau$</td>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, 0)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \mp 1)_\tau</td>
<td>$</td>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \mp 1)_\tau</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \mp 1)_\tau$</td>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \mp 1)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \pm 1)_\tau$</td>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \pm 1)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \pm 1)_\tau$</td>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \pm 1)_\tau$</td>
<td>$\ell$</td>
</tr>
<tr>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \pm 1)_\tau$</td>
<td>$(\pm \mu, 0, 0, \pm \mu, 0, \pm 1)_\tau$</td>
<td>$\ell$</td>
</tr>
</tbody>
</table>

length $\tau$-adic expansion with minimal Hamming weight. We call the $\tau$-adic expansion $\prod_{i=0}^{\ell'-1} e'_{i} \tau^{i}$ ($e'_{i} / D$, $e'_{\ell'-1} = 0$) $\tau$-adic minimal length form ($\tau$-MLF for short).

**Remark 4.1** As described in Corollary 4.1, if $(e_{\ell-1}, \ldots, e_{\ell-6})_\tau = (\mp \mu, 0, \mp \mu, 0, \pm \mu, 0)_\tau$, then the $\tau$-adic expansion $d = \prod_{i=0}^{\ell'-1} e'_{i} \tau^{i}$ is not necessarily a minimal length $\tau$-adic expansion with minimal Hamming weight.

For example, consider $d = 11\mu$. The $\tau$-NAF of $d$ is $(\mu, 0, \mu, 0, \mu, 0, 0, \mu, 0, \mu)_\tau$ and $\ell = 9$. From Corollary 4.1, $\ell' = \ell - 2$ and the $\tau$-MLF of $d$ is $(\mu, 1, \mu, 0, \mu, 0, \mu)_\tau$. However, minimal length $\tau$-adic expansion with minimal Hamming weight of $d$ is $(1, 0, 1, \mu, 1, \mu)_\tau$ and $\ell_{\text{min}}(d) = \ell - 3$. 

71
Another example is \( d = 5 \mu \tau \), where \( \mu = 1 \). The \( \tau \)-NAF of \( d \) is \( (\mu, 0, \mu, 0, \mu, 0, 0, 1, \tau) \), and \( \ell = 8 \). From Corollary 4.1, \( \ell' = \ell \) 2 and the \( \tau \)-MLF of \( d \) is \( (\mu, 1, \mu, 0, 0, 1, \tau) \). However, minimal length \( \tau \)-adic expansion with minimal Hamming weight of \( d \) is \( (1, 0, 1, \mu, 1, \tau) \) and \( \ell_{\min}(d) = \ell \) 3. These issues remain to be discussed.

4.5.2 Cryptographic Application of \( \tau \)-MLF

Suppose that \( P / E_a(\mathbb{F}_p) \) and the point \( P \) has a prime order \( n \). Public key schemes (e.g., ECDSA) using Koblitz curves involve scalar multiplication for the point \( P \) and a random number in \( \mathbb{Z}/n\mathbb{Z}^* \). In general, normal basis (resp. polynomial basis) is used for hardware (resp. software) implementation. The computational cost of the Frobenius map (4.2) is virtually free in normal basis and is relatively inexpensive but not completely free in polynomial basis. As we see in Section 4.5.1, \( \tau \)-MLF has not only the minimal Hamming weight, but also the minimal length among all \( \tau \)-adic expansions of minimal Hamming weight with digit set \( \{0, \pm 1\} \). In order to utilize \( \tau \)-MLF method in the scalar multiplication step, a very small amount of additional memory for storing Table 4.1 and Table 4.2 is required compared to \( \tau \)-NAF method in the scalar multiplication step. However, by using \( \tau \)-MLF in the scalar multiplication step, one can reduce the number of Frobenius map. Namely, scalar multiplication on Koblitz curve using \( \tau \)-MLF method is slightly faster than the original \( \tau \)-NAF method. Thus \( \tau \)-MLF can be used for efficient implementation of ECC.

4.6 Summary of this chapter

In this chapter, we derived an explicit lower bound for the length of minimal Hamming weight \( \tau \)-adic expansions. We also gave a new proof of the minimality of the Hamming weight of the \( \tau \)-NAF on Koblitz curves. Further, by using the proof of the lower bound and the new proof of the minimality of the Hamming weight of the \( \tau \)-NAF, we classified a minimal length \( \tau \)-adic expansion with minimal Hamming weight except for two special cases. The classification shows that the \( \tau \)-NAF has almost minimal length among all \( \tau \)-adic expansions of minimal
Hamming weight and we can easily convert the $\tau$-NAF into a minimal length $\tau$-adic expansion without changing the Hamming weight.
Chapter 5

Batch verification suitable for efficiently verifying a limited number of signatures

5.1 Introduction

Batch verification for digital signature scheme is a method to verify multiple signatures simultaneously in order to significantly speed up signature verification. If signatures are generated by multiple signers, we call it multi-signer batch verification; otherwise, we call it single-signer batch verification in accordance with the work by Guo et al. in [33]. Batch verification was firstly proposed by Naccache, M’Raïhi, Raphaeli, and Vaudenay [61] and Yen and Laih [77], respectively. Although several batch verification methods have been proposed (cf. [39], [40], [52], [65]), almost all methods were broken (cf. [15], [21], [42], [74]).

In 1998, Bellare, Garay, and Rabin [9] introduced the notion of single-signer batch verification (and also introduced screening which is a weaker notion of batch verification) and proposed three general tests for discrete logarithm based signature schemes. The small exponents test was improved by Cheon and Lee [18] to speed up batch verification for elliptic curve based signature schemes, and proposed two batch verification tests. One is called the Sparse Exponent Test and the other is called “the Complex Exponent Test” (CE test for short). So as to
achieve further improvement, Cheon and Yi [19] generalized the idea of CE test in [18].

In 2007, Camenisch, Hohenberger, and Pedersen [17] extended the notion of multi-signer batch verification. Moreover, Camenisch et al. applied the small exponents test to some pairing-based signature schemes. An excellent overview of batch verification can be found in [17].

5.1.1 Motivation

In this chapter, we focus on single-signer batch verification for discrete logarithm based signature schemes. Batch verification is especially appropriate for systems which are required to verify a large amount of signatures. Namely, for many applications of batch verification, it is an important requirement to verify a large amount of signatures efficiently. However, in addition to the above requirement, several types of systems might also require verifying a limited number of digital signatures more and more efficiently. This means that it might be a requirement to verify a limited number of digital signatures generated by a signer, as fast as possible.

- IP camera surveillance system.

One such example which might need the above requirement is IP camera surveillance system. In IP camera surveillance systems, each IP camera captures live images and sends them directly to image storage servers over an IP network. Digital signature provide an effective solution to prevent manipulation of live images. In order to facilitate key management, all IP cameras which are monitoring the same area often use the same signing key.

A typical IP camera supports real time frame rate of $5 \approx 30$ frames per second (fps). If the frame rate is 15 fps and an IP camera captures live images throughout the day, more than one million images are stored in the storage server(s). Normally the storage server verify the signatures which have already stored in the storage server. Thus it is an important requirement to verify a large amount of stored signatures efficiently.

On the other hand, the Internet is inherently insecure and is facing various security threats. So one might verify the signatures immediately after the storage
server received the live images. Therefore, it might be an important requirement to verify a limited number of live images immediately after the storage server received the live images.

- **Wireless sensor network.**

Another example is wireless sensor network (WSN). Wireless sensor networks are widely used in various kinds of fields. Wireless sensor network is typically composed of a large number of sensor nodes and a base station [48]. In WSN, the sensed data is transmit to the base station through a multi-hop network consisting of several sensor nodes. Because multi-hop networks are potentially vulnerable to manipulation, digital signature is used to validate the data integrity of the sensed data. As is the case with the above example, several sensor nodes are used the same signing key. Each sensed data is verify by the base station. In this situation, the base station might verify a limited number of sensed data for some reasons such as efficient verification.

### 5.1.2 Contribution of this chapter

The contribution of this chapter is to propose a new batch verification for discrete logarithm based signature schemes. Whereas the CE test is suitable for systems which need to verify a large amount of signatures, the proposed test is suitable for verifying a limited number of signatures in real-time. The performance analysis
shows that the proposed test is faster than previous methods when a limited number of signatures is verified (See Table 5.8). For example, our test with a 286 bit curve (Table 5.4) is faster than previous methods when the number of signatures is less than or equal to 18 (resp. 6) for security parameter 80 (resp. 140).

The rest of this chapter is organized as follows. Section 5.2 reviews some batch verification tests. In Section 5.3, we propose a new batch verification test. Section 5.5 compares the computational costs of several previous tests and the proposed test. Section 5.6 concludes the chapter.

5.2 Preliminaries

In this section we briefly review batch verification. For detail, refer to [9], [17], [18] and [19].

5.2.1 Batch Verification

Let $G$ be a cyclic group of prime order $n$, and $g$ a generator of $G$. For $a / G$, $\text{ord}_G(a)$ is the order of $a$, namely, $\text{ord}_G(a)$ is the smallest non-negative integer $x$ that holds $a^x = 1_G$, where $1_G$ is the identity element in the group $G$. $X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is a batch instance which we would like to check whether or not $y_i = g^{x_i}$ for all $i = 1, \ldots, N$. We define a Boolean relation $R(\cdot)$ on the set $\{(x, y) \mid x \in \mathbb{Z}_n, y \in G\}$ by the following. We define $R(x, y) = 1$ if $g^x = y$, and $R(x, y) = 0$ otherwise. We say that the batch instance $X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ is correct if $R(x_i, y_i) = 1$ for all $i = 1, \ldots, N$, and incorrect if there exists an index $i \in \{1, \ldots, N\}$ such that $R(x_i, y_i) = 0$. The following definition is the notion of single-signer batch verification introduced by Bellare, Garay, and Rabin [9].

**Definition 5.1 (Batch Verifier (Batch Verification Test) [9])** A batch verifier (also called a batch verification test) for relation $R$ is a probabilistic algorithm $\mathcal{V}$ that takes as input a batch instance $X = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ for $R$ and $k_{\text{br}}$, a security parameter for batch verification, and

1. If $X$ is correct then $\mathcal{V}$ outputs 1.
(2) If \( X \) is incorrect then the probability that \( V \) outputs 1 is at most \( 2^{-k_{br}} \). This probability is called the error probability.

