Doctoral Thesis

A Study of Efficient Algorithms for Computing Cryptosystems Using Elliptic Curve

March, 2022

Yuki Nanjo

Graduate School of
Natural Science and Technology
(Doctor's Course)

OKAYAMA UNIVERSITY

博士論文

楕円曲線を用いた暗号システムのための効率的なアルゴリズムに関する研究

2022年3月

南條由紀

岡山大学大学院自然科学研究科

Doctoral Thesis

A Study of Efficient Algorithms for Computing Cryptosystems Using Elliptic Curve

Author:
Yuki Nanjo

Supervisor:
Yasuyuki Nogami
Co-supervisors:
Yoshitaka Toyota
Minoru Kuribayashi

A dissertation submitted to

Okayama University

in fulfillment of the requirements for the degree of

Doctor of Philosophy in Engineering

in the

Graduate School of Natural Science and Technology

Declaration of Authorship

I, Yuki Nanjo, declare that this thesis titled, "A Study of Efficient Algorithms for Computing Cryptosystems Using Elliptic Curve" and the works presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at Okayama University.
- This work has not been submitted for a degree or any other qualification at this University or any other institution.
- Where I have quoted from the works of others, the sources are always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Signed: Yuki Nanjo

Date: March 8, 2022

Abstract

Cryptography is an essential technology for protecting confidential information in various services via the internet. There are mainly two technologies, i.e., symmetric-key cryptography and public-key cryptography, in which the sender and receiver use the same and different keys, respectively. Currently, public-key cryptographies, such as RSA cryptography and elliptic curve cryptography (ECC), are widely used in familiar situations such as SSL/TLS communication. Since the key size of ECC is much smaller than that of RSA for similar security levels, it is considered that ECC will be the principal technology of public-key cryptography in near the future. Besides, there is innovative cryptography with various functions which is based on a pairing defined over an elliptic curve and is so-called pairing-based cryptography (PBC). It is expected for applying PBC for technologies of secure database systems in cloud services. Furthermore, there is post-quantum cryptography (PQC) based on isogenies between elliptic curves, which is called isogeny-based cryptography (IBC). As an example of IBC, there is a supersingular isogeny Diffie-Hellman (SIDH) key exchange protocol. Currently, an international standardization process for PQC is underway, where the candidates involve a cryptosystem based on the SIDH.

To put the new technologies of the PBC and IBC into practical use, the author works on three studies on speeding up the calculation procedure for these cryptosystems. The following is a summary of the background, motivation, and main contribution of each study.

(i) In PBC, since the calculation amount for computing the pairing is a bottleneck, it is an important issue to improve the efficiency of pairing. Since the pairings are typically computed by two steps, which are called the Miller loop and final exponentiation, the efficiency of the pairing depends on these steps. The optimization of each step is examined corresponding to the elliptic curve in which pairing is defined. However, for some elliptic curves suggested for PBC, efficient algorithms for computing the final exponentiations have not been provided. Therefore, the author newly proposes the algorithms for each curve with explicit calculation costs. The proposal results in the improvement of the efficiency of the final exponentiations, which also means that it contributes to the improvement of the efficiency of pairings. The attractive curves which are suggested for

high-performance PBC are also determined.

(ii) The efficiency of the pairing also depends on the constructions of both elliptic curve and finite field in which pairing is defined. To use the optimum constructions, it is necessary to search for appropriate parameters, however, it typically relies on a brute force method. Indeed, there are several previous works that had clarified conditions for finding such parameters for the pairings on specific curves, however, they have not been clarified for various cases. Therefore, the author explicitly provides such conditions for pairings on various elliptic curves that are expected to be used for a long time by applying the mathematical theory. As a result, the proposal contributes to easily find the appropriate parameters that result in efficient pairings without using the brute force method. Moreover, the results also allow us to update the parameters for PBC more flexibly.

(iii) It is a bottleneck in the SIDH to perform the processes of key generations with large computational complexities. In particular, it is desired to improve the efficiency of the processes for computing the construction of the isogenies and their destination. The efficiency of these processes depends on not only the algorithms based on the Vélu's formula but also the construction of the finite field. Although the algorithms have been improved by the previous work, it has been focused on the finite field of a specific construction. Therefore, the author considers the construction of SIDH using the finite fields with various constructions and confirms its performance by an implementation. As a result, it is clarified the candidates of the constructions of finite fields that can realize efficient SIDH. Furthermore, the new candidates end up expanding a range of choices of elliptic curves which can be used for SIDH.

Abstract (Japanese)

インターネット上で展開される多種多様なサービスにおいて，公開・送信される情報から秘密情報を守るためには，暗号技術が不可欠である．暗号技術は大きく分けて，送信者と受信者が同じ鍵を使用する共通鍵暗号と，異なる鍵を利用する公開鍵暗号に分けられる．公開鍵暗号として，現在では RSA 暗号や楕円曲線暗号が普及しており，インターネット上における SSL/TLS 通信など，身近な場面において活用されている．楕円曲線暗号は，RSA 暗号と同等の安全性をより短い鍵長で実現できるため，今後の公開鍵暗号の中核を担う暗号であると考えられている．また，楕円曲線上で定義されるペアリングと呼ばれる写像を応用することにより，暗号化の機能のみならず，様々な付加機能を持つ高機能な暗号が提案されている．この暗号はペアリング暗号と呼ばれ，主にクラウド上でのデータを安全に管理するための新たな技術として注目されている．さらに，楕円曲線上で定義される同種写像に基づいた，耐量子計算機暗号が提案されている．この暗号は同種写像暗号と呼ばれており，SIDH と呼ばれる鍵共有アルゴリズムがその代表例である．現在では，SIDH を基に構成された暗号を候補に含む，耐量子計算機暗号の国際標準化プロセスが進められている．

本論文では，新たな技術の根幹を担うペアリング暗号と同種写像暗号の実用化のために，これらの暗号に用いられる計算手順の高速化に関する三つの研究に取り組んだ．下記ではそれぞれの研究における研究背景，動機，および主な成果を示す．

(i) ペアリング暗号において，楕円曲線上のペアリングの計算量はボトルネックであるため，ペアリングを効率化することが重要な課題である．ペアリングの計算効率は，Miller ループと最終べきと呼ばれる二つの計算ステップの効率に依存する．これらの計算ステップについては，ペアリングが定義される楕円曲線に応じて，それぞれ最適化の検討がなされる．しかし，ペアリング暗号に推奨されている一部の楕円曲線に対しては，効率的な最終べきのアルゴリズムが明らかにされていない．このため，本研究では，それぞれの曲線に対して効率的な最終べきの計算アルゴリズムを新たに提案し，必要な計算コストを明らかにした．その結果，最終べきの計算アルゴリズムの効率化を実現できたことにより，ペアリングの効率化に貢献した．また，計算コストが明らかになったことで，とくに高効率なペアリング暗号に推奨される楕円曲線を明確にできた．

(ii) ペアリングの計算効率は，ペアリングが定義される楕円曲線と，その楕円曲線が定義される有限体の構成にも依存する．最適な楕円曲線と有限体を利用するためには，それに関連する適切なパラメータを探索する必要があるが，その探索には総当たり的手法を

用いることが主流である．一部の特定の楕円曲線上のペアリングについては，先行研究によりパラメータの探索条件が明らかにされている．しかし，様々な楕円曲線上のペアリングについては，そのような探索条件は明らかにされていない．このため本研究では，将来的に長く利用されると考えられる様々な楕円曲線上のペアリングに対して，数学的な理論に基づいて探索条件を明らかにした．その結果，総当たり的手法を使わずに，高効率なペアリングを実現できるパラメータを容易に探索できるようになった．また，これによりペアリング暗号のパラメータがより柔軟に更新できるようになった．

(iii) 同種写像暗号の一つであるSIDHでは，鍵生成のフェーズで必要となる処理の計算量がボトルネックとなっている．とくに，同種写像と写像先の楕円曲線を構成するための計算量の大きさが問題であるため，これらの処理を効率化することが課題である．これらの処理の効率は，Véluの公式に基づいた計算アルゴリズムの効率や有限体の構成に依存する．これまでの研究により，計算アルゴリズムの改善が行われているが，特定の構成の有限体のみが着目されてきた．このため本研究では，様々な構成による有限体を用いてSIDHを構成することを検討し，その性能を実装により確認した．その結果，高効率なSIDHを実現できる有限体の構成の候補を明らかにした．さらに，新たな候補の有限体を用いることにより，実用的なSIDHを構成できる楕円曲線の選択肢を広げることができた．

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Professor Yasuyuki Nogami for his support throughout my bachelor, master, and doctorial courses at Okayama University. Without his extraordinary understanding and cooperation, I could not be able to finish my doctoral research completely. I also appreciate to co-supervisors, Professor Yoshitaka Toyota and Associate Professor Minoru Kuribayashi who gave me a lot of effort to improve this thesis. They also gave me knowledge of electronics and networks through the classes in my courses.

I sincerely thank Senior Assistant Professor Takuya Kusaka for giving me many pieces of advice for my research and student life. I also would like to thank Assistant Professor Yuta Kodera who is my respected senior and gave me a lot of influence from his great attitude for research. Special thanks also to students members in Information Security Laboratory for creating a great work atmosphere and their generous support.

I would like to thank Associate Professor Masaaki Shirase in Future University Hakodate for giving me great support and allowing me to study in his laboratory for 1 year during my doctoral course. I never forget the precious time of studying in Hakodate and also the knowledge that he had kindly taught me. All of the co-authored research is my irreplaceable treasure.

Thanks to JSPS for employing me as a research fellow DC1. I sincerely acknowledge all the funds that afforded my research activities.

I am grateful to administrative officers, Ms. Yumiko Kurooka, Ms. Midori Onishi, and Ms. Masako Okamoto, who processed my official documents. I would like to thank Ms. Miyuki Tachino and everyone who corporated the works related to the payment processing of JSPS KAKENHI.

Finally, I would like to show my great thanks to my family for allowing me to learn in the doctorial course. They gave me many encouragements which made me motivated to carry out research. I would like to repay the favor for them as much as they have been supporting me for a long time.

Previously Published Materials

The following papers have been published or presented, and contain material based on the content of this thesis.

Peer-Reviewed Journal Papers:

1. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "Efficient Ate-Based Pairing over the Attractive Classes of BN Curves". In: Information Security Applications. WISA 2018. Lecture Notes in Computer Science, vol. 11402, pp. 55–67, 2019.

2. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "A Performance Analysis and Evaluation of SIDH Applied Several Implementation-Friendly Quadratic Extension Fields". In: International Journal of Networking and Computing (IJNC), vol. 10, no. 2, pp. 227–241, 2020.

3. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "A Construction Method of an Isomorphic Map between Quadratic Extension Fields Applicable for SIDH". In: IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E103-A, no. 12, pp.1403–1406, 2020.

4. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "Improvement of Final Exponentiation for Pairings on BLS Curves with Embedding Degree 15". In: IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E104-A, no. 1, pp. 315–318, 2021.

5. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "Restrictions of Integer Parameters for Generating Attractive BLS Subfamilies of Pairing-Friendly Elliptic Curves with Specific Embedding Degrees". In: International Journal of Networking and Computing (IJNC), vol. 11, no. 2, pp. 383–411, 2021.

Peer-Reviewed Conference Papers:

6. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "A Performance Analysis and Evaluation of SIDH with Implementation-Friendly Classes of Quadratic Extension Fields". In: 2019 Seventh International Symposium on Computing and Networking (CANDAR), pp. 178–184, 2019.

7. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "A Technique for Fast Miller's Algorithm of Ate Pairings on Elliptic Curves with Embedding Degrees of Multiple of Three". In: 2020 35th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), pp. 283–287, 2020.

8. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "An Explicit Formula of Cyclotomic Cubing Available for Pairings on Elliptic Curves with Embedding Degrees of Multiple of Three". In: 2020 35th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), pp. 288–292, 2020.

9. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "Specific Congruence Classes of Integer Parameters for Generating BLS Curves for Fast Pairings". In: 2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW), pp. 348–354, 2020.

10. **Yuki Nanjo**, Masaaki Shirase, Yuta Kodera, Takuya Kusaka, Yasuyuki Nogami. "Calculation Costs Estimations of Final Exponentiation for Pairing-Friendly Elliptic Curves Resistant to Special TNFS". In: 2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), pp. 229–232, 2021.

11. **Yuki Nanjo**, Masaaki Shirase, Yuta Kodera, Takuya Kusaka, Yasuyuki Nogami. "A Construction Method of Final Exponentiation for a Specific Cyclotomic Family of Pairing-Friendly Elliptic Curves with Prime Embedding Degrees". In: 2021 Ninth International Symposium on Computing and Networking (CANDAR), pp. 148–154, 2021.

12. **Yuki Nanjo**, Masaaki Shirase, Yuta Kodera, Takuya Kusaka, Yasuyuki Nogami. "Efficient Final Exponentiation for Pairings on Several Curves Resistant to Special TNFS". In: 2021 Ninth International Symposium on Computing and Networking (CANDAR), pp. 48–55, 2021.

Domestic Conference Papers:

13. **Yuki Nanjo**, Masaaki Shirase, Takuya Kusaka, Yasuyuki Nogami. "A Performance Analysis of Supersingular Isogeny Diffie-Hellman with Several Classes of the Quadratic Extension Fields". In: IEICE Technical Report, vol. 119, no. 140, pp. 207–214, 2019.

14. 白勢政明, **南條由紀**. "任意のBLS曲線の最終べきのhard partについて". In: IEICE Technical Report, vol. 120, no. 112, pp. 105–110, 2020.

