九州大学学術情報リポジトリ
Kyushu University Institutional Repository

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：
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

∝ Copyright by Keisuke Hakuta, 2013
All Rights Reserved.

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 Yasuro 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 Furuya, 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 iv

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 τ-NAF on binary Koblitz curves and supersingular Koblitz curves 15

3.3 Proposed Methods (Two classes of ϕ-NAF) . . . 17
3.3.1 ϕ-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ . . . 18
3.3.2 ϕ-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ . . . 28
3.3.3 ϕ-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 τ-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 τ-adic Minimal Length Form and Its Cryptographic Application . . . 69
4.5.1 τ-adic Minimal Length Form . . . 69
4.5.2 Cryptographic Application of τ-MLF . . . 72
4.6 Summary of this chapter . . . 72

5 Batch verification suitable for efficiently verifying a limited number 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

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 ℓ digits ϕ-GNAF representations 25
3.2 Forms of ℓ digits ϕ-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 $\mathbb{F}_{q^m}$-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 42
3.6 Timings of the Finite Field Operations 45
3.7 Timings of Recoding Methods 45
3.8 Timings of Scalar Multiplications on $\mathbb{F}_3$-Koblitz Curve $\mathrm{K}_{(3,1)}$-173 (online precomputation case) 46
3.9 Sample Parameters for $\mathbb{F}_3$-Koblitz Curves of $t = 1$ 47
3.10 Sample Parameters for $\mathbb{F}_3$-Koblitz Curves of $t = \quad 1$ 48

4.1 Conversion of the τ-NAF into the τ-MLF ($\ell < 6$) 70
4.2 Conversion of the τ-NAF into the τ-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 $E_{5,a}/\mathbb{F}_5$ 94
5.4 Sample Parameters for the elliptic curves $E_{7,b}/\mathbb{F}_7$ 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_ω such that $\#S_j \subset 2^{k_{\mathrm{bv}}}$, $\#S_{\mathrm{E}} \subset 2^{k_{\mathrm{bv}}}$ 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 τ-adic expamsions on Koblitz curves. We also give a new proof of the minimality of the Hamming weight of the τ-NAF on Koblitz curves. Furthermore, we classify a minimal length τ-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 $q = p^r$ $(r \sim 1)$, $p = \mathrm{char}(\mathbb{F}_q)$. For any field k, we denote by $p = \mathrm{char}(k)$ the characteristic of the field k. For a field k, the n-dimensional projective space over k is denoted by $\mathbb{P}^n_k$. 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 $x \,/\, \mathbb{C}$, we denote by $\backslash x \backslash$ the absolute value of x. We denote a by $\bar{a}$ for any natural number a

For any non-zero complex number $\psi \,/\, \mathbb{C}$ $\}0\langle$, we denote ψ-adic expansion $\prod_{i=0}^{\ell-1} c_i \psi^i$ with $c_i \,/\, \mathbb{Z}$ by $(c_{\ell-1}, \ldots, c_0)_\psi$, and denote the number of non-zero c_i's by $wt((c_{\ell-1}, \times\!\!\times\!\!\times, c_0)_\psi)$. The symbol '$\pm$ means a non-zero digit of ψ-adic expansions. We call ℓ the length of the ψ-adic expansion. Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, $\phi : E \Leftarrow E, \quad (x, y) \Lsh (x^q, y^q)$ be the qth-power Frobenius map on E. We put $t_m := q^m + 1 \quad \#E(\mathbb{F}_{q^m})$, $t := t_1$, and call t_m the trace of $E(\mathbb{F}_{q^m})$. We can regard ϕ as a complex number which satisfies the characteristic

equation $\phi^2 \quad t\phi + q = 0$. 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. $\mathcal{A}$, $\mathcal{F}$ stand for the computational cost of point addition on $E(\mathbb{F}_{q^m})$, q^{th}-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 τ the Frobenius map on E_a defined by (4.2). Let $\mathcal{D} := \mathcal{D}_2 = \}0, \mp 1\langle$. Remark that for a fixed coefficient $a \,/\, \mathbb{F}_2$ in Equation (4.1), it satisfies that $\mathcal{D} = \}0, \mp\mu\langle$.

For any element α in $\mathbb{Z}[\tau]$, we denote by $\alpha = \prod_{i=0}^{\ell-1} b_i \tau^i$ ($b_i \,/\, \mathcal{D}$) the τ-NAF of α, and by $\alpha = \prod_{i=0}^{\ell'-1} c_i \tau^i$ ($c_i \,/\, \mathcal{D}$) be any τ-adic expansion of α, respectively. The length of τ-NAF of α is denoted by $\ell_{\tau\text{-NAF}}(\alpha)$. We denote by $\ell_{\min}(\alpha)$ the length of τ-adic expansion of minimal length among all τ-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 $c_{\ell'} = c_{\ell'+1} = \cdots = c_{\ell-1} = 0$. Otherwise, put $b_\ell = b_{\ell+1} = \cdots = b_{\ell'-1} = 0$. Furthermore, replace $\max\}\ell, \ell'\langle$ by ℓ if necessary. We put $S_\alpha := \}i \,/\, \}0, 1, \ldots, \ell \quad 1\langle \setminus b_i \neq 0\langle$, and $T_\alpha := \}i \,/\, \}0, 1, \ldots, \ell \quad 1\langle \setminus c_i \neq 0\langle$.

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 $F(x, y, z) = y^2z + (a_1x + a_3z)yz \quad (x^3 + a_2x^2z + a_4xz^2 + a_6z^3)$ 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 $\mathcal{O}_X \Leftarrow \mathcal{O}_X$, $a \not\Leftarrow a^p$ the absolute Frobenius of X.*

Lemma 2.1 ([55, Lemma 3.2.22])

(1) *Let* $f : X \Leftarrow Y$ *be a morphism of schemes over* $\mathbb{F}_p$. *Then* $F_Y \bullet f = f \bullet F_X$.

(2) *For any* $x \,/\, X$, *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 $\}0,1\langle^*$.

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* $\alpha \,/\, \}0,1\langle^*$, *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* $\alpha \,/\, \}0,1\langle^*$ *using the encryption-key* e, *whereas* $D(d, \beta)$ *is the decryption of the ciphertext* β *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* α / $\}0,1\langle^*$, *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 α produced using the signing-key s. Likewise, when $V(v, \alpha, \beta) = 1$, we say that β is a valid signature to α with respect to the verification-key v. This definition asserts that any signature to α produced using the signing-key s is a valid signature to α 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 / $\mathbb{Z}[\phi]$ 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, ϕ is the q-th power Frobenius map on E, d_i / $\}0, \mp 1, \ldots, \mp(q \quad 1)/2\langle$, t is the trace of ϕ, $(q,t) \models (5, \mp 4), (7, \mp 5)$, and ℓ is a non-negative integer. Therefore, scalar multiplication method using ϕ 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 (rNAF) [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 τ-adic NAF (τ-NAF), and Koblitz [47] proposed τ-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 τ-adic NAF on hyperelliptic Koblitz curves. Recently, Blake, Murty, and Xu [11] proposed the radix-τ 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 τ-NAF, that is, two classes of ϕ-adic NAF (ϕ-GNAF and ϕ-rNAF) using the techniques of GNAF and rNAF, 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 $\}0, \mp 1\langle$ and the digit set of the Frobenius expansion is $\}0, \mp 1, \ldots, \mp(q \quad 1)/2\langle$. 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 because of the narrowness of the digit set of NAF. Thus as a natural development, we apply the GNAF and rNAF techniques, which are the generalizations of the ordinary NAF, to apply τ-NAF to elliptic curves in odd characteristic. For the resulting recoding methods, ϕ-GNAF and ϕ-rNAF, if the radix is small (e.g., $3, 5$), the difference between the computational costs for the precomputation tables of ϕ-GNAF and ϕ-rNAF is relatively small (a few elliptic additions). But, if the radix is significantly large, the computational cost for the precomputation table of ϕ-rNAF is quite large compared to that for ϕ-GNAF. However the non-zero density for ϕ-rNAF is significantly smaller than that for ϕ-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 ϕ-NAF, we concentrate on investigating only ϕ-GNAF and ϕ-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, rNAF, and τ-adic NAF on binary Koblitz curves. Section 3.3 shows how to generalize τ-NAF on binary Koblitz curves to two classes of ϕ-adic NAF on a family of subfield elliptic curves and proves some properties of ϕ-GNAF and

ϕ-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, α be relatively prime positive integers. We denote $D_{r,\alpha}$ a set defined as follows.

$$D_{r,\alpha} := \begin{cases} \{0, \mp 1, \ldots, \mp \alpha\} & \text{if } \alpha < r, \\ \{0, \mp 1, \ldots, \mp \alpha\} \setminus \{\mp r, \mp 2r, \ldots, \mp \lfloor \alpha / r \rceil r\} & \text{otherwise.} \end{cases}$$

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 $d = \sum_{i=0}^{\ell-1} e_i r^i$ where $e_i \not\equiv D_{r,r-1}, e_{\ell-1} \not\equiv 0$ 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 $|e_{i+1} + e_i| < r$, (3) if $e_{i+1}e_i < 0$, then $|e_{i+1}| > |e_i|$. The length of the GNAF is ℓ. For $a, b \not\equiv D_{r,r-1}$, if a, b satisfy one of the followings : (1) $ab = 0$, (2) if $ab > 0$, then $|a + b| < r$, (3) if $ab < 0$, then $|a| > |b|$, 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 $d = \sum_{i=0}^{\ell-1} e_i r^i$ where $e_i \not\equiv D_{r,(r^2-1)/2}$, $e_{\ell-1} \not\equiv 0$ and for each i, it satisfies $e_{i+1}e_i = 0$ where we define $e_\ell = 0$. The length of the rNAF is ℓ. For $a, b \not\equiv D_{r,(r^2-1)/2}$, 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 $(r \quad 1)/(r+1)$ (resp. $(r \quad 1)/(2r \quad 1)$).*

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, $q = p^r$ 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*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

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 ϕ the qth-power Frobenius map on E.

$$\phi : E \Leftarrow E, \quad (x, y) \mathrel{\substack{\leftarrow\\\leftarrow}} (x^q, y^q),$$

and set $t_m := q^m + 1 \quad \#E(\mathbb{F}_{q^m})$, $t := t_1$, *where* $E(\mathbb{F}_{q^m})$ *means the set of* $\mathbb{F}_{q^m}$*-rational points on* E. *We can regard* ϕ *as a complex number which satisfies the following characteristic equation :* $\phi^2 \quad t\phi + q = 0$.

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. 36]. The computational cost of Algorithm 1 is approximately

$$\left\lceil \right\rceil \mathcal{D}_{\mathcal{P}} + (2^{w-2} \quad 1)\mathcal{A}_{\mathcal{P}} \left\{ + \left\lceil \right\rceil \frac{\ell}{w+1} \mathcal{A}_{\mathcal{S}} + \ell \mathcal{D}_{\mathcal{S}} \right\{,$$

where $\mathcal{A}_{\mathcal{P}}, \mathcal{D}_{\mathcal{P}}$ (resp. $\mathcal{A}_{\mathcal{S}}, \mathcal{D}_{\mathcal{S}}$) stand for the computational cost of the point addition, point doubling in the precomputational step (resp. scalar multiplication step), respectively.

Algorithm 1 Scalar multiplication (window w-NAF) [37]

Input: $d = (d_{\ell-1}, \ldots, d_0)_{w\text{-NAF}}$ / $\mathbb{Z}_{>0}$,
d_i / $D_{w\text{-NAF}}$
$= \}0, \mp 1, \mp 3, \ldots, \mp(2^{w-1} \quad 1)\langle$,
$\ell \geq \left\lceil \log_2(d) \right\rceil + 1$,
P / $E(\mathbb{F}_{q^m})$
Output: dP

1: Compute $P_i = iP$
for i / $\}1, 3, 5, \ldots, 2^{w-1} \quad 1\langle$
2: $Q \to \mathcal{O}$
3: **for** i from $\ell \quad 1$ downto 0 **do**
4: $\quad Q \to 2Q$
5: $\quad Q \to Q + P_{d_i}$
6: **end for**
7: **return** Q

Algorithm 2 Scalar multiplication (Frobenius expansion)

Input: $d = (d_{\ell-1}, \ldots, d_0)_{\phi}$ / $\mathbb{Z}_{>0}$,
d_i / D_{ϕ}
$= \}0, \mp 1, \mp 2, \ldots, \mp\alpha\langle$,
$\ell \geq \left\lceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}(d)}} \right\rceil + 4$,
P / $E(\mathbb{F}_{q^m})$
Output: dP

1: Compute $P_i = iP$
for i / $\}1, 2, \ldots, \alpha\langle$
2: $Q \to \mathcal{O}$
3: **for** i from $\ell \quad 1$ downto 0 **do**
4: $\quad Q \to \phi(Q)$
5: $\quad Q \to Q + P_{d_i}$
6: **end for**
7: **return** Q

It is well-known that the cost of the Frobenius map ϕ 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 ϕ-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 $\mathbb{F}_q$, ϕ be its qth-power Frobenius map of E, t is the trace of ϕ. We assume that $(q,t) \vDash (5,\mp4), (7,\mp5)$. Let $d \;/\; \mathbb{Z}[\phi]$, then we can write $d = \prod_{i=0}^{\ell-1} c_i\phi^i$, where $c_i \;/\; \}0,\mp1,\ldots,\mp(q \quad 1)/2\langle$ and $\ell \geq \left\rceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\sqrt{}+4$. We denote $\ell_{\phi\text{-}EXP}(d)$ the length of ϕ-adic expansion of $d \;/\; \mathbb{Z}[\phi]$.*

From Theorem 3.1, we can compute dP efficiently using a precomputation table $\}iP \setminus i = 1,2,\ldots,(q \quad 1)/2\langle$ and Horner's method:

$$dP = \Big)_{i=0}^{\tau-1} c_i\phi^i \Big[P = \phi \;\times\!\times\!\times \phi(c_{\ell-1}\phi(P) + c_{\ell-2}P) + \times\!\times\!\times + c_1P| + c_0P.$$

We describe a scalar multiplication algorithm using Horner's method based on Frobenius expansion with digit set $D_\phi = \}0,\mp1,\mp2,\ldots,\mp\alpha\langle$. The computational cost of Horner's method using Smart's Frobenius expansion method (Algorithm 2 with $\alpha = (q \quad 1)/2$, digit set $D_\phi = D_{q,(q-1)/2}$) is approximately

$$\Big]\frac{q \quad 1}{q}\ell\mathcal{A}_{\mathcal{S}} + \ell\mathcal{F}_{\mathcal{S}}\Big\{(q=3), \quad \Big]\mathcal{D}_{\mathcal{P}} + \frac{q \quad 5}{2}\mathcal{A}_{\mathcal{P}}\Big\{+\Big]\frac{q \quad 1}{q}\ell\mathcal{A}_{\mathcal{S}} + \ell\mathcal{F}_{\mathcal{S}}\Big\{(q\sim 5),$$

where $\mathcal{F}_{\mathcal{S}}$ 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 $Q \to Q \quad P_{-d_i}(= Q + P_{d_i})$ and if $d_i = 0$, we do not need to compute $Q \to Q + P_{d_i}$. This is the reason that smaller non-zero densities in the representations of the scalars improve the efficiency.

3.2.3 τ-NAF on binary Koblitz curves and supersingular Koblitz curves

Next, we review τ-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 τ the Frobenius map on E_a:

$$\tau : E_a \Leftarrow E_a, \quad (x, y) \not\Leftarrow (x^2, y^2).$$

We can regard τ as a complex number which satisfies the following characteristic equation: $\tau^2 \quad \mu\tau + 2 = 0$, $\mu = (\quad 1)^{1-a}$. A τ-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 $e_i \;/\; D_{2,1}, e_{\ell-1} \eqsim 0$ and no two consecutive digits e_i are nonzero. The length of the τ-NAF is ℓ.

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 $E_a : y^2 = x^3 \quad x$ $(\quad 1)^a/\mathbb{F}_3$, 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 ϕ-NAF on a family of subfield elliptic curves. In the following, we call these two classes of ϕ-NAF ϕ-GNAF and ϕ-rNAF, respectively.

3.3 Proposed Methods (Two classes of ϕ-NAF)

In this section, we investigate how to expand two classes of ϕ-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 qth-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_a. Since the trace of the Frobenius map τ of E_a is $\mu = (\ \ 1)^{(1-a)} = \mp 1$, $\mathbb{F}_q$-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

$$\phi^2 \bigcirc \phi + q = 0 \quad (1, \bigcirc 1, q)_\phi = 0^{|},$$

and by easy calculation, we have

$$\mp\phi^3 + (q \quad 1)\phi^2 + q^2 = 0 \quad (\mp 1, q \quad 1, 0, q^2)_\phi = 0^{|}.$$

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 $P \,/\, E(\mathbb{F}_{q^m})$, $(\phi^m \quad 1)P = \mathcal{O}$ is satisfied. Hence $dP = (d \bmod (\phi^m \quad 1))\, P$ for any integer scalar d. From [70], there exist $Q, d' \,/\, \mathbb{Z}[\phi]$ such that $d = Q(\phi^m \quad 1) + d'$ with $d' = 0$ or $\Psi(d') < \lambda\Psi(\phi^m \quad 1)$, where Ψ 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 ϕ-GNAF and ϕ-rNAF. For details, refer to [2].

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

$$\phi \setminus \alpha \quad \Rightarrow\infty \quad q \setminus a.$$

In particular, for rational integer $a \,/\, \mathbb{Z}$, we have $\phi \setminus a \;\Rightarrow\infty\; q \setminus a$.

3.3.1 ϕ-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 ϕ-GNAF and prove some properties of this. In this subsection, we assume that $t = 1$. We begin with the definition of ϕ-GNAF on every subfield elliptic curves, and give an algorithm which computes ϕ-GNAF for a given $d \,/\, \mathbb{Z}[\phi]$, where $q \sim 7$.

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

(1) $e_{i+1}e_i = 0$,

(2) *if* $e_{i+1}e_i > 0$, *then* $\backslash e_{i+1} + e_i \backslash < q$,

(3) *if* $e_{i+1}e_i < 0$, *then* $\backslash e_{i+1} \backslash > \backslash e_i \backslash$.

Let $a, b \ / \ D_{q,q-1}$. If a pair $(a,b)_\phi$ satisfies one of the followings: (1) $ab = 0$, (2) *if* $ab > 0$, *then* $\backslash a + b \backslash < q$, (3) *if* $ab < 0$, *then* $\backslash a \backslash > \backslash b \backslash$, *then we call the pair* $(a,b)_\phi$ *a* ϕ*-admissible pair.* [*1)] *Otherwise, we call the pair* $(a,b)_\phi$ *a* ϕ*-non-admissible pair. We denote* $\ell_{\phi\text{-GNAF}}(d)$ *the length of* ϕ*-GNAF 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 ϕ-GNAF is equivalent to the condition that any adjacent digits $(e_{i+1}, e_i)_\phi$ are ϕ-admissible.

Example 3.1 (Example of Frobenius expansion and ϕ-GNAF) *Let* $q = 7$, $d = 314 \ / \ \mathbb{Z}[\phi]$. *By easy calculation, we have the Frobenius expansion* $d = (\bar{1}, 3, \bar{3}, 2, 3, 3, \bar{1})_\phi$. *Using Algorithm 3, we compute the* ϕ*-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 $(3, \bar{1})_\phi$. *This pair is a* ϕ*-admissible pair. We set* $e_0 = \bar{1}$.

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

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

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

In the fifth step, we scan $(3, \bar{3})_\phi$. *This pair is a* ϕ*-non-admissible pair and* $b_4 = \bar{3} < 0$. *We set* $b_6 = \bar{1} + 1 = 0$, $b_5 = 3 \quad 1 = 2$, $e_4 = \bar{3} + 7 = 4$,

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

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

[*1)] In the case that ϕ is a positive integer, a similar notation can be found in [54].

Algorithm 3 ϕ-GNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ ($q \sim 7$)

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

Output: ϕ-GNAF of d

1: $d' \to d \bmod (\phi^m \quad 1)$
2: $\ell \to \left\lceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\sqrt{} + 4 \right.$
3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of d'
4: $b_0 \to c_0$, $b_1 \to c_1, b_\ell \to 0, b_{\ell+1} \to 0$
5: $i \to 0$
6: **while** $i \geq \ell$ **do**
7: **if** (b_{i+1}, b_i) : ϕ-admissible pair **then**
8: $b_{i+2} \to c_{i+2}, e_i \to b_i$
9: **else if** $b_i > 0$ **then**
10: $b_{i+2} \to c_{i+2} \quad 1, b_{i+1} \to b_{i+1} + 1, e_i \to b_i \quad q$
11: **else**
12: $b_{i+2} \to c_{i+2} + 1, b_{i+1} \to b_{i+1} \quad 1, e_i \to b_i + q$
13: **end if**
14: $i \to i + 1$
15: **end while**
16: **return** $(e_{\ell+1}, e_\ell, \ldots, e_1, e_0)_\phi$

The following lemma and theorem show the correctness of Algorithm 3, thus the existence of ϕ-GNAF. From the lemma, for any given Frobenius expansion, we can have a sequence with digits in $D_{q,q-1}$ such that any adjacent digits are ϕ-admissible. Moreover, the theorem below gives the finiteness of the sequence (hence ϕ-GNAF) and evaluates the upper bound of the length of ϕ-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 ϕ-admissible pair and $(b, b')_\phi$ is a ϕ-non-admissible pair. If we convert*

$$(b, b', e)_\phi \not\to \begin{cases} (\bar{1}, c, c', e)_\phi := (\bar{1}, b+1, b' \quad q, e)_\phi & \text{if } b' > 0, \\ (1, c, c', e)_\phi := (1, b \quad 1, b'+q, e)_\phi & \text{otherwise,} \end{cases}$$

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

Theorem 3.3 (Maximum length of ϕ-GNAF) *Let* $d \,/\, \mathbb{Z}[\phi]$ *and* $\ell = \ell_{\phi\text{-EXP}}(d)$ *be the length of Frobenius expansion of* d*. Then* d *has a* ϕ*-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 $d = (c, b, b', e_{\ell-4}, \ldots, e_1, e_0)_\phi$, where $e_i \,/\, D_{q,q-1}$, $c \,/\, D_{q,(q-1)/2}, b \,/\, D_{q,(q+1)/2}, b' \,/\, D_{q,(q+3)/2}$, and $(b', e_{\ell-4}, \ldots, e_1, e_0)_\phi$ is a ϕ-GNAF. Now we scan the two digits $(b, b')_\phi$. If the pairs $(c, b)_\phi$ and $(b, b')_\phi$ are ϕ-admissible, there is nothing to do and $\ell' = \ell$. So we can assume that $(b, b')_\phi$ is ϕ-non-admissible. Then we convert

$$(c, b, b')_\phi \,\not\vdash\, (\tilde{c}, c', e_{\ell-3})_\phi := \left\} \begin{array}{ll} (c \quad 1, b+1, b' \quad q)_\phi & \text{if } b' > 0, \\ (c+1, b \quad 1, b'+q)_\phi & \text{otherwise.} \end{array} \right.$$

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

$$(\tilde{c}, c')_\phi \,\not\vdash\, (a, a', e_{\ell-2})_\phi := \left\} \begin{array}{ll} (\bar{1}, \tilde{c}+1, c' \quad q)_\phi & \text{if } c' > 0, \\ (1, \tilde{c} \quad 1, c'+q)_\phi & \text{otherwise,} \end{array} \right.$$

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

$$(a, a')_\phi \,\not\vdash\, \left\} \begin{array}{ll} (\bar{1}, 0, a' \quad q)_\phi & a' > 0, \\ (1, 0, a'+q)_\phi & a' < 0. \end{array} \right.$$

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

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 $(b_{i+1}, b_i)_\phi \,\not\vdash\, (\ \ b_i/q, b_{i+1} + b_i/q, 0)_\phi$. It is easy to show that if b_i is a non-zero multiple of q, then it satisfies that $b_i = \mp q$. Thus for all i, we always have $\backslash b_{i+1} \backslash \geq (q+1)/2$, $\backslash b_i \backslash \geq (q+3)/2$, 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.

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

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

Output: ϕ-GNAF of d

1: $d' \to d \bmod (\phi^m \quad 1)$
2: $\ell \to \left\lceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\sqrt{} + 4 \right.$
3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of d'
4: $b_0 \to c_0$, $b_1 \to c_1, b_\ell \to 0, b_{\ell+1} \to 0$
5: $i \to 0$
6: **while** $i \geq \ell$ **do**
7: **if** $b_i \bmod q = 0$ **then**
8: $b_{i+1} \to b_{i+1} \quad b_i/q, b_{i+2} \to c_{i+2} + b_i/q, e_i \to 0$
9: **else if** (b_{i+1}, b_i) : ϕ-admissible pair **then**
10: $b_{i+2} \to c_{i+2}, e_i \to b_i$
11: **else if** $b_i > 0$ **then**
12: $b_{i+2} \to c_{i+2} \quad 1, b_{i+1} \to b_{i+1} + 1, e_i \to b_i \quad q$
13: **else**
14: $b_{i+2} \to c_{i+2} + 1, b_{i+1} \to b_{i+1} \quad 1, e_i \to b_i + q$
15: **end if**
16: $i \to i + 1$
17: **end while**
18: **return** $(e_{\ell+1}, e_\ell, \ldots, e_1, e_0)_\phi$

Let ϕ-GNAF$_\ell$ be the set of ϕ-GNAF of length ℓ. We put $A_\ell = \#\phi$-GNAF$_\ell$, $S_\ell = \prod_{d \in \phi\text{-GNAF}_\ell}(\ell \quad w(d))$, and $C_\ell = \#\,\}d$ / ϕ-GNAF$_\ell$ $\setminus w(d) = \ell\langle$, 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

$$1 \quad \lim_{\ell \to \infty} S_\ell/(\ell A_\ell).$$

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

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

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

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

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

where ℓ is the maximum of length of the two representations. Then from Lemma 3.1, we have $q \setminus (e_0 \quad e'_0)$, and by the assumption, $\backslash e_0 \backslash, \backslash e'_0 \backslash \geq q \quad 1$, hence the inequality $\backslash e_0 \quad e'_0 \backslash \geq \backslash e_0 \backslash + \backslash e'_0 \backslash \geq 2q \quad 2$ is satisfied. Therefore we have $e_0 \quad e'_0 = \mp q$. In the following, we will give a proof for the case $e_0 \quad e'_0 = q$. For another case, we can prove similarly, thus we omit it.

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

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

Hence, from Lemma 3.1, we also have $q \setminus (e_1 \quad e'_1 + 1)$. It follows that e'_1 / $\}e_1 + 1 \quad q, e_1 + 1, e_1 + 1 + q\langle$.