### 5.2.2 Small Exponents Test

Bellare, Garay, and Rabin [9] proposed three general batch verification tests, the (Atomic) Random Subset Test, the Small Exponents Test, and the Bucket Test. The atomic random subset test checks whether 
\[ g^{\sum_{i=1}^N s_i x_i} = \prod_{i=1}^N y_i^{s_i}, \]
where each \( s_i \) is randomly chosen from the set \( \{0, 1\} \). Note that the upper bound of the error probability of the atomic random subset test is at most \( 1/2 \). The random subset test is repetition of the atomic random subset test independently \( k_{br} \) times in order to lower the error probability of the random subset test to \( 2^{-k_{br}} \). The small exponents test is an extension of the atomic random subset test. The small exponents test checks whether 
\[ g^{\sum_{i=1}^N s_i x_i} = \prod_{i=1}^N y_i^{s_i}, \]
where each \( s_i \) is randomly chosen from the set \( \{0, 1\}^{k_{br}} \).

### 5.2.3 Complex Exponent Test

Cheon and Lee [18] proposed two improved tests of the small exponents test, one is the Sparse Exponent Test, and the other is the Complex Exponent Test. The complex exponent test can only be applied to a special family of elliptic curves whereas the sparse exponent test can be applied to elliptic curves over prime/extension fields. In the complex exponent test, subfield elliptic curves (cf. [35], [70], [72]) are used.

Digital signature schemes based on an elliptic curve over a finite field (e.g., ECDSA* [1] which is a modified version of ECDSA [62], EC-Schnorr signature [68]), for \( x \in \mathbb{Z}_n, Q \in E \), it is checked whether \( xP = Q \) or not. We consider how to check \( x_i P = Q_i \) (\( i = 1, \ldots, N \)) for a batch instance \( \{(x_1, Q_1), \ldots, (x_N, Q_N)\} \). We define a Boolean relation \( \text{ScMul}_{E,P}(x, Q) \) as follows: \( \text{ScMul}_{E,P}(x, Q) := 1 \) if \( xP = Q \), otherwise \( \text{ScMul}_{E,P}(x, Q) := 0 \).

We prepare some notations to explain the complex exponent test. We define
\[
S_1(\ell_1, k_1, q) \overset{\text{def}}{=} \{ d = \ell_1^{-1} \sum_{i=0}^{\ell_1-1} a_i \phi^i \mid a_i \in \mathbb{Z}, \forall a_i \geq q, 1, a_{i+1}a_i = 0, \text{wt}(d) \geq k_1 \},
\]
for an elliptic curve $E$ defined over $\mathbb{F}_q$. For all $P \in E(\mathbb{F}_{q^m})$, we always have $(\phi^m - 1)P = 0$, the point at infinity. So we sometimes identify $\phi^m$ and 1 as elements in $\mathbb{Z}[\phi]$, or we consider the endomorphisms on $E$ in $\mathbb{Z}[\phi]/(\phi^m - 1)$ instead of $\mathbb{Z}[\phi]$. Thus we have to choose an appropriate value of $\ell_j$ in order to fulfill that for any two different elements in $S_j(\ell_j, k_j, q)$, these two elements are different from each other as endomorphisms on $G = \langle P \rangle$ (e.g., $\prod_{i=0}^{m-1} \phi^i = \prod_{i=0}^{m-1} 0 \phi^i = 0$).

The following theorem guarantees the above situation if the range of $\ell_j$ is chosen appropriately.

**Theorem 5.1 ([18, Theorem 3])** For $j = 1, 2$ and $6 \geq \ell_j \geq (\log_2 n / \log_2 30)/\log_2 q$, each endomorphism on $G$ as an element in $S_j(\ell_j, k_j, q)$ is different from each other, where $M_1 = 2(q - 1)$, $M_2 = 2(q^2 - 1)/2$, and $k_j \geq \ell_j$.

Under the assumption of Theorem 5.1, the cardinalities of $S_j(\ell_j, k_j, q)$ are given by the following theorem.

**Theorem 5.2 ([18, Theorem 4])** $\#S_1(\ell_1, k_1, q) = \prod_{i=0}^{k_1} \ell_1^{i+1-i^2} (2q^2 - 1)^i$ and $\#S_2(\ell_2, k_2, q) = \prod_{i=0}^{k_2} \ell_2^{i+1-i^2} (q^2 - 1)^i$.

We fix an extension degree $m$, and we choose the maximum $\ell_j$ such that the assumption of Theorem 5.1 holds. Moreover, for each $k_{Br} / \mathbb{Z}$, we choose the maximum $k_j$ so that $2^{k_{Br}} \sim \#S_j(\ell_j, k_j, q)$. The complex exponent test check whether

$$
\sum_{i=1}^N s_i x_i \mod n \left[ P = \sum_{i=1}^N s_i Q_i \right] (5.1)
$$

or not for randomly chosen $s_i / S_j(\ell_j, k_j, q) \ (j = 1, 2)$. The naive test verifies whether $x_i P = Q_i$ or not for each $i \ (i / \{ 1, \ldots, N \}$, i.e., $N$ scalar multiplications with $n$ bits long scalars. On the other hand, in order to reduce the
computational cost of the CE test, Cheon and Lee apply BGMW method ([16]) to their CE test (See [18, Fig. 3] for detail). If we rewrite step 4 of [18, Fig. 3] (right-to-left BGMW method [16]) as the left-to-right version of BGMW method, the computational costs are

\[
k_1N + 2(q^2 + 1)A + (q + 1)\ell_1F + (1 \text{ scalar multiplication}),
\]

\[
k_2N + 2\left\lfloor (q^2 + 1)/2 \right\rfloor A + \left\lfloor (q + 1)/2 \right\rfloor \ell_2F + (1 \text{ scalar multiplication}),
\]

respectively. If \(\{x_1, Q_1\}, \ldots, (x_N, Q_N)\) is a correct batch instance, Equation (5.1) is always true. The probability that the complex exponent test accepts an incorrect batch instance \(\{x_1, Q_1\}, \ldots, (x_N, Q_N)\) is at most \(2^{-k_{bv}} \) \((j = 1, 2)\) ([18, Theorem 1]).

### 5.2.4 Improved Complex Exponent Test

Cheon and Yi in [19] generalized the idea of the CE test. In order to improve the efficiency of the CE test, they use the width-\(w\) \(\tau\)-adic non-adjacent form (cf. [37], [36], [72] for the explanation of \(\tau\)-NAF and \(w\)-\(\tau\)-NAF). Although the improved CE test is quite fast for verifying a large amount of signatures, the improved CE test is not suitable for verifying a limited number of signatures, because the precomputation cost is relatively large compared to that of the CE test. Therefore, we omit the details of the improved CE test, and we do not deal with the improved CE test any more. See [19] for details.

### 5.3 Our Proposed Method

In this section, we propose two new methods. In Section 5.3.1, we show some properties of certain types of elliptic curves which will be used in our proposed method. In Section 5.3.2 and Section 5.3.3, we propose our new batch verification tests. Moreover we analyze the security of our batch verification tests, and estimates the computational cost of each proposed method.
5.3.1 Properties of Certain Types of Elliptic Curves

Our target curves are followings. Note that the following curves are not standard curves such as NIST recommended curves [62, Appendix D], but it allows fast computation of multi-scalar multiplication, which is the dominant operation in batch verification for elliptic curve based signature schemes.

**Proposition 5.1 (Automorphism of \(E_{p,b}/F_p : y^2 = x^3 \ b [28, 47, 64, 69]\))**

Let \(p\) be a prime such that \(p \equiv 1 \mod 3\), and \(E_{p,b} : y^2 = x^3 \ b\) be the elliptic curve defined over \(F_p\). Suppose that \(\beta / F_p\) is an element of order 3 (i.e., \(\beta^3 = 1 / F_p\)). Then the curve \(E_{p,b}\) has the automorphism \(\omega : E_{p,b} \rightarrow E_{p,b}\) \((x,y) \mapsto (\beta x, y)\).

We can regard \(\omega\) as a complex number which satisfies the equation \(\omega^2 + \omega + 1 = 0\), namely, \(\omega = \left(1 \pm \sqrt{-3}\right)/2\).

**Proposition 5.2 (Automorphism of \(E_{p,a}/F_p : y^2 = x^3 + ax [28, 47, 64, 69]\))**

Let \(p\) be a prime such that \(p \equiv 1 \mod 4\), and \(E_{p,a} : y^2 = x^3 + ax\) be the elliptic curve defined over \(F_p\). Suppose that \(\alpha / F_p\) is an element of order 4 (i.e., \(\alpha^4 = 1 / F_p\)). Then the curve \(E_{p,a}\) has the automorphism \(\lambda : E_{p,a} \rightarrow E_{p,a}\) \((x,y) \mapsto (x, \alpha y)\).

We can regard \(\lambda\) as a complex number which satisfies the equation \(\lambda^2 + 1 = 0\), namely, \(\lambda = \mp \sqrt{-1}\).

Park, Lee, and Park proved the existence of a Frobenius expansion on \(E_{p,b}\) (resp. \(E_{p,a}\)). In the following, for an elliptic curve \(E\) over \(F_q\), we regard an endomorphism \(\phi\) as a complex number (See [69] for detail).

**Proposition 5.3 (\(D_F\)-Frobenius expansion [64, Theorem 2])** Let \(E_{7,b} : y^2 = x^3 \ b\) be the elliptic curve defined over \(F_7\), \(\beta / F_7\) be an element of order 3, \(\omega : E_{7,b} \rightarrow E_{7,b}\) \((x,y) \mapsto (\beta x, y)\) be the automorphism of \(E_{7,b}\), and \(\phi\) be the 7th-power Frobenius map on \(E_{7,b}\). For any \(d / \mathbb{Z}[\omega]\), we can write \(d = \prod_{i=0}^{t-1} c_i \phi^i\), where \(c_i / D_E := \{0, \pm 1, \pm \omega, \pm \omega^2\}, t \geq 2 \log_7 N_{\mathbb{Z}[\omega]/\mathbb{Z}}(d)\) + 1.

**Proposition 5.4 (\(D_G\)-Frobenius expansion [64, Theorem 1])** Let \(E_{5,a} : y^2 = x^3 + ax\) be the elliptic curve defined over \(F_5\), \(\alpha / F_5\) be an element of order 4, \(\lambda : E_{5,a} \rightarrow E_{5,a}\) \((x,y) \mapsto (x, \alpha y)\) be the automorphism of \(E_{5,a}\), and \(\phi\) be the 5th-power Frobenius map on \(E_{5,a}\). For any \(d / \mathbb{Z}[\sqrt{-1}]\), we can write \(d = \prod_{i=0}^{t-1} c_i \phi^i\), where \(c_i / D_G := \{0, \pm 1, \pm \lambda\}, t \geq 2 \log_5 N_{\mathbb{Z}[\lambda]/\mathbb{Z}}(d)\) + 1.
We define
\[
S_E(\ell_\omega, k_\omega, q) := \left\{ d = \sum_{i=0}^{\ell_\omega - 1} a_i \phi^i : a_i \in \mathbb{D}, \ wt(d) \geq k_\omega \right\},
\]
for an elliptic curve $E_{7,b}$, and
\[
S_G(\ell_\lambda, k_\lambda, q) := \left\{ d = \sum_{i=0}^{\ell_\lambda - 1} a_i \phi^i : a_i \in \mathbb{D}, \ wt(d) \geq k_\lambda \right\},
\]
for an elliptic curve $E_{5,a}$.