Contents

List of Figures

List of Tables

List of Symbols

$\mathbb{C}$	set of all complex numbers
$\mathbb{R}$	set of all real numbers
$\mathbb{Q}$	set of all rational numbers
$\mathbb{Z}$	set of all integers
G	group
H	subgroup of G
G/H	quotient group of G by H
$G_1 \times G_2$	product of two groups G_1 and G_2
R	ring
I	ideal
R/I	quotient ring of R by I
(a)	principal ideal of R generated by an element a in R
F	field
$\mathrm{char}(F)$	characteristic of F
$R[x]$	polynomial ring (or field) of R
$(f(x))$	principal ideal of $R[x]$ generated by a polynomial $f(x)$ in $R[x]$
$R[x]/(f(x))$	quotient ring of $R[x]$ by $(f(x))$
p	prime
$\mathbb{F}_p$	prime field of order p
q	prime or power of prime

$\mathbb{F}_q$	finite field of order q
$\mathbb{F}_q^*$	multiplicative group of $\mathbb{F}_q$
$\overline{\mathbb{F}}_q$	algebraic closure of $\mathbb{F}_q$
$\mathbb{F}_{q^n}$	n-th extension field of $\mathbb{F}_q$
$N_{\mathbb{F}_q/\mathbb{F}_p}(a)$	norm of a in $\mathbb{F}_q$ over $\mathbb{F}_p$
$(\frac{a}{p})$	Legendre symbol
$\mathbb{A}^n$	affine n-space
$\mathbb{P}^n$	projective n-space
E	elliptic curve
$E/\mathbb{F}_q$	elliptic curve defined over $\mathbb{F}_q$
$E(\mathbb{F}_q)$	$\mathbb{F}_q$-rational point group of E
P	point on E
nP	point on E of P multiplied by n
$\mathcal{O}$	point at infinity
$[n]$	multiplication endomorphism in E by n
π_q	q-th power Frobenius endomorphism in E or $\mathbb{F}_q$
r	prime order of a subgroup of $E(\mathbb{F}_q)$
k	embedding degree of E (with respect to q and r)
t	trace of Frobenius
D	CM discriminant
ρ	ρ-value that shows ratio between q and r
$E[r]$	r-torsion subgroup of E
E'	twist of E
d	degree of twist
ϕ_d	twisting isomorphism from E' to E of degree d

$\mathbb{F}_q(E)$	function field of E over $\mathbb{F}_q$
$\mathrm{div}(f)$	divisor of a rational function f in $\mathbb{F}_q(E)$
$f_{n,P}$	function with $\mathrm{div}(f_{n,P}) = n(P) + (nP) - (\mathcal{O})$
l_{P_1,P_2}	sloped line function with $\mathrm{div}(l_{P_1,P_2}) = (P_1)+(P_2)+(-(P_1+P_2)) - 3(\mathcal{O})$ for two points P_1 and P_2
v_P	vertical line function with $\mathrm{div}(v_P) = (P) + (-P) - 2(\mathcal{O})$
$\mathcal{G}_1$	base-field subgroup of $E[r]$
$\mathcal{G}_2$	trace-zero subgroup of $E[r]$
μ_r	subgroup of r-th root of identity in $\mathbb{F}^*_{q^k}$ of order r
ϕ	Euler's totient function
Φ_n	n-th cyclotomic polynomial
$G_{\Phi_k(q)}$	cyclotomic subgroup of $\mathbb{F}^*_{q^k}$ of order $\Phi_k(q)$
e	pairing
e_{W_r}	Weil pairing on E
e_{T_r}	Tate pairing on E
$\tilde{e}_{T_r}$	reduced Tate pairing on E
e_{a_T}	ate pairing on E
$e_{a_{c_i}}$	ate-like pairing on E
e'_{a_T}	ate pairing on E'
φ	isogeny between two elliptic curve
$\mathrm{End}(E)$	Endomorphism ring of E
$\deg(\varphi)$	degree of isogeny
$\ker(\varphi)$	kernel of isogeny
$j(E)$	j-invariant of E
O	Landau symbol

L_q	complexity symbol for variants of number field sieve
gcd	greatest common divisor
ECC	elliptic curve cryptography
PBC	pairing-based cryptography
IBC	isogeny-based cryptography
DH	Diffie-Hellman
DHP	Diffie-Hellman problem
DDHP	decisional Diffie-Hellman problem
DLP	discrete logarithm problem
ECDHP	elliptic curve Diffie-Hellman problem
ECDDHP	elliptic curve decisional Diffie-Hellman problem
ECDLP	elliptic curve discrete logarithm problem
BDHP	bilinear Diffie-Hellman problem
BDDHP	bilinear decisional Diffie-Hellman problem
SSCDHP	supersingular computational Diffie-Hellman problem
SSDDHP	supersingular decision Diffie-Hellman problem
CSSIP	computational supersingular isogeny problem
MNT	Miyaji-Nakabayashi-Takano families
BLS	Barreto-Lynn-Scott families
BN	Barreto-Naehrig family with $k = 12$
KSS	Kachisa-Schaefer-Scott families

For Mum.

Chapter 1

Introduction

This chapter introduces the related literature review, problems, and contributions of this work. This thesis starts to describe cryptology and its roles in information security by referring to the recent textbooks [Shi15; Koj20] written in Japanese.

1.1 Cryptography

In recent years, purchases of goods and services on the internet have been used in everyday life. When using such systems, personal information is sent via web communication for reliable destinations. However, there are risks that the information is eavesdropped on or impersonated by a third party. In order to avoid such risks, it is necessary to discuss information security. It lies on three principles of confidentiality, integrity, and availability, i.e.,

- Confidentiality measures protect information from unauthorized access and misuse;
- Integrity measures protect information from unauthorized alteration;
- Availability measures protect timely and uninterrupted access to the system.

Cryptography is a general term for technologies that guarantee information security. The subject which threatens information security is called an attacker. In cryptography, encryption is the process of encoding information. This process converts the original representation of the information, known as plaintext, into an alternative form known as ciphertext. The operation of restoring the ciphertext is decryption. The information that is used for encryption and decryption is called a key. Historically, various forms of encryption have been used to aid in cryptography. Modern encryption schemes use the concepts of symmetric-key and public-key.

1.1.1 Symmetric-key cryptography

Symmetric-key cryptography, a.k.a, private-key cryptography, uses the same cryptographic private keys for both encryption and decryption. It has a long history, which has been existed at the time of the Greek and Roman empire. Julius Caesar used a simple shift and substitute system, which is known as the Caesar cipher. In modern times, symmetric-key encryption can use either stream ciphers or block ciphers. The stream ciphers encrypt the digits or bytes of a message one at a time, e.g., ChaCha20 [Ber+08] by Bernstein in 2008, which is a modification of Salsa20 [Ber08]. The block ciphers take a certain number of bits of a message and encrypt them as a single unit, by padding the message to be a multiple of the size of the block, e.g., the advanced encryption standard (AES) approved by NIST in 2001, Camellia [Aok+00] developed by Mitsubishi Electric and NTT in 2000, and CLEFIA [Shi+07] developed by Sony in 2007. The block cipher is broken by brute force search for all candidates of keys. In other words, the block cipher with the n-bit key size can be broken in 2^{n-1} trials on average. The security of block ciphers is based on the assumption that the amount of calculation required for an attack cannot be solved with less calculation amount.

1.1.2 Public-key cryptography

Public-key cryptography, or asymmetric cryptography, uses pairs of keys, which was born out of the problem of how to securely send the key of symmetric-key cryptography in around 1970. One key is a public key, which anyone can use to encrypt the plaintext. Another key is a private key, which a receiver needs to decrypt the ciphertext. The security of public-key cryptography is typically based on the assumption that it is computationally difficult to obtain the private key from a given public key and ciphertext. To guarantee such difficulty, mathematical structures are often used for constructing public-key cryptography. In the following, elementary public-key cryptosystems are introduced with a description of security categories.

Key exchange

In 1976, Diffie and Hellman published a method of securely exchanging cryptographic keys over a public channel in [DH76], which is known as the Diffie-Hellman (DH) key exchange. In the DH key exchange, it is needed to use a set of finite numbers of elements, which is called a *finite field*, in which basic arithmetic operations such as addition, subtraction, multiplication, and division are defined. More strictly, we only use a subset of the finite field, which is called a *group*, in which the multiplication and divisions are defined. The private key is securely shared between the two parties, Alice and Bob, by the following steps:

1. Alice and Bob publicly agree to use a finite field F and an element $g \in F$.
2. Alice chooses a private key a, then sends Bob $A = g^a \in F$, and Bob chooses a private key b, then sends Alice $B = g^b \in F$.
3. Alice computes $s_A = B^a \in F$, and Bob computes $s_B = A^b \in F$. Then, Alice and Bob share the same secret $s_A = s_B$, since $s_A = B^a = g^{ba} = g^{ab} = A^b = s_B \in F$.

The security of the DH key exchange is based on an assumption that the problem for computing s from $g, A, B \in F$, which is called the DH problem (DHP), is difficult to solve. The assumption is called the computational DH assumption. The most efficient known way to solve the DHP is to solve a problem for computing x such that $h = g^x$ from $g, h \in F$, which is called a discrete logarithm problem (DLP). At this time, there are no known efficient algorithms for solving DLP with a realistic time in a classical computer.

Encryption

In 1977, Rivest, Shamir, and Adleman firstly introduced public-key encryption, which was published in [RSA78] and is widely known as RSA encryption. The security of RSA encryption relies on a factorization problem. In [Elg85], Elgamal modified the DH key exchange and constructed another public-key encryption. This encryption is known as the ElGamal encryption, in which a message is securely sent from Bob to Alice by the following steps.

- Key generation: Alice uses a finite field F and an element $g \in F$, chooses an integer a, and computes $A = g^a \in F$. Alice sends a public key (F, g, A) and secretly takes a private key a.
- Encryption: Bob maps a plaintext M to an element m in F, chooses a private key b, computes $B = g^b, s_B = A^b, n = m \cdot s_B \in F$, and sends a ciphertext (B, n) to Alice.
- Decryption: Alice computes $s_A = B^a$, obtains m by computing $m = n \cdot s_A^{-1} \in F$, and maps m back to the plaintext M. Then, Alice obtain the correct m since $n \cdot s_A^{-1} = m \cdot s_B \cdot s_A^{-1} = m \cdot A^b \cdot (B^a)^{-1} = m \cdot g^{ab} \cdot (g^{ba})^{-1} = m$.

It is considered that anyone cannot decrypt the ElGamal encryption without the private key under the computational DH assumption. The encryption scheme also needs indistinguishability of ciphertexts. Given two messages M and M' and encryption of either one of the messages, the scheme has to hold property such that anyone cannot guess whether the given ciphertext is encryption of M or M' with better probability than 1/2, which is also known as semantic security. The semantic security of the encryption is based on an assumption that the problem for determining $xy = z$ or not for given $g, X, Y, Z \in F$ such that $Z = g^x$, $Y = g^y$, and $Z = g^z$, which is called the decisional Diffie-Hellman problem (DDHP), is difficult to solve. If DHP is broken, DDHP is also broken.

Signature

If the public key has been tampered with during communication, the security of the public-key cryptosystems is not guaranteed. Such kind of attack is known as a man-in-the-middle attack. Thus, it is necessary to use a digital signature or message authentication code to confirm whether the public key is the correct one or not. The digital signature can be easily constructed by modifying the RSA and ElGamal encryption, which are called the RSA signature and the ElGamal signature, respectively. Currently, a variant of the ElGamal signature, which is developed at the NSA in [KD13] and is known as the digital signature algorithm (DSA), is much more widely used than the original ElGamal signature. However, since the digital signatures also require a public key, we go back to the problem of how to send the public key. This problem is solved by issuing a certificate authority that can trust the certificate for the correctness of the public key.

Security levels

The security of the cryptosystems is typically analyzed by computational complexity or time complexity for executing algorithms for solving mathematical problems where the cryptosystems rely on. Then, the complexities are often expressed by using the Landau symbol O. There are several security levels defined in terms of resources needed for AES such that an attacker requires computational resources comparable to or greater than those required for AES with n-bit keys that offer n-bit security levels. Practically, n is often chosen as $n = \{128, 192, 256\}$. This allows us for meaningful performance comparisons between different cryptosystems. For example, in order to offer the 128-bit security level, RSA must be designed to have 3072-bit keys but elliptic curve cryptography which is introduced in the next subsection requires only 256-bit keys.

1.1.3 Cryptography using elliptic curves

Cryptography can be constructed by using various mathematical structures. Table 1.1 summarizes major cryptosystems with the perspective of their functions and structures. The cryptosystems introduced in Sect. 1.1.2 are based on the problems related to factorization and finite fields. This subsection introduces other cryptosystems which are using elliptic curves and maps defined by using elliptic curves, i.e., pairings and isogenies.

Elliptic curve cryptography

In the middle of the 1980s, Miller [Mil85] and Koblitz [Kob87] independently proposed cryptosystems using an equation $E : y^2 = x^3 + ax + b$, which is called an *elliptic curve*. In the cryptosystems, we use a set of points on E which forms a group under a geometric addition such that the third point on E is given by $R = P + Q$ from two points P and Q

Table 1.1: Cryptosystems.

Structures	Key exchange	Encryption	Signature	Others
Factorization	-	RSA	RSA sig.	-
Finite field	DH	ElGamal	ElGamal sig. DSA	-
Elliptic curve	ECDH	EC ElGamal	ECDSA	-
Pairing	Tripartite DH	ID-based	Short sig.	Attribute-based, searchable enc.
Isogeny	SIDH CSIDH	SIKE	CSI-FiSh SQISign	-

on E. The cryptosystems using the multiplicative group of the finite field can be easily modified by using the group of the elliptic curve. For example, there are the elliptic curve DH (ECDH) key exchange, elliptic curve ElGamal (EC ElGamal) encryption, and EC ElGamal signature/Elliptic curve DSA (ECDSA). The security of these cryptosystems is based on the difficulty of a problem for computing an integer s such that $Q = sP$ for given P and Q of E, which is called the elliptic curve discrete logarithm problem (ECDLP). The cryptography based on the difficulty of ECDLP is called elliptic curve cryptography (ECC). The key size of ECC is much smaller than that of cryptography based on the finite field and factorization for similar security levels since algorithms for solving the factorization problem and DLP cannot be extended for solving ECDLP. Therefore, when high security is required for environments of limited resources, e.g., IoT devices, it is more advantageous to use ECC. Furthermore, ECC will be the principal technology of public-key cryptography in the future.

Pairing-based cryptography

In 2000, Joux [Jou00] and Sakai et al. [SOK00] independently proposed cryptosystems using a map defined over an elliptic curve, which is called a *pairing*. The pairing is a map with two inputs of points on an elliptic curve and one output of an element in a finite field. The important fact is that the pairing has special properties, which are called bilinear and non-degenerate. The properties allow us to realize many cryptosystems with convenient functions involving ones that could not be realized by the previous cryptography. In [Jou00], Joux introduced an one-round DH key exchange for three parties based on the pairing, however, DH/ECDH is a key exchange for two parties. Sakai et al. [SOK00] and Boneh et al. [BF01] introduced ID-based encryption, where use user's unique ID, e.g., email address, can be used as a public key. In [Bar+02], Barreto et al. introduced a short signature, which allows shorter signatures than the previous signatures for a similar level. The short signature has been adopted in Ethereum[1], which is a decentralized, open-source blockchain with smart contract functionality. In addition to this, there are many innovative cryptosystems, e.g., searchable encryption [Bon+04] where the encrypted data can be searched without decryption; attribute-based encryption [SW05], where is a type of public-key encryption in which the secret key of a user and the ciphertext are dependent upon attributes, and so on, which are expected to be applied for the secure database systems in cloud technologies. Such convenient cryptosystems are called pairing-based cryptography (PBC). Note that the security of PBC is typically based on the difficulties of both DLP and ECDLP.