We shall see that in each case, there is a contradiction. (For details, refer to [54, p.177]).

Case 1. Suppose that $e'_1 = e_1 + 1 \quad q$. We have $\quad (q \quad 1) \geq e'_1 \geq 0$ and $0 \geq e_1 \geq (q \quad 1)$. We assume that $\quad (q \quad 1) \geq e'_1 \geq \quad 1$. Then from $e'_0 e'_1 > 0$, we have $\quad e'_0 \quad e'_1 \geq q \quad 1$. By substituting $e_0 \quad q$ for e'_0 and $e_1 \quad (q \quad 1)$ for e'_1, we obtain $e_0 + e_1 \sim q$. On the other hand, since $e_0 > 0$ and $e_1 \sim 0$, we have $e_0 + e_1 \geq q \quad 1$. This is a contradiction. If $e'_1 = 0$, then $e_1 = q \quad 1$. 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 $\quad (q \quad 2) \geq e'_1 \geq q \quad 1$ and $\quad (q \quad 1) \geq e_1 \geq (q \quad 2)$. We assume that $2 \geq e'_1 \geq q \quad 1$. Then $1 \geq e_1 \geq (q \quad 2)$. Since $e'_0 e'_1 < 0$, we have $e'_0 + e'_1 > 0$. By substituting $e_0 \quad q$ for e'_0 and $e_1 + 1$ for e'_1, we obtain $e'_0 + e'_1 = (e_0 \quad q) + e_1 + 1 > 0$. Thus $e_0 + e_1 \sim q$. On the other hand, by $e_0 > 0$, $e_1 > 0$, we have $e_0 + e_1 \geq q \quad 1$. 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 $e'_1 = 1 > \quad e'_0 > 0$. However there does not exist e'_0 which satisfies the above inequality. This is a contradiction. Suppose that $e'_1 = 0$. In this case, $e_1 = \quad 1$. By a similar argument as in the case of $e'_1 = 1$, one can obtain that $\quad e'_1 = 1 > e'_0 > 0$. However there does not exist e'_0 which satisfies the above inequality. This is a contradiction. Finally, we assume that $\quad (q \quad 2) \geq e'_1 \geq \quad 1$. Then $\quad (q \quad 1) \geq e_1 \geq \quad 2$. Since $e'_0 e'_1 > 0$, we have $\quad e'_0 \quad e'_1 \geq q \quad 1$. By substituting $e_0 \quad q$ for e'_0 and $e_1 + 1$ for e'_1, we obtain $\quad e'_0 \quad e'_1 = \quad (e_0 \quad q) \quad (e_1 + 1) \geq q \quad 1$. 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 $2 \geq e'_1 \geq q \quad 1$ and $\quad (q \quad 1) \geq e_1 \geq \quad 2$. Since $e'_0 e'_1 < 0$, we have $e'_0 + e'_1 > 0$. By substituting $e_0 \quad q$ for e'_0 and $e_1 + 1 + q$ for e'_1, we obtain $e_0 \quad q + e_1 + 1 + q > 0$. 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 $C_{\ell+1} = (q \quad 2)C_\ell$, $C_1 = 2(q \quad 1)$. We fix an element $d = (e_{\ell-1}, \ldots, e_0)_\phi$ of $\mathbb{Z}[\phi]$ which is ϕ-GNAF such that each $e_i \eqsim 0$. For the fixed d, we count the number of e which satisfy $1 \geq \backslash e \backslash \geq q \quad 1$, and $(e_{\ell-1}, \ldots, e_0, e)_\phi$ is ϕ-GNAF. We can assume that $e_0 > 0$. If $e_0 e > 0$, e satisfy $1 \geq e \geq q \quad 1 \quad e_0$ and otherwise $1 \geq e < e_0$. Thus we have that the number of e is $q \quad 2$, hence the recurrence equation for C_ℓ. It is trivial that $C_1 = 2(q \quad 1)$.

Next, we show that A_ℓ and S_ℓ satisfy the recurrence equations $A_1 = 2(q \quad 1)$, $A_2 = 2(q \quad 1)^2$, $A_{\ell+2} \quad (q \quad 1)A_{\ell+1} \quad qA_\ell = 0$ $(\ell \sim 1)$, and $S_1 = 0$, $S_2 = 2(q \quad 1)$,

$$S_{\ell+2} \quad (q \quad 1)S_{\ell+1} \quad qS_\ell = 2(q \quad 1)\Big\rfloor 1 + \frac{2(q \quad 1)}{(q+1)}\Big\rfloor \frac{q(q^\ell \quad 1)}{q \quad 1} \quad \frac{(\quad 1)^\ell \quad 1}{2}\Big\lfloor\Big\lfloor .$$

Table 3.1: Forms of ℓ digits ϕ-GNAF representations

forms	# of representations	# of 0 digits
$(\pm 0 \pm \quad \times\times\times \quad)$	$C_1A_{\ell-2}$	$C_1A_{\ell-2} + C_1S_{\ell-2}$
$(\pm 0 0 \pm \quad \times\times\times \quad)$	$C_1A_{\ell-3}$	$2C_1A_{\ell-3} + C_1S_{\ell-3}$
$(\pm 0 0 0 \pm \times\times\times \quad)$	$C_1A_{\ell-4}$	$3C_1A_{\ell-4} + C_1S_{\ell-4}$
$\vdots$	$\vdots$	$\vdots$
$(\pm 0 0 0 0 \quad \times\times 0 \pm)$	C_1A_2	$(\ell \quad 3)C_1A_2 + C_1S_2$
$(\pm 0 0 0 0 \quad \times\times 0 0 \pm)$	C_1A_1	$(\ell \quad 2)C_1A_1 + C_1S_1$
$(\pm 0 0 0 0 \quad \times\times 0 0 0)$	C_1	$(\ell \quad 1)C_1$
$(\pm\pm 0 \pm \quad \times\times\times \quad)$	$C_2A_{\ell-3}$	$C_2A_{\ell-3} + C_2S_{\ell-3}$
$(\pm\pm 0 0 \pm \quad \times\times\times \quad)$	$C_2A_{\ell-4}$	$2C_2A_{\ell-4} + C_2S_{\ell-4}$
$(\pm\pm 0 0 0 \pm \times\times\times \quad)$	$C_2A_{\ell-5}$	$3C_2A_{\ell-5} + C_2S_{\ell-5}$
$\vdots$	$\vdots$	$\vdots$
$(\pm\pm 0 0 0 \quad \times\times 0 \pm)$	C_2A_2	$(\ell \quad 4)C_2A_2 + C_2S_2$
$(\pm\pm 0 0 0 \quad \times\times 0 0 \pm)$	C_2A_1	$(\ell \quad 3)C_2A_1 + C_2S_1$
$(\pm\pm 0 0 0 \quad \times\times 0 0 0)$	C_2	$(\ell \quad 2)C_2$
$\vdots$	$\vdots$	$\vdots$
$(\pm\pm\pm \quad \times\times\times \pm 0 \pm)$	$C_{\ell-2}A_1$	$C_{\ell-2}A_1 + C_{\ell-2}S_1$
$(\pm\pm\pm \quad \times\times\times \pm 0 0)$	$C_{\ell-2}$	$2C_{\ell-2}$
$(\pm\pm\pm \quad \times\times\times \pm \pm 0)$	$C_{\ell-1}$	$C_{\ell-1}$
$(\pm\pm\pm \quad \times\times\times \pm \pm \pm)$	C_ℓ	0

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 ℓ digits ϕ-GNAF, the central column stands for the number of each ϕ-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

$$A_\ell = \left(\sum_{j=1}^{\ell-2} C_j \right) \sum_{i=1}^{\ell \simeq 1-j} A_i + 1 \Bigg[+ C_{\ell-1} + C_\ell,$$
$$S_\ell = \left(\sum_{j=1}^{\ell-2} C_j \right) \sum_{i=1}^{\ell \simeq 1-j} S_i + (\ell \quad i \quad j) A_i \Bigg[+ \sum_{i=1}^{\ell-1} (\ell \quad i) C_i.$$

From these equations, we have the desired recurrence equations, and by elementary calculations, we have $A_\ell = 2(q \quad 1) \ q^\ell \quad (\quad 1)^{\ell|} / (q+1)$, and

$$S_\ell = \frac{2(q \quad 1)}{(q+1)^2} \Big) (2q^\ell \quad (q \quad 1)(\quad 1)^\ell)\ell \quad \frac{3q \quad 1}{q+1}(q^\ell \quad (\quad 1)^\ell) \Big\rfloor .$$

From the equations of A_ℓ, S_ℓ, the average number of non-zero digits for ℓ digits numbers in $\mathbb{Z}[\phi]$ is equal to $\ell \quad S_\ell / A_\ell$. We put

$$f(\ell) := \Big) 1 \quad \frac{S_\ell}{\ell A_\ell} \Big\lfloor \quad \Big) \frac{q \quad 1}{q+1} + \frac{3q \quad 1}{((q+1)^2 \ell)} \Big\rfloor .$$

Then

$$f(\ell) = \frac{q \quad 3}{q+1} \times \Big) \frac{(\quad 1)^\ell}{q^\ell \quad (\quad 1)^\ell} \Big\rfloor \ell.$$

Let us take $g(\ell) := q^{-\ell}$. Since it satisfies that

$$\frac{f(\ell)}{g(\ell)} = \frac{q \quad 3}{q+1} \times \frac{(\quad 1)^\ell}{1 \quad (\quad q)^{-\ell}},$$

we obtain

$$\left(\frac{f(\ell)}{g(\ell)} \right(= \frac{q \quad 3}{q+1} \times \frac{1}{1 \quad (\quad q)^{-\ell}} \Leftarrow \frac{q \quad 3}{q+1} \quad (\ell \Leftarrow \in).$$

Hence $f(\ell) = O(g(\ell))$. Thus $1 \quad \lim_{\ell \to \infty} S_\ell / (\ell A_\ell) = 1 \quad 2/(q+1) = (q \quad 1)/(q+1)$. 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 $\alpha = \quad \phi \quad (q \quad 1) \ / \ \mathbb{Z}[\phi]$. An Frobenius expansion of α with digit set $D_{q,q-1}$ is $(\quad 1, 1 \quad q)_\phi$ and its Hamming weight is 2. But ϕ-GNAF of α is $(1, 0, q \quad 2, 1)_\phi$ and its Hamming weight is 3. Therefore ϕ-GNAF does not have minimal Hamming weight among all Frobenius expansion with digit set $D_{q,q-1}$. $\square$

Algorithm 5 computes the ϕ-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 ϕ-GNAF compared to Algorithm 3 and Algorithm 4. From Theorem 3.3 (especially the finiteness of the length of ϕ-GNAF), it is easy to show the correctness of Algorithm 5.

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

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

Output: ϕ-GNAF of d

1: $d' := d_0 + d_1\phi \to d \bmod (\phi^n \quad 1)(d_0, d_1 \ / \ \mathbb{Z})$
2: $\ell \to \left\lceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\sqrt{} + 4 \right.$
3: Compute $Q, b \ / \ \mathbb{Z}$ such that $d_0 = Qq + b$ $(b \ / \ D_{q,(q-1)/2})$ (see [70])
4: $d_0 \to Q + d_1,\ d_1 \to \quad Q$
5: $i \to 1$
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: $d_0 \to Q + d_1,\ d_1 \to \quad Q$
9: **if** (a, b) : ϕ-admissible pair **then**
10: $e_{i-1} \to b, b \to a$
11: **else if** $b > 0$ **then**
12: $e_{i-1} \to b \quad q, b \to a + 1, d_0 \to d_0 \quad 1$
13: **else**
14: $e_{i-1} \to b + q, b \to a \quad 1, d_0 \to d_0 + q$
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 $1 \geq i \geq q \quad 2$, the computational cost of Horner's method using ϕ-GNAF (Algorithm 2 based on ϕ-GNAF with $\alpha = q \quad 1$, digit set $D_\phi = D_{q,q-1}$) is approximately

$$\left\lceil \mathcal{D}_\mathcal{P} + \frac{q \quad 1}{q+1}\ell\mathcal{A}_\mathcal{S} + \ell\mathcal{F}_\mathcal{S} \right\{ (q = 3), \quad \left\lceil \mathcal{D}_\mathcal{P} + (q \quad 3)\mathcal{A}_\mathcal{P} + \frac{q \quad 1}{q+1}\ell\mathcal{A}_\mathcal{S} + \ell\mathcal{F}_\mathcal{S} \right\{ (q \sim 5).$$

3.3.2 ϕ-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 ϕ-adic rNAF and prove some properties of this. As with the previous subsection, we assume that $t = 1$ and we begin with the definition of ϕ-rNAF on every subfield elliptic curves, and give an algorithm which computes the ϕ-rNAF for a given $d \ / \ \mathbb{Z}[\phi]$, where $q \sim 7$.

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

Example 3.2 (Example of Frobenius expansion and ϕ-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 ϕ-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 ϕ-adjacent pair and $\backslash 3 * \ 7 \quad 1 \backslash = 20 < 24$. We set $b_3 = 2, b_2 = 3 + 3 = 6, e_1 = 0, e_0 = 3 * \ 7 \quad 1 = 20$.

In the second step, we scan $(2, 6)_\phi$. This pair is also a ϕ-adjacent pair and $\backslash 2 * \ 7 + 6 \backslash = 20 < 24$. We set $b_5 = 3, b_4 = \quad 3 + 2 = \quad 1, e_3 = 0, e_2 = 2 * \ 7 + 6 = 20$.

In the third step, we scan $(3, \bar{1})_\phi$. This pair is also a ϕ-adjacent pair and $\backslash 3 * \ 7 \quad 1 \backslash = 20 < 24$. We set $b_7 = 0, b_6 = \quad 1 + 3 = 2, e_5 = 0, e_4 = 3 * \ 7 \quad 1 = 20$.

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

Hence, we obtain the ϕ-rNAF of $d = (2, 0, 20, 0, 20, 0, 20)_\phi$ (we omit the leading 0's). We can see that in the above example the Hamming weight of the ϕ-rNAF of d is smaller than that of the Frobenius expansion of d. Moreover, the Hamming

weight of the ϕ-rNAF of d is also smaller than that of the ϕ-GNAF of d (See Example 3.1).

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

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

Output: ϕ-rNAF of d

1: $d' \to d \bmod (\phi^m \quad 1)$
2: $\ell \to \left\lceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\sqrt{} + 4 \right.$
3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of d'
4: $b_0 \to c_0,\ b_1 \to c_1, b_\ell \to 0, b_{\ell+1} \to 0, b_{\ell+2} \to 0, b_{\ell+3} \to 0$
5: $i \to 0$
6: **while** $i \geq \ell + 2$ **do**
7: **if** $b_i = 0$ **then**
8: $b_{i+2} \to c_{i+2}, e_i \to 0, i \to i + 1$
9: **else if** $\backslash b_{i+1}q + b_i \backslash \geq (q^2 \quad 1)/2$ **then**
10: $b_{i+3} \to c_{i+3}, b_{i+2} \to c_{i+2} + b_{i+1}, e_{i+1} \to 0, e_i \to b_{i+1}q + b_i, i \to i + 2$
11: **else if** $b_{i+1}q + b_i < \quad (q^2 \quad 1)/2$ **then**
12: $b_{i+3} \to c_{i+3} + 1, b_{i+2} \to c_{i+2} + b_{i+1} + (q \quad 1), e_{i+1} \to 0, e_i \to b_{i+1}q + b_i + q^2, i \to i + 2$
13: **else**
14: $b_{i+3} \to c_{i+3} \quad 1, b_{i+2} \to c_{i+2} + b_{i+1} \quad (q \quad 1), e_{i+1} \to 0, e_i \to b_{i+1}q + b_i \quad q^2, i \to i + 2$
15: **end if**
16: **end while**
17: **return** $(e_{\ell+3}, e_{\ell+2}, \ldots, e_1, e_0)_\phi$

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:

$$(b_{i+1}, b_i)_\phi \not\leftarrow \begin{cases} (\bar{b}'_i, b_{i+1} + b'_i, 0)_\phi & \text{if } b_i = b'_i q \text{ for some } b'_i \ / \ \mathbb{Z}\sqrt{}\}0\langle, \\ (\bar{b}'_{i+1}, b'_{i+1}, 0, b_i)_\phi & \text{if } b_{i+1} = b'_{i+1} q \text{ for some } b'_{i+1} \ / \ \mathbb{Z}\sqrt{}\}0\langle. \end{cases}$$

In other words, there is no possibility that b_i or b_{i+1} is a non-zero multiple of q when we scan $(b_{i+1}, b_i)_\phi$. 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 $(c, c', c'', b, b')_\phi$ from right-to-left according to the following rule and we denote the result of the conversion $(a, a', e, e', e'', e''')_\phi$.*
The rule: We assume that we scan consecutive two digits $(a, b)_\phi$, then
Rule 1. *If $a \models 0, b \models 0$, then convert $(a, b)_\phi$*

$$(a,b)_\phi \Vdash \begin{cases} (a, 0, aq+b)_\phi & \text{if } |aq+b| \geq (q^2 \quad 1)/2, \\ (1, a+(q \quad 1), 0, (aq+b)+q^2)_\phi & \text{else if } aq+b < \quad (q^2 \quad 1)/2, \\ (\bar{1}, a \quad (q \quad 1), 0, (aq+b) \quad q^2)_\phi & \text{otherwise.} \end{cases}$$

Rule 2. *If $a \models 0, b = 0$, 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 $(e, e', e'', e''')_\phi$ is a ϕ-rNAF, $a \ / \ D_{q,(q+1)/2}, a' \ / \ D_{q,2q-1}$. In particular, a and a' are not divisible by q.

Proof We apply the rule in the statement of Lemma 3.3 to $(c, c', c'', b, b')_\phi$. We denote the conversion of i-th digit and $(i+1)$-st digit by $\flat$ and the conversion of $(i+2)$-nd digit and $(i+3)$-rd digit by $\sharp$. We assume a' is a non-zero multiple of q. From $|a'| \geq 2q \quad 2$, we have $a' = q$ or $a' = \quad q$. 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 $a' = \quad q$). It is easy to understand the following : the situations which have the possibility of a' is a non-zero multiple of q, is $(\sharp, \flat) = ((1), (2)), ((2), (1)), ((2), (2)), ((3), (2))$, 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. $(\sharp, \flat) = ((1), (2))$. It is easy to show that $a' = c + c' + (q \quad 1)$, i.e. $c' = \quad c+1$. We apply the rule to $(c, c', c'', b, b')_\phi$, We have $e' = (\quad c+1)q+c''+b+q^2$ and it is easy to show that $e' > (q^2 \quad 1)/2$. This is contrary to $|e'| \geq (q^2 \quad 1)/2$.

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

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

Case 4. $(\sharp, \flat) = ((3), (2))$. It is easy to show that $a' = c + c' + q \quad 2$, i.e. $c' = \quad c + 2$. We apply the rule to $(c, c', c'', b, b')_\phi$, We have $e''' = bq + b' \quad q^2$ and $e' = (\quad c + 1)q + c'' + b \quad (q \quad 1) + q^2$. It is easy to show that $b = (q \quad 3)/2, (q \quad 1)/2, (q+1)/2$. It is also easy to see that $\quad cq + c'' < \quad (q^2 \quad 1)/2$. 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 ϕ-rNAF) *Let $d \not/ \mathbb{Z}[\phi]$ and $\ell_{\phi\text{-}EXP}(d)$. Then d has a ϕ-rNAF with digit set $D_{q,(q^2-1)/2}$ such that $\ell_{\phi\text{-}r\mathrm{NAF}}(d)$ is at most $\ell + 4$.*

Proof Let $\ell' = \ell_{\phi\text{-}r\mathrm{NAF}}(d)$. 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 ϕ-rNAF. Now we scan the two digits $(b, b')_\phi$. We can assume that $e_{\ell-3} = 0$ because if $e_{\ell-3} \neq 0$, we can reduce this case to the case 2 (See below for details). If $(b, b')_\phi$ is the ϕ-non-adjacent pair, then $\ell' = \ell$. Otherwise, we convert the pair $(b, b')_\phi$ as follows :

$$(b, b')_\phi \leftarrow \begin{cases} (b, 0, bq + b')_\phi & \flat bq + b' \flat \geq (q^2 \quad 1)/2, \\ (1, b + (q \quad 1), 0, bq + b' + q^2)_\phi & bq + b' < \quad (q^2 \quad 1)/2, \\ (\bar{1}, b \quad (q \quad 1), 0, bq + b' \quad q^2)_\phi & bq + b' > (q^2 \quad 1)/2, \end{cases}$$

and we put the result $(c, c', e_{\ell-1}, e_{\ell-2})_\phi$. If $\lvert bq + b' \rvert \geq (q^2 \quad 1)/2$, then $\ell' \geq \ell + 1$. Otherwise, since $\lvert b \rvert \geq (q+1)/2$,

$$\lvert cq + c' \rvert = \lvert \mp q \mp (q \quad 1) + b \rvert = \lvert \mp (2q \quad 1) + b \rvert \geq (q^2 \quad 1)/2,$$

when $q \sim 5$. We can convert $(1, b + (q \quad 1))_\phi \nleftrightarrow (1, 0, b + (2q \quad 1))_\phi$ or $(\bar{1}, b \quad (q \quad 1))_\phi \nleftrightarrow (\bar{1}, 0, b \quad (2q \quad 1))_\phi$, 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 ϕ-rNAF. Now we scan the two digits $(b, b')_\phi$. We can assume that $b' \neq 0$ 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 $b \neq 0$. Then we convert

$$(c, b, b')_\phi \nleftrightarrow \begin{cases} (c + b, 0, bq + b')_\phi & \lvert bq + b' \rvert \geq (q^2 \quad 1)/2, \\ (1, c + b + (q \quad 1), 0, bq + b' + q^2)_\phi & bq + b' < \quad (q^2 \quad 1)/2, \\ (\bar{1}, c + b \quad (q \quad 1), 0, bq + b' \quad q^2)_\phi & bq + b' > (q^2 \quad 1)/2, \end{cases}$$

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

$$\lvert aq + a' \rvert = \lvert \mp q \mp (q \quad 1) + b + c \rvert = \lvert \mp (2q \quad 1) + b + c \rvert \geq (q^2 \quad 1)/2,$$

when $q \sim 9$. We can convert

$$(\mp 1, \mp(q \quad 1) + b + c)_\phi \quad \nleftrightarrow \quad (\mp 1, 0, \mp(2q \quad 1) + b + c)_\phi.$$

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

$$(\mp 1, b + c \mp (q \quad 1))_\phi \quad \nleftrightarrow \quad (\bigcirc 1, \bigcirc(q \quad 2), 0, b + c \bigcirc(q \quad 1)^2)_\phi.$$

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

$$(\mp 1, b + c \mp (q \quad 1))_\phi \quad \nleftrightarrow \quad (\mp 1, \mp q, 0, b + c \mp (q \quad 1)^2)_\phi.$$

Here, $\mp q \mp q \quad 2 = \mp 2(q \quad 1)$, so $\backslash \mp 2(q \quad 1) \backslash \geq (q^2 \quad 1)/2$. Then $\ell' = \ell + 4$. Therefore $\ell_{\phi\text{-}r\mathrm{NAF}}(d)$ is at most 4 more than $\ell_{\phi\text{-EXP}}(d)$. □

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

$$(b_{i+1}, b_i)_\phi \Lsh (\quad b_i/q, b_{i+1} + b_i/q, 0)_\phi.$$

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

$$\backslash b_{i+1} \backslash \geq (q \quad 1)/2, \ \backslash b_i \backslash \geq q \quad 1 \quad \text{or} \quad \backslash b_{i+1} \backslash \geq (q+1)/2, \ \backslash b_i \backslash \geq 2q \quad 1 \ (\backslash b_i \backslash \models q),$$

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 ϕ-$r\mathrm{NAF}_\ell$ be the set of ϕ-rNAF of the length ℓ. We put $B_\ell = \#\phi\text{-}r\mathrm{NAF}_\ell$, $T_\ell = \prod_{d \in \phi\text{-}r\mathrm{NAF}_\ell} (\ell \quad w(d))$, where $w(d)$ means the Hamming weight of d. Then as is the case with ϕ-GNAF, the non-zero density of ϕ-rNAF is defined by $1 \quad \lim_{\ell \to \infty} T_\ell/(\ell B_\ell)$.

For ϕ-rNAF, we also have similar properties except for the minimality of the Hamming weight. As is the case of ϕ-GNAF, although ϕ-rNAF does not have minimal Hamming weight, asymptotic average non-zero density of ϕ-rNAF is same as rNAF. This indicates that it can be reduced the average number of point additions required by the ϕ-rNAF than the average number of point additions required by the Frobenius expansion.