The proposed method is a modification of the complex exponent test. Whereas each number $s_i$ is randomly chosen from the set $S_1(\ell_1, k_1, q)$ (resp. $S_2(\ell_2, k_2, q)$) in the complex exponent test, each number $s_i$ is randomly chosen from the set $S_E(\ell_\omega, k_\omega, q)$ (resp. $S_G(\ell_\lambda, k_\lambda, q)$) in our proposed method and check whether Equation (5.1) is correct or not. In order to evaluate the security of our method, we have to evaluate an upper bound of $\ell_\omega$ (resp. $\ell_\lambda$) which satisfies that each element in the set $S_E$ (resp. $S_G$) is different from each other as an element in $G = \langle P \rangle$. We also have to calculate the cardinality of the set $S_E$ (resp. $S_G$). Namely, we need analogous results of [18, Theorem 3] and [18, Theorem 4] in Section 5.2.3.

In Section 5.3.2, we evaluate $\#S_E(\ell_\omega, k_\omega, q)$, and in Section 5.3.3, we evaluate $\#S_G(\ell_\lambda, k_\lambda, q)$.

### 5.3.2 Our Proposed Test using the Curve $E_{7,b}$

We show the classification of the elliptic curves of the form $E_{7,b}/\mathbb{F}_7 : y^2 = x^3 - b$ in Table 5.1 (See [47] and [64] for details).

One can show that the Frobenius expansion with coefficients in $D_E$ in $\mathbb{Z}[\omega]$ have unique representation.

**Proposition 5.5 (Uniqueness of the Frobenius expansion with coefficients $D_E$)** Every $d / \mathbb{Z}[\omega]$ has a unique Frobenius expansion with coefficients in $D_E$.

**Proof** Let us assume the contrary and seek a contradiction. We assume that
Table 5.1: Classification of elliptic curves of the form $E_{7,5}$

<table>
<thead>
<tr>
<th>$b$</th>
<th>$t$</th>
<th>$\phi$</th>
<th>relation between $\phi$ and $\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>$2 \pm \sqrt{3}$</td>
<td>$2\omega = 3$ or $\phi + 2\omega = 1$</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>$2 \pm \sqrt{3}$</td>
<td>$2\omega = 1$ or $\phi + 2\omega = 3$</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>$\frac{1 \pm 3\sqrt{-3}}{2}$</td>
<td>$3\omega = 2$ or $\phi + 3\omega = 1$</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>$\frac{-1 \pm 3\sqrt{-3}}{2}$</td>
<td>$3\omega = 1$ or $\phi + 3\omega = 2$</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>$\frac{5 \pm \sqrt{-3}}{2}$</td>
<td>$\omega = 3$ or $\phi + \omega = 2$</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>$\frac{-5 \pm \sqrt{-3}}{2}$</td>
<td>$\omega = 2$ or $\phi + \omega = 3$</td>
</tr>
</tbody>
</table>

there exists an element $d / \mathbb{Z}[\omega]$ which has two different Frobenius expansions with coefficients in $D_E$

$$d \sim e_i \phi^i \sim e_i' \phi^i.$$ 

If $i_0$ is the smallest number such that $e_i = e_i'$ for $0 \leq i \leq i_0$ and $e_i = e_i'$, then we can replace $d$ by $(d \prod_{i=i_0}^{\ell-1} e_i \phi^i)/\phi^{i_0} = \prod_{i=i_0}^{\ell-1} e_i \phi^{i-i_0}$; we have the representations

$$d = \sum_{i=0}^{\tau-1} e_i \phi^i = \sum_{i=0}^{\tau-1} e_i' \phi^i,$$

where $\ell$ is the maximum of length of the two representations. Then we have $\phi \mid (e_0, e_0')$. We put $\kappa := e_0, e_0'$. There exists $\gamma / \mathbb{Z}[\omega]$ such that $\gamma = \phi \gamma$. By taking the square of the usual absolute value of the equation, we obtain $|\gamma|^2 = |\phi|^2 |\gamma|^2$. Remark that $|\gamma|^2$ is a rational integer greater than or equal to 0 and $|\phi|^2 = 7$. Since $\gamma_0 \geq 1$ and $\gamma_0' \geq 1$, we must have $|\gamma|^2 \geq 4$. Hence $|\gamma| = 0$.

Thus we have $e_0 = e_0'$. This is a contradiction. Hence, every $d / \mathbb{Z}[\omega]$ has a unique Frobenius expansion with coefficients in $D_E$. $\square$

Whereas [18, Theorem 3] and [18, Theorem 4] deal with polynomials over $\mathbb{Z}$, our case treat polynomials over $\mathbb{C}$, where each coefficient belongs to $D_E$. Thus, one needs to consider a slightly different strategy to establish analogous results of [18, Theorem 3] and [18, Theorem 4]. Before we turn to establish analogous results, we provide the following remark.

Let $f(X) = \prod_{i=0}^{\ell-1} a_i X^i / \mathbb{C}[X], a_i / D_E$; $0 \geq i \geq \ell, \omega$, 1. Since $\pm \omega^2 = e_0 < 0, \pm \omega^2 > 1$, it is easily seen that $a_i = a_i' + a_i'' \omega$ for some $a_i', a_i'' / \mathbb{Z}$. 

83
By Table 5.1, we can write $\phi = u + v \omega$ for some $u,v \in \mathbb{Z}$. Hence we have $a_i = (a''_i/v)\phi + (a'_{i})(u/v)a''_i$. From this observation, we consider the following transformation: $f(X) = \prod_{i=0}^{\ell-1} a_i X^i \equiv \mathcal{H}_E(f(X)) = \prod_{i=0}^{\ell-1} ((a''_i/v)X + (a'_{i})(u/v)a''_i))X^i \mod Q[X]$. Namely, $\mathcal{H}_E(f(X))$ can be obtained by substitute the coefficients $a_i$ of $f(X)$ for $(a''_i/v)X + (a'_{i})(u/v)a''_i)$. Arranging $\mathcal{H}_E(f(X))$ in descending order yields $\mathcal{H}_E(f(X)) = (a'_0 (u/v)a''_0) + \prod_{i=1}^{\ell-1} ((a'_i (u/v)a''_i) + a''_{i-1})X^i + (a''_{\ell-1}/v)X^{\ell-2}$. We put $C_b = v\setminus$ Then $C_b \ast \mathcal{H}_E(f(X))$ belongs to $\mathbb{Z}[X]$. More precisely, if $b = \mp 1$ then $\mathcal{H}_E(f(X)) / \mathbb{Z}[X]$, and if $b = \mp 2$ (resp. $b = \mp 3$) then $2 * \mathcal{H}_E(f(X)) / \mathbb{Z}[X]$ (resp. $3 * \mathcal{H}_E(f(X)) / \mathbb{Z}[X]$). Since $a'_i, a''_i / \{0,\pm1\} \ast 1 \leq i \leq \ell, a''_{i-1} \sim 0$, we have $\nu_0' (u/v)a''_0 \geq 3, \nu_0' (u/v)a''_i + a''_{i-1}/v \geq 4 \ (1 \leq i \leq \ell, 1)$, and $\nu_{\ell-1} / a''_{i-1} \geq 1$.

For any $D_E$-coefficients Frobenius expansion with appropriate length, it can be proven that each Frobenius expansion as above is different from each other in $\mathbb{Z}[\omega] / (\phi^m 1,n)$.

**Proposition 5.6** Let $P$ be a point on $E_{7,b}(\mathbb{F}_{tm})$, $n$ the order of $P$. Let $M = 24$, $\ell_\omega$ be a positive integer satisfying $3 \geq \ell_\omega \geq m 1$, $C_b / \mathbb{Z}$ a rational integer such that if $b = \mp 1$ then $C_b = 2$, if $b = \mp 2$ then $C_b = 3$, and if $b = \mp 3$ then $C_b = 1$. Let $h_j(X) = \prod_{i=0}^{\ell-2} e_i^{(j)} X^i / \mathbb{C}[X], e_i^{(j)} / D_E$ for $j (j = 1,2)$. We put $f(X) = C_b(h_j(X)) / \mathbb{C}[X]$. Suppose that $n$ is a prime and $2 \ast 7^{2\ell_\omega} \ast M^2 \geq n$. Then $f(\phi)P = \mathcal{O}$ implies that $f(X)$ is divided by $X^2 tX + 7$ in $\mathbb{Z}[X]$.

**Proof** It is sufficient to re-evaluate the inequalities in the proof of [18, Lemma 1] for the case of $q = 7$. By the above observation, there exist $c'_0, \ldots, c'_{\ell_\omega} / \mathbb{Z}$ such that $f(X) = \Pi_{i=0}^{\ell-2} c_i X^i / \mathbb{Z}[X]$. As we explain above, each coefficient of the polynomial $\mathcal{H}_E(h_j(X))$ is equal or less than 4. So $\gamma_i \geq 3 * (4 + 4) = 24 = M$. By dividing $f(X)$ by $X^2 tX + q$, there exist $b_0, b_1, \ldots, b_{\ell-2}, r_0, r_1 / \mathbb{Z}$ such that $f(X) = (X^2 tX + q)(\Pi_{i=0}^{\ell-2} b_i X^i) + r_1 X + r_0$. We evaluate upper bounds of $\gamma_i \ (0 \geq i \leq \ell_\omega 2$) and $\gamma_i \ (i = 0,1)$ using the following number sequence $M_0 = 0, M_1 = M, M_i = M + \gamma_i M_{i-1} + qM_{i-2} (i \sim 2)$. By expanding the right
hand side of \( f(X) \),

\[
\begin{align*}
f(X) = b_{\ell_{-2}}X_{\ell_{-2}} + (b_{\ell_{-3}} + tb_{\ell_{-2}})X_{\ell_{-1}} + & \sum_{i=2}^{\ell_{-2}} (b_{i-2} + tb_{i-1} + qb_{i})X_i \\
+ (tb_0 + qb_1)X + qb_0 + r_1X + r_0.
\end{align*}
\]

(5.6)