Post-quantum isogeny-based cryptography

We have seen that many cryptosystems rely on mathematical problems which are considered to be difficult for solving by using a classical computer. However, in 1994, Shor described that the problems could be solved in a realistic time by using an algorithm executed by a quantum computer [Sho94]. Although the large-scale quantum computer has not been developed at this time, it is required to develop cryptography that cannot be solved even though the quantum computer is applied. Such cryptography is so-called post-quantum cryptography (PQC).

There are several candidates of PQC, e.g., code-base cryptography [McE78] introduced by McEliece in 1978, multivariate cryptography [MI88] by Matsumoto and Imai in 1988, and lattice-based cryptography [AD97] by Ajtai and Dwork in 1997. Some candidates were published with other motivations which are not related to PQC before Shor's result. In 2006, Couveigues [Cou06] and Rostovtsev and Stolvunov [RS06] proposed another candidate of PQC using maps $\varphi : E \to \tilde{E}$ between elliptic curves E and $\tilde{E}$, which are called *isogenies*. The cryptosystems using the isogenies are called isogeny-based cryptography (IBC). The security of IBC is based on the difficulty of a problem for computing an isogeny $\varphi : E \to \tilde{E}$ from given two elliptic curves E and $\tilde{E}$, which is called an isogeny problem. At this time, there is no known efficient algorithm for solving the problem in practical time even though the quantum computer is applied. The problem complexities also make the key sizes of IBC significantly smaller than the other candidates.

In 2011, Jao and De Feo proposed a variant of the DH key exchange based on the isogeny problem for elliptic curves classified into supersingular in [JDF11], which is known as the supersingular isogeny DH (SIDH) key exchange. In 2017, one kind of encryption by using the SIDH, which is called an isogeny-based key encapsulation (SIKE), is proposed by [Aza+17] and is submitted to the NIST standardization process on PQC. The SIKE is the only candidate of which security is based on the isogeny problem and is selected as an alternative candidate for the round 3 submissions at this time. Castryck et al. proposed

[1] https://ethereum.org

another key exchange protocol using the isogenies in [Cas+18], which is a commutative SIDH (CSIDH) in 2018. The CSIDH is more compatible with the original DH key exchange than the SIDH, however, it requires a huge amount of calculation costs. In 2019, Beullens et al. proposed a signature algorithm CSI-FiSh [BKV19] with a short key length and signature length. In 2020, De Feo et al. also presented SQISign [DF+20] which has higher performance in signature verification than CSI-FiSh.

1.2 Previous research and remaining problems

According to recent cryptographic trends, the author focuses on PBC and IBC. Since the pairing on the elliptic curve is the important tool for realizing PBC, it is important to improve its performance and usability. There is a possibility that several techniques for elliptic curve or pairing are extended for the operations related to the SIDH of IBC and contribute to improving its performance. The following summarizes the previous research and remaining problems related to the pairings and SIDH, which motivate this work.

1.2.1 For the pairing on elliptic curve

After the publication of the innovative protocols based on the pairings, researchers have been working on constructing elliptic curves, which have the advantage of pairings. Particularly, the practical pairing should be defined over an elliptic curve with a small embedding degree k, which is one of the parameters for specifying the elliptic curve. Such the curve is called a *pairing-friendly elliptic curve* and is typically not easy to find by random search. One of the construction methods for such curves is the use of the supersingular elliptic curves or Cocks-Pinch method, which was given in the unpublished manuscript [CP01] by Cocks and Pinch. The other remaining constructions of pairing-friendly curves fall into the category of families of curves described in [MNT01; BLS02; BN05; KSS08], which produce currently well-used curves, such as MNT, BLS, BN, and KSS curves. The families of curves have integer parameters for determining unique curves. The research of pairing-friendly curve constructions is extended and collected by Freeman et al. in [FST10]. Note that there are several methods [Per+11; CLN11; Cos12] for generating pairing-friendly curves for certain families which are advantages for the pairings.

Since the pairing computation can be the bottleneck of the protocols, researchers also have been working on optimizing the pairings. The pairings on elliptic curves, such as a reduced Tate pairing and its variants, are typically carried out by two steps, which are the Miller loop and extra exponentiation in the finite field to bring the output of the Miller loop to the unique value. The Miller loop is computed by an iterative algorithm, which is proposed by Miller in [Mil04] and is called Miller's algorithm. To shorten the loop length, Hess et al. [HSV06] and Vercauteren [Ver09] proposed modification of the pairings. Around 2010, Costello et al. [CLN10] and related works [ZL12] presented efficient formulas for computing the Miller loop using another elliptic curve, which is called *twist*. The higher degree twist typically makes the computation of the Miller loop to be faster. The extra exponentiation is called the final exponentiation and it becomes more of a computational bottleneck with the curves with large k. Since the final exponentiation has the specific exponent, Scott et al. [Sco+09] and Fuentes et al. [FCKRH11] provided methods for constructing efficient algorithms using the Frobenius map which is a map in the finite field with the low complexity.

In recent years, the security of the pairings are well-analyzed. In 2016, Kim et al. provided notable developments of the tower number field sieve (TNFS) algorithm [KB16; KJ17], which is one of the best-known algorithms for solving DLP. Particularly, there are many results [FK19; BD19; GS19; Gui20] that show that the special variant TNFS (STNFS) are very efficient in finite fields that are target groups of the pairings on the curves. Although it was previously considered that the BN curve with $k = 12$ is the best choice for the pairings at the 128-bit security level, the results of the analyses expanded the range of the choices of curves. In [RNL19], Barbulescu et al. reported that there are many elliptic curves with various k, e.g., $k = 12$, 14, 15, 16, 24, and 27, which have a good performance of the pairings. Moreover, Fouotsa et al. also suggested using the pairings on the curves with $k = 9$, 15, and 27 in [FMP20]. In 2020, Guillevic also provided a shortlist of the curves with $k = 6$, 8, 10, 11, 13, 14, and 16 that have a resistance to the STNFS in [Gui20]. Their shortlist involves a curve with $k = 12$ which is given by Fotiadis and Konstantinou in [FK19] and is called FK curve, and curves with $k = 6, 8$ found by using Cocks-Pinch method in [GMT20].

There are the following two problems, which motivate this work.

- The first problem: The efficient formulas and algorithms for computing the pairings have to be found corresponding to the curves. However, for several curves that are newly suggested for the pairings in [FMP20; Gui20], there is no work for providing efficient algorithms for computing the final exponentiation or there are possibilities of improvement. It is desired to clarify that and to provide the explicit calculation costs of the pairings for finding one of the best choices of the curves. As one more issue related to the final exponentiation, it is also desired to overcome the problem that the existing methods [Sco+09; FCKRH11] require complicated works for each curve.

- The second problem: The pairings with the family of curves require initial settings such that finding an integer parameter and constructing the curves and finite field. During the search of the parameter, we need to consider that not only the security of the pairings but also the efficiency of their computations strongly depends on the settings. However, since it is typically complicated to handle the favorite constructions of curves and finite fields, it is desired to establish some convenient ways for the settings which are an advantage for the pairings. There exist the previous works [Per+11; CLN11; Cos12] which came from the same motivation, however, such works treated limited curves.

1.2.2 For the SIDH

Since the SIDH has a short history, research on optimizations and security analyses are published in very recent years. In 2011, Jao and De Feo introduced the SIDH together with its practical constructions in [JDF11]. According to [JDF11], the practical SIDH uses the supersingular elliptic curves given by the Montgomery form defined over a specific finite field, which is called an optimal extension field (OEF) given by [BP01]. Then, one can enjoy efficient formulas for computing point arithmetics and small-degree isogenies and are available. The large-degree isogenies required for the SIDH are efficiently computed by decomposed into low-degree isogenies with point multiplications. Around 2017, when the SIKE was submitted for the NIST standardization process, Costello et al. [CLN16; CH17], Faz-Hernández [FH+17], Renes [Ren18] revised the formulas for computing the small-degree isogenies and point arithmetics. Besides, Adj et al. [Adj+18] and Jaques and Schanck [JS19] made the security analyses of the SIDH, which show that the SIKE used rather conservative security estimates. This means that significantly smaller parameters can be used than the SIKE developer thought. Currently, there are mainly four parameters for specifying the curves that are suggested for the SIKE. Note that Matsuo [Mat19] and Costello [Cos20] independently proposed the modification of the SIDH using twist. Their modifications provide new candidates of the curves used for the practical SIDH.

There is the following problem, which also motivates this work.

- The third problem: It is needed to optimize the operations of the isogenies and point arithmetics for the practical SIDH. Although the previous works [CLN16; CH17; FH+17; Ren18] mainly focus on revising the formulas of the operations, there is a possibility that the performance of the operations can be improved by changing the construction of the finite field in which curves are defined. There are several construction methods of the finite fields which have slightly lower computational complexities of the multiplication than the OEF used for the typical SIDH. Although there is a problem that the elliptic curves usable for the SIDH are very limited, the change of the finite field might contribute to expanding the range of the elliptic curves.

1.3 Major contributions

The author tries to overcome the problems which are summarized in the previous section. The summary of major contributions of this thesis is given as follows:

- The first contributions: For the BLS curves with $k = 15$, the algorithm for computing the final exponentiation is improved by using the property related to the family. It is also found that the improvement techniques can be extended for the BLS curves with any k. This contributes to obtaining efficient final exponentiation algorithms for any BLS curves with a small effort. Note that Hayashida et al. achieved similar results for any family of curves in [HHT20] at the same time as this work publication.

  Although there are no efficient algorithms for computing the final exponentiations for the curves with $k = 10, 11, 13$, and 14, the author provides them by applying the existing construction methods. The author also provides explicit calculation costs for executing the algorithms for an estimation of the performance of the pairings on those curves. Comparing the estimation results between the shortlist curves in [Gui20], it is found that the BLS curve with $k = 12$, FK curve with $k = 12$, and Cocks-Pinch curve with $k = 6$ are attractive choices for efficient pairings.

  The author proposes a new method for constructing the final exponentiation algorithm for the specific cyclotomic family of curves with any prime k given by $k = 6n + 1$ for $n > 0$. It is found that the proposed method results in one of the same state-of-the-art algorithms for computing the final exponentiation produced by the previous method [FCKRH11] for the cases of $k = 7, 13$, and 19. Unlike the previous method, the proposed method can immediately produce the algorithms by using mathematical formulas. Moreover, the proposed method can produce the algorithm with $O(n)$, however, the latest method [HHT20] generates the algorithm with $O(n^2)$.

- The second contribution: To overcome the second problem, the author proposes a simple method for generating the BN and BLS curves that have the advantage for the pairings-based cryptography by finding parameters under specific restrictions. The proposed method can generate the curves which automatically give rise to the attractive settings of the curve and finite field for fast pairings. The proposal also contributes to the smooth update of the parameters of the pairings corresponding to the improvement of the security analyses. Moreover, since the proposed method can generate the BLS curves with $k = 2^m \cdot 3$ and 3^n with any $m, n > 0$, the method will be useful for the researcher and implementer of the pairings for a long time. Note that the proposal can result in the same curves as that of the previous method for the BLS curves with $k = 24$ in [CLN11].

- Third contributions: To overcome the third problem, the author focuses on not only OEF by [BP01] but also other construction methods [NSM03; KAH00] which generate an all-one polynomial extension field (AOPF) and extension fields with normal basis representation (EFN). The applicability of the finite fields shows that not only the OEF but also AOPF and EFN can be used for SIDH. Moreover, the EFN allows us to use new curves which have not been used for the previous SIDH. As a result of the implementation of the SIDH with the possible finite fields, it is found that the performances of SIDH with OEF and new candidates are competitive. As one of the additional contributions, a simple map from the OEF to any finite fields is provided for simple parameter settings of the SIDH.

1.4 Organization

The rest of this thesis is organized as follows: Chapter 2 provides the fundamentals of the finite field, elliptic curves, pairings, and isogenies from the definition of the algebraic systems. Although the descriptions of the fundamentals are not short, it is necessary to understand the whole of the contributions. In Chapters 3, 4, and 5, the first, second, and third contributions are provided, respectively. The background and motivations are reviewed and then the previous works are formally described. After that, the proposed contents are described. In Chapter 6, the conclusion is drawn with future works. Additional descriptions of Chapters 3–5 are summarized in Apps. C–F.

Chapter 2

Fundamentals

This chapter describes the mathematical fundamentals of finite fields, elliptic curves, pairings, and isogenies by referring to the textbooks [Sil09; EMJ17]. To make it easier to understand, this thesis provides the descriptions together with some examples.

2.1 Group, ring, and field

This section provides a standard background of algebraic systems, which are materials on sets in which arithmetics are defined. There are descriptions of groups, rings, and fields that are very fundamental for discussing cryptography.

2.1.1 Groups

The definition of the groups are described in below.

Definition 2.1. (Group) Let G be a set and $\circ$ be a binary operation such that $a \circ b \in G$ for all $a, b \in G$, i.e., $\circ$ is defined on G. The pair of G and $\circ$, which is denoted by $(G, \circ)$ and is called a *group* if the following conditions are satisfied:

1. Associative: $(a \circ b) \circ c = a \circ (b \circ c)$ for all $a, b, c \in G$.
2. Identity: There exists $e \in G$ such that $a \circ e = e \circ a = a$ for all $a \in G$.
3. Inverse: Given $a \in G$, there exists $b \in G$ such that $a \circ b = b \circ a = e$.

Then, we say G forms a group under $\circ$.

If we drop condition 3 of Definition 2.1, $(G, \circ)$ is called a *monoid*. If we drop conditions 2 and 3, $(G, \circ)$ is called a *semigroup*. Conversely, if we add the condition such that $a \circ b = b \circ a$ for all $a, b \in G$ to Definition 2.1, $(G, \circ)$ is called an *abelian group* or *commutative group*. Hereafter a group is written by $(G, \circ)$ if the operation is needed to be specified; otherwise, the group is simply referred to by the corresponding set of G.