Algorithm 7 ϕ-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$ ($q = 3$ or 5)

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

Output: ϕ-rNAF of d

1: $d' \to d \bmod (\phi^n \quad 1)$
2: $\ell \to \rceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\{ + 4$
3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of d'
4: $b_0 \to c_0$, $b_1 \to c_1, b_\ell \to 0, b_{\ell+1} \to 0, b_{\ell+2} \to 0, b_{\ell+3} \to 0$
5: $i \to 0$
6: **while** $i \geq \ell + 2$ **do**
7: **if** $b_i \bmod q = 0$ **then**
8: $b_{i+1} \to b_{i+1} \quad b_i/q, b_{i+2} \to c_{i+2} + b_i/q, i \to i + 1$
9: **end if**
10: **if** $b_i = 0$ **then**
11: $b_{i+2} \to c_{i+2}, e_i \to 0, i \to i + 1$
12: **else if** $\backslash b_{i+1}q + b_i \backslash \geq (q^2 \quad 1)/2$ **then**
13: $b_{i+3} \to c_{i+3}, b_{i+2} \to c_{i+2} + b_{i+1}, e_{i+1} \to 0, e_i \to b_{i+1}q + b_i, i \to i + 2$
14: **else if** $b_{i+1}q + b_i < \quad (q^2 \quad 1)/2$ **then**
15: $b_{i+3} \to c_{i+3} + 1, b_{i+2} \to c_{i+2} + b_{i+1} + (q \quad 1), e_{i+1} \to 0, e_i \to b_{i+1}q + b_i + q^2, i \to i + 2$
16: **else**
17: $b_{i+3} \to c_{i+3} \quad 1, b_{i+2} \to c_{i+2} + b_{i+1} \quad (q \quad 1), e_{i+1} \to 0, e_i \to b_{i+1}q + b_i \quad q^2, i \to i + 2$
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 ϕ-rNAF)
(1) *Every d / $\mathbb{Z}[\phi]$ has a unique ϕ-rNAF.*
(2) *The average density of non-zero digits among ϕ-rNAF's of length ℓ is equal to* $(q \quad 1)/(2q \quad 1) + (2q^2 \quad 2q + 1)/((2q \quad 1)^2\ell) + O(((q \quad 1)/q)^\ell)$. *In particular, for a fixed q, the average non-zero density of ϕ-rNAF is asymptotically* $(q \quad 1)/(2q \quad 1)$.
(3) *ϕ-rNAF does not have minimal Hamming weight among all Frobenius expansion with digit set $D_{q,(q^2-1)/2}$.*

Proof

(1) By the same way as Proposition 3.2, we suppose $d \not\equiv \mathbb{Z}[\phi]$ has two such representations

$$d = \sum_{i=0}^{\ell-1} e_i \phi^i = \sum_{i=0}^{\ell-1} e_i' \phi^i, \quad (e_0, e_0') \neq (0, 0).$$

Then from Lemma 3.1, we have $q \mid (e_0 - e_0')$, and by the assumption, $|e_0|, |e_0'| \leq (q^2 - 1)/2$, hence $|e_0 - e_0'| \geq |e_0| + |e_0'| \geq q^2 - 1$. Therefore we have $e_0 - e_0' = \mp q, \mp 2q, \ldots, \mp(q-1)q$. We put $e_0 - e_0' = aq$, where $a \not\equiv \{\mp 1, \mp 2, \ldots, \mp(q-1)\}$. We must have $(e_0, e_0') \neq (0, 0)$. Because if $e_0 = 0$, $e_0' = -(e_0 - e_0')$ is divisible by q but $e_0' \not\equiv D_{q,(q^2-1)/2}$, i.e. $q \nmid e_0'$. This is contrary to $e_0 = 0$. Similarly, we can show that $e_0' \neq 0$. By the definition of ϕ-rNAF, we must have $(e_1, e_1') \neq (0, 0)$. We have $aq = (e_0 - e_0') = \sum_{i=1}^{\ell-1}(e_i' - e_i)\phi^i$ and by the characteristic polynomial of ϕ, we have $a\phi - a\phi^2 = \sum_{i=1}^{\ell-1}(e_i' - e_i)\phi^i$. 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 $q \mid a$. This is contrary to $a \not\equiv \{\mp 1, \mp 2, \ldots, \mp(q-1)\}$.

(2) We will show that B_ℓ and T_ℓ satisfy the recurrence equations $B_1 = B_2 = q^2 - q$, $B_{\ell+2} - B_{\ell+1} - (q^2 - q)B_\ell = 0$ and $T_1 = 0$, $T_2 = q^2 - q$,

$$T_{\ell+2} - T_{\ell+1} - (q^2 - q)T_\ell = (q^2 - q)\left(\sum_{i=1}^{\sim \ell} B_i\right).$$

In Table 3.2, the symbol '$\pm$' means a non-zero digit. We explain how to interpret Table 3.2. The left column stands for forms of ℓ digits ϕ-rNAF, the central column stands for the number of each ϕ-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

$B_\ell = B_1 \Big)1 + \prod_{i=1}^{\ell-2} B_i \Big($, and

$$T_\ell = B_1 \Big)1 + \sum_{i=1}^{\ell-2} T_i \Big[+ B_1 \Big)\sum_{i=1}^{\ell-2} (\ell \quad 1 \quad i)B_i + (\ell \quad 1) \Big[.$$

Table 3.2: Forms of ℓ digits ϕ-rNAF representations

forms	# of representations	# of 0 digits
$(\pm 0 \pm \times\times\times)$	$B_1 B_{\ell-2}$	$B_1 B_{\ell-2} + B_1 T_{\ell-2}$
$(\pm 00 \pm \times\times\times)$	$B_1 B_{\ell-3}$	$2B_1 B_{\ell-3} + B_1 T_{\ell-3}$
$(\pm 000 \pm \times\times\times)$	$B_1 B_{\ell-4}$	$3B_1 B_{\ell-4} + B_1 T_{\ell-4}$
$\vdots$	$\vdots$	$\vdots$
$(\pm 0000 \times\times\times 0 \pm 0)$	$B_1 B_2$	$(\ell \quad 3)B_1 B_2 + B_1 T_2$
$(\pm 0000 \times\times\times 00 \pm)$	$B_1 B_1$	$(\ell \quad 2)B_1 B_1 + B_1 T_1$
$(\pm 0000 \times\times\times 000)$	B_1	$(\ell \quad 1)B_1$

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

$$T_\ell = \frac{q(q \quad 1)}{(2q \quad 1)^2} \Big)(q^{\ell+1} + (1 \quad q)^{\ell+1})\ell \quad \frac{2q^2 \quad 2q+1}{2q \quad 1}(q^\ell \quad (1 \quad q)^\ell) \Big\rfloor ,$$

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

$$f(\ell) := \Big)1 \quad \frac{T_\ell}{\ell B_\ell} \Big\lfloor \quad \Big)\frac{q \quad 1}{2q \quad 1} + \frac{2q^2 \quad 2q+1}{(2q \quad 1)^2 \ell} \Big\rfloor .$$

Then

$$f(\ell) = \frac{1}{2q \quad 1} \times \Big) \frac{(1 \quad q)^\ell}{q^\ell \quad (1 \quad q)^\ell} \Big\rfloor .$$

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

$$\frac{f(\ell)}{g(\ell)} = \frac{1}{2q \quad 1} \times \frac{(\quad 1)^{\ell-1}}{1 \quad (\quad (q \quad 1)/q)^\ell},$$

we obtain

$$\left(\frac{f(\ell)}{g(\ell)}\right) = \frac{1}{2q \quad 1} \times \frac{1}{1 \quad (\quad (q \quad 1)/q)^{\ell}} \Leftarrow \frac{1}{2q \quad 1} \quad (\ell \Leftarrow \in).$$

Hence $f(\ell) = O(g(\ell))$. Thus we have $1 \quad \lim_{\ell \to \infty} T_\ell/(\ell B_\ell) = 1 \quad q/(2q \quad 1) = (q \quad 1)/(2q \quad 1)$ 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 α with digit set $D_{q,(q^2-1)/2}$ is $(3, 8)_\phi$ and its Hamming weight is 2. But ϕ-rNAF of α is $(\quad 1, 0, \quad 6, 0, 2)_\phi$ and its Hamming weight is 3. Therefore ϕ-GNAF does not have minimal Hamming weight among all Frobenius expansion with digit set $D_{q,(q^2-1)/2}$. □

Algorithm 8 computes the ϕ-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 ϕ-rNAF compared to Algorithm 6 and 7. From Theorem 3.4, it is easy to show the correctness of Algorithm 8.

If we compute $2P = P + P, 3P = P + 2P$ ($3P = \phi(P) \quad \phi^2(P)$ when $q = 3$), $4P = P + 3P, (q \quad 1)P = P + (q \quad 2)P, (q + 1)P = P + (q \quad 1)P$, and so on, the computational cost of Horner's method using ϕ-rNAF (Algorithm 2 based on ϕ-rNAF with $\alpha = (q^2 \quad 1)/2$, digit set $D_\phi = D_{q,(q^2-1)/2}$) is approximately

$$\left] \mathcal{D}_{\mathcal{P}} + 2\mathcal{A}_{\mathcal{P}} + 2\mathcal{F}_{\mathcal{P}} + \frac{q \quad 1}{2q \quad 1}\ell\mathcal{A}_{\mathcal{S}} + \ell\mathcal{F}_{\mathcal{S}} \right\{ (q = 3),$$

$$\left] \mathcal{D}_{\mathcal{P}} + \frac{q^2 \quad q \quad 4}{2}\mathcal{A}_{\mathcal{P}} + \frac{q \quad 1}{2q \quad 1}\ell\mathcal{A}_{\mathcal{S}} + \ell\mathcal{F}_{\mathcal{S}} \right\{ (q \sim 5),$$

where $\mathcal{F}_{\mathcal{P}}$ stands for the computational cost of the Frobenius map in the precomputational step.

Algorithm 8 ϕ-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = 1$

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

Output: ϕ-rNAF of d

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

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

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

Contrary to the ϕ-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 ϕ-GNAF of $t = \quad 1$. For example, let $a, b \ / \ D_{q,q-1}$ and we assume that $a, b > 0$. If the pair $(a, b)_\phi$ is

ϕ-non-admissible pair, we have $\backslash a + b \backslash = a + b \sim q$. Even if we convert $(a, b)_\phi$ to $(\ \ 1, a \quad 1, b \quad q)_\phi$, the pair $(a \quad 1, b \quad q)_\phi$ is also ϕ-non-admissible pair when it satisfies that $a+b \quad q = 0$ or $a+b \quad q = 1$. Besides, if we convert $(\ \ 1, a \quad 1, b \quad q)_\phi$ to $(\ \ 2, a \quad 1 \quad q, b \quad q)_\phi$, the pair $(a \quad 1 \quad q, b \quad q)_\phi$ is also ϕ-non-admissible pair when it satisfies that $a + b \quad q = 0$ or $a + b \quad q = 1$.

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

$$(a,b)_\phi \leftarrow \begin{cases} (\ \ a, 0, \quad aq + b)_\phi & \text{if } \backslash \quad aq + b \backslash \geq (q^2 \quad 1)/2, \\ (\bar{1}, \quad a + (q \quad 1), 0, (\ \ aq + b) + q^2)_\phi & \text{else if } \quad aq + b < \quad (q^2 \quad 1)/2, \\ (1, \quad a \quad (q \quad 1), 0, (\ \ aq + b) \quad q^2)_\phi & \text{otherwise.} \end{cases}$$

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 ϕ-rNAF of $t = \quad 1$ is similar to Algorithm 6. Moreover, Theorem 3.4 and Proposition 3.3 is also true for the ϕ-rNAF of $t = \quad 1$. In the case of $t = \quad 1$, 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 ϕ-rNAF of $t = \quad 1$ for a given $d \ / \ \mathbb{Z}[\phi]$, where $q \sim 7$.

Algorithm 9 ϕ-rNAF on $\mathbb{F}_q$-Koblitz curves of $t = \quad 1$ ($q \sim 7$)

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

Output: ϕ-rNAF of d

1: $d' \to d \bmod (\phi^m \quad 1)$
2: $\ell \to \left\lceil 2\log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d)}\sqrt{} + 4 \right.$
3: Compute the Frobenius expansion $(c_{\ell-1}, c_{\ell-2}, \ldots, c_1, c_0)_\phi$ of d'
4: $b_0 \to c_0,\ b_1 \to c_1, b_\ell \to 0, b_{\ell+1} \to 0, b_{\ell+2} \to 0, b_{\ell+3} \to 0$
5: $i \to 0$
6: **while** $i \geq \ell + 2$ **do**
7: **if** $b_i = 0$ **then**
8: $b_{i+2} \to c_{i+2}, e_i \to 0, i \to i + 1$
9: **else if** $\backslash \quad b_{i+1}q + b_i\backslash \geq (q^2 \quad 1)/2$ **then**
10: $b_{i+3} \to c_{i+3}, b_{i+2} \to c_{i+2} \quad b_{i+1}, e_{i+1} \to 0, e_i \to \quad b_{i+1}q + b_i, i \to i + 2$
11: **else if** $\quad b_{i+1}q + b_i < \quad (q^2 \quad 1)/2$ **then**
12: $b_{i+3} \to c_{i+3} \quad 1, b_{i+2} \to c_{i+2} \quad b_{i+1} + (q \quad 1), e_{i+1} \to 0, e_i \to \quad b_{i+1}q + b_i + q^2, i \to i + 2$
13: **else**
14: $b_{i+3} \to c_{i+3} + 1, b_{i+2} \to c_{i+2} \quad b_{i+1} \quad (q \quad 1), e_{i+1} \to 0, e_i \to \quad b_{i+1}q + b_i \quad q^2, i \to i + 2$
15: **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 $D = \}0, \mp 1, \ldots, \mp(q \quad 1)\langle$ and $d_0 = d \bmod (\phi^m \quad 1)$,

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 $m_q = \lceil \log_q d \{ \subset \lceil \log_q q^m \{ = m$ and $\ell \geq \left\lceil 2 \log_q 2 \quad \overline{N_{\mathbb{Z}[\phi]/\mathbb{Z}}(d_0)} \sqrt{+} \right.$ $4 \subset \left\lceil 2 \log_q 2q^{(m+1)/2} \sqrt{+} \right. 4 \geq m + 6$. 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 $\ell_{\phi\text{-}r\text{NAF}}(d_0)$ are equal to each other (Strictly speaking, we should analyze each average of $\ell_{\phi\text{-EXP}}(\not{\ast}$, $\ell_{\phi\text{-GNAF}}(\not{\ast}$ and $\ell_{\phi\text{-}r\text{NAF}}(\not{\ast}$ among positive integers in the range $[1, \#E(\mathbb{F}_{q^m}) \quad 1]$ 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 $(\phi^2 \quad \phi)$, i.e., the computational cost of a point multiplication by q is $\mathcal{A}_{\mathcal{S}} + 2 * \mathcal{F}_{\mathcal{S}}$. 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 ($q = 3$)

Method	#Table	$\mathcal{A}_{\mathcal{P}}$	$\mathcal{D}_{\mathcal{P}}$	$\mathcal{F}_{\mathcal{P}}$	$\mathcal{A}_{\mathcal{S}}$	$\mathcal{D}_{\mathcal{S}}$	$\mathcal{F}_{\mathcal{S}}$
w-NAF [72]	2^{w-2}	$(2^{w-2} \quad 1)$	1	0	$\frac{m_2}{w+1}$	ℓ	0
Frobenius expansion [70]	1	0	0	0	0.67ℓ	0	ℓ
ϕ-GNAF	2	0	1	0	0.5ℓ	0	ℓ
ϕ-rNAF	3	2	1	2	0.4ℓ	0	ℓ

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

Method	#Table	$\mathcal{A}_{\mathcal{P}}$	$\mathcal{D}_{\mathcal{P}}$	$\mathcal{F}_{\mathcal{P}}$	$\mathcal{A}_{\mathcal{S}}$	$\mathcal{D}_{\mathcal{S}}$	$\mathcal{F}_{\mathcal{S}}$
w-NAF [72]	2^{w-2}	$(2^{w-2} \quad 1)$	1	0	$\frac{m_2}{w+1}$	ℓ	0
Frobenius expansion [70]	$\frac{q-1}{2}$	$\frac{q-5}{2}$	1	0	$\frac{q-1}{q}\ell$	0	ℓ
ϕ-GNAF	$q \quad 1$	$q \quad 3$	1	0	$\frac{q-1}{q+1}\ell$	0	ℓ
ϕ-rNAF	$\frac{q^2-1}{2}$	$\frac{q^2-q-4}{2}$	1	0	$\frac{q-1}{2q-1}\ell$	0	ℓ

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 $\mathcal{A}$ (resp.$\mathcal{J}$). If we choose

$$\equiv \mathcal{A}_{\mathcal{P}} : \mathcal{A} + \mathcal{A} \Leftarrow \mathcal{A} \text{ (2M + S + I)}, \quad \mathcal{D}_{\mathcal{P}} : 2\mathcal{A} \Leftarrow \mathcal{A} \text{ (2M + 2S + I)},$$

$$\equiv \mathcal{A}_{\mathcal{S}} : \mathcal{J} + \mathcal{A} \Leftarrow \mathcal{J} \text{ (8M + 3S)}, \quad \mathcal{D}_{\mathcal{S}} : 2\mathcal{J} \Leftarrow \mathcal{J} \text{ (4M + 4S)},$$

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

Method	#Table	M	S	I	# of M
192bit					
w-NAF $(w = 6)$	16	1019.4	850.3	1	1721.3
Frobenius expansion $(q = 3)$	1	650.7	244.0	0	845.9
ϕ-GNAF $(q = 3)$	2	490.0	185.0	1	646.0
ϕ-rNAF $(q = 3)$	3	396.4	150.4	3	424.6
Frobenius expansion $(q = 5)$	2	533.2	201.2	1	629.0
ϕ-GNAF $(q = 5)$	4	448.7	170.0	3	608.7
ϕ-rNAF $(q = 5)$	12	295.1	120.7	9	481.5
Frobenius expansion $(q = 7)$	3	477.3	180.3	2	592.6
ϕ-GNAF $(q = 7)$	6	424.0	161.3	5	593.0
ϕ-rNAF $(q = 7)$	24	294.8	119.5	20	548.0
224bit					
w-NAF $(w = 6)$	16	1184.0	1009.0	1	1999.2
Frobenius expansion $(q = 3)$	1	757.3	284.0	0	984.5
ϕ-GNAF $(q = 3)$	2	570.0	215.0	1	742.0
ϕ-rNAF $(q = 3)$	3	460.4	174.4	3	599.9
Frobenius expansion $(q = 5)$	2	622.8	234.8	1	818.6

Method	#Table	M	S	I	# of M
ϕ-GNAF $(q = 5)$	4	523.3	198.0	3	705.7
ϕ-rNAF $(q = 5)$	12	362.9	139.3	9	546.4
Frobenius expansion $(q = 7)$	3	552.6	208.7	2	735.5
ϕ-GNAF $(q = 7)$	6	490.0	186.0	5	678.8
ϕ-rNAF $(q = 7)$	24	335.4	131.8	20	600.8
256bit					
w-NAF $(w = 6)$	16	1348.6	1150.7	1	2277.1
Frobenius expansion $(q = 3)$	1	864.0	324.0	0	1123.2
ϕ-GNAF $(q = 3)$	2	650.0	245.0	1	846.0
ϕ-rNAF $(q = 3)$	3	524.4	198.4	3	683.1
Frobenius expansion $(q = 5)$	2	712.4	268.4	1	935.1
ϕ-GNAF $(q = 5)$	4	598.0	226.0	3	802.8
ϕ-rNAF $(q = 5)$	12	412.7	158.0	9	611.1
Frobenius expansion $(q = 7)$	3	634.9	239.6	2	842.5
ϕ-GNAF $(q = 7)$	6	562.0	213.0	5	772.4
ϕ-rNAF $(q = 7)$	24	379.7	148.4	20	658.4
384bit					
w-NAF $(w = 6)$	16	2006.9	1717.6	1	3389.0
Frobenius expansion $(q = 3)$	1	1296.0	486.0	0	1684.8
ϕ-GNAF $(q = 3)$	2	974.0	366.5	1	1267.2
ϕ-rNAF $(q = 3)$	3	783.6	295.6	3	1020.1
Frobenius expansion $(q = 5)$	2	1064.4	400.4	1	1392.7
ϕ-GNAF $(q = 5)$	4	891.3	336.0	3	1184.1
ϕ-rNAF $(q = 5)$	12	608.2	231.0	9	865.3
Frobenius expansion $(q = 7)$	3	878.8	330.8	2	1151.4
ϕ-GNAF $(q = 7)$	6	832.0	314.3	5	1123.4
ϕ-rNAF $(q = 7)$	24	545.8	210.7	20	874.4

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 $\mathbb{F}_3$-Koblitz Curve $K_{(3,1)}$-173 (See Table 3.9 in Section 3.5) for binary method, w-NAF, rNAF, the Frobenius expansion, and ϕ-rNAF. 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 rNAF, [70, Theorem 3] and modular reduction in $\mathbb{Z}[\phi]$ ([70, pp.146–147]) for the Frobebnius expansion, Algorithm 7 for ϕ-rNAF, respectively.

In the precomputation step, we use Weierstrass affine coordinates. Precomputation points are stored in Weierstrass affine coordinates for rNAF, the Frobenius expansion, and ϕ-rNAF. 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 rNAF, the Frobenius expansion, and ϕ-rNAF.

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 rNAF, the Frobenius expansion, and ϕ-rNAF. Additionaly, we use in [37, Algorithm 3. 27] for binary method, Algorithm 1 for w-NAF, Triple and Add Algorithm in [71] for rNAF, Algorithm 2 with $\alpha = (q \quad 1)/2$ for the Frobebnius expansion, Algorithm 2 with $\alpha = (q^2 \quad 1)/2$, $\ell = \ell + 4$ for ϕ-rNAF, respectively.

The plat form 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-framepointer-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 $d = n \quad 2$. For each method, the representation of $n \quad 2$ has average Hamming weight. Note that the timing of each method in Table 3.8 does not constain the timing of the recoding method.

Table 3.6: Timings of the Finite Field Operations

Finite Field Operations	timing (in μs)
Addition	0.02
Cube	0.51
Multiplication	3.81
Inversion	65.77

Table 3.7: Timings of Recoding Methods

Recoding Methods	timing (in μs)
binary	n/a
w-NAF ($w = 6$)	53.42
rNAF ($r = 3$)	159.54
Frobenius expansion	169.92
ϕ-rNAF	177.75

In Table 3.8, $\mathcal{A}_{\mathcal{W}}$, $\mathcal{P}_{\mathcal{W}}$, $\mathcal{A}_{\mathcal{H}}$, $\mathcal{P}_{\mathcal{H}}$ 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)

Methods	Coordinates (Precomputation)	Coordinates (Scalar Multiplication)		timing (in ms)
		Point Addition	Point Doubling Point Tripling Frobenius Map	
binary	$\mathcal{A}_\mathcal{W} + \mathcal{A}_\mathcal{W} \to \mathcal{A}_\mathcal{W}$ $2\mathcal{A}_\mathcal{W} \to \mathcal{A}_\mathcal{W}$	$\mathcal{P}_\mathcal{H} + \mathcal{A}_\mathcal{H} \to \mathcal{P}_\mathcal{H}$	$2\mathcal{P}_\mathcal{H} \to \mathcal{P}_\mathcal{H}$	**7.77**
w-NAF ($w = 6$)				**5.75**
rNAF ($r = 3$)		$\mathcal{P}_\mathcal{W} + \mathcal{A}_\mathcal{W} \to \mathcal{P}_\mathcal{W}$	$3\mathcal{P}_\mathcal{W} \to \mathcal{P}_\mathcal{W}$	**8.14**
Frobenius expansion			$\phi\mathcal{P}_\mathcal{W} \to \mathcal{P}_\mathcal{W}$	**6.20**
ϕ-rNAF				**3.90**

timing of ϕ-rNAF is between 1.5 and 2 times faster than those of binary method, w-NAF, rNAF, 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 $(a, b) = (2, 1)$ $(t = \quad 1)$. 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 / $E(\mathbb{F}_{3^m})$. $P = (x, y)$.

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 0D7A2C0B 42D347DA 07AFBAAD
$x =$ 0x 0000036D E36A98A8 D56DDACF C011EC49
$y =$ 0x 000000A9 2FE6BE42 8DC74805 7B640ECB
$K_{(3,1)}$-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 00818AD9 3E1DDF77 1C305C95
$x =$ 0x 00004FF6 583FB8D7 5F959EE5 14B11CB3
$y =$ 0x 0000B903 A6F791B9 B1B45197 28BDCEA7
$K_{(3,1)}$-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 23D0435F D52C298D
$x =$ 0x 0005C879 CE7EF384 FE7118F9 68394290
$y =$ 0x 00024CDA BA42B22E 6D48E8AD 85A81F1D
$K_{(3,1)}$-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 C009C029
$x =$ 0x 239F6D7B F71E7F65 5DC8BA17 B038D640
$y =$ 0x 1055DDF1 29D8B12D 46CBE4D0 F4951F9D
$K_{(3,1)}$-89: $m = 89$, $f(x) = x^{89} + 2x^{13} + 1$, $a = 1$, $b = 2$, $h = 3 \times 179^2$, $\|h\| = 17$, $\|n\| = 125$
$n =$ 0x 00000000 16C5250D 8B657F3E 274AB564 81F60CA7
$x =$ 0x 00000B21 E2ADF87B A018D443 9A947D5B ED64B48C
$y =$ 0x 00000E9B 57923010 7FDA3B49 22F7E4FB 236D2381
$K_{(3,1)}$-131: $m = 131$, $f(x) = x^{131} + 2x^{27} + 1$, $a = 1$, $b = 2$, $h = 3 \times 103322059$, $\|h\| = 29$, $\|n\| = 180$
$n =$ 0x 00000000 000AB8ED 4C070E99 F70B469E 3DCB40AC C6B1D8C1 DE40A849
$x =$ 0x 00003D21 EB4D1339 567FEB9B 610C1E35 646FF22A A6F2B3A6 8C58EE80
$y =$ 0x 00007072 8F598DB2 A7E18F01 F9701125 8ADAA39E 41C52435 BCB2F36E
$K_{(3,1)}$-137: $m = 137$, $f(x) = x^{137} + 2x + 1$, $a = 1$, $b = 2$, $h = 3 \times 82349398913$, $\|h\| = 37$, $\|n\| = 181$
$n =$ 0x 00000000 0013DD81 7356A56F 7344A743 74EF4124 EAEFADB0 CE570C99
$x =$ 0x 012AABA1 C9645237 8D48957F 0D658D80 312201A9 B0BCEA23 128EA426
$y =$ 0x 0209BA61 C4956DD9 4B2EBF12 484209F2 628FA8E0 7F22BCC7 C5F74964
$K_{(3,1)}$-157: $m = 157$, $f(x) = x^{157} + 2x^{22} + 1$, $a = 1$, $b = 2$, $h = 3 \times 12424039 \times 196520041$, $\|h\| = 53$, $\|n\| = 197$
$n =$ 0x 00000000 00000011 995535CF E334EF32 FBE8EF37 C8429B9A 1D0A9C1A 6300F7A7
$x =$ 0x 01C444E6 95E2CAD1 A4E9C4A3 50004F3A 49D9C0B5 04073A47 3A7E587A 7E2BC4BB
$y =$ 0x 005F258F 76BC2455 30C425F9 12E52AB6 C33BCAE5 3E2C1307 177CDCA3 CD6F171A
$K_{(3,1)}$-173: $m = 173$, $f(x) = x^{173} + 2x^7 + 1$, $a = 1$, $b = 2$, $h = 3 \times 13841$, $\|h\| = 16$, $\|n\| = 259$
$n =$ 0x 00000007 3E98AF5C 5582BFF6 488420E2 204975CB 9DD3F6BB 7FD35200 460A5265\ 06CAD487
$x =$ 0x 0001DC27 001037F6 B71AD345 80795836 F56494FD 68EFD9D0 B30AB2F1 1F116B3E\ 65C1B782
$y =$ 0x 0000093C 414D0E94 54E95C94 1EA65879 3DAF3112 ABD06704 DD117F34 41B00AFE\ 67D3FAFA
$K_{(3,1)}$-271: $m = 271$, $f(x) = x^{271} + 2x^{50} + 1$, $a = 1$, $b = 2$, $h = 3$, $\|h\| = 2$, $\|n\| = 428$
$n =$ 0x 00000F58 CE524B47 41035912 5CF1CB97 5B1BDA26 8BE9AE0D 0E08088B 96AEF224\ DB049A5F 9CCA8541 1FE21CED 5286BCD3 AA5789E8 6DB00E45
$x =$ 0x 000013D4 3729B9A3 495A2426 B6A3358D 75A66E71 AFF97595 B19A7C21 3F45478E\ 6C65B6B9 62304FE3 389310AF 5FB413C7 9FF6B7C3 5DD62DA7