By comparing the both sides of Equation (5.6), we have

\[
\begin{align*}
c'_{\ell_{-2}} &= b_{\ell_{-2}} \\
c'_{\ell_{-1}} &= b_{\ell_{-3}} + tb_{\ell_{-2}} \\
c'_i &= b_{i-2} + tb_{i-1} + qb_{i} \quad (2 \geq i \geq \ell_{\omega} - 2) \\
c'_0 &= tb_0 + qb_1 + r_1 \\
c'_0 &= qb_0 + r_0.
\end{align*}
\]

(5.7)

From the definition of the sequence of numbers \( M_i \) and Equation (5.7),

\[
\begin{align*}
\psi_{\ell_{-2}} &\geq |c'_{\ell_{-2}}| \geq M_1 \\
\psi_{\ell_{-3}} &\geq |c'_{\ell_{-3}} + \psi_{\ell_{-2}}| \geq M + \psi_{M_1} = M_2 \\
\psi_{\ell_{i-2}} &\geq |c'_{\ell_{i-2}} + \psi_{\ell_{i-1}} + q\psi_{\ell_{i-1}}| \\
&\geq M + \psi_{M_i} + qM_{i-1} = M_{i+1} \quad (2 \geq i \geq \ell_{\omega}) \\
\psi_1 &\geq |c_1 + \psi_{b_0} + q\psi_0| \\
&\geq M + \psi_{M_{\ell_{-1}} + qM_{\ell_{-2}}} = M_{\ell_{\omega}} \\
\psi_0 &\geq |c_0 + q\psi_0| \geq M + qM_{\ell_{-1}}.
\end{align*}
\]

(5.8)

Next we evaluate upper bounds of \( \psi_0 \) and \( \psi_1 \) by using \( q \) and \( M \). We put \( M' := M/(\psi + q - 1) \), \( M'_i := M_i + M' \ (i \sim 0) \). Then, we have \( M'_i = c\alpha_t^i + d\beta_t^i \).

Here, \( \alpha_t, \beta_t \in \mathbb{C} \) are two roots of the quadratic equation \( X^2 - \psi X + q = 0 \), \( c := (M + M' - \beta_t M')/(\alpha_t - \beta_t) \), and \( d := (M + M' - \alpha_t M')/(\alpha_t - \beta_t) \). Therefore, it satisfies that

\[
|\psi_i| \geq |c\alpha_t^i + d\beta_t^i| + |M'| \\
\geq \max_t \{|c| \psi([i \leq t] * 2 * \max_{t \pm 1, \pm 4} \alpha_t^i \beta_t^i) + M/(\psi + q - 1)\}.
\]

(5.9)
By elementary calculation, we obtain

\[
\max_{t=\pm 1, \pm 4, \pm 5} \left\{ |\alpha_t|, |\beta_t| \right\} = \left( \frac{5 + \sqrt{53}}{2} \right)^i \leq q^i.
\]

In addition,

\[
\|e\| = \sqrt{M + M' - \beta_t M'} |\alpha_t|, \quad \|e\| \geq \left( \sqrt{M + M' + \beta_t M'} \right) / (t^2 + 4q)
\]
\[
< \left( M + \frac{M}{\|\| + q} \right) \left( \frac{1}{\|\| + q} \right) \leq \left( \frac{M}{t^2 + 4q} \right).
\]

Similarly, we have

\[
\|d\| \geq \left( \frac{M}{t^2 + 4q} \right) \left( \frac{q + 1}{\|\| + q} \right) + 1
\]

From the above two inequality, we obtain

\[
\max \left\{ \|e\|, \|d\| \right\} \geq \left( \frac{M}{t^2 + 4q} \right) \left( \frac{q + 1}{\|\| + q} \right) + 1
\]

(5.10)

By combining the estimates (5.9) and (5.10), we obtain

\[
\|M_t\| < \left( \frac{2q^4 M}{t^2 + 4q} \right) \left( \frac{q + 1}{\|\| + q} \right) + 1
\]

Since \( E_{7,b} \) is non-supersingular, we have

\[
1 / (t^2 + 4q) \geq 1 / (t^2 + 4q) \text{ and } 1 / (\|\| + 4q)
\]
$q \geq 1/q$. From $q = 7$ and $1/q < q^i/q$ ($i \sim 1$), we have

$$\begin{aligned}
\|M_i\| &< \left( \frac{2q^1M}{4q+1} \right) \left( \frac{q+1}{q} \right) + 1 \left\lfloor \frac{M}{q} \right\rfloor + 1 \\
&< \left( \frac{2}{4q+1} \right) \left( \frac{q+1}{q} \right) + 1 \left\lfloor \frac{1}{q} \right\rfloor q^iM \\
&< \frac{94}{100} q^iM.
\end{aligned}$$

(5.11)

The above inequality yields

$$\begin{aligned}
\|\gamma_0\| &\geq \|M + q|\lambda_{\omega-1}\| < \left( 1 + q \star \frac{94}{100} \star q^{(\omega-1)} \right) M \\
&< \left( \frac{1}{100} q^\omega + \frac{94}{100} q^{\omega} \right) M \\
&= \frac{95}{100} q^\omega M, \ (\omega \sim 3)
\end{aligned}$$

(5.12)

and

$$\|\gamma_1\| \geq \|M_{\omega}\| < \frac{94}{100} q^\omega M.$$

(5.13)

By Hasse-Weil bound, and the two inequalities (5.12) and (5.13), we obtain

$$\begin{aligned}
N(f(\phi)) &= N(r_1 \phi + r_0) \geq r_0^2 + \|\gamma_0\| r_1^2 + q r_1^2 \\
&\geq r_0^2 + 2 \|\gamma_0\| r_1 + q r_1^2 = (r_0 + \|r_1\|)^2 \\
&< \left( \frac{95}{100} + \frac{94}{100} \right) q^2 \left( q^{\ell_\omega} M \right)^2 \\
&< 3.5 \times 7^{2\ell_\omega} M^2.
\end{aligned}$$

(5.14)

Here, we assume that $f(\phi)P = \emptyset$. Then, $N(f(\phi))P = (f(\phi)f(\phi))P = \emptyset$, and $N(f(\phi)) \subseteq \mathbb{Z}$. Thus we obtain $n \setminus N(f(\phi))$. Hence $N(f(\phi)) < 3.5 \times 7^{2\ell_\omega} M^2 \geq n$ implies $N(f(\phi)) = 0$. Therefore $r_0 = r_1 = 0$. This completes the proof.

If $\ell_\omega$ satisfies the condition of Proposition 5.6, each element in $S_E(\ell_\omega, k_\omega, 7)$ is different from each other.

**Theorem 5.3 (Upper bound of the length of the $D_E$-coefficients Frobe-**
\textbf{nius expansion) } Let \( 3 \geq \omega \geq (\log_2 n \ 2 \log_2 M \ \log_2 3.5)/(2 \log_2 7) \), and \( M = 24 \). Then each element in \( S_E(\ell, k, 7) \) is different endomorphism on \( G \).

**Proof** Let \( h_j(X) = \prod_{i=0}^{\ell-1} e_i^{(j)} X^i \) / \( C[X] \), \( e_i^{(j)} / D \) for \( j = 1, 2 \). Suppose that \( h_1(\phi), h_2(\phi) / S_E(\ell, k, 7) \), and \( h_1(\phi) = h_2(\phi) \). We assume \( h_1(\phi)P = h_2(\phi)P \) and seek a contradiction. Without loss of generality, we may assume that \( e_0^{(1)} = e_0^{(2)} \). We take \( f(X) = C_b(\mathcal{H}_E(h_1(X)) \ 5 \mathcal{H}_E(h_2(X))) \). By Proposition 5.6, \( f(X) \) is divisible by \( X^2 - tX + 7 \). Put \( e_0^{(j)} := e_0^{(j)} + e_0^{(j)} \omega \) for \( j = 1, 2 \). Then \( (e_0^{(j)} e_0^{(j)}) (u/v)(e_0^{(j)} e_0^{(j)}) \) must be divisible by \( 7 \). However, one can easily check that there do not exist \( e_0^{(0)}, e_0^{(1)} / D \) such that \( 7 (e_0^{(0)} e_0^{(1)}) (u/v)(e_0^{(0)} e_0^{(1)}) \). This is a contradiction. Thus we have the desired result. \( \square \)

By Theorem 5.3, if \( 3.5 \times 7^{2\ell} \times 24^2 \geq n \), the cardinality of the set \( S_E \) is given by the following theorem.

**Theorem 5.4 [Cardinality of the set \( S_E \)]**

\[
\#S_E(\ell, k, 7) = \sum_{i=0}^{\ell} \ell \omega i 6^i.
\]

**Proof** By elementary combinatorics, we obtain the above equation. \( \square \)

Our proposed test using the curve \( E_{T,b} \) is described in Algorithm 10.

By the same argument as Section 5.2.3, the computational cost of the proposed test using the Curve \( E_{T,b} \) is given by

\[
N + 2|A| + 3\ell M + 2M + (1 \text{ scalar multiplication}.)
\] (5.15)

**5.3.3 Our Proposed Test using the Curve \( E_{5,a} \)**

By similar argument, we can show the results in Section 5.3.2.

We show the classification of the elliptic curves of the form \( E_{5,a} / \mathbb{F}_5 : y^2 = x^3 + ax \) in Table 5.2 (See [64] for details).
Table 5.2: Classification of elliptic curves of the form $E_{5,a}$

<table>
<thead>
<tr>
<th>$a$</th>
<th>$t$</th>
<th>$\phi$</th>
<th>relation between $\phi$ and $\lambda$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>$1 \mp 2 \pm 1$</td>
<td>$\phi \mp 2\lambda = 1$</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>$1 \mp 2 \pm 1$</td>
<td>$\phi \mp 2\lambda = 1$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$2 \mp 1$</td>
<td>$\phi \mp \lambda = 2$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$2 \mp 1$</td>
<td>$\phi \mp \lambda = 2$</td>
</tr>
</tbody>
</table>

Algorithm 10 Proposed Batch Verification Test using the curve $E_{7,b}$

**Input:** batch instance $\{(x_1, Q_1), \ldots, (x_N, Q_N)\}$

**Output:** Accept or Reject

1: Choose $N$ random elements $s_1, \ldots, s_N$ from $S_E(\ell_\omega, k_\omega, 7)$. Write $s_i = \prod_{j=0}^{\ell_\omega-1} c_{i,j} \phi^j$ and $\epsilon_{i,j} = 0$ (if $c_{i,j} = 0$), $\epsilon_{i,j} = 1$ (if $c_{i,j} \neq 0$, $\omega$, $\omega^2$) for nonzero $c_{i,j}$ for each $i$.