The number of elements of G, i.e., $\#G$, is called an *order* of the group G. If $\#G$ is a finite number, G is called a *finite group*; otherwise, G is an *infinite group.* For $a \in G$ and positive integer m, let $[m]a$ be an element in G by applying $\circ$ for m-terms of a, i.e., $[m]a = a \circ a \circ \cdots \circ a$. An *order* of a is the smallest number n such that $[n]a = e$. If G is a set given by $\langle a \rangle = \{e, a, [2]a, \ldots, [n-1]a\}$ for an element $a \in G$, G is called a *cyclic group* and a is a *generator* of G. In this thesis, there appears a set given by $\langle a, b \rangle = \{[m_1]a + [m_2]b\}$ for elements $a, b \in G$ and any integers m_1, m_2.

Several examples of infinite groups are provided below. To present finite groups, it is needed to study quotient groups described in the next subsection.

Example 2.2. (Group of integers) Let $\mathbb{Z}$ be a set of integers and let $+$ be a natural addition defined on $\mathbb{Z}$. Then, $(\mathbb{Z}, +)$ is an abelian group. It is an infinite cyclic group of which 1 or -1 can be generators.

Example 2.3. (Group of real numbers) Let $\mathbb{R}^*$ be a set of real numbers excluding 0 and let $\cdot$ be a natural multiplication defined on $\mathbb{R}$. Then, $(\mathbb{R}^*, \cdot)$ is an abelian group. It is not a cyclic group since there do not exist generators.

2.1.2 Quotient groups and homomorphisms

Suppose that G is an abelian group since the commutative property gives rise to simplifying discussions. Let H be a subset of G such that H forms a group under the same binary operation of G. Then, H is called a *subgroup* of G. Given a single element $a \in G$, a *coset* is a subset of G defined by

$$a \circ H = \{a \circ h : h \in H\}, \tag{2.1}$$

which is a subset of G. Note that actually we need to distinguish the direction to apply a from left and right but it can be ignored as long as we work on the commutative group. The number of cosets is called the *index of H in G*, written as $[G : H]$. Then, it is obtained $\#G = [G : H]\#H$ from Lagrange's theorem if G is a finite group. Let G/H be a set of all cosets defined by

$$G/H = \{a \circ H : a \in G\}. \tag{2.2}$$

For all $a \circ H, b \circ H \in G/H$, a binary operation $*$ can be defined by $a \circ H * b \circ H = (a \circ b) \circ H \in G/H$. Then, G/H forms a group under $*$, which is called a *quotient group* or *factor group.* Note that the order of G/H is explicitly given by $\#(G/H) = \#G/\#H$ with a finite group G.

There is an easier understanding of the quotient groups with the following description. If two element $a_1, a_2 \in G$ produce the same coset, i.e., $a_1 \circ H = a_2 \circ H$, a_1 and a_2 are becomes to be *equivalence*, i.e., $a_1 \sim a_2$. This equivalence relation leads to a set defined by $\overline{a} = \{x \in G : x \sim a\}$ for $a \in G$ which is called an *equivalence class*. The important fact is that this equivalence class $\overline{a}$ is exactly the same as the coset $a \circ H$. Thus, the set of the quotient group G/H is the same as the set of all equivalence classes, i.e., $G/H = \{a \circ H : a \in G\} = \{\overline{a} : a \in G\}$. A binary operation $*$ can be naturally defined by $\overline{a} * \overline{b} = \overline{a \circ b} \in G/H$ for all $\overline{a}, \overline{b} \in G/H$.

Example 2.4. (Quotient group of integers) Let $(\mathbb{Z}, +)$ be an abelian group. Then, $n\mathbb{Z} = \{n \cdot a : a \in \mathbb{Z}\}$ is a subgroup of $\mathbb{Z}$. For a given $a \in \mathbb{Z}$, a coset is given by $a + n\mathbb{Z} = \{a + h : h \in n\mathbb{Z}\}$, which produce an equivalence relation $a \sim a + n \sim a + 2n \sim \cdots$. Assuming $\overline{a}$ is a equivalence class of a with this relation, there is a quotient group defined by $\mathbb{Z}/n\mathbb{Z} = \{a + n\mathbb{Z} : a \in \mathbb{Z}\} = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$ with a binary operation $+$ defined by $\overline{a} + \overline{b} = \overline{a+b}$ for all $a, b \in \mathbb{Z}/n\mathbb{Z}$. The quotient group is a finite abelian group of order n.

Example 2.5. Let $\mathbb{Z}/n\mathbb{Z}$ be a finite abelian group described in Example 2.4 and let n be a composite number. In this example, suppose that $n = 12$ and $G = \mathbb{Z}/12\mathbb{Z} = \{\overline{0}, \overline{1}, \ldots, \overline{11}\}$. Then, there exists a subgroup $H = \{\overline{0}, \overline{3}, \overline{6}, \overline{9}\}$ of which the order $\#H = 4$ divides $\#G = 12$. There are only three cosets given by $\overline{\overline{0}} = \overline{0}+H = \overline{3}+H = \overline{6}+H = \overline{9}+H$, $\overline{\overline{1}} = \overline{1}+H = \overline{4}+H = \overline{7}+H = \overline{10}+H$, and $\overline{\overline{2}} = \overline{2}+H = \overline{5}+H = \overline{8}+H = \overline{11}+H$, which is $[G : H] = 3$ and thus $\#G = [G : H] \cdot \#H = 3 \cdot 4 = 12$. Then, it is obtained a quotient group $G/H = \{\overline{\overline{0}}, \overline{\overline{1}}, \overline{\overline{2}}\}$ of order 3 and confirm $\#G/H = \#G/\#H = 12/4 = 3$.

Much of the importance of quotient groups is derived from their relation to homomorphisms, which is a structure-preserving map defined as follows:

Definition 2.6. (Group homomorphism) Let $(G, \circ)$ and $(G', \circ')$ be abelian groups of which identities e and e', respectively. A *group homomorphism* $\varphi : G \to G'$ be a map such that for all $a, b \in G$ it holds that $\varphi(a \circ b) = \varphi(a) \circ' \varphi(b)$ and $\varphi(e) = e'$. A set $\ker(\varphi) = \{a \in G : \varphi(a) = e'\}$ is especially called a *kernel* of φ.

The homomorphisms φ are classified into several cases corresponding to their properties. If φ is a bijection, i.e., φ is a one-to-one map, we say φ is an *isomorphism* and G and G' are *isomorphic*, which is denoted by $G \cong G'$. If $G = G'$, φ is called an *endomorphism*. If φ is an isomorphism and $G = G'$, φ is called an *automorphism*.

In what follows, let us consider a map ϕ from a group G to a quotient group G/H by a subgroup H of G. Then, $\phi : G \to G/H$ is a homomorphism such that $\phi(a) = aH \in G/H$ for $a \in G$. Its kernel is given by $\ker(\phi) = H$. The correctness of the fact can be confirmed by the following example.

Example 2.7. Let $G = \{\overline{0}, \overline{1}, \ldots, \overline{11}\}$, $H = \{\overline{0}, \overline{3}, \overline{6}, \overline{9}\}$, and $G/H = \{\overline{\overline{0}}, \overline{\overline{1}}, \overline{\overline{2}}\}$ be abelian groups defined in Example 2.5, respectively. A map $\psi : G \to G/H$ is a homomorphism such that $\overline{0} + H = \{\overline{0}, \overline{3}, \overline{6}, \overline{9}\} \to \{\overline{\overline{0}}\}$, $\overline{1} + H = \{\overline{1}, \overline{4}, \overline{7}, \overline{10}\} \to \{\overline{\overline{1}}\}$, and $\overline{2} + H =$

$\{\overline{2}, \overline{5}, \overline{8}, \overline{11}\} \rightarrow \{\overline{\overline{2}}\}$. From the definition, it is found that the kernel of ψ is $\ker(\psi) = \overline{0} + H = H$.

2.1.3 Direct product of groups

This subsection introduces one of the construction methods to create a larger group from the given plural groups. Let G_1 and G_2 be abelian groups under binary operations $+_1$ and $+_2$ with identities e_1 and e_2, respectively. A *direct product of* G_1 *and* G_2 is a set

$$G_1 \times G_2 = \{(g_1, g_2) : g_1 \in G_1, g_2 \in G_2\}. \tag{2.3}$$

Then, $G_1 \times G_2$ forms a group under a binary operation $*$ such that $(g_1, g_2) * (g_1', g_2') = (g_1 +_1 g_1', g_2 +_2 g_2') \in G_1 \times G_2$ for all $(g_1, g_2), (g_1', g_2') \in G_1 \times G_2$ with the identity (e_1, e_2). Such a group is called the *direct product group*. If G_1 and G_2 are finite groups, the order of the direct product group is given by $\#G_1\#G_2$. In this thesis, the direct product group $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ is often used to indicate a group structure.

2.1.4 Pairing

In the following, a map from a product of two groups to one group is described.

Definition 2.8. (Pairing) Let G_1 and G_2 be abelian groups under binary operations $+_1$ and $+_2$ with identities e_1 and e_2, respectively, and let G_T be an abelian group under a binary operation $\cdot$ with identity e_T. A *pairing* is a map from a product of G_1 and G_2 to G_T defined by

$$e : G_1 \times G_2 \rightarrow G_T, \tag{2.4}$$

which has the following properties:

1. Bilinear: For all $g_1, g_1' \in G_1$ and $g_2, g_2' \in G_2$,

$$e(g_1 +_1 g_1', g_2) = e(g_1, g_2) \cdot e(g_1', g_2), \tag{2.5}$$

$$e(g_1, g_2 +_2 g_2') = e(g_1, g_2) \cdot e(g_1, g_2'). \tag{2.6}$$

2. Non-degenerate: For all $g_1 \in G_1$, $e(g_1, g_2) = e_T$ if and only if $g_2 = e_2 \in G_2$. For all $g_2 \in G_2$, $e(g_1, g_2) = e_T$ if and only if $g_1 = e_1 \in G_1$.

Note that the bilinear map is not a group homomorphism out of the direct product group. For some integer m, let $[m]g_1$ and $[m]g_2$ denote elements G_1 and G_2 applying m-term operations for $g_1 \in G_1$ and $g_2 \in G_2$, respectively. Besides, let a_T^m denote an element in G_T applying m-term operations for $g_T \in \mathbb{G}_T$. Then, the property of the bilinear map gives rise to the following fact for all $a, b \in \mathbb{Z}$:

$$e([a]g_1, [b]g_2) = e([b]g_1, [a]g_2) = e(g_1, g_2)^{ab}. \tag{2.7}$$

In cryptography, the pairing is typically defined on an elliptic curve over a finite field. The details are introduced in the later sections.

2.1.5 Rings and fields

In the groups, there is only one binary operation. This subsection considers another operation and studies the structure that results from their interaction.

Definition 2.9. (Ring) Let R be a set and $+$ and $\cdot$ be binary operations defined on R. The triple of R and two operations, which is denoted by $(R, +, \cdot)$ and is called a *ring* if the following conditions are satisfied:

1. Assosiative: $(a + b) + c = a + (b + c)$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in R$.
2. Commutative: $a + b = b + a$ for all $a, b \in R$.
3. Distributive: $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(a + b) \cdot c = a \cdot c + b \cdot c$ for all $a, b, c \in R$.
4. Identities: There exist $0_R, 1_R \in R$ such that $a+0_R = 0_R+a = a$ and $a \cdot 1_R = 1_R \cdot a = a$ for all $a \in R$.
5. Inverse: Given $a \in R$, there exists $b \in R$ such that $a + b = 0_R$.

Then, we say R forms a ring under $+$ and $\cdot$.

The conditions of $(R, +, \cdot)$ to be a ring are equivalence to ones that $(R, +)$ is an abelian group, $(R, \cdot)$ is a monoid, and there is the distributive property. If there is one more condition of commutative $a \cdot b = b \cdot a$ for all $a, b \in R$ to Definition 2.9, $(R, +, \cdot)$ is called a *commutative ring.* If there is a unique 0_R in R, $(R, +, \cdot)$ is called an *integral domain.* For the integral domain $(R, +, \cdot)$, a *characteristic*, which is denoted as $\mathrm{char}(R)$, is the smallest positive number n such that $[n]1_R = 0_R$ if such a number n exists; and 0 otherwise.

If a requirement for the existence of multiplicative inverses to the commutative ring is joined, a field of the following definition is obtained.

Definition 2.10. (Field) Let F be a set and $+$ and $\cdot$ be binary operations defined on F. The triple of F and two operations, which is denoted by $(F, +, \cdot)$ and is called a *field* if the following conditions are satisfied:

1. Assosiative: $(a + b) + c = a + (b + c)$ and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for all $a, b, c \in F$.
2. Commutative: $a + b = b + a$ and $a \cdot b = b \cdot a$ for all $a, b \in F$.
3. Distributive: $a \cdot (b + c) = a \cdot b + a \cdot c$ for all $a, b, c \in F$.
4. Identities: There exist $0_R, 1_R \in R$ such that $a+0_R = 0_R+a = a$ and $a\cdot 1_R = a\cdot 1_R = a$ for all $a \in F$.
5. Additive inverse: Given $a \in R$, there exists $b \in R$ such that $a + b = 0_R$.
6. Multiplicative inverse: Given $a \in R$ such that $a \neq 0_R$, there exists $b \in R$ such that $a \cdot b = 1_R$.

Then, we say F forms a field under $+$ and $\cdot$.

Let F^* be a set of elements of F excluding 0_R, which is defined by $F^* = F\backslash\{0_R\}$. Then, the conditions of $(F, +, \cdot)$ being a field are equivalence to ones that $(F, +)$ and $(F^*, \cdot)$ are abelian groups with the distributive property. Hereafter a ring involving a field is written by $(R, +, \cdot)$ if the operations are needed to be specified; otherwise, the ring is simply referred to by the corresponding set of R. The number of the set R is also called an *order* of the ring (or field) and is denoted by $\#R$. If $\#R$ is a finite number, R is called a *finite ring* (or *finite field*).

The following provides examples of infinite rings and fields. For similar reasons with the groups, examples of finite rings and fields are provided after describing the quotient rings.

Example 2.11. (Ring of integers) As defined in Example 2.2, let $\mathbb{Z}$ be a set of integers. Let $+$ and $\cdot$ be a natural addition and multiplication defined on $\mathbb{Z}$. Then, $(\mathbb{Z}, +, \cdot)$ is an infinite commutative ring.