$y =$ 0x 0000241B 6436A572 A19CDF26 C40422FD B32E6902 07A9D50F 430566BD 4B595497\ 34C3B21B 69DD4479 65395AFE 3D6ACD9E 85D617AA 65D9DCEE
$K_{(3,1)}$-409: $m = 409$, $f(x) = x^{409} + 2x^{99} + 1$, $a = 1$, $b = 2$, $h = 3$, $\|h\| = 2$, $\|n\| = 647$
$n =$ 0x 00000065 748DF2AB 667F95B3 7CAF05D2 3486DE2D E347D045 28E47CDE 99932CBC\ 6011F56B 8DAD3499 A57DF4F9 A30D7371 C8920A4A 599BC22C D5826376 86203BAC\ 851A2E2A F858FC0B 7CE12764 E47B2F27 C42C7331
$x =$ 0x 00000080 C84D3105 28E2A719 8B9FA2DC 972A17F6 532FFD1B FDB6E578 2D629980\ D0939D10 9A12CA93 808CE70C ABA7B992 AA230700 4207E5A4 8BAB477C DB05C035\ D817853E FF036178 CDD64D07 82E2EE1E DDC3C75A
$y =$ 0x 0000003C 68B3AEF3 9AE9F9BA 08E7BFE2 8402CB2F 785E5BE4 13C9FBEA B5F07FD0\ B109772D D55BEEE7 79B15F24 97984554 C7E62067 91ABD6EF 21B1A067 DEBDC815\ 2A67163F CBE1426B A78EAA3B B266498D D02C834F
$K_{(3,1)}$-449: $m = 449$, $f(x) = x^{449} + 2x^{52} + 1$, $a = 1$, $b = 2$, $h = 3$, $\|h\| = 2$, $\|n\| = 711$
$n =$ 0x 00000042 DDB063CA 56DD7CAC 6D58162F D7870343 84422069 9FADED04 42B08C10\ 8F349CB6 234DB8E7 67F17D8A 9C22BA7D 238E935D 8B9DAF46 0C80CE84 F164BB99\ 93CEFB3F A449A5F5 BFE3CA1C BF45457E D66183F9 48244053 E4CCB06F
$x =$ 0x 00000043 146CCB64 4BB79CF5 85A948E6 3593A1EB B3439F7A 40A416E3 AF6FCE48\ 27CED6B9 ECD87B9D 7220BD9C 3CF13AAB 0E738D98 8BB637F9 EDFC207D A1A6C100\ 85DB9016 E6F2B9B5 37D27A9A C30154E7 3718966B 42368D90 222B0739
$y =$ 0x 00000098 53EC043B 05AA8E7F 2FE6F5B9 13760170 CE8437C8 F89BBB78 2C8BF4F1\ A1B297BA D52F5DE3 02C7B85F FEA71E66 C992AAAD 48C6D06A ED3C3C2C C8BBC54C\ C6B999E7 79CF484E 91A17267 5FD6A795 5BBFEC57 8E98C0FB 622F769B

Table 3.10: Sample Parameters for $\mathbb{F}_3$-Koblitz Curves of $t = \quad 1$

$K_{(3,-1)}$-103: $m = 103$, $f(x) = x^{103} + x^{50} + 2$, $a = 2$, $b = 1$, $h = 5 \times 4327 \times 1036799$, $\|h\| = 35$, $\|n\| = 129$
$n =$ 0x 00000000 00000001 D2B35574 33F08C49 9322D80B C5F9092D
$x =$ 0x 00000007 284B450E 869B7E1F 306F457B CF6AA41D B3D6E2E9
$y =$ 0x 00000005 E0DE463B A058B2D2 7F1E9865 391F1A51 E1247771
$K_{(3,-1)}$-131: $m = 131$, $f(x) = x^{131} + 2x^{27} + 1$, $a = 2$, $b = 1$, $h = 5 \times 1049 \times 8647$, $\|h\| = 26$, $\|n\| = 183$
$n =$ 0x 00000000 0049480E 7FDADB8F 76EC9979 EF332739 F1F41326 9503B62D
$x =$ 0x 00007D74 1FB4D2AF 4B8D3E2D 460DFE4C 981CCCBA E485350F DEBEB7BA
$y =$ 0x 00009F9E 48BAC488 A4C72100 7AFD71C5 93A7744C 982EBAC9 EC9509F2
$K_{(3,-1)}$-139: $m = 139$, $f(x) = x^{139} + 2x^{59} + 1$, $a = 2$, $b = 1$, $h = 5 \times 49207$, $\|h\| = 18$, $\|n\| = 203$
$n =$ 0x 00000548 60F76059 D7493A96 8FAFE340 7E156352 AADD0D14 A672BB1B
$x =$ 0x 0DAB0EC2 793934D7 7346EA27 1BBC3AE4 82AB9EDE 6AFEFEDD B3701348
$y =$ 0x 1014DCC0 1C16200A F052A716 088B55AA 84D65D5D 5EB1BF32 FC57DAB8
$K_{(3,-1)}$-179: $m = 179$, $f(x) = x^{179} + 2x^{59} + 1$, $a = 2$, $b = 1$, $h = 5$, $\|h\| = 3$, $\|n\| = 282$
$n =$ 0x 029D3B06 85310E1C 772BA466 37C33F01 8A69D95B 45AEDEFC 6073BE02 2C7C0DC7\ 45BD3A23
$x =$ 0x 07BBC98A 2D42704F 77663B23 D1110AF5 1557E002 8A8882E4 A5FDD9C3 9FDB3151\ 199B6116
$y =$ 0x 069EBE86 4F28DBF7 8D2E5BF9 E95E17BE 0E712CC1 94133E13 BC4E7C0F 390F3987\ 99420B9B
$K_{(3,-1)}$-193: $m = 193$, $f(x) = x^{193} + 2x^{12} + 1$, $a = 2$, $b = 1$, $h = 5 \times 773$, $\|h\| = 12$, $\|n\| = 294$
$n =$ 0x 0000003F 2F5B6990 37A4EF79 11AB67DC 332B8C29 2460ADF3 6562517E C9DAE9AA\ A01A216B 0FA39F4D

$x =$ 0x	00006045	73E844EB	561E0972	8AD4B750	3AA5AD53	BD1D4EC0	E4752FEC	54AE6B69\
	604D0373	11298E22						
$y =$ 0x	00016680	15BF3C50	0B99293A	152F07E2	890298FA	5EEACCEF	DD207F1B	05F8C4B6\
	91AACE9A	E9378D4E						
$K_{(3,-1)}$-227: $m = 227$, $f(x) = x^{227} + 2x^{11} + 1$, $a = 2$, $b = 1$, $h = 5 \times 94433$, $\|h\| = 19$, $\|n\| = 341$								
$n =$ 0x	00000000	001EA4FD	B56B5F0E	980F8E70	AB5D6F9D	16B7DFA1	A5458383	7CF0F34D\
	53AFDA42	E031268C	7E6BCF93	E2E86C23				
$x =$ 0x	00000043	3529AE60	D7DAD310	441BC57B	0C7829C3	BF7A3798	09356A57	BD9FCE84\
	745056D0	D2D9551A	3113A630	31ECBB2C				
$y =$ 0x	00000098	733D95CB	430E7691	68CAC5C8	5A4DA04C	953D15F9	F94EC570	FD089FB9\
	032E0658	D33994A5	CF5F69A3	AFB893AB				
$K_{(3,-1)}$-269: $m = 269$, $f(x) = x^{269} + 2x^7 + 1$, $a = 2$, $b = 1$, $h = 5$, $\|h\| = 3$, $\|n\| = 425$								
$n =$ 0x	00000105	EB9F1615	D1225B45	7DA9B83D	4A5730AD	3C870B9A	7866EF80	C5C78EA6\
	24E1753F	56FFE6AD	4BEAC262	13C08485	861C9E11	748B8967		
$x =$ 0x	000003B7	F9D56F5D	509EADF8	CCEF88B5	77C7E651	5FFEF860	6E73FA03	E06FA4EA\
	BB06571A	BFFD694A	4DCE8A8E	10F0EDEA	B012EA0B	3FF90FC4		
$y =$ 0x	000001F9	05967EB1	64636A6E	B76552F4	092CB885	A055E564	3EC61706	C5CE87F8\
	422FFA34	FEFFC4A7	4E7AF00D	293FF202	03824693	FE900E5A		
$K_{(3,-1)}$-349: $m = 349$, $f(x) = x^{349} + 2x^{54} + 1$, $a = 2$, $b = 1$, $h = 5 \times 48163 \times 1423921$, $\|h\| = 39$, $\|n\| = 515$								
$n =$ 0x	00000000	00000007	20064187	5613800A	31B61B54	CBF86B26	B0D51A79	DC994D31\
	412D0F47	DBED2770	59502899	18C0F639	727C493E	AFF7A460	083BB928	8FD46F24\
	20C6FBB6	33627523						
$x =$ 0x	000000AA	55C1AF64	24B8F496	F4F32E61	B918A260	B8B40A13	F11D49AB	75719F06\
	D17E154B	4750BA39	F010025A	FEF6EFE6	AE167CA3	6612F5F8	23376CD4	4ABEACEC\
	379BE2CA	874763C4						
$y =$ 0x	000001A3	084992C7	72A3143F	E3300FB6	4C77C9D7	4995755C	BA89F275	FF73F83A\
	9056EF05	05645142	ADD72C35	89296AFF	A156BACB	10A3A612	FC25BB09	A5426FDF\
	8E386585	58AB3412						
$K_{(3,-1)}$-383: $m = 383$, $f(x) = x^{383} + 2x^{80} + 1$, $a = 2$, $b = 1$, $h = 5$, $\|h\| = 3$, $\|n\| = 605$								
$n =$ 0x	1A54D378	1C37C976	43012FF1	65F899DF	87FE3CF6	4DD81134	DF15418F	7C1614C5\
	9CCE5DF0	59DD1C85	7C8FEA30	E7D28D54	4FC8A2EE	CD392854	2E887272	E44A6EC4\
	86F4C4EA	CA38D2BB	0D36B7AB					
$x =$ 0x	5E1B131D	7EC7B977	BDF0B470	811D5A0B	392C1E27	3010AA06	C4BBAB88	C5607F14\
	B92586E2	3303E8DB	1BA48A51	40FB7F40	46AC61EE	D4223848	D3BA57BD	B851ACEC\
	89DD08EB	269303C0	2C303C40					
$y =$ 0x	0313F939	AB78280B	A634271B	714BCE18	143A4009	2F65D8D2	76326A60	FCB0DCF0\
	25F07E4C	0B1D60B6	632B8DA5	5913A6DE	B7F2226B	759BD4C8	344E2F0E	11598621\
	9CF3C9F2	2B6D04C3	58BFDB5E					

3.6 Summary of this chapter

It has been an unsolved problem to generalize τ-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 ϕ-NAF (ϕ-GNAF and ϕ-rNAF).

ϕ-GNAF can be applied to every subfield elliptic curves where the trace of the Frobenius map equals to 1, and ϕ-rNAF can be applied to every subfield elliptic curves where the trace of the Frobenius map equals to ∓ 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 τ-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 $P + P + \cdots + P$, 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:

$$dP = 2 \times\!\!\times 2(d_{\ell-1}2P + d_{\ell-2}P) + \times\!\!\times\!\!\times + d_1P| + d_0P,$$

where $\prod_{i=0}^{\ell-1} d_i 2^i = (d_{\ell-1}, d_{\ell-2}, \ldots, d_1, d_0)_2$ 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

$$E_a : y^2 + xy = x^3 + ax^2 + 1, \quad a \,/\, \mathbb{F}_2 \tag{4.1}$$

over a finite field $\mathbb{F}_2$. We identify $\}0, 1\langle (\leftarrow \mathbb{Z})$ with $\mathbb{F}_2$ via the natural map $f : \}0, 1\langle \Leftarrow \mathbb{F}_2,\ a \mapsto a \bmod 2$. For some cryptographic usage, we focus on the group of $\mathbb{F}_{2^m}$-rational points $E_a(\mathbb{F}_{2^m})$ 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 τ be the Frobenius map on E_a,

$$\tau : E_a(\mathbb{F}_{2^m}) \Leftarrow E_a(\mathbb{F}_{2^m}), \quad (x, y) \mapsto (x^2, y^2). \tag{4.2}$$

We can regard τ as a complex number which satisfies the following characteristic equation

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

[*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 $\tau = (\mu \mp \quad \overline{7})/2$, that is, the Koblitz curve has complex multiplication by τ [*1)]. Since the cost of the Frobenius map τ is cheaper than that of point doubling, and a scalar can be written as a τ-adic expansion, the Frobenius map allows for scalar multiplication without the need for point doubling [46].

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

$$e_i \eqslant 0 \text{ implies } e_{i+w-1} = \times\!\times\!\times = e_{i+1} = 0 \tag{4.4}$$

and $e_i \;/\; \mathcal{D}_w$ for all i, where $\mathcal{D}_w$ 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, $\mathcal{D}_w = \}0, \mp 1, \mp 3, \ldots, \mp(2^{w-1} \quad 1)\langle$ [*2)]. Solinas proved some desired properties [*3)] of the τ-NAF with respect to the Hamming weight, namely, the τ-NAF has the existence and uniqueness, and the non-zero density of the τ-NAF is asymptotically $1/3$ [72]. Subsequently, Avanzi, Heuberger, and Prodinger [4] have proven the minimality of the Hamming weight of the 2-τ-NAF (or τ-NAF [*4)]).

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

In order to take advantage of the efficiency of the τ-NAF, it is necessary that the τ-NAF has appropriate length. The length of the τ-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 non-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 $w \;/\; \}4, 5, 6\langle$ [41]. Similar results for the τ-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 τ-NAF. By using modular reduction in $\mathbb{Z}[\tau]$, we can reduce the length ℓ 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 τ-adic expansions with digit set $\}0, \mp 1\langle$, is not known yet. If the lower bound is quite small compared to the length of the τ-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 τ-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 τ-NAF. Secondly, we derive an explicit lower bound for the length of minimal Hamming weight τ-adic expansions based on the lemma. We also give a new proof of the minimality of the Hamming weight of the τ-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 τ-NAF, we classify a minimal length τ-adic expansion with minimal Hamming weight except for two special cases. The classification shows the following two facts. One is that the τ-NAF has *almost* minimal length among all τ-adic expansions of minimal Hamming weight with digit set $\}0, \mp 1\langle$. The other is that we can easily convert the τ-NAF into a minimal length τ-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 τ-NAF. Section 4.3 derives an explicit lower bound for the length of minimal Hamming weight τ-adic expansions. Section 4.4 gives the new proof of minimality of the Hamming weight of the τ-NAF on Koblitz curves. Section 4.5 classifies a minimal length τ-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 τ-adic expansion into the τ-NAF. The following lemma is useful to obtain such recursive formulas.

Lemma 4.1 *Let* ℓ / $\mathbb{N}$ *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 ($0 \geq i \geq \ell$ 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 / $\mathcal{D}$, we have $a_0 = 0$. We put $\alpha' := \prod_{i=0}^{\ell-2} a_{i+1}\tau^{i|}$. 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. □

The τ-adic expansion $\prod_{i=0}^{\ell-1}(b_i \quad c_i)\tau^i$ is not necessarily τ-adic expansion with $\mathcal{D}$, because $(b_i \quad c_i)$ / $\}0, \mp 1, \mp 2\langle$. 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 $\prod_{i=0}^{\ell-1}(b_i \quad c_i)\tau^i$ into τ-adic expansion $\prod_{i=0}^{\ell-1} a_i\tau^i$ (a_i / $\mathcal{D}$). For each i ($i = 0, 1, 2, ..., \ell$ 1), a_i's are obtained by the following recursive formulas:

$$a_i = (b_i \quad c_i) \quad \mu D^*_{i-1} + D^*_{i-2} + D_i, \tag{4.5}$$

where $D^*_{-1} = D^*_{-2} = 0$, and for all i,

$$D_i := \begin{cases} \left\lfloor \frac{(b_i - c_i) - \mu D^*_{i-1} + D^*_{i-2}}{2} \right\rfloor * 2 & \\ \qquad\qquad (\text{if } (b_i \quad c_i) \quad \mu D^*_{i-1} + D^*_{i-2} \sim 0), & \\ -\left\lfloor \frac{(b_i - c_i) - \mu D^*_{i-1} + D^*_{i-2}}{2} \right\rfloor * 2 \qquad (\text{otherwise}), & \end{cases} \tag{4.6}$$

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 $\}0, \mp 1\langle$. From Lemma 4.1, we have $a_i = 0$ for all i. In other words, for any α / $\mathbb{Z}[\tau]$ and any τ-adic expansion $\alpha = \prod_{i=0}^{\ell'-1} c_i\tau^i$ with digit set $\mathcal{D}$, we can compute the τ-NAF of α ($\alpha = \prod_{i=0}^{\ell-1} b_i\tau^i$)

using the recursive formulas (4.5), (4.6), and (4.7).

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

$$(b_{29}, \ldots, b_0)_\tau = (1,0,1,0,0,1,0,0,\bar{1},0,1,0,\bar{1},0,1,0,1,0,1,0,\bar{1},0,1,0,1,0,1,0,0,\bar{1})_\tau,$$
$$(c_{29}, \ldots, c_0)_\tau = (0,1,\bar{1},\bar{1},1,0,0,0,1,0,\bar{1},1,1,0,0,0,\bar{1},0,\bar{1},0,1,\bar{1},\bar{1},0,0,1,0,0,1,1)_\tau$$

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

$$\begin{aligned}
&(0,1,\bar{1},\bar{1},1,0,0,0,1,0,\bar{1},1,1,0,0,0,\bar{1},0,\bar{1},0,1,\bar{1},\bar{1},0,0,1,0,0,1,1)_\tau \\
&(\bar{1},1,\bar{2})_\tau \\
&(\bar{1},1,\bar{2},0)_\tau \\
&(1,\bar{1},2,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0)_\tau \\
&(\bar{1},1,\bar{2},0,0,0,0,0,0,0,0)_\tau \\
&(\bar{1},1,\bar{2},0,0,0,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(\bar{1},1,\bar{2},0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(\bar{1},1,\bar{2},0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(\bar{1},1,\bar{2},0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(\bar{1},1,\bar{2},0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
&(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
+)\quad &(1,\bar{1},2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)_\tau \\
\hline
&(1,0,1,0,0,1,0,0,\bar{1},0,1,0,\bar{1},0,1,0,1,0,1,0,\bar{1},0,1,0,1,0,1,0,0,\bar{1})_\tau
\end{aligned}$$

The lower bound and our new proof of the minimality of the Hamming weight of the τ-NAF are based on the following lemma.

Lemma 4.2 (Key Lemma)

Let $S := \}0, \mp 1\langle * \}0, \mp 1\langle * \mathbb{Z} * \mathbb{Z} * \mathbb{Z}$ *be the direct product of two copies of* $\}0, \mp 1\langle (\leftarrow \mathbb{Z})$ *and* $\mathbb{Z}^3$. *Let* $H_i := (b_i, c_i, D^*_{i-1}, D^*_{i-2}, D_i) \ / \ S$ *for* i $(0 \geq i \geq \ell \quad 1)$, *where* b_i, c_i, D^*_{i-1}, D^*_{i-2}, *and* D_i *satisfy Equation* (4.6) *and* (4.7). *Let* $A_1 := \}(\mu, 0, 1, 0, 0), (\ \mu, 0, \ \ 1, 0, 0)\langle$, $A_2 := \}(\mu, 0, 0, \ \ \mu, 0), (\ \mu, 0, 0, \mu, 0)\langle$, $A_3 := \}(\mu, 0, \ \ 1, 0, \ \ 2\mu), (\ \mu, 0, 1, 0, 2\mu)\langle$, $A_4 := \}(\mu, 0, 0, \mu, \ \ 2\mu), \ (\ \mu, 0, 0, \ \ \mu, 2\mu)\langle$ *be the subset of* S, *respectively.*

(1) $D_i = 0, \mp 2$ $(D^*_i = 0, \mp 1)$ *for all* i.

(2) *If* $c_{i+1} = 0, b_{i+1} \models 0$ *for some* $i \sim 0$, *then* $H_{i+1} \ / \ A_1 \lfloor A_2 \lfloor A_3 \lfloor A_4$.

(3) *If* $H_{i+1} \ / \ A_1 \lfloor A_3$, *then* $b_i = 0, c_i \models 0$. *If* $H_{i+1} \ / \ A_4$, *then it hold* $b_{i+2} = 0$ *and* $c_{i+2} \models 0$. *In particular, if* $H_i \ / \ A_4$, *then* $H_{i+2} \not/ \ A_1 \lfloor A_3$.

(4) *For* $i_0 \ / \ \}0, 1, \ldots, \ell \quad 1\langle$, *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* $(D_{j+1}, D_j) \models (0, 0)$ *for all* j $(\ \ 1 \geq j \geq i \quad 1)$, *then* $i \sim 2$ *and*

$$\rangle \begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix} [$$

satisfies one of the following two cases:

$$\rangle \begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix} [\ / \ \Gamma_1, \tag{4.8}$$

or

$$\rangle \begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix} [\ / \ \Gamma_2, \tag{4.9}$$

where

$$\Gamma_1 = \left\{\left(\begin{matrix} 0 & 0 & 1 \\ 0 & \mu & 1 \end{matrix}\right], \left(\begin{matrix} 0 & 0 & 1 \\ 0 & \mu & 1 \end{matrix}\right]\right.$$

$$\Gamma_2 = \left\{\left(\begin{matrix} \pm & 0 & 1 \\ \pm & \mu & 1 \end{matrix}\right], \left(\begin{matrix} \pm & 0 & 1 \\ \pm & \mu & 1 \end{matrix}\right], \left(\begin{matrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{matrix}\right], \left(\begin{matrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{matrix}\right]\right.$$

In particular, if

$$\left(\begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix}\right] / \left\}\left(\begin{matrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{matrix}\right], \left(\begin{matrix} 1 & 0 & 1 \\ 0 & \mu & 1 \end{matrix}\right]\right. \leftarrow \Gamma_2$$

holds, then $i \sim 3$, $H_2 \not\sim A_4$, *and*

$$\left(\begin{matrix} b_3 & b_2 & b_1 & b_0 \\ c_3 & c_2 & c_1 & c_0 \end{matrix}\right] \not\sim \Gamma_3, \tag{4.10}$$

where

$$\Gamma_3 = \left\{\left(\begin{matrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{matrix}\right], \left(\begin{matrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{matrix}\right], \right.$$
$$\left(\begin{matrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{matrix}\right], \left(\begin{matrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{matrix}\right]\right. \tag{4.11}$$

Proof

(1) Let us assume the contrary and seek a contradiction. Suppose that there exists $i \not\sim \{0, 1, 2, \ldots, \ell - 1\}$ such that i does not satisfy $D_i = 0, \mp 2$, and i_0 be the minimal such $i \not\sim \{0, 1, 2, \ldots, \ell - 1\}$. 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

$$\begin{aligned} & |b_{i_0} \quad c_{i_0} \quad \mu D_{i_0-1}^* + D_{i_0-2}^*| \\ \geq\ & |b_{i_0}| + |c_{i_0}| + |\mu D_{i_0-1}^*| + |D_{i_0-2}^*| \\ \geq\ & 1 + 1 + 1 + 1 = 4. \end{aligned}$$

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 $\backslash b_{i_0} \backslash, \backslash c_{i_0} \backslash \geq 1, \backslash D^*_{i_0-1} \backslash, \backslash D^*_{i_0-2} \backslash \geq 1$, we must have $b_{i_0} = 1$, $c_{i_0} = \quad 1$, $D^*_{i_0-1} = \quad \mu$, $D^*_{i_0-2} = 1$ in order to satisfy $0 = a_{i_0} = b_{i_0} \quad c_{i_0} \quad \mu D^*_{i_0-1} + D^*_{i_0-2} + D_{i_0}$. So $D_{i_0-1} = 2D^*_{i_0-1} = \quad 2\mu$. On the other hand, $(b_{\ell-1}, \ldots, b_1, b_0)_\tau$ is the τ-NAF, so $b_{i_0} = 1$ implies $b_{i_0-1} = 0$. Hence

$$\begin{aligned} 0 = a_{i_0-1} &= b_{i_0-1} \quad c_{i_0-1} \quad \mu D^*_{i_0-2} + D^*_{i_0-3} + D_{i_0-1} \\ &= \quad c_{i_0-1} \quad \mu * 1 + D^*_{i_0-3} \quad 2\mu \\ &= \quad c_{i_0-1} + D^*_{i_0-3} \quad 3\mu, \end{aligned}$$

we obtain $c_{i_0-1} = D^*_{i_0-3} \quad 3\mu$. However, $D^*_{i_0-3} \not/ \}0, \mp 1\langle = \}0, \mp\mu\langle$, we have $\backslash c_{i_0-1} \backslash = \backslash D^*_{i_0-3} \quad 3\mu \backslash \sim \backslash \big(D^*_{i_0-3} \big(\quad \backslash 3\mu \backslash \backslash \sim \backslash \big(D^*_{i_0-3} \big(\quad 3 \backslash \sim 2$. 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 $D_{i+1} \models 0$ separately.

Case 1. $D_{i+1} = 0$.

It is easy to see that $H_{i+1} \not/ A_1 \lfloor A_2$.

Case 2. $D_{i+1} \models 0$.

If the sign of b_{i+1} is same as that of D_{i+1}, we must have $\backslash b_{i+1} + D_{i+1} \backslash = 3$. 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 $H_{i+1} \not/ A_3 \lfloor A_4$.

Therefore, if $c_{i+1} = 0, b_{i+1} \models 0$ then $H_{i+1} \not/ A_1 \lfloor A_2 \lfloor A_3 \lfloor A_4$.

(3) First, we assume that $H_{i+1} \not/ A_1$. Since $(b_{\ell-1}, \ldots, b_1, b_0)_\tau$ is the τ-NAF and $b_{i+1} \models 0$, we have $b_i = 0$. We substitute $b_i = 0$ into $a_i = (b_i \quad c_i) \quad \mu D^*_{i-1} + D^*_{i-2} + D_i = 0$, we have $c_i = \quad \mu D^*_{i-1} + D^*_{i-2} + D_i$. Since $H_{i+1} \not/ A_1$ and $c_i = \quad \mu \pm 0 + D^*_{i-2} \mp 2 = D^*_{i-2} \mp 2 \models 0$, we have $c_i \models 0$.