2: $s \rightarrow \prod_{i=1}^{N} s_i x_i \mod (\phi^{m-1} + \phi^{m-2} + \cdots + \phi + 1)$

3: $R[i] \rightarrow \emptyset$ for $i \in \{1, \omega, \omega^2\}$

4: for $j$ from $\ell_\omega$ downto 0 do

5: $R[1] \rightarrow \phi(R[1])$, $R[\omega] \rightarrow \phi(R[\omega])$, $R[\omega^2] \rightarrow \phi(R[\omega^2])$

6: for $i$ from 1 to $N$ do

7: if $c_{i,j} = 0$ then

8: if $c_{i,j} = \mp 1$ then

9: $R[1] \rightarrow R[1] + \epsilon_{i,j}(Q_i)$

10: else if $c_{i,j} = \mp \omega$ then

11: $R[\omega] \rightarrow R[\omega] + \epsilon_{i,j}(Q_i)$

12: else

13: $R[\omega^2] \rightarrow R[\omega^2] + \epsilon_{i,j}(Q_i)$

14: end if

15: end if

16: end for

17: end for

18: $Q \rightarrow \omega(R[\omega^2])$, $Q \rightarrow \omega(R[\omega] + Q)$, $Q \rightarrow Q + R[1]$

19: if $Q = sP$ then

20: return Accept

21: else

22: return Reject

23: end if
One can show that the Frobenius expansion with coefficients in $D_G$ in $\mathbb{Z}[\lambda]$ have unique representation.

**Proposition 5.7 (Uniqueness of the Frobenius expansion with coefficients $D_G$)** Every $d / \mathbb{Z}[\lambda]$ has a unique Frobenius expansion with coefficients in $D_G$.

**Proof** Let us assume the contrary and seek a contradiction. We assume that there exists an element $d / \mathbb{Z}[\lambda]$ which has two different Frobenius expansions with coefficients in $D_G$

$$d = e_i \phi^i = e'_i \phi'^i.$$  

If $i_0$ is the smallest number such that $e_i = e'_i$ for $0 \geq i \geq i_0$ and $e_{i_0} = e'_{i_0}$, then we can replace $d$ by $(d \prod_{i=i_0}^{\ell-1} e_i \phi^i) / \phi^i = \prod_{i=i_0}^{\ell-1} e_i \phi^{i-i_0}$, we have the representations

$$d = e_i \phi^i = e'_i \phi'^i, \quad e_0 = e'_0,$$

where $\ell$ is the maximum of length of the two representations. Then we have $\phi \setminus (e_0, e'_0)$. We put $\kappa := e_0, e'_0$. There exists $\gamma / \mathbb{Z}[\lambda]$ such that $\kappa = \phi \gamma$. By taking the square of the usual absolute value of the equation, we obtain $|\gamma|^2 = |\phi|^2 |\gamma|^2$. Remark that $|\gamma|^2$ is a rational integer greater than or equal to 0 and $|\phi|^2 = 5$. Since $\varphi_0 \geq 1$ and $\varphi'_0 \geq 1$, we must have $|\gamma|^2 \geq 4$. Hence $|\gamma| = 0$.

Thus we have $e_{i_0} = e'_{i_0}$. This is a contradiction. Hence, every $d / \mathbb{Z}[\lambda]$ has a unique Frobenius expansion with coefficients in $D_G$. \hfill $\square$

As in Section 5.3.2, we prepare the following transformation. Let $f(X) = \prod_{i=0}^{\ell-1} a_i X^i \in \mathbb{C}[X]$, $a_i \in D_G (0 \geq i \geq \ell_{\lambda} - 1)$. It is easily seen that $a_i = a'_i + a''_i \lambda$ for some $a'_i, a''_i \in \mathbb{Z}[\lambda]$ satisfying $a_i a''_i = 0$. By Table 5.2, we can write $\phi = u + v\lambda$ for some $u, v / \mathbb{Z}$. Hence we have $a_i = (a''_i / v) \phi + (a'_i + u/v) a''_i$. From this observation, we consider the following transformation:

$$f(X) = \prod_{i=0}^{\ell-1} a_i X^i \in \mathbb{H}_G(f(X)) = \prod_{i=0}^{\ell-1} ((a''_i / v) X + (a'_i / v) a''_i) X^i \in \mathbb{Q}[X].$$

Namely, $\mathbb{H}_G(f(X))$ can be obtained by substitute the coefficients $a_i$ of $f(X)$ for $(a''_i / v) X + (a'_i / v) a''_i$. Arranging $\mathbb{H}_G(f(X))$ in descending order yields $\mathbb{H}_G(f(X)) = (a'_0 (u/v) a''_0) + \prod_{i=0}^{\ell-1} ((a'_i (u/v) a''_i) + a''_{i-1} / v) X^i + (a''_{\ell-1} / v) X^\ell$. We put $C_a = \varphi \setminus C_a$. Then $C_a \ast \mathbb{H}_G(f(X))$ belongs to $\mathbb{Z}[X]$. More precisely, if $a = \pm 2$
then $\mathcal{H}_G(f(X)) \in \mathbb{Z}[X]$, and if $a = \pm 1$ then $2 \ast \mathcal{H}_G(f(X)) \in \mathbb{Z}[X]$. Since $a'i, a''i / 0, \pm 1 \leftarrow \mathbb{Z}$ and $a'a'' = 0$, we have $|a'_0| (u/v)a'' \geq 2, |a'_i (u/v)a''| + a''i-1/v \geq 3$ (1 $\leq i \leq \ell$, 1), and $|a''i-1/v| \geq 1$.

For $D_G$-coefficients Frobenius expansion with appropriate length, it can be proven that each above Frobenius expansion is different from each other in $\mathbb{Z}[X]/(\phi^m 1, n)$.

**Proposition 5.8** Let $P$ be a point on $E_{5,a}(\mathbb{F}_{5^m})$, $n$ the order of $P$. Let $M = 12, \ell_\lambda$ be a positive integer satisfying $3 \geq \ell_\lambda \geq m - 1$, $C_a / \mathbb{Z}$ be a rational integer such that if $a = \pm 1$ then $C_a = 2$, if $a = \pm 2$ then $C_a = 1$. Let $h_j(X) = \prod_{i=0}^{\ell_\lambda - 1} c^{(j)}_i X^t / \mathbb{C}[X]$, $c^{(j)}_i / D_G$ for $j (j = 1, 2)$. We put $f(X) = C_a (\mathcal{H}_G(h_1(X)) \mathcal{H}_F(h_2(X))) / \mathbb{Z}[X]$. Suppose that $n$ is a prime and $5^{2\ell_\lambda} \ast M^2 \geq n$. Then $f(\phi)P = \emptyset$ implies that $f(X)$ is divided by $X^2 \ tX + 5$ in $\mathbb{Z}[X]$.

**Proof** The proof is very similar to that of Proposition 5.6. By the above observation, there exist $c_0, \ldots, c_{\ell_\lambda} / \mathbb{Z}$ such that $f(X) = \prod_{i=0}^{\ell_\lambda} c_i X^t / \mathbb{Z}[X]$. As we explain above, each coefficient of the polynomial $\mathcal{H}_G(h_j(X))$ is equal or less than 3. So $|c^{(j)}_i| \geq 2 \ast (3 + 3) = 12 = M$. By dividing $f(X)$ by $X^2 \ tX + q$, there exist $b_0, b_1, \ldots, b_{\ell_\lambda - 2}, r_0, r_1 / \mathbb{Z}$ such that $f(X) = (X^2 \ tX + q)(\prod_{i=0}^{\ell_\lambda - 2} bX^t) + r_1 X + r_0$. By similar argument as in the proof of Proposition 5.6, we have Equation (5.7) and Inequality (5.8). By the same discussion as in the proof of Proposition 5.6, we have $M'_t = ca'_t + d\beta^t_i$, where $\alpha_t, \beta_t / \mathbb{C}$ are two roots of the quadratic equation $X^2 \ qX = 0, c = (M + M' \beta_t M')/(\alpha_t \beta_t)$, and $d = (M + M' \alpha_t M')/(\alpha_t \beta_t)$. Therefore, it satisfies that

$$|M'_t| \geq |M'_t| + |M'| = |c'_t + d\beta^t_i| + |M'| \geq \max_{t=\pm 2, 4 \pm 1} |c'_t + \beta^t_i| + M'(q + q 1). \quad (5.16)$$

By elementary calculation, we obtain

$$\max_{t=\pm 2, 4 \pm 1} |\alpha'_t| \sqrt{v_2} \geq 5^t = q^t.$$ 

By the same reason in the proof of Proposition 5.6, we obtain Inequality (5.10).
By combining the estimates (5.10) and (5.16), we obtain

\[ |M_i| \geq \left( \frac{2q^iM}{t^2+4q} \right)^* \left( \frac{q+1}{q} \right) + 1 + \frac{M}{q^{i+1}}. \]

Since \( E_{5,a} \) is non-supersingular, we have \( \frac{1}{t^2+4q} \geq 1/1+4q \) and \( 1/((q^i+q^i) \geq 1/q \). From \( q = 5 \) and \( 1/q < q^i/q \) \((i \sim 1)\), we have

\[ |M_i| \geq \left( \frac{2q^iM}{4q+1} \right)^* \left( \frac{q+1}{q} \right) + 1 + \frac{M}{q^{i+1}} \]
\[ < \left( \frac{2}{4q+1} \right) \left( \frac{q+1}{q} \right) + 1 + \frac{1}{q} q^i M \]
\[ < \frac{117}{100} q^i M. \] \( (5.17) \)

The above inequality yields

\[ |r_0| \geq \left| M \right| + q |M_{\ell-1}| < 1 + q \times \left( \frac{117}{100} \times q^{\ell-1} \right) M \]
\[ < \left( \frac{1}{100} q^{\ell} + \frac{117}{100} \times q^{\ell} \right) M \]
\[ = \frac{118}{100} q^{\ell} M, \ (\ell \sim 3) \] \( (5.18) \)

and

\[ |r_1| \geq \left| M_{\ell} \right| < \frac{117}{100} q^{\ell} M. \] \( (5.19) \)

By Hasse-Weil bound, and the two inequalities (5.18) and (5.19), we obtain

\[ N(f(\phi)) = N(r_1 \phi + r_0) \geq r_0^2 + q |r_0| |r_1| + q r_1^2 \]
\[ \geq r_0^2 + 2 q |r_0| |r_1| + q r_1^2 = (r_0 + q r_1)^2 \]
\[ < \left( \frac{118}{100} + \frac{117}{100} \right) q^2 (q^{\ell} M)^2 \]
\[ < 3.8 \times q^{2\ell} M^2. \] \( (5.20) \)

Here, we assume that \( f(\phi)P = \emptyset \). Then, \( N(f(\phi))P = (f(\bar{\phi})f(\phi))P = \emptyset \), and
\[ N(f(\phi)) \in \mathbb{Z}. \] Thus we obtain \( n \setminus N(f(\phi)) \). Hence \( N(f(\phi)) < 3.8 \times 5^{2\lambda} M^2 \geq n \) implies \( N(f(\phi)) = 0 \). Therefore \( r_0 = r_1 = 0 \). This completes the proof. \( \square \)

If \( \ell_\lambda \) satisfies the condition of Proposition 5.8, each element in \( S_G(\ell_\lambda, k_\lambda, 5) \) is different from each other.