Example 2.12. (Ring of rational numbers) Let $\mathbb{Q}$ be a set of rational numbers and let $+$ and $\cdot$ be a natural addition and multiplication defined on $\mathbb{Q}$. Then, $(\mathbb{Q}, +, \cdot)$ is an infinite field.

2.1.6 Ideals, quotient rings, and homomorphisms

If a subset S of R forms a ring (or field) under the same laws of R, S is said to be a *subring* (or *subfield*) of R. This subsection describes a special class of subrings.

Definition 2.13. (Ideal) Let $(R, +, \cdot)$ be a commutative ring and let I be a subset of R. We say I is an *ideal* of R if the following conditions are satisfied:

1. $(I, +)$ is a subgroup of $(R, +)$.

2. $r \cdot i \in I$ for all $r \in R$ and $i \in I$.

Among the ideals, an ideal I of R is called a *principal ideal* if there is an element if I is generated by a single element $a \in R$, i.e., $I = aR = \{a \cdot r : r \in R\}$. The principal ideal I generated by $a \in R$ is denoted as $I = (a)$. Given $a \in R$ and ideal I of R, let $a + I$ be a coset of I defined by $a + I = \{a + i : i \in I\}$. Let R/I be a set of the cosets defined by

$$R/I = \{a + I : a \in R\}. \tag{2.8}$$

For the set R/I, one can define binary operations $*$ and $\star$ such that $a+I*b+I = (a+b)+I$ and $a + I \star b + I = (a \cdot b) + I$ for all $a + I, b + I \in R/I$. Then, R/I forms a ring under $*$ and $\star$, which is called a *quotient ring.*

In the same manner as the groups, there appear an equivalence relation $a_1 \sim a_2$ for $a_1, a_2 \in R$ which produce the same coset $a_1 + I = a_2 + I$. Assuming $\overline{a} = \{x \in G : x \sim a\}$ is a equivalence class for $a \in R$ with the equivalence relation, one can define the quotient ring by the set of all equivalence classes given by $R/I = \{\overline{a} : a \in G\}$ with operations $*$ and $\star$ given by $\overline{a} * \overline{a} = \overline{a + b} = (a + b) + I$ and $\overline{a} \star \overline{b} = \overline{a \cdot b} = (a \cdot b) + I$ for all $\overline{a}, \overline{b} \in R/I$, respectively.

Example 2.14. (Quotient ring of integers) As shown in Example 2.11, $(\mathbb{Z}, +, \cdot)$ is a commutative ring. A subring of $\mathbb{Z}$ can be easily found by $n\mathbb{Z} = \{n \cdot a : a \in \mathbb{Z}\}$, which is a principal ideal of $\mathbb{Z}$ denoted by $(n) = n\mathbb{Z}$. For a given $a \in G$, a coset is given by $a + (n) = \{a+i : i \in (n)\}$ and generates an equivalence relation $a \sim a+n \sim a+2n \sim \cdots$. Then, one can also define the quotient ring by $\mathbb{Z}/(n) = \{a + (n) : a \in R\} = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$, with operations $+$ and $\cdot$ defined by $\overline{a}+\overline{b} = \overline{a + b}$ and $\overline{a}\cdot\overline{b} = \overline{a \cdot b}$ for all $\overline{a}, \overline{b} \in \mathbb{Z}$, respectively. The quotient ring is a finite commutative ring of order n. From the properties of the ring, note that $(\mathbb{Z}/(n), +)$ is a finite abelian group, and $(\mathbb{Z}/(n)\backslash\{\overline{0}\}, \cdot)$ is a commutative semigroup. It is easily confirmed that the group $(\mathbb{Z}/(n), +)$ is exactly the same as $(\mathbb{Z}/n\mathbb{Z}, +)$ given in Example 2.4. To make $\mathbb{Z}/(n)$ being a field, $(\mathbb{Z}/(n)\backslash\{\overline{0}\}, \cdot)$ must be an abelian group.

Example 2.15. Let $\mathbb{Z}/(n)$ be a commutative ring described in Example 2.14 and let n be a composite number. In this example, suppose that $n = 12$ and $R = \mathbb{Z}/(12) = \{\overline{0}, \overline{1}, \ldots, \overline{11}\}$. Then, there exists a subring $I = \{\overline{0}, \overline{4}, \overline{8}\}$ with the multiplicative identity $\overline{4}$. Note that the identity elements of the subring and original ring do not always the same. The subring I is a principal ideal denoted by $I = (\overline{4}) = \overline{4}R = \{\overline{4} \cdot r : r \in R\}$. Then, there are only four cosets $\overline{\overline{0}} = \overline{0} + I = \overline{4} + I = \overline{8} + I$, $\overline{\overline{1}} = \overline{1} + I = \overline{5} + I = \overline{9} + I$, and $\overline{\overline{2}} = \overline{2} + I = \overline{6} + I = \overline{10} + I$, $\overline{\overline{3}} = \overline{3} + I = \overline{7} + I = \overline{11} + I$. Then, it is obtained a quotient ring $R/I = \{\overline{\overline{0}}, \overline{\overline{1}}, \overline{\overline{2}}, \overline{\overline{3}}\}$ under an addition $+$ and multiplication $\cdot$ defined by $\overline{\overline{a}} + \overline{\overline{b}} = \overline{\overline{a} + \overline{b}} = \overline{a + b} + I$ and $\overline{\overline{a}} \cdot \overline{\overline{b}} = \overline{\overline{a} \cdot \overline{b}} = \overline{a \cdot b} + I$ for all $\overline{\overline{a}}, \overline{\overline{b}} \in R/I$.

Similar to the case of quotient groups, the quotient ring is also related to homomorphisms.

Definition 2.16. (Ring homomorphism) Let $(R, +, \cdot)$ and $(R', +', \cdot')$ be abelian groups of which identities $0_R, 1_R$ and $0_{R'}, 1_{R'}$, respectively. A *ring homomorphism* $\varphi : R \to R'$ be a map satisfying the following.

1. For all $a, b \in G$, $\varphi(a + b) = \varphi(a) +' \varphi(b)$.

2. For all $a, b \in G$, $\varphi(a \cdot b) = \varphi(a) \cdot' \varphi(b)$.

3. $\varphi(1_R) = 1_{R'}$.

A set $\ker(\varphi) = \{a \in R : \varphi(a) = 0_{R'}\}$ is a *kernel* of φ.

The homomorphisms φ are classified into the following cases if φ satisfies the specific conditions. If φ is a bijection, φ is called an *isomorphism*, and R and R' are *isomorphic*, which is denoted by $R \cong R'$. If $R = R'$, the isomorphism is called an *endomorphism*. If φ is an isomorphism and $R = R'$, φ is called an *automorphism*.

Let us consider a map ψ from a ring R to a quotient ring R/I by an ideal I of R. Then, $\psi : R \to R/I$ is a homomorphism such that $\psi(a) = a + I \in R/I$ for $a \in G$. Its kernel is given by $\ker(\psi) = I$.

Example 2.17. Let $R = \mathbb{Z}/(12) = \{\overline{0}, \overline{1}, \ldots, \overline{11}\}$, $I = \{\overline{0}, \overline{4}, \overline{8}\}$, and $R/I = \{\overline{\overline{0}}, \overline{\overline{1}}, \overline{\overline{2}}, \overline{\overline{3}}\}$ be rings defined in Example 2.15, respectively. Then, $\phi : R \to R/I$ is a homomorphism such that $\overline{0} + I = \{\overline{0}, \overline{4}, \overline{8}\} \to \{\overline{\overline{0}}\}$, $\overline{1} + I = \{\overline{1}, \overline{5}, \overline{9}\} \to \{\overline{\overline{1}}\}$, $\overline{2} + I = \{\overline{2}, \overline{6}, \overline{10}\} \to \{\overline{\overline{2}}\}$, and $\overline{3} + I = \{\overline{3}, \overline{7}, \overline{11}\} \to \{\overline{\overline{3}}\}$. The kernel of ψ is given by $\ker(\psi) = \overline{0} + I = I$.

A subring S of a ring R is discussed above. Then, a ring R is called an *extension ring* (or *extension field*) of a subring S. A dimension of R as a vector space on S is said to be an *extension degree* and denoted by $[R : S]$. A *ring (or field) extension* is a ring homomorphism such that $S \to R$, which plays an important role to discuss the finite fields in cryptography. The context of the extension is explained at the same time on presenting more details about the finite fields in Sect. 2.2.

2.2 Finite fields

This section provides details about the finite fields that are applied for cryptography.

2.2.1 Prime field

In this section, more details of the operations defined on the ring $\mathbb{Z}/(n)$ given in Example 2.14 are described with a definition of a prime field.

We have seen that $\mathbb{Z}/(n) = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$ forms a ring under the binary operations $+$ and $\cdot$ such that $\overline{a} + \overline{b} = \overline{a+b}$ and $\overline{a} \cdot \overline{b} = \overline{a \cdot b}$ for all $\overline{a}, \overline{b} \in \mathbb{Z}/(n)$. We also have studied that there is the equivalence relation $a \sim a+n \sim a+2n \sim \cdots$ generated by the elements of $\mathbb{Z}/(n)$. This relation can be more formally defined as follows: For $a, b \in \mathbb{Z}$, if n divides $a-b$, i.e., a can be denoted by $a = n \cdot l + b$ with $l \in \mathbb{Z}$, we say *a and b are congruent modulo n* and denote as follows:

$$a \equiv b \ (\mathrm{mod}\ n). \tag{2.9}$$

The congruence relation gives rise to an equivalence class of a modulo n, which is defined by $\overline{a} = \{a \in \mathbb{Z} : x \equiv a \ (\mathrm{mod}\ n)\}$. For example, if $a \equiv b \equiv c \ (\mathrm{mod}\ n)$ for $a, b, c \in \mathbb{Z}$, then $a, b, c \in \overline{a}$. Since the congruence relation is compatible with the addition and multiplication, one can have the operations $\overline{a} + \overline{b} = \overline{a+b}$ and $\overline{a} \cdot \overline{b} = \overline{a \cdot b}$ for $\overline{a}, \overline{b} \in \mathbb{Z}/(n)$. As a result, there is a quotient ring $\mathbb{Z}/(n) = \{\overline{a} : a \in \mathbb{Z}\} = \{\overline{0}, \overline{1}, \ldots, \overline{n-1}\}$ under the above operations.

As mentioned in Example 2.14, to make $\mathbb{Z}/(n)$ being a field, it is required to make $\mathbb{Z}/(n)\backslash\{\overline{0}\}$ to be an abelian group under $\cdot$. However, $\mathbb{Z}/(n)\backslash\{\overline{0}\}$ do not always form a group for any n. This is because that $\cdot$ is not always defined on $\mathbb{Z}/(n)\backslash\{\overline{0}\}$ since there are possibilities such that $\overline{a} \cdot \overline{b} = \overline{0} \notin \mathbb{Z}/(n)\backslash\{\overline{0}\}$ for several $\overline{a}, \overline{b} \in \mathbb{Z}/n\mathbb{Z}\backslash\{\overline{0}\}$, e.g., $\overline{2} \cdot \overline{2} = \overline{0} \notin \mathbb{Z}/(4)\backslash\{\overline{0}\}$ and $\overline{3} \cdot \overline{5} = \overline{0} \notin \mathbb{Z}/(15)\backslash\{\overline{0}\}$. Fortunately, if n is a prime p, one can avoid the possibilities and can make $\mathbb{Z}/(p)\backslash\{\overline{0}\}$ being an abelian group under $\cdot$. Then, there is a field $\mathbb{Z}/(p)$ of order p and $\mathrm{char}(\mathbb{Z}/(p)) = p$. The field $\mathbb{Z}/(p)$ is especially called a *prime field*. Note that the prime field and its isomorphic fields are the smallest subfields of finite fields.

Although the elements of $\mathbb{Z}/(p)$ are equivalence classes of integers in $\mathbb{Z}$ modulo p, it is preferred to operate $\mathbb{Z}/(p)$ by using only the limited elements in an environment with limited resources. Therefore, we work on a field that is isomorphic to $\mathbb{Z}/(p)$. Indeed, let $\mathbb{F}_p$ be a set defined by

$$\mathbb{F}_p = \{0, 1, \ldots, p-1\}. \tag{2.10}$$

Let us define binary operations $a + b = (a+b)\%p \in \mathbb{F}_p$ and $a \cdot b = (a \cdot b)\%p \in \mathbb{F}_p$ for all $a, b \in \mathbb{F}_p$ where % is a reminder operation. It can be confirmed that $\mathbb{F}_p$ forms a field under these operations. The multiplicative group of $\mathbb{F}_p$ is referred to as the set $\mathbb{F}_p^* = \mathbb{F}_p\backslash\{0\}$. Then, $\mathbb{F}_p^*$ is a cyclic group of order $p-1$, i.e., there is a generator $a \in \mathbb{F}_p^*$ such that $\mathbb{F}_p^* = \langle a \rangle$. An example of $\mathbb{F}_p$ with a concrete p is provided in the following.

Example 2.18. (Prime field of order 5) Let $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$ and let $+$ and $\cdot$ be multiplication and addition operation taking a reminder divided by 5. The operations table in $\mathbb{F}_5$ is given in Table 2.1. From the table, it is found that $\mathbb{F}_5$ and $\mathbb{F}_5^*$ form abelian groups under $+$ and $\cdot$, respectively. Thus, $(\mathbb{F}_5, +, \cdot)$ is a prime field of order 5.

$+$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

$\cdot$	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Table 2.1: Operation tables in $\mathbb{F}_5$.