Next, suppose that $H_{i+1} \not/ A_3$. Similar to the above case, since $c_i = \quad \mu \pm 0 + D^*_{i-2} \bigcirc 2 = D^*_{i-2} \bigcirc 2 \models 0$, we also have $b_i = 0, c_i \models 0$.

We assume that $H_{i+1} \not/ A_4$. Since $(b_{\ell-1}, \ldots, b_1, b_0)_\tau$ is the τ-NAF and $b_{i+1} \models 0$, we have $b_{i+2} = 0$. From $a_{i+2} = b_{i+2} \quad c_{i+2} \quad \mu D^*_{i+1} + D^*_i + D_{i+2} = 0$ and $c_{i+2} = \mp 1 + D_{i+2} \models 0$, we have $c_{i+2} \models 0$. Moreover, if $H_i \not/ A_4$, from $D_i = \bigcirc 2\mu$, we have $D^*_i = \bigcirc\mu$. Therefore it does not occur that $H_{i+2} \not/ A_1 \lfloor A_3$.

(4)

$$
\begin{aligned}
& \sum_{i=0}^{\sim i_0} b_i\tau^i = \sum_{i=0}^{\sim i_0} c_i\tau^i \\
\Rightarrow\infty \quad & \sum_{i=0}^{\sim i_0} (b_i \quad c_i)\tau^i = 0 \\
\Rightarrow\infty \quad & \sum_{i=0}^{\sim i_0} (b_i \quad c_i)\tau^i + \sum_{i=0}^{\sim i_0} (\tau^2 \quad \mu\tau + 2)D_i^*\tau^i = 0 \\
\Rightarrow\infty \quad & \sum_{i=0}^{\sim i_0} \}(b_i \quad c_i) \quad \mu D_{i-1}^* + D_{i-2}^* + D_i\langle\tau^i \\
& \qquad +(D_{i_0}^*\tau \quad \mu D_{i_0}^* + D_{i_0-1}^*)\tau^{i_0+1} = 0 \\
\Rightarrow\infty \quad & \sum_{i=0}^{\sim i_0} a_i\tau^i + (D_{i_0}^*\tau \quad \mu D_{i_0}^* + D_{i_0-1}^*)\tau^{i_0+1} = 0 \\
\Rightarrow\infty \quad & (D_{i_0}^*\tau \quad \mu D_{i_0}^* + D_{i_0-1}^*)\tau^{i_0+1} = 0 \\
\Rightarrow\infty \quad & D_{i_0}^*\tau + (\ \ \mu D_{i_0}^* + D_{i_0-1}^*) = 0 \\
\Rightarrow\infty \quad & D_{i_0-1} = 0, D_{i_0} = 0
\end{aligned}
$$

(5) Since $(D_0, D_{-1}) \vDash (0,0)$ and $D_{-1} = 0$, we must have $D_0 \vDash 0$. 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 $(b_{\ell-1}, \ldots, b_1, b_0)_\tau$ is the τ-NAF and $b_0 \vDash 0$, we have $b_1 = 0$. Hence from $b_1 \quad c_1 \quad \mu D_0^* + D_{-1}^* + D_1 = \ \ c_1 + \mu + D_1 = 0$, we have $(c_1, D_1) = (\ \ \mu, \ \ 2\mu)$ or $(\mu, 0)$.

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

By $b_2 \quad c_2 \quad \mu D_1^* + D_0^* + D_2 = b_2 \quad c_2 + D_2 = 0$, we have $(b_2, c_2, D_2) = (0,0,0)$, $(\mp 1, \mp 1, 0)$, or $(\mp\mu, \bigcirc\mu, \bigcirc 2\mu)$, where double signs are taken in the same order. Thus $\binom{b_2 \ b_1 \ b_0}{c_2 \ c_1 \ c_0} = \binom{0 \ 0 \ 1}{0 \ \mu \ 1}, \binom{1 \ 0 \ 1}{1 \ \mu \ 1}, \binom{1 \ 0 \ 1}{1 \ \mu \ 1}, \binom{\mu \ 0 \ 1}{\mu \ \mu \ 1}$, or $\binom{\mu \ 0 \ 1}{\mu \ \mu \ 1}$.

Case 2. $(c_1, D_1) = (\mu, 0)$.

By $b_2 \quad c_2 \quad \mu D_1^* + D_0^* + D_2 = b_2 \quad c_2 + D_2 \quad 1 = 0$, we have $(b_2, c_2, D_2) = (0, 1, 2)$, $(0, \quad 1, 0)$, $(\quad 1, 0, 2)$, or $(1, 0, 0)$. Thus, in the cases of $(b_2, c_2, D_2) = (0, 1, 2)$ or $(0, \quad 1, 0)$, we have

$$\left(\begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix}\right[= \left(\begin{matrix} 0 & 0 & 1 \\ 1 & \mu & 1 \end{matrix}\right[\text{ or } \left(\begin{matrix} 0 & 0 & 1 \\ 1 & \mu & 1 \end{matrix}\right[.$$

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

$$\left(\begin{matrix} b_3 & b_2 & b_1 & b_0 \\ c_3 & c_2 & c_1 & c_0 \end{matrix}\right[= \left(\begin{matrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{matrix}\right[\text{ or } \left(\begin{matrix} 0 & 1 & 0 & 1 \\ \mu & 0 & \mu & 1 \end{matrix}\right[.$$

Moreover, if $(c_2, D_3) = (\mu, 2\mu)$, then $H_2 = (\quad 1, 0, 0, \quad 1, 2) \ / \ A_4$, and if $(c_2, D_3) = (\quad \mu, 0)$, then $H_2 = (\quad \mu, 0, 0, \quad \mu, 2\mu) \ / \ A_4$. 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

$$\left(\begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix}\right[\ / \ \Gamma_1 \ \lfloor \ \Gamma_2,$$

or

$$\left(\begin{matrix} b_3 & b_2 & b_1 & b_0 \\ c_3 & c_2 & c_1 & c_0 \end{matrix}\right[\ / \ \Gamma_3.$$

In particular,

$$\left(\begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix}\right[= \left(\begin{matrix} 0 & 0 & \pm \\ 0 & \pm & \pm \end{matrix}\right[, \left(\begin{matrix} \pm & 0 & \pm \\ \pm & \pm & \pm \end{matrix}\right[, \left(\begin{matrix} 0 & 0 & \pm \\ \pm & \pm & \pm \end{matrix}\right[$$

or

$$\left(\begin{matrix} b_3 & b_2 & b_1 & b_0 \\ c_3 & c_2 & c_1 & c_0 \end{matrix}\right[= \left(\begin{matrix} 0 & \pm & 0 & \pm \\ \pm & 0 & \pm & \pm \end{matrix}\right[.$$

It is easy to see that $i \sim 2$ when

$$\left(\begin{matrix} b_2 & b_1 & b_0 \\ c_2 & c_1 & c_0 \end{matrix}\right[\ / \ \Gamma_1 \ \lfloor \ \Gamma_2,$$

and $i \sim 3$ when

$$\left.\begin{matrix} b_3 & b_2 & b_1 & b_0 \\ c_3 & c_2 & c_1 & c_0 \end{matrix}\right] \not/ \Gamma_3.$$

□

4.3 Lower Bound for the Length

This section derives an explicit lower bound for the length of minimal Hamming weight τ-adic expansions. From the definition of $\ell_{\min}$, the following upper bound for $\ell_{\min}$ is trivially true for all $\alpha \not/ \mathbb{Z}[\tau]$:

$$\ell_{\min}(\alpha) \geq \ell_{\tau\text{-NAF}}(\alpha).$$

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

Theorem 4.1 (Lower Bound for $\ell_{\min}(\alpha)$)
Suppose that $\ell' < \ell$. Then for any $\alpha \not/ \mathbb{Z}[\tau]$,

$$\ell_{\tau\text{-NAF}}(\alpha) \quad 3 \ (= \ell \quad 3) \geq \ell'. \tag{4.12}$$

In particular,

$$\ell_{\tau\text{-NAF}}(\alpha) \quad 3 \geq \ell_{\min}(\alpha). \tag{4.13}$$

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 $b_{\ell-1} \models 0$ and $c_{\ell-1} = 0$. From Lemma 4.2 (2), it satisfies that $H_{\ell-1} \not/ A_1 \lfloor A_2 \lfloor A_3 \lfloor A_4$. 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 $i \sim \ell \quad 2$. It follows that $H_{\ell-1} \not/ A_1 \lfloor A_3 \lfloor A_4$. We only deal with the case of $H_{\ell-1} \not/ A_2$. There are two cases to consider, $H_{\ell-1} = (\mu, 0, 0, \quad \mu, 0)$ and $H_{\ell-1} = (\quad \mu, 0, 0, \mu, 0)$. Without loss of generality, we may assume that $H_{\ell-1} = (\mu, 0, 0, \quad \mu, 0)$ because the latter may

be treated in exactly the same way. By $b_{\ell-1} \models 0$, it satisfies $b_{\ell-2} = 0$. From

$$\begin{aligned} a_{\ell-2} &= (b_{\ell-2} \quad c_{\ell-2}) \quad \mu D^*_{\ell-3} + D^*_{\ell-4} + D_{\ell-2} \\ &= \quad c_{\ell-2} + 1 + D^*_{\ell-4} \\ &= 0, \end{aligned}$$

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

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

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

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

It holds that

$$\begin{aligned} a_{\ell-3} &= (b_{\ell-3} \quad c_{\ell-3}) \quad \mu D^*_{\ell-4} + D^*_{\ell-5} + D_{\ell-3} \\ &= (b_{\ell-3} \quad c_{\ell-3}) + \mu + D^*_{\ell-5} \quad 2\mu \\ &= (b_{\ell-3} \quad c_{\ell-3}) + D^*_{\ell-5} \quad \mu \\ &= 0. \end{aligned}$$

So

$$\begin{aligned} (b_{\ell-3}, c_{\ell-3}, D^*_{\ell-5}) = (0, 0, \mu), (0, \quad \mu, 0), (\mu, 0, 0), \\ (\mu, \mu, \mu), (\mu, \quad \mu, \quad \mu), \text{ or } (\quad \mu, \quad \mu, \mu). \end{aligned}$$

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

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

$$\begin{aligned} a_{\ell-4} &= (b_{\ell-4} \quad c_{\ell-4}) \quad \mu D^*_{\ell-5} + D^*_{\ell-6} + D_{\ell-4} \\ &= (b_{\ell-4} \quad c_{\ell-4}) + D^*_{\ell-6} \quad 3 \\ &= 0, \end{aligned}$$

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

$$\begin{aligned} a_{\ell-4} &= (b_{\ell-4} \quad c_{\ell-4}) \quad \mu D^*_{\ell-5} + D^*_{\ell-6} + D_{\ell-4} \\ &= \quad c_{\ell-4} + D^*_{\ell-6} \quad 2 \\ &= 0. \end{aligned}$$

Hence we obtain $(c_{\ell-4}, D^*_{\ell-6}) = (\quad 1, 1)$. Thus $\ell' = \ell \quad 3$.

Combining case 1 and case 2, we obtain Inequality (4.12). □

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

4.4 Another Proof of the Minimality

In this section, we give a new proof of minimality of the Hamming weight of the τ-NAF on Koblitz curves.

4.4.1 The Main Idea of Our New Proof

The minimality of the Hamming weight of the τ-NAF on Koblitz curves was first proved by Avanzi, Heuberger, and Prodinger [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 τ-NAF from any τ-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_α into T_α. Notice that if it is possible to construct an injection map from S_α to T_α for any α and any τ-adic expansion of α, then the Hamming weight of the τ-NAF of α is always smaller than that of the τ-adic expansion, that is, the τ-NAF minimizes the Hamming weight with digit set $\}0, \mp 1\langle$.

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

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, β be any element of $\mathbb{Z}_{>0}$. We denote by $\beta = \sum_{i=0}^{\ell-1} g_i r^i$ $(g_i \ / \ \mathcal{D}_{\mathrm{G}})$ the GNAF of β, where $\mathcal{D}_{\mathrm{G}} = \}0, \mp 1, \ldots, \mp(r \quad 1)\langle$. Let $\beta = \sum_{i=0}^{\ell'-1} h_i r^i$ $(h_i \ / \ \mathcal{D}_{\mathrm{G}})$ be any r-adic expansion of β. If $\ell > \ell'$, then put $h_{\ell'} = h_{\ell'+1} = \times\times\times = h_{\ell-1} = 0$. Otherwise, put $g_\ell = g_{\ell+1} = \times\times\times = g_{\ell'-1} = 0$. Furthermore, replace $\max\}\ell, \ell'\langle$ by ℓ if necessary. We put $S_\beta := \}i \ / \ \}0, 1, \ldots, \ell \quad 1\langle \ \backslash g_i \vDash 0\langle$, and $T_\beta := \}i \ / \ \}0, 1, \ldots, \ell \quad 1\langle \ \backslash h_i \vDash 0\langle$.

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 $h_i \vDash 0$ and $g_i = 0$.*

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

$$\begin{array}{rccccl} \varphi_\beta : & S_\beta & \Leftarrow & T_\beta & & \\ & \diagdown & & \diagdown & & \\ & i & \not\Leftarrow & i & & (g_i \vDash 0,\ h_i \vDash 0), \\ & i & \not\Leftarrow & i & 1 & (g_i \vDash 0,\ h_i = 0). \end{array} \tag{4.14}$$

We can see that Lemma 4.2 is analogous result for τ-adic expansion.

Example 4.2 *We put $\beta = 686204583$.*

$$(b_{29}, \ldots, b_0)_2 = (1,0,1,0,0,1,0,0,\bar{1},0,1,0,\bar{1},0,1,0,1,0,1,0,\bar{1},0,1,0,1,0,1,0,0,\bar{1})_2,$$
$$(c_{29}, \ldots, c_0)_2 = (1,0,1,0,0,0,1,1,1,0,0,1,1,0,1,0,1,0,0,1,1,0,1,0,1,0,0,1,1,1)_2$$

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

$T_\beta = \}0, 1, 2, 5, 7, 9, 10, 13, 15, 17, 18, 21, 22, 23, 27, 29\langle$. *We construct a map*

$\varphi_\beta : S_\beta \Leftarrow T_\beta$ *using* (4.14):

$$
\begin{array}{rccc}
\varphi_\beta : & S_\beta & \Leftarrow & T_\beta \\
& \diagdown & & \diagdown \\
& 0 & \not\Leftarrow & 0 \\
& 3 & \not\Leftarrow & 2 \\
& 5 & \not\Leftarrow & 5 \\
& 7 & \not\Leftarrow & 7 \\
& 9 & \not\Leftarrow & 9 \\
& 11 & \not\Leftarrow & 10 \\
& 13 & \not\Leftarrow & 13 \\
& 15 & \not\Leftarrow & 15 \\
& 17 & \not\Leftarrow & 17 \\
& 19 & \not\Leftarrow & 18 \\
& 21 & \not\Leftarrow & 21 \\
& 24 & \not\Leftarrow & 23 \\
& 27 & \not\Leftarrow & 27 \\
& 29 & \not\Leftarrow & 29.
\end{array}
$$

Thus the map φ_β 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 τ-NAF on Koblitz curves.

Our New Proof of the Minimality.

With notation as above, we directly construct an injection map $\varphi_\alpha : S_\alpha \Leftarrow T_\alpha$ for each case.

Case 1. $H_i \not/ A_2$ for all i $(0 \geq i \geq \ell \quad 1)$.

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

$$\begin{array}{rccl} \varphi_\alpha : & S_\alpha & \Leftarrow & T_\alpha \\ & \diagdown & & \diagdown \\ & i & \not\Leftarrow & i \quad (b_i \models 0,\ c_i \models 0), \\ & i & \not\Leftarrow & i \quad 1 \quad (H_i \ / \ A_1 \lfloor\ A_3), \\ & i & \not\Leftarrow & i+1 \quad (H_i \ / \ A_4). \end{array} \tag{4.15}$$

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

Case 2. $H_i \ / \ A_2$ for some i $(0 \geq i \geq \ell \quad 1)$.

Let $\}i_1, i_2, \ldots, i_k\langle$ 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 $i_0 := \quad 1$ 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 $\prod_{i=i_j+1}^{i_{j+1}} b_i \tau^i = \prod_{i=i_j+1}^{i_{j+1}} c_i \tau^i$ for $0 \geq j \geq k \quad 1$.

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

$$M_j := \}n \ / \ \mathbb{Z} \setminus i_{j-1} \geq n \geq i_j \quad 1, (D_n, D_{n-1}) = (0,0)\langle. \tag{4.16}$$

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 $(D_n, D_{n-1}) \models (0,0)$ for $m_j + 1 \geq n \geq i_j \quad 1$. By Lemma 4.2 (5), we have $(b_{m_j+2}, c_{m_j+2}) = (0, \pm)$. Moreover, from Lemma 4.2 (5), if

$$\left)\begin{matrix} b_{m_j+4} & b_{m_j+3} & b_{m_j+2} & b_{m_j+1} \\ c_{m_j+4} & c_{m_j+3} & c_{m_j+2} & c_{m_j+1} \end{matrix}\right[\ / \ \Gamma_3,$$

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

$$i_0 \geq m_1 \geq i_1 \geq \gtrsim m_j \geq i_j \geq \gtrsim m_k \geq i_k. \tag{4.17}$$

Furthermore, by Lemma 4.2 (5), if

$$\Big)\begin{matrix} b_{m_j+3} & b_{m_j+2} & b_{m_j+1} \\ c_{m_j+3} & c_{m_j+2} & c_{m_j+1} \end{matrix}\Big[\ / \ \Gamma_1 \ \lfloor \ \Gamma_2,$$

then $i_j \sim m_j + 3$, and if

$$\Big)\begin{matrix} b_{m_j+4} & b_{m_j+3} & b_{m_j+2} & b_{m_j+1} \\ c_{m_j+4} & c_{m_j+3} & c_{m_j+2} & c_{m_j+1} \end{matrix}\Big[\ / \ \Gamma_3,$$

then $i_j \sim m_j + 4$. We define a map $\varphi_\alpha : S_\alpha \Leftarrow T_\alpha$ as follows.

$$\begin{array}{rccccl} \varphi_\alpha : & S_\alpha & \Leftarrow & T_\alpha & & \\ & \diagdown & & \diagdown & & \\ & i & \not\Leftarrow & i & & (b_i \eqcirc 0, \ c_i \eqcirc 0), \\ & i & \not\Leftarrow & i & 1 & (H_i \ / \ A_1 \ \lfloor \ A_3), \\ & i & \not\Leftarrow & i+1 & & (H_i \ / \ A_4), \\ & i_j & \not\Leftarrow & m_j+2 & & (H_{i_j} \ / \ A_2). \end{array} \tag{4.18}$$

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 $i \not/ \ \}i_1, i_2, \ldots, i_k\langle$, 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* $\mu = 1$, $\beta = \quad 3597 \quad 11571\tau$.

$$(b_{29}, \ldots, b_0)_\tau = (1,0,1,0,0,1,0,0,\bar{1},0,1,0,\bar{1},0,1,0,1,0,1,0,\bar{1},0,1,0,1,0,1,0,0,\bar{1})_\tau,$$
$$(c_{29}, \ldots, c_0)_\tau = (0,1,\bar{1},\bar{1},1,0,0,0,1,0,\bar{1},1,1,0,0,0,\bar{1},0,\bar{1},0,1,\bar{1},\bar{1},0,0,1,0,0,1,1)_\tau$$

are the τ*-NAF of* α*, and a* τ*-adic expansion of* α*, respectively.*

With the above notation, $S_\alpha = \}0,3,5,7,9,11,13,15,17,19,21,24,27,29\langle$ *and* $T_\alpha = \}0,1,4,7,8,9,11,13,17,18,19,21,25,26,27,28\langle$. *We construct a map*

$\varphi_\alpha : S_\alpha \Leftarrow T_\alpha$ *using* (4.18):

$$\begin{array}{rccc} \varphi_\alpha : & S_\alpha & \Leftarrow & T_\alpha \\ & \diagdown & & \diagdown \\ & 0 & \not\Leftarrow & 0 \\ & 3 & \not\Leftarrow & 4 \\ & 5 & \not\Leftarrow & 1 \\ & 7 & \not\Leftarrow & 7 \\ & 9 & \not\Leftarrow & 9 \\ & 11 & \not\Leftarrow & 11 \\ & 13 & \not\Leftarrow & 13 \\ & 15 & \not\Leftarrow & 8 \\ & 17 & \not\Leftarrow & 17 \\ & 19 & \not\Leftarrow & 19 \\ & 21 & \not\Leftarrow & 21 \\ & 24 & \not\Leftarrow & 25 \\ & 27 & \not\Leftarrow & 27 \\ & 29 & \not\Leftarrow & 28. \end{array}$$

Thus the map φ_α is injective.

4.5 τ-adic Minimal Length Form and Its Cryptographic Application

4.5.1 τ-adic Minimal Length Form

This section classifies a minimal length τ-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 τ-adic expansion. Corollary 4.1 shows that we can convert τ-NAF into a minimal length τ-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 τ-NAF.

Corollary 4.1 (τ-adic Minimal Length Expansion)

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

Case 1. $\ell < 6$.

If $(e_{\ell-1}, \ldots, e_0)_\tau$ is equal to one of the τ-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 τ-NAF into the τ-MLF ($\ell < 6$)

$(e_{\ell-1}, \ldots, e_0)_\tau$	$(e'_{\ell'-1}, \ldots, e'_0)_\tau$	ℓ	ℓ'
$(\mp\mu, 0, 0, \mp 1)_\tau$	$(\bigcirc\mu, \bigcirc 1)_\tau$	4	2
$(\mp\mu, 0, 0, \mp 1, 0)_\tau$	$(\bigcirc\mu, \bigcirc 1, 0)_\tau$	5	3
$(\mp\mu, 0, \mp\mu)_\tau$	$(\mp 1, \bigcirc\mu)_\tau$	3	2
$(\mp\mu, 0, \mp\mu, 0)_\tau$	$(\mp 1, \bigcirc\mu, 0)_\tau$	4	3
$(\mp\mu, 0, \mp\mu, 0, 0)_\tau$	$(\mp 1, \bigcirc\mu, 0, 0)_\tau$	5	4
$(\mp\mu, 0, \mp\mu, 0, \mp\mu)_\tau$	$(\bigcirc\mu, \bigcirc 1, \bigcirc\mu)_\tau$	5	3
$(\mp\mu, 0, \mp\mu, 0, \bigcirc\mu)_\tau$	$(\mp 1, \bigcirc\mu, 0, \bigcirc\mu)_\tau$	5	4

Case 2. $\ell = 6$.

We convert $(e_{\ell-1}, \ldots, e_{\ell-6})_\tau$ into $(e'_{\ell'-1}, \ldots, e'_{\ell-6})_\tau$ using Table 4.2 *(double signs are taken in the same order).*

Case 3. $\ell \sim 7$.

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

(ii) $(e_{\ell-1}, \ldots, e_{\ell-7})_\tau \neq (\mp\mu, 0, \mp\mu, 0, 0, 0, \bigcirc\mu)_\tau$ *and* $(\mp\mu, 0, \mp\mu, 0, \bigcirc\mu, 0, \bigcirc\mu)_\tau$. $e_i = e'_i$ *for all* $i \geq \ell \quad 7$ *and we convert* $(e_{\ell-1}, \ldots, e_{\ell-6})_\tau$ *into* $(e'_{\ell'-1}, \ldots, e'_{\ell-6})_\tau$ *using* Table 4.2.

Then, except for the cases that $(e_{\ell-1}, \ldots, e_{\ell-6})_\tau = (\mu, 0, \mu, 0, \mu, 0)_\tau$ *and* $(e_{\ell-1}, \ldots, e_{\ell-6})_\tau = (\quad \mu, 0, \quad \mu, 0, \quad \mu, 0)_\tau$, *the τ-adic expansion $d = \prod_{i=0}^{\ell'-1} e'_i\tau^i$ is a minimal*

Table 4.2: Conversion of the τ-NAF into the τ-MLF ($\ell \sim 6$)

$(e_{\ell-1}, \ldots, e_{\ell-6})_\tau$	$(e'_{\ell'-1}, \ldots, e'_{\ell-6})_\tau$	ℓ'
$(\mp\mu, 0, 0, 0, 0, 0)_\tau$	$(\mp\mu, 0, 0, 0, 0, 0)_\tau$	ℓ
$(\mp\mu, 0, 0, 0, 0, \mp\mu)_\tau$	$(\mp\mu, 0, 0, 0, 0, \mp\mu)_\tau$	ℓ
$(\mp\mu, 0, 0, 0, 0, \bigcirc\!\!\!\!/\,\mu)_\tau$	$(\mp\mu, 0, 0, 0, 0, \bigcirc\!\!\!\!/\,\mu)_\tau$	ℓ
$(\mp\mu, 0, 0, 0, \mp\mu, 0)_\tau$	$(\mp\mu, 0, 0, 0, \mp\mu, 0)_\tau$	ℓ
$(\mp\mu, 0, 0, 0, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	$(\mp\mu, 0, 0, 0, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	ℓ
$(\mp\mu, 0, 0, \mp 1, 0, 0)_\tau$	$(\bigcirc\!\!\!\!/\,\mu, \bigcirc\!\!\!\!/\,1, 0, 0)_\tau$	$\ell - 2$
$(\mp\mu, 0, 0, \mp 1, 0, \mp 1)_\tau$	$(\bigcirc\!\!\!\!/\,\mu, \bigcirc\!\!\!\!/\,1, 0, \mp 1)_\tau$	$\ell - 2$
$(\mp\mu, 0, 0, \mp 1, 0, \bigcirc\!\!\!\!/\,1)_\tau$	$(\bigcirc\!\!\!\!/\,1, \mp\mu, \mp 1)_\tau$	$\ell - 3$
$(\mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1, 0, 0)_\tau$	$(\mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1, 0, 0)_\tau$	ℓ
$(\mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1, 0, \mp 1)_\tau$	$(\mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1, 0, \mp 1)_\tau$	ℓ
$(\mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1, 0, \bigcirc\!\!\!\!/\,1)_\tau$	$(\mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1, 0, \bigcirc\!\!\!\!/\,1)_\tau$	ℓ
$(\mp\mu, 0, \mp\mu, 0, 0, 0)_\tau$	$(\mp 1, \bigcirc\!\!\!\!/\,\mu, 0, 0, 0)_\tau$	$\ell - 1$
$(\mp\mu, 0, \mp\mu, 0, 0, \mp 1)_\tau$	$(\mp 1, \bigcirc\!\!\!\!/\,\mu, 0, 0, \mp 1)_\tau$	$\ell - 1$
$(\mp\mu, 0, \mp\mu, 0, 0, \bigcirc\!\!\!\!/\,1)_\tau$	$(\bigcirc\!\!\!\!/\,1, \bigcirc\!\!\!\!/\,\mu, \mp 1)_\tau$	$\ell - 3$
$(\mp\mu, 0, \mp\mu, 0, \mp\mu, 0)_\tau$	$(\bigcirc\!\!\!\!/\,\mu, \bigcirc\!\!\!\!/\,1, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	$\ell - 2$
$(\mp\mu, 0, \mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	$(\mp 1, \bigcirc\!\!\!\!/\,\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	$\ell - 1$
$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, 0, 0)_\tau$	$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, 0, 0)_\tau$	ℓ
$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, 0, \mp\mu)_\tau$	$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, 0, \mp\mu)_\tau$	ℓ
$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, 0, \bigcirc\!\!\!\!/\,\mu)_\tau$	$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, 0, \bigcirc\!\!\!\!/\,\mu)_\tau$	ℓ
$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, \mp\mu, 0)_\tau$	$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, \mp\mu, 0)_\tau$	ℓ
$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	$(\mp\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0, \bigcirc\!\!\!\!/\,\mu, 0)_\tau$	ℓ

length τ-adic expansion with minimal Hamming weight. We call the τ-adic expansion $\prod_{i=0}^{\ell'-1} e'_i \tau^i$ ($e'_i \not\equiv \mathcal{D}$, $e'_{\ell'-1} \eqsim 0$) τ-adic minimal length form (τ-MLF for short).