**Theorem 5.5 (Upper bound of the length of the \( D_G \)-coefficients Frobenius expansion)** Let \( 3 \geq \ell_\lambda \geq (\log_2 n - 2 \log_2 M - \log_2 3.8)/\log_2 5 \), and \( M = 12 \).

Then each element in \( S_G(\ell_\lambda, k_\lambda, 5) \) is different endomorphism on \( G \).

**Proof** Let \( h_j(X) = \prod_{i=0}^{\ell_j-1} e_i^{(j)} X^i / \mathbb{C}[X] \), \( e_i^{(j)} / D_\xi \) for \( j \) (\( j = 1, 2 \)). Suppose that \( h_1(\phi), h_2(\phi) / S_G(\ell_\lambda, k_\lambda, 7) \), and \( h_1(\phi) = h_2(\phi) \). We assume \( h_1(\phi)P = h_2(\phi)P \) and seek a contradiction. Without loss of generality, we may assume that \( e_0^{(1)} = e_0^{(2)} \). We take \( f(X) = h_1(X) = h_2(X) \). By Proposition 5.8, \( C_a \ast f(X) \) is divisible by \( X^2 \), \( tX + 5 \). Put \( e_0^{(j)} := e_0^{(j)^\prime} + e_0^{(j)^{\prime\prime}} \lambda \) for \( j \) (\( j = 1, 2 \)). Then \( C_a \ast (e_0^{(0)} e_0^{(1)^\prime}) (u/v)(e_0^{(0)^{\prime\prime}} e_0^{(1)^{\prime\prime}}) \) must be divisible by 5. Recall that \( e_0^{(j)} e_0^{(j)^{\prime\prime}} = 0 \) (\( j = 1, 2 \)). From this fact, one can easily check that there do not exist \( e_0^{(0)}, e_0^{(1)} / D_\xi \) such that \( 5 \left( C_a \ast (e_0^{(0)^\prime} e_0^{(1)^\prime}) (u/v)(e_0^{(0)^{\prime\prime}} e_0^{(1)^{\prime\prime}}) \right) \).

This is a contradiction. Thus we have the desired result. \( \square \)

By Theorem 5.5, if \( 3.8 \times 5^{2\lambda} \times 12^2 \geq n \), the cardinality of the set \( S_G \) is given by the following theorem.

**Theorem 5.6 (Cardinality of the set \( S_G \))**

\[
\#S_G(\ell_\lambda, k_\lambda, 5) = \sum_{i=0}^{k_\lambda} \ell_\lambda^i 4^i.
\]

Our proposed test using the curve \( E_{5,a} \) is similar to Algorithm 10 and one can easily describe the algorithm using the curve \( E_{5,a} \) by modifying Algorithm 10.

By the same argument as Section 5.2.3 and Section 5.3.2, the computational cost of the proposed test using the Curve \( E_{5,a} \) is given by

\[
k_{by} N + 1^1 A + 2\ell_\lambda F + M + (1 \text{ scalar multiplication}). \tag{5.21}
\]
5.4 Sample Parameters

Table 5.3 and Table 5.4 list sample parameters for an elliptic curve $E_{5,a}/\mathbb{F}_5$ and an elliptic curve $E_{7,a}/\mathbb{F}_7$, respectively. The extension degree $m$'s are prime and are selected so that there exists an elliptic curve $E_{5,b}/\mathbb{F}_5$ or an elliptic curve $E_{7,b}/\mathbb{F}_7$ having the cofactor $\#E_{5,a}(\mathbb{F}_5)$ or $\#E_{7,b}(\mathbb{F}_7)$, respectively. The prime order $n$ is presented in hexadecimal form. A backslash at the end of a line indicates that the number (hexadecimal) is continued in the next line.

- $m$ The extension degree of the finite field $\mathbb{F}_5^m$ (resp. $\mathbb{F}_7^m$).
- $f(x)$ The irreducible trinomial of degree $m$ in $\mathbb{F}_5[x]$ (resp. $\mathbb{F}_7[x]$).
- $h$ The cofactor. $h = \#E(\mathbb{F}_5)$ (resp. $\#E(\mathbb{F}_7)$).
- $n$ $n = \#E(\mathbb{F}_5)/h$ (resp. $\#E(\mathbb{F}_7)/h$).
- bitlen($x$) bit length of a positive integer $x$ / $\mathbb{Z}$.

| $E_{5,2}$-97: $m = 97$, $f(x) = x^{97} + x^{58} + 2$, $a = 2$, $h = 2$, bitlen($h$) = 2, bitlen($n$) = 225 |
| n = 0x 00000001 2BA095DC 7701D9CB 7743E3A2 B0E3DBDC E284B04F 367F1914 ED5FBA01 |
| $E_{5,2}$-107: $m = 107$, $f(x) = x^{107} + x^{9} + 1$, $a = 2$, $h = 2$, bitlen($h$) = 2, bitlen($n$) = 248 |
| n = 0x 00AE67F1 E9AEC071 87EDB959 0680A3AA 02A9DFE5 BD41A836 BA41FAF8 6FAE279D |
| $E_{5,-2}$-137: $m = 137$, $f(x) = x^{137} + 4x^{41} + 1$, $a = 2$, $h = 10$, bitlen($h$) = 4, bitlen($n$) = 315 |
| n = 0x 06E10D08 B8EA1322 EBB383AA 1FA7D0FB 9D92FC04 1360D417 5A223E6C CB8DB93B C8B3F993 E09D2061 |
| $E_{5,-2}$-151: $m = 151$, $f(x) = x^{151} + x^{61} + 1$, $a = 2$, $h = 10$, bitlen($h$) = 4, bitlen($n$) = 348 |
| n = 0x 09C69A97 28AB578D 7FF2A760 414536EF BCA758CB F4FBCBOA 32CC8363 855BB99A 84122DF7 C08209C8 5F656D4D |
| $E_{5,-2}$-167: $m = 167$, $f(x) = x^{167} + x^{66} + 1$, $a = 2$, $h = 10$, bitlen($h$) = 4, bitlen($n$) = 385 |
| n = 0x 00000001 5BAE5998 400A95D3 5EAC354F 34215CD4 6E417018 FB1DC739 C5E736BD C153819F 71B64393 465FC46B F4AB38FF A407344D |
| \(E_{5,1\cdot227}: \) & \(m = 227, \) & \(f(x) = x^{227} + x^{53} + 1, \) & \(a = 1, \) & \(h = 4, \) & \(\text{bitlen}(h) = 3, \) & \(\text{bitlen}(n) = 526 \) |
|---|---|---|---|---|---|---|
| \(n = \text{0x } 000021C5\) & 29DD78FA & 571E196B & 3EBB0D20 & 429C476A & 1848CAB5 & \checkmark |
| & E0E8A121 & 378DE187 & 888F99D2 & 99F404EE & 4F9BC974 & D5035A62 & \checkmark |
| & AC9F5E1E & 0DA29A51 & 0B4012E2 & 3ECD1590 & 9A4B1065 & |

| \(E_{5,2\cdot239}: \) & \(m = 239, \) & \(f(x) = x^{239} + 4x^{124} + 1, \) & \(a = 2, \) & \(h = 2, \) & \(\text{bitlen}(h) = 2, \) & \(\text{bitlen}(n) = 554 \) |
|---|---|---|---|---|---|---|
| \(n = \text{0x } 000003D6\) & D75FB22B & 2D628AB8 & 42BDA59D & D2C76CE2 & C5B53D88 & |
| & A4349EFF & F107E218 & 706735C4 & 7CC53BFD & E966A0AC & 0346C5EE & \checkmark |
| & 1F25BD2D & 2FA79AC & F4DA7B45 & 947F51BB & 5E33A643 & A74A470D & \checkmark |

| \(E_{5,3\cdot317}: \) & \(m = 317, \) & \(f(x) = x^{317} + x^{24} + 4, \) & \(a = 2, \) & \(h = 10, \) & \(\text{bitlen}(h) = 4, \) & \(\text{bitlen}(n) = 733 \) |
|---|---|---|---|---|---|---|
| \(n = \text{0x } 1A8662F3\) & B3919708 & 2BF4C02D & 548C2A3D & 779053F3 & CF932023 & |
| & 7E5EC14C & 922CB561 & 85BD758D & 2B99F28C & ADBC9799 & 509367D4 & \checkmark |
| & E379DA56 & D0582105 & 28C0C11C & 7B00363E & 904270F & 4F3A60D4 & \checkmark |
| & 8CE325D4 & 4171E2E9 & 68316505 & D41F503A & 36E361D1 & |

| \(E_{5,3\cdot353}: \) & \(m = 353, \) & \(f(x) = x^{353} + x^{42} + 2, \) & \(a = 1, \) & \(h = 4, \) & \(\text{bitlen}(h) = 3, \) & \(\text{bitlen}(n) = 818 \) |
|---|---|---|---|---|---|---|
| \(n = \text{0x } 00031E34\) & C30449D0 & 361BD9BB & 7B921C55 & EDFE15C3 & 63B2DB07 & |
| & E102C653 & 7B1E4A4D & B5613C98 & 5D75D2B6 & 9C6D5CCE & AD8A0A8F & \checkmark |
| & 01E021FB & 5DCA0A1F & 6B4BE3D8 & FE3A5043 & 09BF6B34 & 81B13AF6 & \checkmark |
| & C4DB17A1 & ED39F3BA & E85063F6 & A98D2DCF & 776DAB2E & 24EB92CE & \checkmark |
| & 6819B5A7 & 32EB3841 & |