2.2.2 Polynomial rings and field extensions

This subsection introduces polynomial rings to describe the field extension from a subfield to a field. Let R be a commutative ring. Let n be a positive integer and let $f(x)$ be a polynomial defined by

$$f(x) = \sum_{i=0}^{n} a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0, \tag{2.11}$$

where $a_i \in R$. Then, we say $f(x)$ is defined over R. If $a_n \neq 0$, n is called degree of $f(x)$ and is denoted by $n = \deg(f)$. If $f(x)$ has $a_i = 0$ for all i, i.e., $f(x) = 0$, let us define $\deg(f) = -\infty$. Let m be a positive integer such that $m \leq n$ and let $g(x)$ be a polynomial defined by

$$g(x) = \sum_{j=0}^{m} b_j x^j = b_m x^m + a_{m-1} x^{m-1} + \cdots b_1 x + b_0, \tag{2.12}$$

where $b_i \in R$. Let us define $f(x) = g(x)$ if $a_i = b_i$ for $0 \leq i \leq m$ and $a_i = 0$ for $m < i \leq n$. For $f(x), g(x)$, an addition and multiplication are defined as follows:

$$f(x) + g(x) = \sum_{i=0}^{n} a_i x^i + \sum_{j=0}^{m} b_j x^j = \sum_{i=0}^{m} (a_i + b_i) x^i + \sum_{i=m+1}^{n} a_i x^i, \tag{2.13}$$

$$f(x) \cdot g(x) = \left(\sum_{i=0}^{n} a_i x^i \right) \cdot \left(\sum_{j=0}^{m} b_j x^j \right) = \sum_{k=0}^{m+n} \left(\sum_{i+j=k} a_i b_j \right) x^k, \tag{2.14}$$

which $f(x) + g(x)$ and $f(x) \cdot g(x)$ are also polynomials defined over R. Then, it can be easily found that a set of polynomials defined over R forms a commutative ring under the above operations. The ring is called a *polynomial ring* and is denoted by $R[x]$. Note that a set of polynomial defined over R with variables $x_1, x_2, \ldots, x_n$ also forms a polynomial ring $R[x_1, x_2, \ldots, x_n]$, however, here we only focus on $R[x]$.

Let F be a field and let $F[x]$ be a polynomial ring over F. Then, for all $f(x), g(x) \in F[x]$ such that $g(x) \neq 0$, there exist polynomials $q(x), r(x) \in F[x]$ such that $f(x) =$

$+$	$\overline{0}$	$\overline{1}$	$\overline{x}$	$\overline{x+1}$
$\overline{0}$	$\overline{0}$	$\overline{1}$	$\overline{x}$	$\overline{x+1}$
$\overline{1}$	$\overline{1}$	$\overline{0}$	$\overline{x+1}$	$\overline{x}$
$\overline{x}$	$\overline{x}$	$\overline{x+1}$	$\overline{0}$	$\overline{1}$
$\overline{x+1}$	$\overline{x+1}$	$\overline{x}$	$\overline{1}$	$\overline{0}$

$\cdot$	$\overline{0}$	$\overline{1}$	$\overline{x}$	$\overline{x+1}$
$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$	$\overline{0}$
$\overline{1}$	$\overline{0}$	$\overline{1}$	$\overline{x}$	$\overline{x+1}$
$\overline{x}$	$\overline{0}$	$\overline{x}$	$\overline{x+1}$	$\overline{1}$
$\overline{x+1}$	$\overline{0}$	$\overline{x+1}$	$\overline{1}$	$\overline{x}$

Table 2.2: Operation tables in $\mathbb{F}_2[x]/(x^2+x+1)$.

$q(x) \cdot g(x) + r(x)$ where $r(x)$ is 0 or a polynomial of degree such that $\deg(r) < \deg(g)$. Moreover, $q(x)$ and $r(x)$ are uniquely determined corresponding to $f(x)$ and $g(x)$. The fact gives rise to an equivalence relation on $F[x]$ producing an equivalence class of $a(x) \in F[x]$ modulo $f(x) \neq 0 \in F[x]$, which is denoted by $\overline{a(x)}$. In fact, $\overline{a(x)}$ is denoted by $\overline{a(x)} = a(x) + (f(x))$ where $(f(x))$ plays a principal ideal of $R[x]$, there is a quotient ring $F[x]/(f(x))$ consists of the set of all equivalence classes modulo $f(x)$ under the addition and multiplication defined by $\overline{a(x)} + \overline{b(x)} = \overline{a(x) + b(x)}$ and $\overline{a(x)} \cdot \overline{b(x)} = \overline{a(x) \cdot b(x)}$ for $\overline{a(x)}, \overline{b} \in F[x]/(f(x))$, respectively. Considering the fact that the ring $Z/(n)$ can be a field if n is a prime, a similar discussion can be held in this case. If $f(x)$ is decomposed into at least two polynomials of degree at least 1, we say $f(x)$ is *reducible*; otherwise, $f(x)$ is *irreducible*. Then, there is a fact that $F[x]/(f(x))$ is a field if $f(x)$ is irreducible. The following shows an example.

Example 2.19. (Field of order 4) Let $\mathbb{F}_2 = \{0, 1\}$ be a prime field of order 2 under the operations $+$ and $\cdot$ taking a reminder divided by 2. Then, there is an irreducible polynomial given by $x^2 + x + 1$ in $\mathbb{F}_2[x]$. The principal ideal $(x^2 + x + 1)$ produce four costs given by $\overline{0} = 0 + (x^2 + x + 1)$, $\overline{1} = 1 + (x^2 + x + 1)$, $\overline{x} = x + (x^2 + x + 1)$, and $\overline{x+1} = x + 1 + (x^2 + x + 1)$. From the above, there is a quotient ring $\mathbb{F}_2[x]/(x^2+x+1) = \{\overline{0}, \overline{1}, \overline{x}, \overline{x+1}\}$ under the addition and multiplication for $\overline{a}, \overline{b} \in \mathbb{F}_2[x]/(x^2 + x + 1)$, e.g., $\overline{1} + \overline{x+1} = \overline{x+2} = \overline{x}$ and $\overline{x} \cdot \overline{x+1} = \overline{x^2+x} = \overline{-x-1+x} = \overline{1}$. It is observed that one can compute a reminder divided by not only 2 but also $x^2 + x + 1$. The operation tables for $\mathbb{F}_2[x]/(x^2 + x + 1)$ are given in Table 2.2. From the table, it can be confirmed $\mathbb{F}_2[x]/(x^2 + x + 1)$ forms a field of order 4.

Suppose that $f(x)$ is an irreducible polynomial of degree $n > 1$ and let α be a root of the irreducible polynomial $f(x)$. Then, $F[x]/(f(x))$ is isomorphic to a field $F(\alpha)$, which is obtained from F by adjoining α, i.e., every element of $F(\alpha)$ can be uniquely expressed in the form

$$a_{n-1}\alpha^{n-1} + a_{n-2}\alpha^{n-2} + \cdots + a_1\alpha + a_0, \tag{2.15}$$

where $a_i \in F$ for $0 \leq i \leq n-1$ and which is a m-th dimensional vector space of F with a basis $\{1, \alpha, \ldots \alpha^{n-1}\}$. Notice that $F(\alpha)$ is an extension field of F of degree $[F(\alpha) : F] = n$

$+$	0	1	α	$\alpha+1$
0	0	1	α	$\alpha+1$
1	1	0	$\alpha+1$	α
α	α	$\alpha+1$	0	1
$\alpha+1$	$\alpha+1$	α	1	0

$\cdot$	0	1	α	$\alpha+1$
0	0	0	0	0
1	0	1	α	$\alpha+1$
α	0	α	$\alpha+1$	1
$\alpha+1$	0	$\alpha+1$	1	$\alpha+1$

Table 2.3: Operation tables in $\mathbb{F}_2(\alpha)$.

and thus the number of elements of $F(\alpha)$ is given by $(\#F)^n$. Since the smallest subfield of an extension field is a field isomorphic to $\mathbb{F}_p$ of order p, the extension field must have an order p or power of p.

Example 2.20. Let $\mathbb{F}_2$ and $\mathbb{F}_2[x]/(x^2+x+1)$ be fields defined in Example 2.19. The irreducible polynomial x^2+x+1 has a root $\alpha = \frac{\pm\sqrt{-3}+1}{2}$ which is an element not in $\mathbb{F}_2$. Considering the field $\mathbb{F}_2(\alpha)$, there are four elements 0, 1, α, and $\alpha+1$ of the form $a_1\alpha + a_0 \in \mathbb{F}_2(\alpha)$ where $a_1, a_1 \in \mathbb{F}_2$. Since α is a root of the irreducible polynomial, there is a relation $\alpha^2+\alpha+1=0$, i.e., $\alpha^2 = \alpha+1$. The operation tables are given in Table 2.3. As seen in the table, $\mathbb{F}_2[x]$ and $\mathbb{F}_2[x]/(x^2+x+1)$ are isomorphic since there is an isomorphism $\mathbb{F}_2[x]/(x^2+x+1) \to \mathbb{F}_2(\alpha)$ such that $\overline{0} \mapsto 0$, $\overline{1} \mapsto 1$, $\overline{x} \mapsto \alpha$, and $\overline{x+1} \mapsto \alpha+1$. It is also found $[\mathbb{F}_2(\alpha) : \mathbb{F}_2] = 2$ and $\#\mathbb{F}_2(\alpha) = 4 = 2^2$.

A finite field order $q = p^m$ $(m > 0)$ is denoted as $\mathbb{F}_q$. Note that every field of order q is isomorphic to $\mathbb{F}_q$. The multiplicative group of $\mathbb{F}_q$ is referred to as the set $\mathbb{F}_q^* = \mathbb{F}_q \backslash \{0\}$. Similar to the case of $\mathbb{F}_p$, the multiplicative group $\mathbb{F}_q^*$ is a cyclic group of order $q-1$, i.e., there is a generator $a \in \mathbb{F}_q^*$ such that $\mathbb{F}_q^* = \langle a \rangle$.

For positive integers $m, n, o, \ldots$, let us consider a sequence of finite fields such that $\mathbb{F}_{q^m} \cong \mathbb{F}_q[x]/(f_m(x))$, $\mathbb{F}_{(q^m)^n} \cong \mathbb{F}_{q^m}[x]/(f_n(x))$, $\mathbb{F}_{((q^m)^n)^o} \cong \mathbb{F}_{(q^m)^n}[x]/(f_o(x)), \ldots$ where $f_m(x)$, $f_n(x)$, $f_o(x), \ldots$ are irreducible polynomials in $\mathbb{F}_q[x]$, $\mathbb{F}_{q^m}[x]$, $\mathbb{F}_{(q^m)^n}[x], \ldots$. We say the sequence is a *tower of fields*. Assuming $\alpha_m, \alpha_n, \alpha_o, \ldots$ be a root of $f_m(x)$, $f_n(x)$, $f_o(x), \ldots$, the field $\mathbb{F}_{(\cdots(((q^m)^n)^o)\cdots)}$ is isomorphic to a field F adjoining $\alpha_m, \alpha_n, \alpha_o, \ldots$, i.e., $F(\alpha_m)(\alpha_n)(\alpha_o)\cdots$, of which elements can be written by using the following basis.

$$\{1, \alpha_m, \ldots, \alpha_m^{m-1}\} \times \{1, \alpha_n, \ldots, \alpha_n^{n-1}\} \times \{1, \alpha_o, \ldots, \alpha_o^{o-1}\} \times \cdots . \tag{2.16}$$

A field adjoining roots of all polynomials over $\mathbb{F}_q$ is said to be an *algebraic closure* of $\mathbb{F}_q$ and is denoted by $\overline{\mathbb{F}}_q$.

Example 2.21. (Field of order 16) Let $\mathbb{F}_2$ and $\mathbb{F}_2[x]/(x^2+x+1) \cong \mathbb{F}_2(\alpha)$ be fields found in Examples 2.19 and 2.20. There is an irreducible polynomial $x^2+\alpha x+1$ in $\mathbb{F}_2(\alpha)[x]$. Then, the principal ideal $(x^2+\alpha x+1)$ produce 16 cosets given by the form $\overline{a_1x+a_0} = a_1x + a_0 + (x^2+\alpha x+1)$ with $a_0, a_1 \in \mathbb{F}_2(\alpha)$ and gives rise to a finite field $\mathbb{F}_2(\alpha)[x]/(x^2+\alpha x+1)$ of order 16. Since the irreducible polynomial $x^2+\alpha x+1$ has a root β which is not an element in $\mathbb{F}_2(\alpha)$, the field $\mathbb{F}_2(\alpha)[x]/(x^2+\alpha x+1)$ is isomorphic to $\mathbb{F}_2(\alpha)(\beta)$ having 16 elements of the form $a_1\beta+a_0 \in \mathbb{F}_2(\alpha)(\beta)$ where $a_1, a_1 \in \mathbb{F}_2(\alpha)$. Since there are the representations $a_0 = a_{01}\alpha + a_{00} \in \mathbb{F}_2(\alpha)$ and $a_1 = a_{11}\alpha + a_{10} \in \mathbb{F}_2(\alpha)$ with $a_{00}, a_{01}, a_{10}, a_{11} \in \mathbb{F}_2$, every element in $\mathbb{F}_2(\alpha)(\beta)$ is given by $a_1\beta + a_0 = a_{11}\alpha\beta + a_{10}\beta + a_{01}\alpha + a_{00}$ of which a basis is $\{1, \alpha, \beta, \alpha\beta\} = \{1, \alpha\} \times \{1, \beta\}$.

2.2.3 Frobenius endomorphism and conjugates

This subsection introduces the properties of the finite fields. The definition of the important endomorphism in the finite fields is provided below.

Definition 2.22. (Frobenius endomorphism) Let $\mathbb{F}_q$ be a finite field and let $p = \mathrm{char}(\mathbb{F}_q)$. Then, the *Frobenius endomorphism* is a map defined by

$$\pi_p : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q, a \mapsto a^p. \tag{2.17}$$

The map π_p is exactly endomorphism since it holds $\pi_p(a \cdot b) = \pi_p(a) \cdot \pi_p(b)$ and $\pi_p(a+b) = \pi_p(a) + \pi_p(b)$ for all $a, b \in \mathbb{F}_q$. From Fermat's little theorem, $\pi_p(a) = a^p = a$ for all $a \in \mathbb{F}_p \subseteq \mathbb{F}_q$. A similar property is enjoyed on $\mathbb{F}_q$ by the m-th iterate of the Frobenius endomorphisms, i.e., $\pi_p^m(a) = a^{p^m} = a$ for all $a \in \mathbb{F}_q$. It is more often used the q-th power Frobenius endomorphism defined by $\pi_q : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q, a \to a^q$. Note that it is an automorphism in certain contexts, however, this is not true in general.