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

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

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

4.5.2 Cryptographic Application of τ-MLF

Suppose that $P \ / \ E_a(\mathbb{F}_{2^m})$ 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, τ-MLF has not only the minimal Hamming weight, but also the minimal length among all τ-adic expansions of minimal Hamming weight with digit set $\}0, \mp 1\langle$. In order to utilize τ-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 τ-NAF method in the scalar multiplication step. However, by using τ-MLF in the scalar multiplication step, one can reduce the number of Frobenius map. Namely, scalar multiplication on Koblitz curve using τ-MLF method is slightly faster than the original τ-NAF method. Thus τ-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 τ-adic expansions. We also gave a new proof of the minimality of the Hamming weight of the τ-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 τ-NAF, we classified a minimal length τ-adic expansion with minimal Hamming weight except for two special cases. The classification shows that the τ-NAF has *almost* minimal length among all τ-adic expansions of minimal

Hamming weight and we can easily convert the τ-NAF into a minimal length τ-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 threads. So one might verify the signatures immediately after the storage

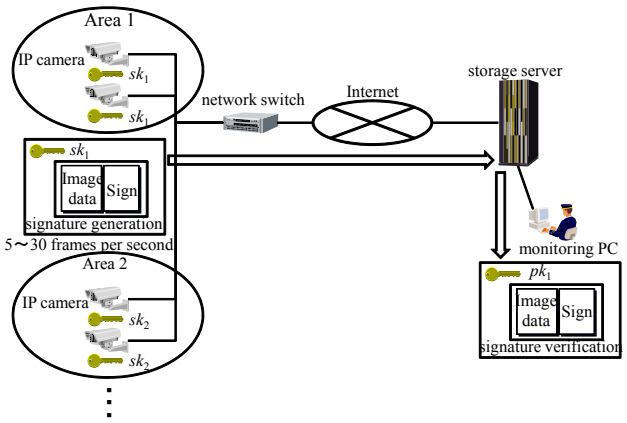

Figure 5.1: A typical IP camera surveillance system

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 sigantures 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 $\mathbb{G}$ be a cyclic group of prime order n, and g a generator of $\mathbb{G}$. For $a \in \mathbb{G}$, $\mathrm{ord}_{\mathbb{G}}(a)$ is the order of a, namely, $\mathrm{ord}_{\mathbb{G}}(a)$ is the smallest non-negative integer x that holds $a^x = 1_G$, where 1_G is the identity element in the group $\mathbb{G}$. $X = \{(x_1, y_1), \cdots, (x_N, y_N)\} \subset (x_i \in \mathbb{Z}_n, y_i \in \mathbb{G})$ is a batch instance which we would like to check whether or not $y_i = g^{x_i}$ for all $i = 1, \cdots, N$. We define a Boolean relation $R(\not\in)$ on the set $\{(x, y) \setminus x \in \mathbb{Z}_n, y \in \mathbb{G}\}$ 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 $X = \{(x_1, y_1), \cdots, (x_N, y_N)\} \subset (x_i \in \mathbb{Z}_n, y_i \in \mathbb{G})$ 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_{bv} 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 $\mathcal{V}$ outputs 1 is at most $2^{-k_{\mathrm{bv}}}$. 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 $g^{\sum_{i=1}^{N} s_i x_i} = \sum_{i=1}^{N} y_i^{s_i}$, 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_{bv} times in order to lower the error probability of the random subset test to $2^{-k_{\mathrm{bv}}}$. The small exponents test is an extension of the atomic random subset test. The small exponents test checks whether $g^{\sum_{i=1}^{N} s_i x_i} = \sum_{i=1}^{N} y_i^{s_i}$, where each s_i is randomly chosen from the set $\{0,1\}^{k_{\mathrm{bv}}}$.

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 $\mathrm{ScMul}_{E,P}(x, Q)$ as follows: $\mathrm{ScMul}_{E,P}(x, Q) := 1$ if $xP = Q$, otherwise $\mathrm{ScMul}_{E,P}(x, Q) := 0$.

We prepare some notations to explain the complex exponent test. We define

$$S_1(\ell_1, k_1, q) \stackrel{\mathrm{def}}{=} \left\{ d = \sum_{i=0}^{\ell_1 - 1} a_i \phi^i \;\middle|\; a_i \in \mathbb{Z},\ |a_i| \geq q \quad 1,\ a_{i+1} a_i = 0,\ wt(d) \geq k_1 \right\},$$

$$S_2(\ell_2, k_2, q) \stackrel{\text{def}}{=} \left\{ d = \sum_{i=0}^{\ell_2 - 1} a_i \phi^i \;\middle|\; a_i \in \mathbb{Z},\ |a_i| < q^2/2, \right.$$
$$\left. a_i = 0 \Longrightarrow q \nmid a_i,\ a_{i+1} a_i = 0,\ wt(d) \geq k_2 \right\},$$

for an elliptic curve E defined over $\mathbb{F}_q$. For all $P \in E(\mathbb{F}_{q^m})$, we always have $(\phi^m - 1)P = \mathcal{O}$, the point at infinity. So we sometimes identify ϕ^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 ℓ_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 ℓ_j is chosen appropriately.

Theorem 5.1 ([18, Theorem 3]) *For $j = 1, 2$ and $6 \geq \ell_j \geq (\log_2 n - 2\log_2 M_j - \log_2 30)/\log_2 q$, each endomorphism on G as an element in $S_j(\ell_j, k_j, q)$ is different from each other, where $M_1 = 2(q - 1)$, $M_2 = 2\lfloor (q^2 - 1)/2 \rceil$, 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]) *$\#S_1(\ell_1, k_1, q) = \prod_{i=0}^{k_1} \binom{\ell_1 + 1 - i}{i} (2q - 2)^i$ and $\#S_2(\ell_2, k_2, q) = \prod_{i=0}^{k_2} \binom{\ell_2 + 1 - i}{i} (q^2 - q)^i$.*

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

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

or not for randomly chosen $s_i \in S_j(\ell_j, k_j, q)$ $(j = 1, 2)$. The naive test verifies whether $x_i P = Q_i$ or not for each i $(i \in \{1, \ldots, N\})$, 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

$$k_1 N + 2(q \quad 2)^{\mid}\mathcal{A} + (q \quad 1)\ell_1\mathcal{F} + (1 \text{ scalar multiplication}), \tag{5.2}$$

$$\begin{aligned} k_2 N + 2(\rfloor(q^2 \quad 1)/2\lceil + \rfloor(q \quad 1)/2\lceil)^{\mid}\mathcal{A} \\ + (\rfloor(q^2 \quad 1)/2\lceil + \rfloor(q \quad 1)/2\lceil)\ell_2\mathcal{F} + (1 \text{ scalar multiplication}), \end{aligned} \tag{5.3}$$

respectively. If $\}(x_1, Q_1), \ldots, (x_N, Q_N)\langle$ is a correct batch instance, Equation (5.1) is always true. The probability that the complex exponent test accepts an incorrect batch instance $\}(x_1, Q_1), \ldots, (x_N, Q_N)\langle$ is at most $2^{-k_{\mathrm{bv}}}$ $(j = 1, 2)$ ([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 τ-adic non-adjacent form* (cf. [37], [36], [72] for the explanation of τ-NAF and w-τ-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 $E_{p,b}/\mathbb{F}_p : y^2 = x^3 \quad b$ [28, 47, 64, 69]) *Let p be a prime such that $p \leq 1 \bmod 3$, and $E_{p,b} : y^2 = x^3 \quad b$ be the elliptic curve defined over $\mathbb{F}_p$. Suppose that $\beta \,/\, \mathbb{F}_p$ is an element of order 3 (i.e., $\beta^3 = 1 \,/\, \mathbb{F}_p$). Then the curve $E_{p,b}$ has the automorphism $\omega : E_{p,b} \Leftarrow E_{p,b}$, $(x, y) \mapsto (\beta x, y)$. We can regard ω as a complex number which satisfies the equation $\omega^2 + \omega + 1 = 0$, namely, $\omega = (\quad 1 \mp \quad \overline{3})/2$.*

Proposition 5.2 (Automorphism of $E_{p,a}/\mathbb{F}_p : y^2 = x^3 + ax$ [28, 47, 64, 69]) *Let p be a prime such that $p \leq 1 \bmod 4$, and $E_{p,a} : y^2 = x^3 + ax$ be the elliptic curve defined over $\mathbb{F}_p$. Suppose that $\alpha \,/\, \mathbb{F}_p$ is an element of order 4 (i.e., $\alpha^4 = 1 \,/\, \mathbb{F}_p$). Then the curve $E_{p,a}$ has the automorphism $\lambda : E_{p,a} \Leftarrow E_{p,a}$, $(x, y) \mapsto (\quad x, \alpha y)$. We can regard λ as a complex number which satisfies the equation $\lambda^2 + 1 = 0$, namely, $\lambda = \mp \quad \overline{1}$.*

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 $\mathbb{F}_q$, we regard an endomorphism ϕ as a complex number (See [69] for detail).

Proposition 5.3 ($D_{\mathbf{E}}$-Frobenius expansion [64, Theorem 2]) *Let $E_{7,b} : y^2 = x^3 \quad b$ be the elliptic curve defined over $\mathbb{F}_7$, $\beta \,/\, \mathbb{F}_7$ be an element of order 3, $\omega : E_{7,b} \Leftarrow E_{7,b}$, $(x, y) \mapsto (\beta x, y)$ be the automorphism of $E_{7,b}$, and ϕ be the 7th-power Frobenius map on $E_{7,b}$. For any $d \,/\, \mathbb{Z}[\omega]$, we can write $d = \sum_{i=0}^{\ell-1} c_i \phi^i$, where $c_i \,/\, D_E := \{0, \mp 1, \mp\omega, \mp\omega^2\}$, $\ell \geq \lceil 2 \log_7 \quad \overline{N_{\mathbb{Z}[\omega]/\mathbb{Z}}(d)} \rceil + 1$.*

Proposition 5.4 ($D_{\mathbf{G}}$-Frobenius expansion [64, Theorem 1]) *Let $E_{5,a} : y^2 = x^3 + ax$ be the elliptic curve defined over $\mathbb{F}_5$, $\alpha \,/\, \mathbb{F}_5$ be an element of order 4, $\lambda : E_{5,a} \Leftarrow E_{5,a}$, $(x, y) \mapsto (\quad x, \alpha y)$ be the automorphism of $E_{5,a}$, and ϕ be the 5th-power Frobenius map on $E_{5,a}$. For any $d \,/\, \mathbb{Z}[\quad \overline{1}]$, we can write $d = \sum_{i=0}^{\ell-1} c_i \phi^i$, where $c_i \,/\, D_G := \{0, \mp 1, \mp\lambda\}$, $\ell \geq \lceil 2 \log_5 \quad \overline{N_{\mathbb{Z}[\lambda]/\mathbb{Z}}(d)} \rceil + 1$.*

We define

$$S_{\mathrm{E}}(\ell_\omega, k_\omega, q) := \left\{ d = \sum_{i=0}^{\ell_\omega - 1} a_i \phi^i \;\middle/\; a_i \in D_{\mathrm{E}},\ wt(d) \geq k_\omega \right\}, \tag{5.4}$$

for an elliptic curve $E_{7,b}$, and

$$S_{\mathrm{G}}(\ell_\lambda, k_\lambda, q) := \left\{ d = \sum_{i=0}^{\ell_\lambda - 1} a_i \phi^i \;\middle/\; a_i \in D_{\mathrm{G}},\ wt(d) \geq k_\lambda \right\}, \tag{5.5}$$

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 $S_{\mathrm{E}}(\ell_\omega, k_\omega, q)$ (resp. $S_{\mathrm{G}}(\ell_\lambda, k_\lambda, q)$) 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 ℓ_ω (resp. ℓ_λ) which satisfies that each element in the set S_{E} (resp. S_{G}) is different from each other as an element in $G = \langle P |$. 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 $\#S_{\mathrm{E}}(\ell_\omega, k_\omega, q)$, and in Section 5.3.3, we evaluate $\#S_{\mathrm{G}}(\ell_\lambda, k_\lambda, q)$.

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

We show the classification of the elliptic curves of the form $E_{7,b}/\mathbb{F}_7 : y^2 = x^3 \quad b$ 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 representaion.

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 $E_{7,b}$

b	t	ϕ	relation between ϕ and ω
1	4	$2 \mp \sqrt{3}$	$\phi \quad 2\omega = 3$ or $\phi + 2\omega = 1$
1	4	$2 \mp \sqrt{3}$	$\phi \quad 2\omega = \quad 1$ or $\phi + 2\omega = \quad 3$
2	1	$\frac{1 \pm 3\sqrt{-3}}{2}$	$\phi \quad 3\omega = 2$ or $\phi + 3\omega = \quad 1$
2	1	$\frac{-1 \pm 3\sqrt{-3}}{2}$	$\phi \quad 3\omega = 1$ or $\phi + 3\omega = \quad 2$
3	5	$\frac{5 \pm \sqrt{-3}}{2}$	$\phi \quad \omega = 3$ or $\phi + \omega = 2$
3	5	$\frac{-5 \pm \sqrt{-3}}{2}$	$\phi \quad \omega = \quad 2$ or $\phi + \omega = \quad 3$

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 $0 \geq i \geq i_0 \quad 1$ and $e_{i_0} \vDash e'_{i_0}$, then we can replace d by $(d \quad \prod_{i=0}^{i_0-1} e_i \phi^i)/\phi^{i_0} = \prod_{i=i_0}^{\ell-1} e_i \phi^{i-i_0}$, we have the representations

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

where ℓ is the maximum of length of the two representations. Then we have $\phi \setminus (e_0 \quad e'_0)$. We put $\kappa := e_0 \quad e'_0$. There exists $\gamma \ / \ \mathbb{Z}[\omega]$ such that $\kappa = \phi\gamma$. By taking the square of the usual absolute value of the equation, we obtain $|\kappa|^2 = |\phi|^2|\gamma|^2$. Remark that $|\gamma|^2$ is a rational integer greater than or equal to 0 and $|\phi|^2 = 7$. Since $|e_0| \geq 1$ and $|e'_0| \geq 1$, we must have $|\kappa|^2 \geq 4$. Hence $|\kappa| = 0$.

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}. □

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 $f(X) = \prod_{i=0}^{\ell_\omega - 1} a_i X^i \ / \ \mathbb{C}[X]$, $a_i \ / \ D_{\mathrm{E}}$ $(0 \geq i \geq \ell_\omega \quad 1)$. Since $\mp\omega^2 = \bigcirc(\omega + 1)$, it is easily seen that $a_i = a'_i + a''_i\omega$ for some $a'_i, a''_i \ / \ \}0, \mp 1\langle \leftarrow \mathbb{Z}$.

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: $f(X) = \sum_{i=0}^{\ell_\omega - 1} a_i X^i \mapsto \mathcal{H}_{\mathrm{E}}(f(X)) = \sum_{i=0}^{\ell_\omega - 1}((a_i''/v)X + (a_i' - (u/v)a_i''))X^i \in \mathbb{Q}[X]$. Namely, $\mathcal{H}_{\mathrm{E}}(f(X))$ can be obtained by substitute the coefficients a_i of $f(X)$ for $(a_i''/v)X + (a_i' - (u/v)a_i'')$. Arranging $\mathcal{H}_{\mathrm{E}}(f(X))$ in descending order yields $\mathcal{H}_{\mathrm{E}}(f(X)) = (a_0' - (u/v)a_0'') + \sum_{i=1}^{\ell_\omega - 1}((a_i' - (u/v)a_i'') + a_{i-1}''/v)X^i + (a_{\ell_\omega - 1}''/v)X^{\ell_\omega}$. We put $C_b = |v|$. Then $C_b * \mathcal{H}_{\mathrm{E}}(f(X))$ belongs to $\mathbb{Z}[X]$. More precisely, if $b = \mp 3$ then $\mathcal{H}_{\mathrm{E}}(f(X)) \in \mathbb{Z}[X]$, and if $b = \mp 1$ (resp. $b = \mp 2$) then $2 * \mathcal{H}_{\mathrm{E}}(f(X)) \in \mathbb{Z}[X]$ (resp. $3 * \mathcal{H}_{\mathrm{E}}(f(X)) \in \mathbb{Z}[X]$). Since $a_i', a_i'' \in \{0, \mp 1\} \subset \mathbb{Z}$ and $a_i' a_i'' \sim 0$, we have $|a_0' - (u/v)a_0''| \geq 3$, $|(a_i' - (u/v)a_i'') + a_{i-1}''/v| \geq 4$ $(1 \geq i \geq \ell_\omega - 1)$, and $|a_{\ell_\omega - 1}''/v| \geq 1$.

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 $E_{7,b}(\mathbb{F}_{7^m})$, n the order of P. Let $M = 24$, ℓ_ω be a positive integer satisfying $3 \geq \ell_\omega \geq m - 1$, $C_b \in \mathbb{Z}$ 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 $h_j(X) = \sum_{i=0}^{\ell_\omega - 1} e_i^{(j)} X^i \in \mathbb{C}[X]$, $e_i^{(j)} \in D_E$ for j $(j = 1, 2)$. We put $f(X) = C_b(\mathcal{H}_E(h_1(X)) - \mathcal{H}_E(h_2(X))) \in \mathbb{Z}[X]$. Suppose that n is a prime and $2.6 * 7^{2\ell_\omega} * M^2 \geq n$. 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 $c_0', \ldots, c_{\ell_\omega}' \in \mathbb{Z}$ such that $f(X) = \sum_{i=0}^{\ell_\omega} c_i' X^i \in \mathbb{Z}[X]$. As we explain above, each coefficient of the polynomial $\mathcal{H}_{\mathrm{E}}(h_j(X))$ is equal or less than 4. So $|c_i'| \geq 3 * (4 + 4) = 24 = M$. By dividing $f(X)$ by $X^2 - tX + q$, there exist $b_0, b_1, \ldots, b_{\ell_\omega - 2}, r_0, r_1 \in \mathbb{Z}$ such that $f(X) = (X^2 - tX + q)(\sum_{i=0}^{\ell_\omega - 2} b_i X^i) + r_1 X + r_0$. We evaluate upper bounds of $|b_i|$ $(0 \geq i \geq \ell_\omega - 2)$ and $|r_i|$ $(i = 0, 1)$ using the following number sequence $M_0 = 0$, $M_1 = M$, $M_i = M + |t|M_{i-1} + qM_{i-2}$ $(i \sim 2)$. By expanding the right

hand side of $f(X)$,

$$\begin{aligned} f(X) &= b_{\ell_\omega-2}X^{\ell_\omega} + (b_{\ell_\omega-3} \quad tb_{\ell_\omega-2})X^{\ell_\omega-1} + \sum_{i=2}^{\ell_\omega-2} (b_{i-2} \quad tb_{i-1} + qb_i)X^i \\ &\quad +(\quad tb_0 + qb_1)X + qb_0 + r_1X + r_0. \end{aligned} \tag{5.6}$$

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

$$\begin{aligned} c'_{\ell_\omega} &= b_{\ell_\omega-2} \\ c'_{\ell_\omega-1} &= b_{\ell_\omega-3} \quad tb_{\ell_\omega-2} \\ c'_i &= b_{i-2} \quad tb_{i-1} + qb_i \ (2 \geq i \geq \ell_\omega \quad 2) \\ c'_1 &= \quad tb_0 + qb_1 + r_1 \\ c'_0 &= qb_0 + r_0. \end{aligned} \tag{5.7}$$

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

$$\begin{aligned} \|b_{\ell_\omega-2}\| &= \|c'_{\ell_\omega}\| \geq M_1 \\ \|b_{\ell_\omega-3}\| &\geq \|c'_{\ell_\omega-1}\| + \|t\|\|b_{\ell_\omega-2}\| \geq M + \|t\|M_1 = M_2 \\ \|b_{\ell_\omega-i-2}\| &\geq \|c'_{\ell_\omega-i}\| + \|t\|\|b_{\ell_\omega-i-1}\| + q\|b_{\ell_\omega-i}\| \\ &\geq M + \|t\|M_i + qM_{i-1} = M_{i+1} \ (2 \geq i \geq \ell_\omega \quad 2) \\ \|r_1\| &\geq \|c'_1\| + \|t\|\|b_0\| + q\|b_1\| \\ &\geq M + \|t\|M_{\ell_\omega-1} + qM_{\ell_\omega-2} = M_{\ell_\omega} \\ \|r_0\| &\geq \|c_0\| + q\|b_0\| \geq M + qM_{\ell_\omega-1}. \end{aligned} \tag{5.8}$$

Next we evaluate upper bounds of $\|r_0\|$ and $\|r_1\|$ by using q and M. We put $M' := M/(\|t\| + q \quad 1)$, $M_i' := M_i + M' \ (i \sim 0)$. Then, we have $M_i' = c\alpha_t^i + d\beta_t^i$. Here, $\alpha_t, \beta_t \ / \ \mathbb{C}$ are two roots of the quadratic equation $X^2 \quad \|t\|X \quad q = 0$, $c = (M + M' \quad \beta_t M')/(\alpha_t \quad \beta_t)$, and $d = \quad (M + M' \quad \alpha_t M')/(\alpha_t \quad \beta_t)$. Therefore, it satisfies that

$$\begin{aligned} \|M_i\| &\geq \|M_i'\| + \|M'\| = \|c\alpha_t^i + d\beta_t^i\| + \|M'\| \\ &\geq \max\{\|c\|, \|d\|\} * 2 * \max_{t=\pm1,\pm4,\pm5}\{\|\alpha_t^i\|, \|\beta_t^i\|\} + M/(\|t\| + q \quad 1). \end{aligned} \tag{5.9}$$

By elementary calculation, we obtain

$$\max_{t=\pm1,\pm4,\pm5}\{|\alpha_t^i|, |\beta_t^i|\} = \left(\frac{5+\sqrt{53}}{2}\right)^i < q^i.$$

In addition,

$$\begin{aligned}
|c| &= |(M+M' \quad \beta_t M')(|\alpha_t \quad \beta_t|) \geq (|M|+|M'|+|\beta_t M'|)/\sqrt{t^2+4q} \\
&= \left(M + \frac{M}{|t|+q \quad 1} + \frac{|\beta_t| M}{|t|+q \quad 1}\right) * \frac{1}{\sqrt{t^2+4q}} \\
&< \left(1 + \frac{1}{|t|+q \quad 1} + \frac{q}{|t|+q \quad 1}\right) * \frac{M}{\sqrt{t^2+4q}} \\
&= \frac{M}{\sqrt{t^2+4q}} * \left(\frac{q+1}{|t|+q \quad 1} + 1\right).
\end{aligned}$$

Similary, we have

$$|d| \geq \frac{M}{\sqrt{t^2+4q}} * \left(\frac{q+1}{|t|+q \quad 1} + 1\right).$$

From the above two inequality, we obtain

$$\max\{|c|, |d|\} \geq \frac{M}{\sqrt{t^2+4q}}\left(\frac{q+1}{|t|+q \quad 1} + 1\right). \tag{5.10}$$

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

$$|M_i| < \frac{2q^i M}{\sqrt{t^2+4q}} * \left(\frac{q+1}{|t|+q \quad 1} + 1\right) + \frac{M}{|t|+q \quad 1}.$$

Since $E_{7,b}$ is non-supersingular, we have $1/\sqrt{t^2+4q} \geq 1/\sqrt{1+4q}$ and $1/(|t|+$

$q \quad 1) \geq 1/q$. From $q = 7$ and $1/q < q^i/q$ $(i \sim 1)$, we have

$$\begin{aligned} \backslash M_i \backslash &< \frac{2q^i M}{4q+1} * \Big) \frac{q+1}{q} + 1 \Big\lfloor + \frac{M}{q} \\ &< \Big) \frac{2}{4q+1} \Big) \frac{q+1}{q} + 1 \Big\lfloor + \frac{1}{q} \Big\lfloor q^i M \\ &< \frac{94}{100} q^i M. \end{aligned} \tag{5.11}$$

The above inequality yields

$$\begin{aligned} \backslash \gamma_0 \backslash &\geq \backslash M \backslash + q \backslash M_{\ell_\omega - 1} \backslash < \Big) 1 + q * \frac{94}{100} * q^{(\ell_\omega - 1)} \Big\lfloor M \\ &< \Big) \frac{1}{100} q^{\ell_\omega} + \frac{94}{100} * q^{\ell_\omega} \Big\lfloor M \\ &= \frac{95}{100} q^{\ell_\omega} M, \ (\ell_\omega \sim 3) \end{aligned} \tag{5.12}$$

and

$$\backslash \gamma_1 \backslash \geq \backslash M_{\ell_\omega} \backslash < \frac{94}{100} q^{\ell_\omega} M. \tag{5.13}$$

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

$$\begin{aligned} N(f(\phi)) &= N(r_1 \phi + r_0) \geq {r_0}^2 + \not{t} \backslash \backslash \gamma_0 \backslash \backslash \gamma_1 \backslash + q {r_1}^2 \\ &\geq {r_0}^2 + 2 \ \ \overline{q} \backslash \gamma_0 \backslash \backslash \gamma_1 \backslash + q {r_1}^2 = (r_0 + \ \ \overline{q} r_1)^2 \\ &< \Big) \frac{95}{100} + \frac{94}{100} \ \ \overline{q} \Big\lfloor^2 * (q^{\ell_\omega} M)^2 \\ &< 3.5 * q^{2\ell_\omega} M^2. \end{aligned} \tag{5.14}$$

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

If ℓ_ω satisfies the condition of Proposition 5.6, each element in $S_{\mathrm{E}}(\ell_\omega, k_\omega, 7)$ is different from each other.