| \(E_{5,3\cdot397}: \) & \(m = 397, \) & \(f(x) = x^{397} + x^{64} + 4, \) & \(a = 2, \) & \(h = 10, \) & \(\text{bitlen}(h) = 4, \) & \(\text{bitlen}(n) = 919 \) |
|---|---|---|---|---|---|---|
| \(n = \text{0x } 00597B64\) & 43800D18 & F17B748F & F893C1B1 & 43EF63CB & 0FBE159E & \checkmark |
| & 81A88F44 & 5A61A016 & 77803E71 & 6BDF2E24 & E0E1C364 & 07217F0E & \checkmark |
| & OC5AE445 & 12F85C59 & 7570046F & F08D5D1B & 9214AE06 & 3105AB29 & \checkmark |
| & 53E07DE4 & DED5B4C2 & A8FC11B6 & 0DA9FF06 & 724F75F7 & 5936C6D3 & \checkmark |
| & 122C03F3 & 6C7BC7F3 & A29A1678 & CCF178A0 & A81A1791 & \checkmark |

| \(E_{5,3\cdot401}: \) & \(m = 401, \) & \(f(x) = x^{401} + x^{47} + 1, \) & \(a = 1, \) & \(h = 4, \) & \(\text{bitlen}(h) = 3, \) & \(\text{bitlen}(n) = 930 \) |
|---|---|---|---|---|---|---|
| \(n = \text{0x } 00000022\) & 22279F75 & FD0FF0BD & E402F0BA & B1D5B471 & 249E96E3 & \checkmark |
| & 95AEF3F2 & 5F4E6E81 & A4DB9200 & 603D1F44 & E5AF2416 & 91F31205 & \checkmark |
| & 85F1FB3E & 68C23F98 & 48DA803 & 483E223B & F6475435 & 908078EE & \checkmark |
| & 74D28C7D & 7AEDFDC & 9B759941 & 699DB218 & 11FB4156 & 492C66D9 & \checkmark |
| & AF41C992 & CB1A08F9 & E89ABC23 & C39054AC & 9C4704F34 & 67A26B61 & \checkmark |

\(E_{5,3\cdot439}: \) & \(m = 439, \) & \(f(x) = x^{439} + 4x^{35} + 1, \) & \(a = 2, \) & \(h = 10, \) & \(\text{bitlen}(h) = 4, \) & \(\text{bitlen}(n) = 1017 \) |
|---|---|---|---|---|---|---|

\(\)
<table>
<thead>
<tr>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
<th>$\mathcal{E}_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n = 0x$ 0100CCFD 6D32E361 2A7F4535 9B775336 206E4053 5E362EC5</td>
<td>7D5C0F1E B832C209 C2B8B468 3B54F5EC 018C1A10 A8DECBFE</td>
<td>C96B211E E977B7C4 6DCEF7E2 C81A8F0E 7DDFC28A 051C3168</td>
<td>8DA47D5C 7E6B9E9F F1F5A336 8CAFE22B 9A5A9357 7377B7A8</td>
<td>342935DB 09E38487 A25E300C 5F68178B B3A4E79A 331AE55C</td>
<td>5091770A</td>
<td>DF9A5B4D</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| $E_{7,-59}$: $m = 59$, $f(x) = x^{59} + x^4 + 3$, $b = 1$, $h = 12$, \text{bitlen}(h) = 4$ | $E_{7,61}$: $m = 61$, $f(x) = x^{61} + x^4 + 1$, $b = 3$, $h = 3$, \text{bitlen}(h) = 2$ | $E_{7,71}$: $m = 71$, $f(x) = x^{71} + x^{10} + 4$, $b = 3$, $h = 3$, \text{bitlen}(h) = 2$ | $E_{7,103}$: $m = 103$, $f(x) = x^{103} + x^{54} + 6$, $b = 3$, $h = 13$, \text{bitlen}(h) = 4$ | $E_{7,286}$: $m = 286$, $f(x) = x^{286} + x^{107} + x^{15} + 6$, $b = 2$, $h = 7$, \text{bitlen}(h) = 3$ | $E_{7,298}$: $m = 298$, $f(x) = x^{298} + x^{109} + 6x^{23} + 6$, $b = 1$, $h = 12$, \text{bitlen}(h) = 4$ | $E_{7,303}$: $m = 303$, $f(x) = x^{303} + x^{127} + x^2 + 6$, $b = 1$, $h = 4$, \text{bitlen}(h) = 3$ | $E_{7,355}$: $m = 355$, $f(x) = x^{355} + 3x^{59} + 1$, $b = 3$, $h = 3$, \text{bitlen}(h) = 2$ | $E_{7,468}$: $m = 468$, $f(x) = x^{468} + x^{167} + 3x^{39} + 1$, $b = 3$, $h = 3$, \text{bitlen}(h) = 2$ |
| \text{bitlen}(n) = 163 | \text{bitlen}(n) = 170 | \text{bitlen}(n) = 198 | \text{bitlen}(n) = 286 | \text{bitlen}(n) = 287 | \text{bitlen}(n) = 298 | \text{bitlen}(n) = 303 | \text{bitlen}(n) = 355 | \text{bitlen}(n) = 468 |

Table 5.4: Sample Parameters for the elliptic curves $E_{7,b}/\mathbb{F}_7$
5.5 Comparison

5.5.1 Timings

In order to evaluate the computational costs of the CE test and the proposed test, we implement arithmetic operations a finite field $\mathbb{F}_{7^{103}}$ (See Table 5.4 in Section 5.4). We use $f(x) = x^{103} + x^{54} + 6$ and $E_{7,-3-103}$ (Table 5.4) for the irreducible trinomial of degree 103 and the estimation of the computational costs, respectively. The bit length of the prime order $n$ of this curve is 286. For the finite field implementation, we use polynomial basis.

The platform is an Intel® *1) Core™ 2 Duo Processor E8400 (2.99GHz) with *1) Windows is a registered trademark of Microsoft Corporation in the United States
2GB RAM computer, Windows® XP. Programs are all written in ANSI C language with gcc 3.4.4 compiler using the flags “-O3”.

In the case of \( p = 3 \), the three elements in \( \mathbb{F}_3 \) are represented by two bits, and it is known that the method to construct the addition in \( \mathbb{F}_3 \) by using bitwise operations such as “AND” operation or “XOR” operation (cf. [38]). In our implementation, the seven elements in \( \mathbb{F}_7 \) are represented by three bits, and we extend the method in [38] to implement the arithmetic operations in \( \mathbb{F}_{7^{16}} \). The timings are listed in Table 5.5.

<table>
<thead>
<tr>
<th>Finite Field Operations</th>
<th>timing (in ( \mu \text{sec} ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addition</td>
<td>0.063</td>
</tr>
<tr>
<td>Subtraction</td>
<td>0.063</td>
</tr>
<tr>
<td>7-th powering</td>
<td>1.400</td>
</tr>
<tr>
<td>Multiplication</td>
<td>15.230</td>
</tr>
</tbody>
</table>

### 5.5.2 Estimate

We estimate batch verification costs of the CE test and our proposed test in the case of \( E_{7,-3}\text{-103} \) (See Table 5.4 in Section 5.4). The computational cost of the CE tests are given by (5.2), (5.3), respectively. The computational cost of the proposed test using \( E_{7,b} \) is given by (5.15). From Theorem 5.1 and Theorem 5.3, for \( n \) for \( E_{7,-3}\text{-103} \) in Table 5.4, the maximal value of \( \ell_1, \ell_2, \ell_\omega \) are \( \ell_1 = 97, \ell_2 = 96, \ell_\omega = 98 \), respectively. We use the Jacobian coordinate for point addition (12 times finite field multiplications and four times squarings), and the Jacobian coordinate for Frobenius map (three times 7\textsuperscript{th}-powerings) for the estimation. Moreover, it is assumed that

\[ \equiv \text{the costs of finite field multiplication and finite field squaring are equal to each other.} \]

and other countries.

\(^{1)}\) Intel, Core are trademarks or a registered trademarks of Intel Corporation in the United States and other countries.
\[ \equiv \text{the cost of a } 7\text{th powering and } 0.092 \text{ times finite field multiplication are equal (from the result in Table 5.5).} \]

We show the numbers of finite field multiplications that need for the CE test and the proposed test in Table 5.6 by using (5.2), (5.3), and (5.15) \(^{1)}\).

<table>
<thead>
<tr>
<th>Method</th>
<th># of M</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE test ((S_1))</td>
<td>(16k_1N + 284.21)</td>
</tr>
<tr>
<td>CE test ((S_2))</td>
<td>(16k_2N + 1222.72)</td>
</tr>
<tr>
<td>Our test ((S_E))</td>
<td>(16k_\omega N + 59.04)</td>
</tr>
</tbody>
</table>

For \(\ell_1 = 97, \ell_2 = 96, \ell_\omega = 98\) and for a given \(k_{\text{bw}}\), we describe the values \(k_1, k_2, k_\omega\) so that \(#S_j \subset 2^{k_{\text{bw}}} , #S_E \subset 2^{k_{\text{bw}}}\).

<table>
<thead>
<tr>
<th>Method</th>
<th># of M</th>
</tr>
</thead>
<tbody>
<tr>
<td>CE test ((S_1))</td>
<td>5 8 11 14 18 22</td>
</tr>
<tr>
<td>CE test ((S_2))</td>
<td>4 6 9 11 14 16</td>
</tr>
<tr>
<td>Our test ((S_E))</td>
<td>6 9 12 16 21 25</td>
</tr>
</tbody>
</table>

For a given method in Table 5.6, a given number of signatures \(N\), and a given security parameter \(k_{\text{bw}}\), we can estimate the number of multiplications in \(\mathbb{F}_{\text{res}}\) by substituting \(N\) and \(k_j\) or \(k_\omega\) in Table 5.7 into the method in Table 5.6. Our method is faster than the complex exponent test when the number of signatures \(N\) satisfy Table 5.8:

\(^{1)}\) In order to simplify the computational costs of the algorithm in [18, Fig. 3] and our proposed test (Algorithm 10), we use left-to-right version of BGMW method. However, even if we compare the computational costs of the algorithm in [18, Fig. 3] and the right-to-left version of Algorithm 10, our algorithm is faster than the algorithm [18, Fig. 3] for verifying a limited number of signatures because the number of the Frobenius map in the right-to-left version of Algorithm 10 is larger than that of [18, Fig. 3].
Table 5.8: the number of signatures $N$ for which our method is faster than CE test

<table>
<thead>
<tr>
<th>Security Param. $k_{iv}$</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>140</th>
</tr>
</thead>
<tbody>
<tr>
<td># of signatures</td>
<td>$N \geq 18$</td>
<td>$N \geq 18$</td>
<td>$N \geq 18$</td>
<td>$N \geq 9$</td>
<td>$N \geq 6$</td>
<td>$N \geq 6$</td>
</tr>
</tbody>
</table>

From Table 5.8, we can see that our test is faster than the CE test when the number of signatures is relatively small. Thus, our proposed test is suitable for verifying a limited number of signatures in real-time such as IP camera surveillance systems.