Applying the Frobenius endomorphisms, there are $a^p, a^{p^2}, \ldots, a^{p^m} = a$, which are *conjugates* of a over $\mathbb{F}_q$. Note that the explicit definition of the conjugates comes from a minimal polynomial of a over $\mathbb{F}_p$, which is a polynomial $f(x) \in \mathbb{F}_p[x]$ of the smallest degree m satisfying $f(a) = 0$. If $f(x)$ is a monic polynomial (a polynomial of which a coefficient of the highest degree is 1), all of the roots of $f(x)$ involving a are said to be conjecture of a over $\mathbb{F}_p$ and are given by the form a^{p^i} with an integer $1 \leq i \leq m$. The product of all the conjugates of a is called a *norm* of a and is defined as follows:

$$N_{\mathbb{F}_q/\mathbb{F}_p}(a) = \prod_{i=1}^{n} a^{p^i} \in \mathbb{F}_p. \tag{2.18}$$

The important fact is that $N_{\mathbb{F}_q/\mathbb{F}_p}(a)$ becomes to be an element in the prime field $\mathbb{F}_p$.

Example 2.23. Let $\mathbb{F}_{16} \cong \mathbb{F}_2(\alpha)(\beta)$ be a finite field of order 16 described in Example 2.21. The conjugates of β are $\beta^2 = \beta\alpha + 1$, $\beta^4 = \beta + \alpha$, $\beta^8 = \beta\alpha + \alpha$, and $\beta^{16} = \beta$, which are computed by using the relations $\alpha^2 = \alpha + 1$ and $\beta^2 = \beta\alpha + 1$. Then, the norm of β is computed by $N_{\mathbb{F}_{16}/\mathbb{F}_2}(\beta) = (\beta\alpha + 1) \cdot (\beta + \alpha) \cdot (\beta\alpha + \alpha) \cdot \beta = \alpha^2\beta^4 + (\alpha^3 + \alpha^2 + \alpha)\beta^3 + (\alpha^3 + \alpha^2 + \alpha)\beta^2 + \alpha^2\beta$. Since $\alpha^3 + \alpha^2 + \alpha = 0$, it is obtained $N_{\mathbb{F}_{16}/\mathbb{F}_2}(\beta) = \alpha^2\beta^4 + \alpha^2\beta = (\alpha+1)(\beta+\alpha) + (\alpha+1)\beta = (2\alpha+2)\beta + \alpha^2 + \alpha = 1 \in \mathbb{F}_2$.

2.2.4 Power residue properties

This subsection presents the power residue properties in $\mathbb{F}_p$. Let d be a cofactor of $\#\mathbb{F}_p^* = p - 1$ such that $d \mid (p-1)$[1]. Then, there exists a d-th root of the identity 1 in $\mathbb{F}_p^*$. If there exists $g \in \mathbb{F}_p^*$ such that $a = g^d$ for $a \in \mathbb{F}_p^*$, we say that *a is d-th residue in* $\mathbb{F}_p^*$; otherwise, *a is d-th non-residue in* $\mathbb{F}_p^*$.

Particularly, quadratic residue properties of $d = 2$ are well-studied. There is a convenience symbol that indicates the quadratic residue properties by the values $1, -1, 0$, which was introduced by Legendre and is so-called the *Legendre symbol.*

Definition 2.24. (Legendre symbol) For $a \in \mathbb{F}_p$ and characteristic p, the *Legendre symbol* is defined by

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{if } a \text{ is quadratic residue in } \mathbb{F}_p \text{ and } a \neq 0, \\ -1 & \text{if } a \text{ is quadratic non-residue in } \mathbb{F}_p, \\ 0 & \text{if } a = 0. \end{cases} \tag{2.19}$$

Legendre's original definition is given by using the explicit formula $(\frac{a}{p}) = a^{(p-1)/2}$. There is the following theorem

Theorem 2.25. For an odd prime p, the following is true.

$$\left(\frac{-1}{p}\right) = \begin{cases} 1 & \text{if } p \equiv 1 \pmod 4, \\ -1 & \text{if } p \equiv 3 \pmod 4. \end{cases} \tag{2.20}$$

$$\left(\frac{2}{p}\right) = \begin{cases} 1 & \text{if } p \equiv \pm 1 \pmod 8, \\ -1 & \text{if } p \equiv \pm 3 \pmod 8. \end{cases} \tag{2.21}$$

$$\left(\frac{-3}{p}\right) = \begin{cases} 1 & \text{if } p \equiv 1 \pmod 3, \\ -1 & \text{if } p \equiv 2 \pmod 3. \end{cases} \tag{2.22}$$

Proof. Please refer to [Kob94]. □

Example 2.26. Let $\mathbb{F}_7$ be a field of order 7. Since $p = 7$ satisfies $p \equiv 3 \pmod 4$, $p \equiv -1 \pmod 8$, and $p \equiv 1 \pmod 3$, it is determined $(\frac{-1}{p}) = (\frac{6}{p}) = -1$, $(\frac{2}{p}) = 1$, and $(\frac{-3}{p}) = (\frac{4}{p}) = 1$. The correctness is found from the facts $1^2 = 6^2 = 1$, $2^2 = 5^2 = 4$, $3^2 = 4^2 = 2$, which indicates $1, 2, 4$ are quadratic residue in $\mathbb{F}_7$, but $3, 5, 6$ are quadratic non-residue in $\mathbb{F}_7$.

In addition to the above, cubic residue properties of $d = 3$ are also important in this thesis. Following the definition of the Legendre symbol, let us define a symbol that indicates the cubic residue properties as follows:

[1] For integers a, b, note that $a \mid b$ means that a divides b.

Definition 2.27. Let p be a prime such that $3 \mid (p-1)$. Let ϵ be a cube root of 1 in $\mathbb{F}_p^*$. Then, let us define the symbol

$$\left(\frac{a}{p}\right)_3 = \begin{cases} 1 & \text{if } a \text{ is cubic residue in } \mathbb{F}_p \text{ and } a \neq 0, \\ \epsilon, \epsilon^2 & \text{if } a \text{ is cubic non-residue in } \mathbb{F}_p, \\ 0 & \text{if } a = 0. \end{cases} \tag{2.23}$$

The symbol is explicitly given by $(\frac{a}{p})_3 = a^{(p-1)/3}$. The properties of the cubic residue properties are firstly studied by Euler who provided the following conjecture, which has already been proven today.

Theorem 2.28. *(Euler's conjecture)* Let p be a prime $p \equiv 1 \pmod 3$. Then, p is written by $p = a^2 + 3b^2$ with integers a and b, and the following is true.

$$\left(\frac{2}{p}\right)_3 \begin{cases} = 1 & \text{if } 3 \mid b, \\ \neq 1 & \text{otherwise.} \end{cases} \tag{2.24}$$

$$\left(\frac{3}{p}\right)_3 \begin{cases} = 1 & \text{if either } 9 \mid b,\ 9 \mid (a+b), \text{ or } 9 \mid (a-b), \\ \neq 1 & \text{otherwise.} \end{cases} \tag{2.25}$$

$$\left(\frac{6}{p}\right)_3 \begin{cases} = 1 & \text{if either } 9 \mid b,\ 9 \mid (a+2b), \text{ or } 9 \mid (a-2b), \\ \neq 1 & \text{otherwise.} \end{cases} \tag{2.26}$$

Proof. Please refer to [Lem13]. □

Example 2.29. Let $\mathbb{F}_7$ be a field of order 7. Since $p = 7$ is decomposed into $p = a^2 + 3b^2$ with $a = 2$ and $b = 1$, it is determined $(\frac{2}{p})_3 \neq 1$, $(\frac{3}{p})_3 \neq 1$, and $(\frac{6}{p})_3 = 1$. The correctness is found from the facts $1^3 = 2^3 = 4^3 = 1$ and $3^3 = 5^3 = 6^3 = 6$, which indicates $1, 6$ are cubic residue in $\mathbb{F}_7$, but $2, 3, 4, 5$ are cubic non-residue in $\mathbb{F}_7$.

The power residue properties can also be extended for an extension field $\mathbb{F}_q$ of $\mathbb{F}_p$. For a positive integer d such that $d \mid (p-1)$, the d-th power residue properties of $a \in \mathbb{F}_q$ can be regarded as d-th residue properties of the norm $N_{\mathbb{F}_q/\mathbb{F}_p}(a) \in \mathbb{F}_p$ of a (see Eq. (2.18)). This is because that there is a relation

$$a^{(q-1)/d} = (N_{\mathbb{F}_q/\mathbb{F}_p}(a))^{(p-1)/d}. \tag{2.27}$$

The power residue properties in $\mathbb{F}_q$ can help for the extensions of $\mathbb{F}_q$.

2.2.5 Computational problems

This subsection introduces computational problems in the finite field $\mathbb{F}_q$, which are two of them are related to the DH key exchange protocol, which is introduced in Sect. 1.1.2.

Definition 2.30. (Diffie-Hellman problem (DHP)) Given $g, g^x, g^y \in \mathbb{F}_q^*$ with $x, y \in \mathbb{Z}/n\mathbb{Z}$, compute g^{xy}.

Definition 2.31. (Decisional Diffie-Hellman problem (DDHP)) Given $g, g^x, g^y, g^z \in \mathbb{F}_q^*$ with $x, y, z \in \mathbb{Z}/n\mathbb{Z}$, determine if $g^{xy} = g^z$ or not.

Definition 2.32. (Discrete logarithm problem (DLP)) Given $g, h \in \mathbb{F}_q^*$, compute x such that $h = g^x$.

It is believed that if DLP is difficult, both problems are difficult. If there is an efficient algorithm for solving DHP, it is trivial to solve DDHP by computing g^{xy} from g, g^x, g^y and taking comparison with g^z. If there is an efficient algorithm for solving DLP, it is possible to solve DHP by computing x from g and g^x and then computing $(g^y)^x = g^{xy}$. The algorithm for solving the DLP and its complexity are described in the following.

As one of the methods for solving the DLP, there are the number field sieve (NFS) and its variants. Indeed, the NFS is firstly proposed as an algorithm for solving a factoring problem of a special form by Lenstra in [Len+93] based on an idea by Pollard in [Pol93]. The original NFS is generalized for factoring any composite number. To classify the form of the factorized number, the former and latter methods are called special NFS (SNFS) and generalized NFS (GNFS), respectively.

As the first variant of GNFS, in [Sch93], Schirokauer provided the tower number field sieve (TNFS) for computing DLP in fields $\mathbb{F}_q$. The TNFS is classified into an algorithm called an index calculus method introduced by Kraitchi and is the most efficient classical algorithm for computing the DLP. Its complexity, i.e., the number of steps, for computing the DLP in $\mathbb{F}_q$ is given by the form

$$L_q(\alpha, c) = \exp(c(\ln q)^{\alpha}(\ln \ln q)^{1-\alpha}), \tag{2.28}$$

where α and c are positive real numbers. For a finite field $\mathbb{F}_q$, the complexity is typically given by $L_q(1/3, \sqrt[3]{64/9}) \approx L_q(1/3, 1.923)$. On the other hand, if $\mathbb{F}_q$ with a characteristic p with a very sparse representation, the complexity is reduced to $L_q(1/3, \sqrt[3]{32/9}) \approx L_q(1/3, 1.526)$.

In recent years, it turned out that prime fields and extension fields of the same size of q and p^n with a prime p and integer $n > 1$ do not offer the same security. Indeed, in [BGK15; KB16; KJ17], Barbulescu and Kim et al. revised the TNFS by applying a new setting to finite fields of composite extension degree n, which is called the extended TNFS (exTNFS). Then, the complexity with this new algorithm decreased significantly to $L_q(1/3, \sqrt[3]{32/9}) \approx L_q(1/3, 1.526)$. There are many analyses of the special variant of TNFS (STNFS) of which p is restricted by a special form for PBC [FK19; MSS16; BD19; BD19; FM19; GMT20; Gui20]. Although there is some variation in the complexity corresponding to the improvement of the variant of NFS, the running time is sub-exponential. Currently, to achieve security comparable to AES-128, which is so-called the 128-bit security level, it is required to use around 5,000-bit sizes of q for the finite field used for pairings [Gui20].

2.3 Elliptic curves over finite fields

In mathematics, an *elliptic curve* is a smooth, projective, algebraic curve of genus one, on which there is a specified point. Although not all terms in the definition are explained, this section describes necessary and basic facts of the elliptic curves over finite fields, which are used for cryptography. This section is written by referring to the textbook [Sil09] by Silverman.

2.3.1 Algebraic varieties

The basic objects that arise in the study of algebraic geometry is briefly described. The following shows the definitions of the affine and projective n-spaces over a finite field $\mathbb{F}_q$.

Definition 2.33. (Affine n-space over $\mathbb{F}_q$) Affine n-space over $\mathbb{F}_q$ is the set of n-tuples

$$\mathbb{A}^n = \mathbb{A}^n(\overline{\mathbb{F}}_q) = \{(x_1, x_2, \ldots, x_n) : x_i \in \overline{\mathbb{F}}_q\}. \tag{2.29}$$

Definition 2.34. (Projective n-space over $\mathbb{F}_q$) Projective n-space over $\mathbb{F}_q$, which is denoted by $\mathbb{P}^n$ or $\mathbb{P}^n(\overline{\mathbb{F}}_q)$, is the set of all $(n+1)$-tuples

$$(x_0, x_1, \ldots, x_n) \in \mathbb{A}^{n+1}, \tag{2.30}$$

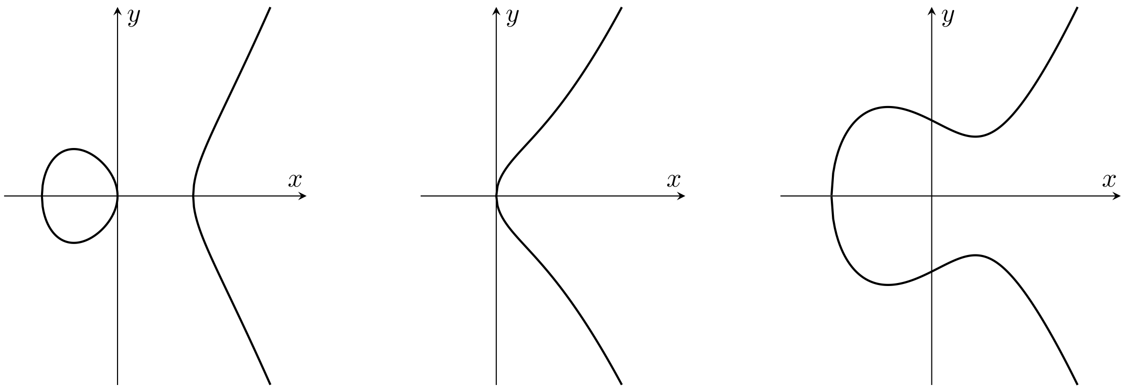
such that at least one x_i is nonzero and $(x_0, x_1, \ldots, x_n)$ is equivalence to $(x'_0, x'_1, \ldots, x'_n)$ if there exists a $\lambda \in \overline{\mathbb{F}}_q^*$ such that $x_i = \lambda x'_i$ for all i.