Theorem 5.3 (Upper bound of the length of the D_{E}-coefficients Frobe-

nius expansion) *Let* $3 \geq \ell_\omega \geq (\log_2 n \quad 2\log_2 M \quad \log_2 3.5)/(2\log_2 7)$, *and* $M = 24$. *Then each element in* $S_E(\ell_\omega, k_\omega, 7)$ *is different endomorphism on* G.

Proof Let $h_j(X) = \prod_{i=0}^{\ell_\omega - 1} e_i^{(j)} X^i \ / \ \mathbb{C}[X]$, $e_i^{(j)} \ / \ D_{\mathrm{E}}$ for j $(j = 1, 2)$. Suppose that $h_1(\phi), h_2(\phi) \ / \ S_{\mathrm{E}}(\ell_\omega, k_\omega, 7)$, and $h_1(\phi) \nvDash h_2(\phi)$. We assume $h_1(\phi)P = h_2(\phi)P$ and seek a contradiction. Without loss of generality, we may assume that $e_0^{(1)} \nvDash e_0^{(2)}$. We take $f(X) = C_b\,(\mathcal{H}_{\mathrm{E}}(h_1(X)) \quad \mathcal{H}_{\mathrm{E}}(h_2(X)))$. By Proposition 5.6, $f(X)$ is divisible by $X^2 \quad tX + 7$. Put $e_0^{(j)} := e_0^{(j)'} + e_0^{(j)''}\omega$ for j $(j = 1, 2)$. Then $(e_0^{(0)'} \quad e_0^{(1)'}) \quad (u/v)(e_0^{(0)''} \quad e_0^{(1)''})|$ must be divisible by 7. However, one can easily check that there do not exist $e_0^{(0)}, e_0^{(1)} \ / \ D_{\mathrm{E}}$ such that $7 \mid (e_0^{(0)'} \quad e_0^{(1)'}) \quad (u/v)(e_0^{(0)''} \quad e_0^{(1)''})|$. This is a contradiction. Thus we have the desired result. □

By Theorem 5.3, if $3.5 * \ 7^{2\ell_\omega} * \ 24^2 \geq n$, the cardinality of the set S_{E} is given by the following theorem.

Theorem 5.4 [Cardinality of the set S_{E}]

$$\#S_E(\ell_\omega, k_\omega, 7) = \sum_{i=0}^{k_\omega} \binom{\ell_\omega}{i} \Big\rfloor 6^i.$$

Proof By elementary combinatorics, we obtain the above equation. □

Our proposed test using the curve $E_{7,b}$ is described in Algorithm 10.

By the same argument as Section 5.2.3, the computational cost of the proposed test using the Curve $E_{7,b}$ is given by

$$k_\omega N + 2|\mathcal{A} + 3\ell_\omega \mathcal{F} + 2M + (1 \text{ scalar multiplication}). \tag{5.15}$$

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 $E_{5,a}$

a	t	ϕ	relation between ϕ and λ
1	2	$1 \mp 2 \quad \overline{1}$	$\phi \mp 2\lambda = 1$
1	2	$1 \mp 2 \quad \overline{1}$	$\phi \mp 2\lambda = \quad 1$
2	4	$2 \mp \quad \overline{1}$	$\phi \mp \lambda = 2$
2	4	$2 \mp \quad \overline{1}$	$\phi \mp \lambda = \quad 2$

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

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

Output: Accept or Reject

1: Choose N random elements $s_1, \ldots, s_N$ from $S_{\mathrm{E}}(\ell_\omega, k_\omega, 7)$. 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 $c_{i,j} \ / \ \}1, \omega, \omega^2\langle$), $\epsilon_{i,j} = \quad 1$ (if $c_{i,j} \ / \ \} \quad 1, \quad \omega, \quad \omega^2\langle$) for nonzero $c_{i,j}$ for each i.
2: $s \to \prod_{i=1}^{N} s_i x_i \bmod (\phi^{m-1} + \phi^{m-2} + \times\times\times + \phi + 1)$
3: $R[i] \to \mathcal{O}$ for $i \ / \ \}1, \omega, \omega^2\langle$
4: **for** j from $\ell_\omega \quad 1$ downto 0 **do**
5: $\quad R[1] \to \phi(R[1])$, $R[\omega] \to \phi(R[\omega])$, $R[\omega^2] \to \phi(R[\omega^2])$
6: $\quad$ **for** i from 1 to N **do**
7: $\quad\quad$ **if** $c_{i,j} \models 0$ **then**
8: $\quad\quad\quad$ **if** $c_{i,j} = \mp 1$ **then**
9: $\quad\quad\quad\quad R[1] \to R[1] + \epsilon_{i,j}(Q_i)$
10: $\quad\quad\quad$ **else if** $c_{i,j} = \mp\omega$ **then**
11: $\quad\quad\quad\quad R[\omega] \to R[\omega] + \epsilon_{i,j}(Q_i)$
12: $\quad\quad\quad$ **else**
13: $\quad\quad\quad\quad R[\omega^2] \to R[\omega^2] + \epsilon_{i,j}(Q_i)$
14: $\quad\quad\quad$ **end if**
15: $\quad\quad$ **end if**
16: $\quad$ **end for**
17: **end for**
18: $Q \to \omega(R[\omega^2])$, $Q \to \omega(R[\omega] + Q)$, $Q \to Q + R[1]$
19: **if** $Q = sP$ **then**
20: $\quad$ **return** Accept
21: **else**
22: $\quad$ **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}

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

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

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

where ℓ 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 γ / $\mathbb{Z}[\lambda]$ such that $\kappa = \phi\gamma$. By taking the square of the usual absolute value of the equation, we obtain $\lvert\kappa\rvert^2 = \lvert\phi\rvert^2\lvert\gamma\rvert^2$. Remark that $\lvert\gamma\rvert^2$ is a rational integer greater than or equal to 0 and $\lvert\phi\rvert^2 = 5$. Since $\lvert e_0\rvert \geq 1$ and $\lvert e'_0\rvert \geq 1$, we must have $\lvert\kappa\rvert^2 \geq 4$. Hence $\lvert\kappa\rvert = 0$.

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}. □

As in Section 5.3.2, we prepare the following transformation. Let $f(X) = \prod_{i=0}^{\ell_\lambda-1} a_iX^i$ / $\mathbb{C}[X]$, a_i / 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 / $\}0, \mp1\langle \leftarrow \mathbb{Z}$ satisfying $a'_ia''_i = 0$. 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: $f(X) = \prod_{i=0}^{\ell_\lambda-1} a_iX^i \not\mapsto \mathcal{H}_{\mathrm{G}}(f(X)) = \prod_{i=0}^{\ell_\lambda-1}((a''_i/v)X + (a'_i (u/v)a''_i))X^i$ / $\mathbb{Q}[X]$. Namely, $\mathcal{H}_{\mathrm{G}}(f(X))$ can be obtained by substitute the coefficients a_i of $f(X)$ for $(a''_i/v)X + (a'_i (u/v)a''_i)$. Arranging $\mathcal{H}_{\mathrm{G}}(f(X))$ in descending order yields $\mathcal{H}_{\mathrm{G}}(f(X)) = (a'_0 (u/v)a''_0) + \prod_{i=1}^{\ell_\lambda-1}((a'_i (u/v)a''_i) + a''_{i-1}/v)X^i + (a''_{i-1}/v)X^{\ell_\lambda}$. We put $C_a = \lvert\psi\rvert$ Then $C_a * \mathcal{H}_{\mathrm{G}}(f(X))$ belongs to $\mathbb{Z}[X]$. More precisely, if $a = \mp2$

then $\mathcal{H}_{\mathrm{G}}(f(X))$ / $\mathbb{Z}[X]$, and if $a = \mp 1$ then $2 * \mathcal{H}_{\mathrm{G}}(f(X))$ / $\mathbb{Z}[X]$. Since a'_i, a''_i / $\}0, \mp 1\langle$ $\leftarrow \mathbb{Z}$ and $a'_i a''_i = 0$, we have $\backslash a'_0$ $(u/v)a''_0 \backslash \geq 2$, $\backslash (a'_i$ $(u/v)a''_i) + a''_{i-1}/v \backslash \geq 3$ $(1 \geq i \geq \ell_\lambda$ 1), and $\backslash a''_{\ell_\lambda - 1}/v \backslash \geq 1$.

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

Proposition 5.8 *Let P be a point on $E_{5,a}(\mathbb{F}_{5^m})$, n the order of P. Let $M = 12$, ℓ_λ be a positive integer satisfying $3 \geq \ell_\lambda \geq m$ 1, C_a / $\mathbb{Z}$ be a rational integer such that if $a = \mp 1$ then $C_a = 2$, if $a = \mp 2$ then $C_a = 1$. Let $h_j(X) = \prod_{i=0}^{\ell_\lambda - 1} e_i^{(j)} X^i$ / $\mathbb{C}[X]$, $e_i^{(j)}$ / D_G for j $(j = 1, 2)$. We put $f(X) = C_a\,(\mathcal{H}_G(h_1(X))$ $\mathcal{H}_E(h_2(X)))$ / $\mathbb{Z}[X]$. Suppose that n is a prime and $5^{2\ell_\lambda} *$ $M^2 \geq n$. Then $f(\phi)P = \mathcal{O}$ 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 $f(X) = \prod_{i=0}^{\ell_\lambda} c'_i X^i$ / $\mathbb{Z}[X]$. As we explain above, each coefficient of the polynomial $\mathcal{H}_{\mathrm{G}}(h_j(X))$ is equal or less than 3. So $\backslash c'_i \backslash \geq 2 * (3 + 3) = 12 = M$. By dividing $f(X)$ by X^2 $tX + q$, there exist $b_0, b_1, \times\times\times, b_{\ell_\lambda - 2}, r_0, r_1$ / $\mathbb{Z}$ such that $f(X) = (X^2$ $tX + q)(\prod_{i=0}^{\ell_\lambda - 2} b_i X^i) + r_1 X + r_0$. 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 $M'_i = c\alpha_t^i + d\beta_t^i$, where α_t, β_t / $\mathbb{C}$ are two roots of the quadratic equation X^2 $\backslash t \backslash X$ $q = 0$, $c = (M + M'$ $\beta_t M')/(\alpha_t$ $\beta_t)$, and $d =$ $(M + M'$ $\alpha_t M')/(\alpha_t$ $\beta_t)$. Therefore, it satisfies that

$$\begin{aligned} \backslash M_i \backslash &\geq \backslash M'_i \backslash + \backslash M' \backslash = \backslash c\alpha_t^i + d\beta_t^i \backslash + \backslash M' \backslash \\ &\geq \max\}\backslash c \backslash, \backslash d \backslash\langle * 2 * \max_{t=\pm 2, \pm 4} \}\backslash \alpha_t^i \backslash, \backslash \beta_t^i \backslash\langle + M/(\backslash t \backslash + q \quad 1). \quad (5.16) \end{aligned}$$

By elementary calculation, we obtain

$$\max_{t=\pm 2, \pm 4} \}\backslash \alpha_t^i \backslash, \backslash \beta_t^i \backslash\langle = 5^i = q^i.$$

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

$$\backslash M_i \backslash \ \geq \ \frac{2q^i M}{\overline{t^2+4q}} * \ \Big) \frac{q+1}{\backslash t \backslash + q \quad 1} + 1 \Big\lfloor \ + \frac{M}{\backslash t \backslash + q \quad 1}.$$

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

$$\begin{aligned} \backslash M_i \backslash \ &\geq \ \frac{2q^i M}{\overline{4q+1}} * \ \Big) \frac{q+1}{q} + 1 \Big\lfloor \ + \frac{M}{q} \\ &< \ \Big) \frac{2}{\overline{4q+1}} \Big) \frac{q+1}{q} + 1 \Big\lfloor \ + \frac{1}{q} \Big\lfloor \ q^i M \\ &< \ \frac{117}{100} q^i M. \end{aligned} \tag{5.17}$$

The above inequality yields

$$\begin{aligned} \backslash r_0 \backslash \ &\geq \ \backslash M \backslash + q \backslash M_{\ell_\lambda - 1} \backslash < \Big) 1 + q * \ \frac{117}{100} * \ q^{(\ell_\lambda - 1)} \Big\lfloor \ M \\ &< \ \Big) \frac{1}{100} q^{\ell_\lambda} + \frac{117}{100} * \ q^{\ell_\lambda} \Big\lfloor \ M \\ &= \ \frac{118}{100} q^{\ell_\lambda} M, \ (\ell_\lambda \sim 3) \end{aligned} \tag{5.18}$$

and

$$\backslash r_1 \backslash \geq \backslash M_{\ell_\lambda} \backslash < \frac{117}{100} q^{\ell_\lambda} M. \tag{5.19}$$

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

$$\begin{aligned} N(f(\phi)) \ &= \ N(r_1 \phi + r_0) \geq {r_0}^2 + \backslash t \backslash \backslash r_0 \backslash \backslash r_1 \backslash + q {r_1}^2 \\ &\geq \ {r_0}^2 + 2 \ \ \overline{q} \backslash r_0 \backslash \backslash r_1 \backslash + q {r_1}^2 = (r_0 + \ \ \overline{q} r_1)^2 \\ &< \ \Big) \frac{118}{100} + \frac{117}{100} \ \ \overline{q} \Big\lfloor \ ^2 * \ (q^{\ell_\lambda} M)^2 \\ &< \ 3.8 * \ q^{2\ell_\lambda} M^2. \end{aligned} \tag{5.20}$$

Here, we assume that $f(\phi)P = \mathcal{O}$. Then, $N(f(\phi))P = (f(\bar{\phi}) f(\phi))P = \mathcal{O}$, and

$N(f(\phi))$ / $\mathbb{Z}$. Thus we obtain $n \setminus N(f(\phi))$. Hence $N(f(\phi)) < 3.8 * 5^{2\ell_\lambda} M^2 \geq n$ implies $N(f(\phi)) = 0$. Therefore $r_0 = r_1 = 0$. This completes the proof. □

If ℓ_λ satisfies the condition of Proposition 5.8, each element in $S_{\mathrm{G}}(\ell_\lambda, k_\lambda, 5)$ is different from each other.

Theorem 5.5 (Upper bound of the length of the D_{G}-coefficients Frobenius expansion) *Let* $3 \geq \ell_\lambda \geq (\log_2 n \quad 2\log_2 M \quad \log_2 3.8)/\log_2 5$, *and* $M = 12$. *Then each element in* $S_G(\ell_\lambda, k_\lambda, 5)$ *is different endomorphism on* G.

Proof Let $h_j(X) = \prod_{i=0}^{\ell_\lambda - 1} e_i^{(j)} X^i$ / $\mathbb{C}[X]$, $e_i^{(j)}$ / D_{G} for j $(j = 1, 2)$. Suppose that $h_1(\phi), h_2(\phi)$ / $S_{\mathrm{G}}(\ell_\lambda, k_\lambda, 7)$, and $h_1(\phi) \vDash h_2(\phi)$. We assume $h_1(\phi)P = h_2(\phi)P$ and seek a contradiction. Without loss of generality, we may assume that $e_0^{(1)} \vDash e_0^{(2)}$. We take $f(X) = \mathcal{H}_{\mathrm{G}}(h_1(X)) \quad \mathcal{H}_{\mathrm{G}}(h_2(X))$. By Proposition 5.8, $C_a * f(X)$ is divisible by $X^2 \quad tX + 5$. Put $e_0^{(j)} := e_0^{(j)'} + e_0^{(j)''}\lambda$ for j $(j = 1, 2)$. Then $C_a * \big)(e_0^{(0)'} \quad e_0^{(1)'}) \quad (u/v)(e_0^{(0)''} \quad e_0^{(1)''})\big($ must be divisible by 5. Recall that $e_0^{(j)'} e_0^{(j)''} = 0$ $(j = 1, 2)$. From this fact, one can easily check that there do not exist $e_0^{(0)}, e_0^{(1)}$ / D_{G} such that $5 \big| C_a * (e_0^{(0)'} \quad e_0^{(1)'}) \quad (u/v)(e_0^{(0)''} \quad e_0^{(1)''})\big|$. This is a contradiction. Thus we have the desired result. □

By Theorem 5.5, if $3.8 * 5^{2\ell_\lambda} * 12^2 \geq n$, the cardinality of the set S_{G} is given by the following theorem.

Theorem 5.6 (Cardinality of the set S_{G})

$$\#S_G(\ell_\lambda, k_\lambda, 5) = \sum_{i=0}^{k_\lambda} \binom{\ell_\lambda}{i} \Big\lfloor 4^i.$$

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

$$k_{\mathrm{bv}} N + 1 \big| \mathcal{A} + 2\ell_\lambda \mathcal{F} + M + (1 \text{ scalar multiplication}). \tag{5.21}$$

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 $E_{7,a}/\mathbb{F}_7$ 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	$n = \#E(\mathbb{F}_{5^m})/h$ (resp. $\#E(\mathbb{F}_{7^m})/h$).
$bitlen(x)$	bit length of a positive integer x / $\mathbb{Z}$.

Table 5.3: Sample Parameters for the elliptic curves $E_{5,a}/\mathbb{F}_5$

$E_{5,2}$-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 B0E3BDBC 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 87ECD859 0680A3AA 02A9DFE5 BD41A836 BA41FAF8 6F4E279D √
$E_{5,-2}$-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 9D82FCC4 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 284B578D 7FF2A760 414536EF BCA758CB F4FBCB0A 32CC8363 8555BB9A 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 5B4E5998 400A95D3 5EAC354F 34215CD4 6E417018 √ FB1DC739 C5E736BD C153819F 71B64393 465FC46B F4AB38FF √ A407344D

$E_{5,1}$-227: $m = 227$, $f(x) = x^{227} + x^{53} + 1$, $a = 1$, $h = 4$, $bitlen(h) = 3$, $bitlen(n) = 526$

$n =$ 0x 000021C5 29DD78FA 571E196B 3EBB0D20 429C476A 1848CAB5 √
E0E8A121 378DE187 888F99D2 99F404EE 4F9BC974 D5035A62 √
AC9F5E1E 0DA29A51 0B4012E2 3ECD1590 9A4B1065 √

$E_{5,2}$-239: $m = 239$, $f(x) = x^{239} + 4x^{124} + 1$, $a = 2$, $h = 2$, $bitlen(h) = 2$, $bitlen(n) = 554$

$n =$ 0x 000003D6 D75FB22B 2D628AB8 42BDA59D D2C76CE2 C5B53D88 √
A4349EFF F107E218 706735C4 7CC53BFD E966A0AC 0346C5EE √
1F25BD2D 2FAC79AC F4DA7B45 947F51BB 5E33A643 A74A470D √

$E_{5,-2}$-317: $m = 317$, $f(x) = x^{317} + x^{24} + 4$, $a = \quad 2$, $h = 10$, $bitlen(h) = 4$, $bitlen(n) = 733$

$n =$ 0x 1A8662F3 B3919708 2BF4C0D2 548C2A3D 779053F3 CF932023 √
7E5EC14C 922CB561 85BD758D 2B99F28C ADBC9799 509367D4 √
E379DA56 D0582105 28C0C11C 7B00363E 9042A70F 4F3A60D4 √
8CE325D4 4171E2E9 68316505 D41F503A 36E361D1 √

$E_{5,1}$-353: $m = 353$, $f(x) = x^{353} + x^{42} + 2$, $a = 1$, $h = 4$, $bitlen(h) = 3$, $bitlen(n) = 818$

$n =$ 0x 00031E34 C30449D0 361BD9BB 7B921C55 EDFE15C3 63B2DB07 √
E102C653 7B1E4A4D B5613C98 5D75D2B6 9C6D5CEE AD8A0A8F √
01E021FB 5DCA0A1F 6B4BE8D8 FE3A5043 09BF6B34 81B13AFC √
C4DB17A1 ED39F3BA E85063F6 A98D2DCF 776DAB2E 24EB92CE √
6819B5A7 32EB3841 √

$E_{5,-2}$-397: $m = 397$, $f(x) = x^{397} + x^{64} + 4$, $a = \quad 2$, $h = 10$, $bitlen(h) = 4$, $bitlen(n) = 919$

$n =$ 0x 00597B64 43800D18 F17B748F F893C1B1 43EF63CB 0FBE159E √
81A88F44 5A61A016 77803E71 6BDF2E24 E0E1C364 07217F0E √
0C5AE445 12F85C59 7570046F F08D5D1B 9214AE06 3105AB29 √
53E07DE4 DEDB8C42 ABFC11B6 0DA9FF06 724F75F7 5936C6D3 √
122CDF3C 6C7BC7F3 A29A1678 CCF178A0 A8A17191 √

$E_{5,1}$-401: $m = 401$, $f(x) = x^{401} + x^{47} + 1$, $a = 1$, $h = 4$, $bitlen(h) = 3$, $bitlen(n) = 930$

$n =$ 0x 00000002 22279F75 FD0FF0BD E402F0BA B1D5B471 249E96E3 √
95AEF3F2 5F4E6EB1 A4DB9920 603D1F44 E5AF2416 91F31205 √
85F1FB3E 68C23F98 48DFA803 483E223B F6475435 908078EE √
74D28CDF 7AEDED0C 9B759941 699DB218 11FB4156 492C66D9 √
AF41C992 CB1A08F9 E89ABC23 C39054AC 9C470F34 67A26B61 √

$E_{5,-2}$-439: $m = 439$, $f(x) = x^{439} + 4x^{35} + 1$, $a = \quad 2$, $h = 10$, $bitlen(h) = 4$, $bitlen(n) = 1017$

$n =$ 0x	0100CCFD 6D32E361 2A7F4535 9B775336 206E4053 5E362EC5
	7D5C0F1E B832C2D9 C2B8B468 3B54F5EC 018C1A10 A8DECBFE ✓
	C96B211E E977B7C4 6DCEF7E2 C81A8F0E 7DDFC28A 051C3168 ✓
	8DA47D5C 7E6B890F F1F5A336 8CAFE22B 9A5A9357 7377B7A8 ✓
	342935DB 09E38487 A25E300C 5F681788 B3A4E79A 331AE55C ✓
	5091770A DF9A5B4D

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

$E_{7,-1}$-59: $m = 59$, $f(x) = x^{59} + x^4 + 3$, $b = \quad 1$, $h = 12$, $bitlen(h) = 4$, $bitlen(n) = 163$
$n =$ 0x 00000004 235C4976 E80807AB CA35A9F1 81D3ED3E 61445E53
$E_{7,3}$-61: $m = 61$, $f(x) = x^{61} + x^4 + 1$, $b = 3$, $h = 3$, $bitlen(h) = 2$, $bitlen(n) = 170$
$n =$ 0x 0000032B 12A83F09 A625DF86 D157F30E 0B1CDFB1 3FD4ED99
$E_{7,3}$-71: $m = 71$, $f(x) = x^{71} + x^{10} + 4$, $b = 3$, $h = 3$, $bitlen(h) = 2$, $bitlen(n) = 198$
$n =$ 0x 00000035 57E6DA5B FB1EC95B 4614A377 DBAF4B91 89919638 ✓
64505C87
$E_{7,-3}$-103: $m = 103$, $f(x) = x^{103} + x^{54} + 6$, $b = \quad 3$, $h = 13$, $bitlen(h) = 4$, $bitlen(n) = 286$
$n =$ 0x 2BEDF1E0 41D10B7B 55B514D7 9FE59CB4 0B53BF6B 9DB9FE62 ✓
3B89B385 159E5565 1BA268D5
$E_{7,2}$-103: $m = 103$, $f(x) = x^{103} + 3x^9 + 1$, $b = 2$, $h = 7$, $bitlen(h) = 3$, $bitlen(n) = 287$
$n =$ 0x 5195537B E7F1F0C0 7A996FD9 96AA6C29 CBE5609D 63D7EABF ✓
9ADF4C63 94735FEE E78111CD
$E_{7,2}$-107: $m = 107$, $f(x) = x^{107} + x^{15} + 6$, $b = 2$, $h = 7$, $bitlen(h) = 3$, $bitlen(n) = 298$
$n =$ 0x 000002FD 2983FD1A 6422FD3D D91207BE 14607401 4FB6235C ✓
33B027AB EBCDF55A 49A828DE 311D033F
$E_{7,-1}$-109: $m = 109$, $f(x) = x^{109} + 6x^{23} + 6$, $b = \quad 1$, $h = 12$, $bitlen(h) = 4$, $bitlen(n) = 303$
$n =$ 0x 0000556E E2A757DD 0392C67D 27435D4E 711A4925 792E091D ✓
B3B0409D 29FAD758 F7AC5FD4 27A74F4D
$E_{7,1}$-127: $m = 127$, $f(x) = x^{127} + x^2 + 6$, $b = 1$, $h = 4$, $bitlen(h) = 3$, $bitlen(n) = 355$
$n =$ 0x 00000005 CAC4104D 859A6DF5 82D57312 11D9947A 4AE9CFD1 ✓
F4E36489 97D050DC E03624B8 91381F19 AA1824CF 98DE5637
$E_{7,3}$-167: $m = 167$, $f(x) = x^{167} + 3x^{39} + 1$, $b = 3$, $h = 3$, $bitlen(h) = 2$, $bitlen(n) = 468$
$n =$ 0x 0009783A 48EDC313 BA408497 7B4E4849 883BF3D0 8EC233E6 ✓

	2CB37954	33A61505	385CAF38	732C2337	EDA85316	2CA773D5 √
	1053D953	7248F3EA	7C290A47			
$E_{7,2}$-181: $m = 181$, $f(x) = x^{181} + x^5 + 3$, $b = 2$, $h = 7$, $bitlen(h) = 3$, $bitlen(n) = 506$						
$n =$ 0x	0280DE7F	5E056B02	C82C2CFE	C4038DDB	6CA57BF4	2FBB56FB √
	F74DAADB	65B85853	0E19FF02	B0B2748C	C32514A1	4F5975AB √
	53254133	EC718BF1	50C86198	19A6FF59		
$E_{7,-3}$-191: $m = 191$, $f(x) = x^{191} + 3x^{31} + 1$, $b = \quad 3$, $h = 13$, $bitlen(h) = 4$, $bitlen(n) = 533$						
$n =$ 0x	0016B21C	CFE2FB4E	0B541889	60D8B31F	5711D7D5	A8AE5602 √
	AEEDEA73	33DAAC30	8706656A	62F57B75	1A9FFBD9	D470E482 √
	0D7CE237	021A828C	B8BD9503	4AE2D557	ECD12537	
$E_{7,1}$-223: $m = 223$, $f(x) = x^{223} + 4x^{16} + 6$, $b = 1$, $h = 4$, $bitlen(h) = 3$, $bitlen(n) = 625$						
$n =$ 0x	00010739	5CC42964	0E1C6468	886BEC6F	6478026C	3BB31334 √
	297DC5EC	9BB482E1	748405E2	807FB6D7	1CB4BE06	F6CD1379 √
	5ABC7679	0E24ABC6	0B4CF9BB	65600766	528640F7	EE46D8EF √
	61ECB724	6B049C97				
$E_{7,1}$-383: $m = 383$, $f(x) = x^{383} + x^{24} + 6$, $b = 1$, $h = 4$, $bitlen(h) = 3$, $bitlen(n) = 1074$						
$n =$ 0x	00025313	FC8FCB0A	1214B25F	E5235E4E	98F78982	632A525B √
	57FC01BB	C71B6491	C54FFF95	E9822C33	9B81C0A3	57472671 √
	0F6B4A58	9FB61465	A3450E13	FF323D32	ECD4F0F5	EC177CD6 √
	6EA567B4	EC0BDD2F	6DF59C4C	2CC17741	373453DD	06EB3E66 √
	63D4D191	86294806	59E26EA1	20C39651	43F3B8AC	66FC3262 √
	7061D995	046ADEA9	198BB1B1	A1E92737		

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® [*1)] 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 represened 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 represened by three bits, and we extend the method in [38] to implement the arithmetic operations in $\mathbb{F}_{7^{103}}$. The timings are listed in Table 5.5.