5.6 Summary of this chapter

In this chapter, we described a new batch verification test based on elliptic curves defined over fields with characteristics 5 and 7. Our proposed method is a modification of the complex exponent test (CE test) proposed by Cheon and Lee. The difference between the new test and the CE test is that our test use another digit set of Frobenius expansion in order to accelerate batch verification. We evaluated the security of our new test, and the proof of the security of our new test can be treated similar to the one of CE test. We also evaluated the computational cost of the new test. Our estimation indicates that the proposed test is suitable for verifying a limited number of signatures.
Chapter 6

Conclusion and Open Problems

6.1 Conclusion

This thesis motivated the need for efficient algorithms for elliptic curve cryptosystems. Our main contributions are as follows:

1. We develop efficient scalar multiplication algorithms for subfield elliptic curves in odd characteristic where the traces of Frobenius are equal to 1 or 1. These are two generalizations of $\tau$-NAF on Koblitz curves. The first generalization is $\phi$-GNAF. $\phi$-GNAF can be applied to subfield elliptic curves where the traces of Frobenius are equal to 1. The second generalization is $\phi$-rNAF. $\phi$-rNAF can be applied to subfield elliptic curves where the traces of Frobenius are equal to 1 or 1. The digit set of $\phi$-rNAF is larger than that of $\phi$-GNAF. If the size of the ground field $q$ is small (e.g., 3, 5), the difference between the computational costs for the precomputation tables of $\phi$-GNAF and $\phi$-rNAF is relatively small (a few elliptic additions). But, if $q$ is significantly large, the computational cost for the precomputation table of $\phi$-rNAF is quite large compared to that for $\phi$-GNAF. However the non-zero density for $\phi$-rNAF is significantly smaller than that for $\phi$-GNAF. Thus, these two generalizations are complementary. This result is published in [35].

2. We derive an explicit lower bound for the length of minimal Hamming weight $\tau$-adic expansions. We also give a new proof of the minimality of
the Hamming weight of the $\tau$-NAF. Further, by using the proof of the lower bound and the new proof of the minimality, we classify a minimal length $\tau$-adic expansion ($\tau$-MLF) with minimal Hamming weight on Koblitz curves. The classification shows that the $\tau$-NAF has almost minimal length among all $\tau$-adic expansions of minimal Hamming weight and we can easily convert the $\tau$-NAF into a minimal length $\tau$-adic expansion without changing the Hamming weight. Scalar multiplication on Koblitz curve using $\tau$-MLF method is slightly faster than the original $\tau$-NAF method. Thus $\tau$-MLF can be used for efficient implementation of ECC. This result is published in [36].

3. We construct two batch verification tests for elliptic curve based signature schemes. The first (resp. second) test is based on non-supersingular elliptic curves in characteristic seven (resp. five). The target curves have efficiently computable automorphisms. The new tests are modifications of the previous method (complex exponent test). The new tests are suitable for verifying a limited number of signatures in real-time. This result is published in [34].

6.2 Open Problems

We conclude this thesis by highlighting some open problems.

Hyperelliptic Koblitz curves. As we see in previous sections, $\tau$-NAF on elliptic Koblitz curves has three properties, namely, the existence, uniqueness, and the minimality of the Hamming weight. The existence must be satisfied for concrete cryptographic implementations. The uniqueness and the minimality are not only of intrinsic mathematical interest, but also desirable in some cryptographic applications such as batch verification. On the other hand, Günther, Lange, and Stein in [31], [32] have proposed two generalizations of $\tau$-NAF for a family of hyperelliptic curves

$$C_a : y^2 + xy = x^5 + ax^2 + 1, \quad a \in \mathbb{F}_2$$

(6.1)
over a finite field $\mathbb{F}_2$. The curves are listed in [45, Table 1]. We call the curves as 
hyperelliptic Kobližt curves.

The two generalizations are as follows:

(i) GLS $\tau$-adic expansion: One is the $\tau$-adic expansion with the strategy “at least one of four consecutive coefficients is zero”. The expansion of $\alpha / \mathbb{Z}[\tau]$ is a $\tau$-adic expansion $\alpha = \prod_{i=0}^{\ell-1} c_i \tau^i$ such that $c_{i+3}c_{i+2}c_{i+1}c_i = 0$ for all $i$ ($0 \leq i \leq \ell$ 4) and $c_i / \mathcal{D} = \{0, \mp 1, \mp 2, \mp 3\}$ for all $i$ ($0 \leq i \leq \ell$ 1). We will call the above expansion as **GLS $\tau$-adic expansion.**

(ii) $\tau$-NAF: The other is a sparse $\tau$-adic expansion. The expansion of $\alpha / \mathbb{Z}[\tau]$ is a $\tau$-adic expansion $\alpha = \prod_{i=0}^{\ell-1} c_i \tau^i$ such that $c_{i+1}c_i = 0$ for all $i$ ($0 \leq i \leq \ell$ 2) and $c_i / \mathcal{D} = \{0, \mp 1, \mp 2, \mp (1 + \tau), \mp (1 - \tau), \mp (1 + 2\tau), \mp 2 + \tau\}$ for all $i$ ($0 \leq i \leq \ell$ 1). We will call the above expansion as **$\tau$-NAF.**

To our knowledge, there are no proofs of these existences in the relevant literature (cf. [2], [31], [32], [49], [50]). This raises the question of whether or not the three properties are true for hyperelliptic Kobližt curves.

**Efficiently computable endomorphisms on some other forms of elliptic curves.**

Elliptic curves in Weierstrass form are widely used in ECC. Over the last decade, some other forms of elliptic curves have been proposed to improve the efficiency of group law on elliptic curves.

(i) Edwards form [26]: An elliptic curve in Edwards form over a field $K$ is defined by $E_{Ed,c,d} : x^2 + y^2 = c^2(1+dx^2y^2)$, where $\text{char}(K) = 2$, $c,d \neq K$, and $cd(1-c^4d) = 0$.

(ii) Jacobi quartic form [10]: An elliptic curve in Jacobi quartic form over a field $K$ is defined by $E_{JQ,k} : y^2 = (1-x^2)(1-k^2x^2)$, where $\text{char}(K) = 2,3$, $k \neq K$, and $k = 0, \mp 1$.

(iii) Jacobi intersection form [53]: An elliptic curve in Jacobi intersection form
over a field $K$ is defined by

\[ E_{1, a} : \begin{cases} x^2 + y^2 = 1 \\ ax^2 + z^2 = 1 \end{cases} \quad (6.2a) \]

where $\text{char}(K) = 2$, $a / K$ and $a(1-a) = 0$.

(iv) Huff form [43]: An elliptic curve in Huff form over a field $K$ is defined by

\[ E_{H_{a, b}} : ax(y^2 - 1) = by(x^2 - 1), \quad \text{where } \text{char}(K) = 2, a, b / K^*, \text{ and } a^2 = b^2. \]

However, so far little is known about efficiently computable endomorphisms on such forms of elliptic curves. To find efficiently computable endomorphisms will be a very attractive object in order to construct efficient scalar multiplication algorithms.

**Double base number system and efficiently computable endomorphisms.**

A double base number system (DBNS) [24] is a special way of representing integers as a sum or difference of mixed powers of two different integers. Several numerical evidences indicate that the number of non-zero digits in DBNS is smaller than that of NAF. The use of Frobenius endomorphism on Kobliitz curves in DBNS have been proposed in [3], [25]. However, some conjectures remain to be discussed. The results in Chapter 4 may be useful to explore the relation between $\tau$-DBNS and $\tau$-NAF.

Since efficiently computable endomorphisms have a large potential to be explored, the above open problems are not limited by this list.
List of Related Publications

Peer Reviewed Journal Publications


Refereed Conference Proceedings


Non-Refereed Conference Proceedings and Workshops


Bibliography


BIBLIOGRAPHY

Vol.658 of Lecture Notes in Computer Science (1993), Springer-Verlag, 200–207. 80


[31] C. G"unther, T. Lange, and A. Stein, Speeding up the Arithmetic on Koblitz Curves of Genus Two, Research Report CORR #2000-04, Department of Combinatorics and Optimization, Faculty of Mathematics, University of Waterloo, Canada, 2000. 102, 103


[66] G.W. Reitwiesner, Binary Arithmetic, Advances in Computers, 1 (1960), 231–308. 10, 52


Index

\(\tau\)-adic expansion, 53
\(\tau\)-and-add method, 53
\(F_q\)-Koblitz curve, 17
\(\phi\)-r-NAF, 11, 28
\(\phi\)-GNAF, 11, 18
\(\tau\)-adic minimal length form (\(\tau\)-MLF), 71
\(\tau\)-adic non adjacent form (\(\tau\)-NAF), 2, 10, 16, 53, 80
anomalous binary curve, 9, 52
atomic random subset test, 78
automorphism, 81
batch verification, 2, 74, 77
batch verifier, 77
boolean relation, 77
bucket test, 78
complex exponent test, 74, 78, 79, 82
digital signature scheme, 6, 74
discrete logarithm problem, 1, 51
double-and-add method, 52
EC-ELGamal, 9
EC-Schnorr, 78
ECDH, 9
ECDSA, 9, 72, 78
ECDSA*, 78
elliptic curve, 5
elliptic curve cryptosystems (ECC), 1, 7, 9, 51
elliptic curve discrete logarithm problem, 51
endomorphism, 1, 81
endomorphism ring, 5
endomorphism-and-and-add method, 52
Frobenius endomorphism, 2
Frobenius expansion, 2, 9, 15, 81, 82, 90
Frobenius map, 4, 5, 9, 52, 81
generalized non-adjacent form (GNAF), 10, 12, 65
GHS attack, 18
Hamming weight, 2, 53
Hasse-Weil bound, 87, 92
Horner’s method, 15
hyperelliptic Koblitz curve, 10
integer factoring problem, 1, 51
Koblitz curve, 2, 10, 16, 52
non-adjacent form (NAF), 10
normal basis, 14, 72
number system, 52

point multiplication, 1, 7, 51
polynomial basis, 14, 72
projective space, 4
public key encryption scheme, 6
public key scheme, 1

radix-$r$ non-adjacent form ($r$-NAF), 10,
  12
RSA cryptosystems, 9, 51

scalar multiplication, 1, 7, 9, 51
small exponents test, 78
sparse exponent test, 78
subfield elliptic curve, 2, 9, 13

Weil descent attack, 18
width-$w$ $\tau$-adic non-adjacent form ($w$-
  $\tau$-NAF), 53, 80