Table 5.5: Timings of the Finite Field Operations.

Finite Field Operations	timing (in μsec)
Addition	0.063
Subtraction	0.063
7-th powering	1.400
Multiplication	15.230

5.5.2 Estimate

We estimate batch verification costs of the CE test and our proposed test in the case of $E_{7,-3}$-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 $E_{7,-3}$-103 in Table 5.4, the maximal value of ℓ_1, ℓ_2, ℓ_ω 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 7th-powerings) for the estimation. Moreover, it is assumed that

≡ 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.

$\equiv$ the cost of a 7th powering and 0.092 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 5.6: Estimate the computational costs of CE test and our test

Method	# of M
CE test (S_1)	$16k_1N + 284.21$
CE test (S_2)	$16k_2N + 1222.72$
Our test (S_{E})	$16k_\omega N + 59.04$

For $\ell_1 = 97$, $\ell_2 = 96$, $\ell_\omega = 98$ and for a given k_{bv}, we describe the values k_1, k_2, k_ω so that $\#S_j \subset 2^{k_{\mathrm{bv}}}$, $\#S_{\mathrm{E}} \subset 2^{k_{\mathrm{bv}}}$.

Table 5.7: Values k_j and k_ω such that $\#S_j \subset 2^{k_{\mathrm{bv}}}$, $\#S_{\mathrm{E}} \subset 2^{k_{\mathrm{bv}}}$

Method \ Security Param. k_{bv}	40	60	80	100	120	140
CE test (S_1)	5	8	11	14	18	22
CE test (S_2)	4	6	9	11	14	16
Our test (S_{E})	6	9	12	16	21	25

For a given method in Table 5.6, a given number of signatures N, and a given security parameter k_{bv}, we can estimate the number of multiplications in $\mathbb{F}_{7^{103}}$ by substituting N and k_j or k_ω 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

Security Param. k_{bv}	40	60	80	100	120	140
# of signatures	$N \geq 18$	$N \geq 18$	$N \geq 18$	$N \geq 9$	$N \geq 6$	$N \geq 6$

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 τ-NAF on Koblitz curves. The first generalization is ϕ-GNAF. ϕ-GNAF can be applied to subfield elliptic curves where the traces of Frobenius are equal to 1. The second generalization is ϕ-rNAF. ϕ-rNAF can be applied to subfield elliptic curves where the traces of Frobenius are equal to 1 or 1. The digit set of ϕ-rNAF is larger than that of ϕ-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 ϕ-GNAF and ϕ-rNAF is relatively small (a few elliptic additions). But, if q is significantly large, the computational cost for the precomputation table of ϕ-rNAF is quite large compared to that for ϕ-GNAF. However the non-zero density for ϕ-rNAF is significantly smaller than that for ϕ-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 τ-adic expansions. We also give a new proof of the minimality of

the Hamming weight of the τ-NAF. Further, by using the proof of the lower bound and the new proof of the minimality, we classify a minimal length τ-adic expansion (τ-MLF) with minimal Hamming weight on Koblitz curves. The classification shows that the τ-NAF has almost minimal length among all τ-adic expansions of minimal Hamming weight and we can easily convert the τ-NAF into a minimal length τ-adic expansion without changing the Hamming weight. Scalar multiplication on Koblitz curve using τ-MLF method is slightly faster than the original τ-NAF method. Thus τ-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, τ-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 τ-NAF for a family of hyperelliptic curves

$$C_a : y^2 + xy = x^5 + ax^2 + 1, \quad a \ / \mathbb{F}_2 \tag{6.1}$$

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

The two generalizations are as follows:

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

(ii) τ-NAF: The other is a sparse τ-adic expansion. The expansion of $\alpha \ / \ \mathbb{Z}[\tau]$ is a τ-adic expansion $\alpha = \sum_{i=0}^{\ell-1} c_i \tau^i$ such that $c_{i+1}c_i = 0$ for all i $(0 \geq i \geq \ell \quad 2)$ and $c_i \ / \ \mathcal{D} = \}0, \mp 1, \mp 2, \mp(1+\tau), \mp(1 \quad \tau), \mp(1 \quad 2\tau), \mp 2 + \tau\langle$ for all i $(0 \geq i \geq \ell \quad 1)$. We will call the above expansion as *τ-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 Koblitz 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 $E_{\mathrm{Ed},c,d} : x^2 + y^2 = c^2(1 + dx^2y^2)$, where $\mathrm{char}(K) \nmid 2$, $c, d \ / \ K$, and $cd(1 \quad c^4d) \nmid 0$.

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

(iii) Jacobi intersection form [53]: An elliptic curve in Jacobi intersection form

over a field K is defined by

$$E_{\mathrm{JI},a}: \left\{ \begin{array}{ll} x^2 + y^2 = 1 & (6.2a) \\ ax^2 + z^2 = 1 & (6.2b) \end{array} \right.$$

where $\mathrm{char}(K) \neq 2$, $a \neq 0$ in K and $a(1 - a) \neq 0$.

(iv) Huff form [43]: An elliptic curve in Huff form over a field K is defined by $E_{\mathrm{Hu},a,b}: ax(y^2 - 1) = by(x^2 - 1)$, where $\mathrm{char}(K) \neq 2$, $a, b \neq 0$ in K^*, and $a^2 \neq b^2$.

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 Koblitz 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 τ-DBNS and τ-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

[1] Keisuke Hakuta, Hisayoshi Sato, and Tsuyoshi Takagi, Efficient arithmetic on subfield elliptic curves over small finite fields of odd characteristic, *J. Math. Cryptol.* **4** (2010), No.3, 199–238.

[2] Keisuke Hakuta, Hisayoshi Sato, and Tsuyoshi Takagi, Explicit lower bound for the length of minimal weight τ-adic expansions on Koblitz curves, *J. Math-for-Ind.* **2A** (2010), 75–83.

Refereed Conference Proceedings

[1] Keisuke Hakuta, Hisayoshi Sato, and Tsuyoshi Takagi, Efficient Arithmetic on Subfield Elliptic Curves over Small Finite Fields of Odd Characteristic, In *the 4th Information Security Practice and Experience Conference, ISPEC 2008*, Vol.4991 of Lecture Notes in Computer Science (2008), Springer-Verlag, 304–318.

[2] Keisuke Hakuta, Yosuke Katoh, Hisayoshi Sato, and Tsuyoshi Takagi, Batch Verification Suitable for Efficiently Verifying a Limited Number of Signatures, In *the 15th Annual International Conference on Information Security and Cryptology, ICISC 2012*, Vol.7839 of Lecture Notes in Computer Science (2013), Springer-Verlag, 425–440.

Non-Refereed Conference Proceedings and Workshops

[1] Keisuke Hakuta, Hisayoshi Sato, and Tsuyoshi Takagi, Efficient Arithmetic of Scalar Multiplication on Subfield Elliptic Curves over Small Odd Characteristics (in Japanese), In *Proc. of Computer Security Symposium 2005* (*CSS 2005*), 6C-3, pp.469–474, 2005.

[2] Keisuke Hakuta, Yuta Sasaki, Toru Owada, and Tsuyoshi Takagi, Fast Software Implementation of Scalar Multiplication on Koblitz Curves over $\mathbb{F}_3$ using ϕ-rNAF (in Japanese), In *Proc. of the 26th Symposium on Cryptography and Information Security 2009* (*SCIS 2009*), 1C-1, 2009.

[3] Keisuke Hakuta, Hisayoshi Sato, and Tsuyoshi Takagi, Explicit lower bound for the length of minimal weight τ-adic expansions on Koblitz curves, In *Proc. of the 27th Symposium on Cryptography and Information Security 2010* (*SCIS 2010*), 2D4-2, 2010.

[4] Keisuke Hakuta, Efficient scalar multiplication on subfield elliptic curves using low Hamming weight Frobenius expansion, Workshop "Algorithmic Number Theory and Its Application to Cryptography", Kyushu University, Japan, June 11th, 2010.

[5] Keisuke Hakuta, Yosuke Katoh, Hisayoshi Sato, and Tsuyoshi Takagi, Fast Batch Verification of Elliptic Curve-based Digital Signatures using Efficiently Computable Endomorphisms (in Japanese), In *Proc. of the 28th Symposium on Cryptography and Information Security 2011* (*SCIS 2011*), 3A4-3, 2011.

Bibliography

[1] A. Antipa, D. Brown, R. Gallant, R. Lambert, R. Struik, and S. Vanstone, Accelerated Verification of ECDSA Signatures, In *the 12th Annual International Workshop on Selected Areas in Cryptography, SAC 2005*, Vol.3897 of Lecture Notes in Computer Science (2006), Springer-Verlag, 307–318. 78

[2] R.M. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren, *The Handbook of Elliptic and Hyperelliptic Curve Cryptography*, CRC Press, 2005. 11, 18, 103

[3] R.M. Avanzi, V.S. Dimitrov, C. Doche, and F. Sica, Extending Scalar Multiplication using Double Bases, In *Advances in Cryptology – ASIACRYPT 2006*, Vol.4284 of Lecture Notes in Computer Science (2006), Springer-Verlag, 130–144. 104

[4] R.M. Avanzi, C. Heuberger, and H. Prodinger, Minimality of the Hamming Weight of the τ-NAF for Koblitz Curves and Improved Combination with Point Halving, In *the 12th Annual International Workshop on Selected Areas in Cryptography, SAC 2005*, Vol.3897 of Lecture Notes in Computer Science (2006), Springer-Verlag, 332–344. 16, 53, 64

[5] R.M. Avanzi, C. Heuberger, and H. Prodinger, Scalar Multiplication on Koblitz Curves Using the Frobenius Endomorphism and Its Combination with Point Halving: Extensions and Mathematical Analysis, *Algorithmica*, **46** (2006), No.3-4, 249–270. 64

[6] R.M. Avanzi, C. Heuberger, and H. Prodinger, On Redundant τ-adic Expansions and Non-Adjacent Digit Sets, In *the 13th Annual International Workshop on Selected Areas in Cryptography, SAC 2006*, Vol.4356 of Lecture Notes in Computer Science (2007), Springer-Verlag, 285–301. 53

[7] R.M. Avanzi, C. Heuberger, and H. Prodinger, Redundant τ-adic Expansions I: Non-Adjacent Digit Sets and their Applications to Scalar Multiplication, *Des. Codes Cryptogr.*, **58** (2011), No.2, 173–202. 53

[8] G. Avoine, J. Monnerat, and T. Peyrin, Advances in Alternative Non-adjacent Form Representations, *Progress in Cryptology - INDOCRYPT 2004, the 5th International Conference on Cryptology in India*, Vol.3348 of Lecture Notes in Computer Science (2004), Springer-Verlag, 260–274. 53

[9] M. Bellare, J. Garay and T. Rabin, Fast Batch Verification for Modular Exponentiation and Digital Signatures, In *Advances in Cryptology – EUROCRYPT '98*, Vol.1403 of Lecture Notes in Computer Science (1998), Springer-Verlag, 236–250. Full version is available at `http://www-cse.ucsd.edu/users/mihir` 74, 77, 78

[10] O. Billet and M. Joye, The Jacobi Model of an Elliptic Curve and Side-Channel Analysis, In *the 15th International Symposium on Applied Algebra, Algorithms and Error-Correcting Codes*, Vol.2643 of Lecture Notes in Computer Science (2003), Springer-Verlag, 34–42. 103

[11] I.F. Blake, V.K. Murty and G. Xu, Nonadjacent radix-τ expansions of integers in Euclidean imaginary quadratic number fields, *Canad. J. of Math.*, **60** (2008), No.6, 1267–1282. 10, 16, 53

[12] I.F. Blake, G. Seroussi, and N.P. Smart, *Elliptic Curves in Cryptography*, LMS Lecture Note Series 265, Cambridge University Press, 1999. 9, 13

[13] I.F. Blake, G. Seroussi, and N.P. Smart, *Advances in Elliptic Curve Cryptography*, LMS Lecture Note Series 317, Cambridge University Press, 2005. 18

[14] W. Bosma, Signed Bits and Fast Exponentiation, *Journal de Théorie des Nombres de Bordeaux*, **13** (2001), No.1, 27–41. 69

[15] C. Boyd, and C. Pavlovski, Attacking and Repairing Batch Verification Schemes, In *Advances in Cryptology – ASIACRYPT 2000*, Vol.1976 of Lecture Notes in Computer Science (2000), Springer-Verlag, 58–71. 74

[16] E. Brickell, D. Gordon, K. McCurley, and D. Wilson, Fast Exponentiation with Precomputation, In *Advances in Cryptology – EUROCRYPT '92*,

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

[17] J. Camenisch, S. Hohenberger, and M. Pedersen, Batch Verification of Short Signatures, *Advances in Cryptology – EUROCRYPT 2007*, Vol.4515 of Lecture Notes in Computer Science (2007), Springer-Verlag, 246–263. 75, 77

[18] J.H. Cheon and D.H. Lee, Use of Sparse and/or Complex Exponents in Batch Verification of Exponentiations, *IEEE Transactions on Computers*, **55** (2006), No.12, 1536–1542. 74, 75, 77, 78, 79, 80, 82, 83, 84, 99

[19] J.H. Cheon and J.H. Yi, Fast Batch Verification of Multiple Signatures, In *the 10th International Conference on Practice and Theory in Public Key Cryptography, PKC 2007*, Vol.4450 of Lecture Notes in Computer Science (2007), Springer-Verlag, 442–457. 75, 77, 80

[20] W.E. Clark and J.J. Liang, On arithmetic weight for a general radix representation of integers, *IEEE Transactions on Information Theory*, **19** (1973), No.6, 823–826. 10, 12, 13, 52, 65

[21] J.S. Coron and D. Naccache, On the Security of RSA Screening, In *the Second International Conference on Practice and Theory in Public Key Cryptography, PKC 1999*, Vol.1560 of Lecture Notes in Computer Science (1999), Springer-Verlag, 197–203. 74

[22] C. Diem, *A Study on Theoretical and Practical Aspects of Weil-Restriction of Varieties*, Ph.D. thesis, Universität Gesamthochschule Essen, 2001. 18

[23] C. Diem, The GHS-Attack in Odd Characteristic, *Journal of the Ramanujan Mathematical Society*, **18** (2003), No.1, 1–32. 18

[24] V. Dimitrov, L. Imbert, and P.K. Mishra, The double-base number system and its application to elliptic curve cryptography, *Math. Comp.*, **77** (2008), No.262, 1075–1104. 104

[25] V.S. Dimitrov, K. Jarvinen, M.J. Jacobson, Jr, W.F. Chan, and Z. Huang, FPGA Implementation of Point Multiplication on Koblitz Curves Using Kleinian Integers, In *the 8th International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2006*, Vol.4249 of Lecture Notes in Computer Science (2006), Springer-Verlag, 445–459. 104

[26] H.M. Edwards, A normal form for elliptic curves, *Bull. Amer. Math. Soc.*, **44** (2007), No.3, 393–422. 103

[27] A. Enge, Bilinear pairings on elliptic curves, Preprint (2013), ArXiv:1301.5520. Available at http://arxiv.org/abs/1301.5520 8

[28] R. Gallant, R. Lambert, and S. Vanstone, Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms, In *Advances in Cryptology – CRYPTO 2001*, Vol.2139 of Lecture Notes in Computer Science (2001), Springer-Verlag, 190–200. 7, 10, 52, 81

[29] O. Goldreich, *Foundations of Cryptography: Volume 1 – Basic Tools*, Cambridge University Press, 2001. 7

[30] O. Goldreich, *Foundations of Cryptography: Volume 2 – Basic Applications*, Cambridge University Press, 2004. 6, 7

[31] C. Günther, 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

[32] C. Günther, T. Lange, and A. Stein, Speeding up the Arithmetic on Koblitz Curves of Genus Two, In *the 7th Annual International Workshop on Selected Areas in Cryptography, SAC 2000*, Vol.2012 of Lecture Notes in Computer Science (2001), Springer-Verlag, 106–117. 10, 102, 103

[33] F. Guo, Y. Mu, and Z. Chen, Efficient Batch Verification of Short Signatures for a Single-Signer Setting without Random Oracles, In *the Third International Workshop on Security, IWSEC 2008*, Vol.5312 of Lecture Notes in Computer Science (2008), Springer-Verlag, 49–63. 74

[34] K. Hakuta, Y. Katoh, H. Sato, and T. Takagi, Batch Verification Suitable for Efficiently Verifying a Limited Number of Signatures, In *the 15th Annual International Conference on Information Security and Cryptology, ICISC 2012*, Vol.7839 of Lecture Notes in Computer Science (2013), Springer-Verlag, 425–440. 102

[35] K. Hakuta, H. Sato, and T. Takagi, Efficient arithmetic on subfield elliptic curves over small finite fields of odd characteristic, *J. Math. Cryptol.* **4** (2010), No.3, 199–238. 53, 78, 101

[36] K. Hakuta, H. Sato, and T. Takagi, Explicit lower bound for the length of minimal weight τ-adic expansions on Koblitz curves, *J. Math-for-Ind.* **2A** (2010), 75–83. 80, 102

[37] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag, New York, 2004. 14, 40, 44, 80

[38] K. Harrison, D. Page and N.P. Smart, Software Implementation of Finite Fields of Characteristic Three, for Use in Pairing-based Cryptosystems, In *LMS Journal of Computation and Mathematics*, **5** (2002), No.1, 181–193. 98

[39] L. Harn, Batch Verifying Multiple DSA Digital Signatures, *Electronics Letters*, **34** (1998), No.9, 870–871. 74

[40] L. Harn, Batch Verifying Multiple RSA Digital Signatures, *Electronics Letters*, **34** (1998), No.12, 1219–1220. 74

[41] C. Heuberger, Redundant τ-adic Expansions II: Non-Optimality and Chaotic Behaviour, *Math. Comput. Sci.* **3** (2010), 141–157. 53

[42] M.S Hwang, C.C Lee, and E.J. Lu, Cryptanalysis of the Batch Verifying Multiple DSA-Type Digital Signatures, *Pakistan Journal of Applied Sciences*, **1** (2001), No.3, 287–288. 74

[43] M. Joye, M. Tibouchi, and D. Vergnaud, Huff's model for elliptic curves, In *the 9th Algorithmic Number Theory Symposium, ANTS-IX*, Vol.6197 of Lecture Notes in Computer Science (2010), Springer-Verlag, 234–250. 104

[44] N. Koblitz, Elliptic curve cryptosystems, *Math. Comp.*, **48** (1987), No.177, 203–209. 9, 51

[45] N. Koblitz, Hyperelliptic Cryptosystems, *J. Cryptology*, **1** (1989), No.3, 139–150. 103

[46] N. Koblitz, CM-curves with good cryptographic properties, In *Advances in Cryptology – CRYPTO 1991*, Vol.576 of Lecture Notes in Computer Science (1992), Springer-Verlag, 279–287. 7, 9, 52, 53

[47] N. Koblitz, An Elliptic Curve Implementation of the Finite Field Digital Signature Algorithm, In *Advances in Cryptology – CRYPTO 1998*, Vol.1462 of Lecture Notes in Computer Science (1998), Springer-Verlag, 327–337. 7, 10, 16, 81, 82

[48] V. Kumer and S. Madria, Secure Data Aggregation in Wireless Sensor Networks, Wireless Sensor Network Technologies for the Information Explosion Era (Book Series: Studies in Computational Intelligence), Vol.278 (2010), Springer-Verlag, 77–107. 76

[49] T. Lange, *Efficient Arithmetic on Hyperelliptic Koblitz Curves*, Ph.D. thesis, Universität Gesamthochschule Essen, 2001. 103

[50] T. Lange, Koblitz curve cryptosystems, *Finite Fields Appl.*, **11** (2005), No.2, 200–229. 7, 103

[51] T. Lange and I.E. Shparlinski, Distribution of some sequences of points on elliptic curves, *J. Math. Cryptol.* **1** (2007), No.1, 1–11. 8

[52] S. Lee, S. Cho, J. Choi, and J. Cho, Efficient Identification of Bad Signatures in RSA-Type Batch Signatures, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E89-A, No.1 (2006), 74–80. 74

[53] P. Liardet and N.P. Smart, Preventing SPA/DPA in ECC Systems using the Jacobi Form, In *the Third International Workshop on Cryptographic Hardware and Embedded Systems, CHES 2001*, Vol.2162 of Lecture Notes in Computer Science (2001), Springer-Verlag, 391–401. 103

[54] J.H. van Lint, *Introduction to coding theory*, GTM 86 (3rd edition), Springer-Verlag, 1999. 19, 20, 23

[55] Q. Liu, *Algebraic Geometry and Arithmetic Curves*, Oxford Graduate Texts in Mathematics, Vol.6, Oxford University Press, Oxford, 2002. 5, 6

[56] V. Miller, Use of elliptic curves in cryptography, In *Advances in Cryptology – CRYPTO 1985*, Vol.218 of Lecture Notes in Computer Science (1986), Springer-Verlag, 417–426. 9, 51

[57] A. Miyaji, T. Ono, and H. Cohen, Efficient elliptic curve exponentiation, In *the 1st International Conference on Information and Communication Security,*

ICICS 1997, Vol.1334 of Lecture Notes in Computer Science (1997), Springer-Verlag, 282–290. 53

[58] J.A. Muir and D.R. Stinson, Alternative Digit Sets for Nonadjacent Representations, *SIAM Journal on Discrete Mathematics*, **19** (2005), No.1, 165–191. 53

[59] J.A. Muir and D.R. Stinson, Minimality and Other Properties of the width-w Nonadjacent Form, *Math. Comp.*, **75** (2006), No.253, 369–384. 53

[60] V. Müller, Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two, *J. Cryptology*, **11** (1998), No.4, 219–234. 9, 11, 52

[61] D. Naccache, D. M'Raithi, S. Vaudenay and D. Raphaeli, Can D.S.A be Improved? Complexity trade-offs with the Digital Signature Standard, In *Advances in Cryptology – EUROCRYPT '94*, Vol.950 of Lecture Notes in Computer Science (1994), Springer-Verlag, 77–85. 74

[62] National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication 186-3, June 2009. Available at `http://csrc.nist.gov/publications/PubsFIPS.html` 78, 81

[63] National Institute of Standards and Technology, Recommendation for Key Management – Part 1: General (Revised), Special Publication 800-57, March 2007. Available at `http://csrc.nist.gov/publications/PubsSPs.html` 51

[64] T.J. Park, M.K. Lee and K. Park, New Frobenius Expansions for Elliptic Curves with Efficient Endomorphisms, In *the 5th International Conference on Information Security and Cryptology, ICISC 2002*, Vol.2587 of Lecture Notes in Computer Science (2003), Springer-Verlag, 264–282. 10, 52, 81, 82, 88

[65] J. Pastuszak, D. Michalek, J. Pieprzyk, and J. Seberry, Identification of Bad Signatures in Batches, In *the Third International Conference on Theory and Practice in Public Key Cryptography, PKC 2000*, Vol.1751 of Lecture Notes in Computer Science (2000), Springer-Verlag, 28–45. 74

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

[67] R. Rivest, A. Shamir, and L. Adleman, A Method for Obtaining Digital Signatures and Public Key Cryptosystems, In *Communications of the ACM*, **21** (1978), No.2, 120–126. 51

[68] C.P. Schnorr, Efficient signature generation by smart cards, *J. Cryptology*, **4** (1991), No.3, 161–174. 78

[69] J.H. Silverman, *The Arithmetic of Elliptic Curves*, GTM 106, Springer-Verlag, 1986. 5, 13, 53, 81

[70] N.P. Smart, Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic, *J. Cryptology*, **12** (1999), No.2, 141–151. 7, 9, 11, 13, 14, 15, 18, 27, 38, 41, 44, 52, 78

[71] N.P. Smart, and E.J. Westwood, Point Multiplication on Ordinary Elliptic Curves over Fields of Characteristic Three, Cryptology ePrint Archive, Report 2002/114, 2002. Available at http://eprint.iacr.org/2002/114 44

[72] J.A. Solinas, Efficient Arithmetic on Koblitz Curves, *Des. Codes Cryptogr*, **19** (2000), No.2-3, 195–249. 10, 16, 41, 52, 53, 54, 78, 80

[73] Standards for Efficient Cryptography Group, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000. Available at `http://www.secg.org` 52

[74] M. Stanek, Attacking LCCC Batch Verification of RSA Signatures, *International Journal of Network Security*, **6** (2008), No.2, 238–240. 74

[75] T. Takagi, D. Reis, Jr., S.M. Yen and B.C. Wu, Radix-r Non-adjacent Form and Its Application to Pairing-Based Cryptosystem, *IEICE Transactions*, Vol.E89-A, No.1 (2006), 115–123. 10, 12, 13, 30, 44, 52

[76] L.C. Washington, *Elliptic Curves: Number Theory and Cryptography*, CRC Press, Boca Raton, 2003. 18

[77] S.M. Yen and C.S. Laih, Improved Digital Signature Suitable for Batch Verification, *IEEE Transactions on Computers*, **44** (1995), No.7, 957–959. 74

Index

τ-adic expansion, 53
τ-and-add method, 53
$\mathbb{F}_q$-Koblitz curve, 17
ϕ-rNAF, 11, 28
ϕ-GNAF, 11, 18
τ-adic minimal length form (τ-MLF), 71
τ-adic non adjacent form (τ-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-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 (rNAF), 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 τ-adic non-adjacent form (w-τ-NAF), 53, 80