The tuples, which are elements in $\mathbb{A}^n$ and $\mathbb{P}^n$, are called *rational points* or *points.* In this thesis, an equivalence class of a rational point $(x_0, x_2, \ldots, x_n)$ in $\mathbb{P}^2$ is denoted as $(x_0 : x_1 : \ldots : x_n)$. Then, the individual $x_0, x_1, \ldots, x_n$ are called *homogeneous coordinates* for the corresponding points in $\mathbb{P}^n$. One can embed $\mathbb{A}^n$ into $\mathbb{P}^n$ by sending the coordinate $(x_1, x_2, \ldots, x_n) \mapsto (x_1 : x_2 : \ldots : x_n : 1)$. Notice that there are additional homogenous points $(x_1 : x_2 : \ldots : x_n : 0)$ in $\mathbb{P}^n$, which are called *points at infinity* in $\mathbb{A}^n$.

2.3.2 Weierstrass equations

Every elliptic curve over a finite field $\mathbb{F}_q$ can be written by a cubic equation of the homogeneous coordinate in the 2-space $\mathbb{P}^2$ over $\overline{\mathbb{F}}_q$, where are written by the form $(X : Y : Z)$ with $X, Y, Z \in \overline{\mathbb{F}}_q$. In this context, the equation is given by

$$Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3. \tag{2.31}$$



(a) $y^2 = x^3 - x$ over $\mathbb{R}$ (b) $y^2 = x^3 + x$ over $\mathbb{R}$ (c) $y^2 = x^3 - x + 1$ over $\mathbb{R}$

Figure 2.1: Three elliptic curves.

where $a_1, a_2, \dots, a_6 \in \mathbb{F}_q$. Then, a point at infinity satisfying the above equation is written as $\mathcal{O} = (0 : 1 : 0)$.

To ease notation, the equation is generally written by non-homogeneous coordinates $x = X/Z$ and $y = Y/Z$,

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{2.32}$$

which lies in the affine 2-space $\mathbb{A}^2$ over $\overline{\mathbb{F}}_q$. Aside from all the solutions of Eq. (2.32), there is one extra point $\mathcal{O}$ out at infinity. As usual, if the coefficients of E are in $\mathbb{F}_q$, E is said to be defined over $\mathbb{F}_q$ and is denoted as $E/\mathbb{F}_q$.

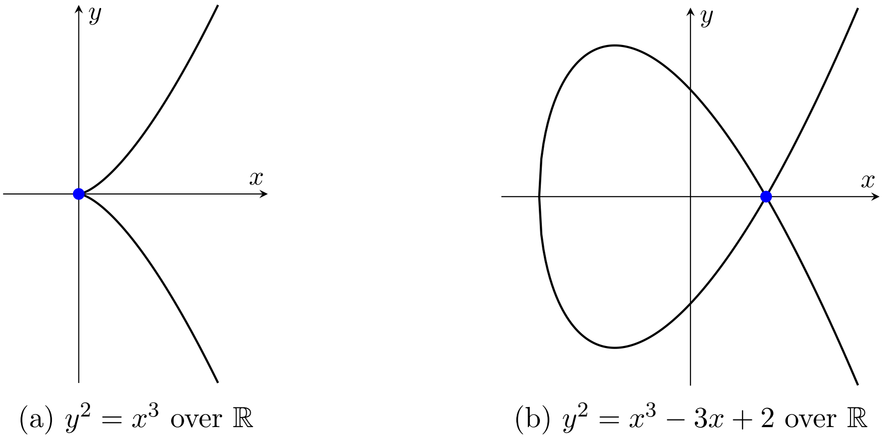
If the field characteristic char$(\mathbb{F}_q)$ is not 2 and 3, one can simplify Eq. (2.32). The restriction permits the substitution $y \mapsto (y - a_1x - a_3)/2$ that gives rise to an equation of the form $E : y^2 = 4x^3 + b_2x^2 + 2b_4x + b_6$ where $b_2, b_4, b_6 \in \mathbb{F}_q$. Then, one more substitution $(x, y) \mapsto ((x - 3b_2)/36, y/108)$ results in

$$E : y^2 = x^3 + ax + b, \tag{2.33}$$

where $a, b \in \mathbb{F}_q$. Eq. (2.33) is called the *short Weierstrass equation.*

Let C be a curve defined by $f(x, y) = y^2 - (x^3 + ax + b) = 0$. Not all a and b gives rise to a curve C being an elliptic curve. If there exists a point P on C such that it is not differentiable at P, the point P is called a *singular point*, and C is also said to be *singular.* If C does not admit such points, we say C is *non-singular* or *smooth* and is an elliptic curve. If and only if the discriminant defined by $\Delta = -16(4a^3 + 27b^2)$ is not 0, C is non-singular and elliptic curve. As the quantity related to Δ, we say $j = -1728 \cdot (4a)^3/\Delta$ is the *j-invariant* of an elliptic curve.

To support the understanding, Figures 2.1 and 2.2 illustrate the singular and non-singular elliptic curves over a set $\mathbb{R}$ of real numbers, respectively. Note that the elliptic



(a) $y^2 = x^3$ over $\mathbb{R}$ (b) $y^2 = x^3 - 3x + 2$ over $\mathbb{R}$

Figure 2.2: Two singular curves.

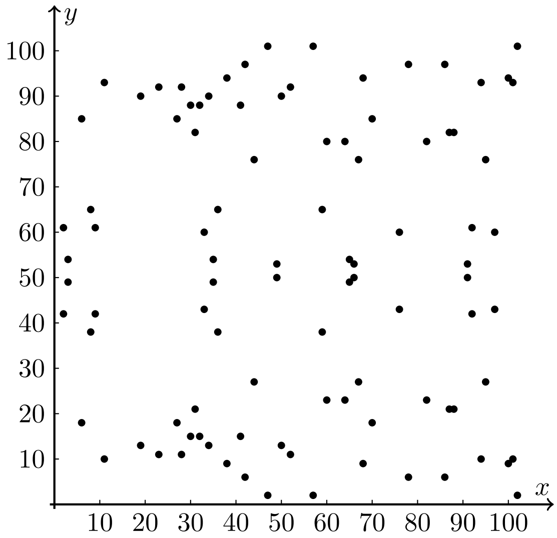


Figure 2.3: $y^2 = x^3 + 5$ over $\mathbb{F}_{103}$.

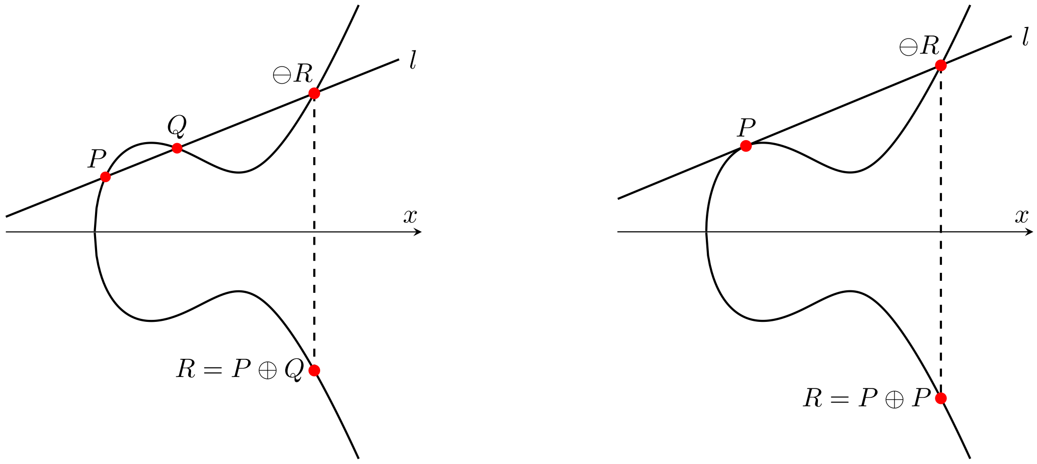
curves over $\mathbb{R}$ will appear in the following descriptions. Figure 2.3 shows rational points on the elliptic curve over a prime field $\mathbb{F}_p$ excluding $\mathcal{O}$, which are actually used for cryptography.

2.3.3 Group law

Let E be an elliptic curve given by the short Weierstrass equation. Let $E(\mathbb{F}_q)$ be a set of rational points of E over $\mathbb{F}_q$ defined by

$$E(\mathbb{F}_q) = \{(x, y) \in \mathbb{F}_q \times \mathbb{F}_q : (x, y) \text{ satisfies Eq. (2.33)}\} \cup \{\mathcal{O}\}. \tag{2.34}$$

Note that sometimes the set $E(\overline{\mathbb{F}}_q)$ is referred to E. There is a law $\oplus$ making the set to be an abelian group, which is called a $\mathbb{F}_q$-*rational point group*. The law comes from the fact that a line l intersects E at exactly three points which are not necessary to be distinct.



(a) Elliptic curve addition.

(b) Elliptic curve doubling.

Figure 2.4: The composition law.

The fact is a special case of Bázout's theorem [Har+75]. Let us define a composition law $\oplus$ by the following rule:

Definition 2.35. (Group law) Let l be a line passing through P and Q on E. If $P = Q$, let l be a tangent line to E at P. Then, one can find the third point of intersection of l with E, which is denoted as $\ominus R$. Let us take a point R of x-axis symmetry of $\ominus R$ and define that as $P \oplus Q$.

The instances of the law $\oplus$ are illustrated in Figure 2.4. The group law has the following properties.

- The law $\oplus$ has the properties of the associative and commutative.
- The point $\mathcal{O}$ at infinity plays a role of the identity, i.e., $P \oplus \mathcal{O} = P$ for any P.
- For any P, there exists a point on E defined as $\ominus P$ such that $P \oplus (\ominus P) = \mathcal{O}$. In fact, $\ominus P$ is P's reflected image over the x-axis.

The properties indicate that $\oplus$ makes $E(\mathbb{F}_q)$ into an abelian group with the identity $\mathcal{O}$. The properties also show that the result of the addition of all three points of the intersections of a line and E becomes to be the identity $\mathcal{O}$. Thus, the intersections are given by points P, Q, and $\ominus(P \oplus Q)$. Even though the line is vertical, one can regard that there are three intersections which consist of P, $\ominus P$, and $\mathcal{O}$. For these reasons, the notation $\ominus R$ is used in Definition 2.35 and Figure 2.4.

From the definition, one can obtain point addition and doubling formulas as follows:

Lemma 2.36. (Doubling and addition formulas) Let $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ be affine points on E. If $Q \neq \ominus P$, point addition and doubling formulas for computing $R = P \oplus Q = (x_R, y_R)$ are given as follows:

- Addition formula ($P \neq Q$):

$$(x_R, y_R) = (\lambda^2 - x_P - x_Q, \lambda(x_P - x_R) - y_P), \lambda = \frac{y_Q - y_P}{x_Q - x_P}. \tag{2.35}$$

- Doubling formula ($P = Q$):

$$(x_R, y_R) = (\lambda^2 - 2x_P, \lambda(x_P - x_R) - y_P), \lambda = \frac{3x_P^2 + a}{2y_P}. \tag{2.36}$$

If $Q = \ominus P$, it is clearly $R = P \oplus Q = \mathcal{O}$.

In the following, the derivation of the formulas is briefly described. Let l be a line passing through P and Q defined by $l : y = \lambda x + \nu$ where $\lambda, \nu \in \mathbb{F}_q$. Then, the gradient λ of l is given by $\lambda = (y_Q - y_P)/(x_Q - x_P)$ if $P \neq Q$; $\lambda = (3x_P^2 + a)/2y_P$ if $P = Q$. Note that the denominator of λ cannot be 0 since we work on the assumption $Q \neq \ominus P$, i.e., $(x_Q, y_Q) \neq (x_P, -y_P)$, which gives rise to $x_Q - y_P \neq 0$ and $y_Q + y_P \neq 0$. The y-intercept ν of l is also given as $\nu = -\lambda x_P + y_P$. Assuming $R = (x_R, y_R)$, the third point of the intersections of l and E is written as $\ominus R = (x_R, -y_R)$. Since P, Q, and R (or $\ominus R$) are on E, one can represent $x^3 + ax + b - y^2 = (x - x_P)(x - x_Q)(x - x_R)$. When substituting l into the left side and expanding the right side, it is obtained $x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x + b - \nu^2 = x^3 + (-x_P - x_Q - x_R)x^2 + (x_P x_Q + x_Q x_R + x_R x_P)x - x_P x_Q x_R$. Looking the coefficient of x^2, we have $-x_P - x_Q - x_R = -\lambda^2$, which leads to $x_R = \lambda^2 - x_P - x_Q$. Since $\ominus R = (x_R, -y_R)$ is on l, $y_R = \lambda(x_P - x_R) - y_P$. As a result, the above formulas are derived.

The following example confirms how $\oplus$ does work on the curves on a finite field.

Example 2.37. (Point addition) Let E be an elliptic curve defined by $E/\mathbb{F}_{11} : y^2 = x^3 + 2x + 1$. Then, there are points $P = (1, 9)$ and $Q = (3, 10)$ in $E(\mathbb{F}_{11})$. A line passing through P and Q is easily obtained as $l : y = (1/2)x + 17/2$, which is equivalence to $l : y = 6x + 3$ over $\mathbb{F}_{11}$. Applying the addition formula given in Eq. (2.35), it is obtained $R = P \oplus Q = (x_R, y_R)$ where $x_R = 6^2 - 1 - 3 \equiv 10 \pmod{11}$ and $y_R = 6(1 - 10) - 9 \equiv 3 \pmod{11}$, i.e., $R = (10, 3)$. Figure 2.5 illustrates that the points on $E(\mathbb{F}_{11})$ excluding $\mathcal{O}$ and process of the addition $(1, 9) \oplus (3, 10) = (10, 3)$.

Repeating to use $\oplus$ for $P \in E$ leads to the definition of a point mP which is P multiplied by $m \in \mathbb{Z}$. For $m > 0$, let

$$mP = \underbrace{P \oplus P \oplus \cdots \oplus P}_{m\text{-terms}}. \tag{2.37}$$

For $m < 0$, set $mP = -m(\ominus P)$ and define $0P = \mathcal{O}$. The point multiplication can be performed as shown in the following example.

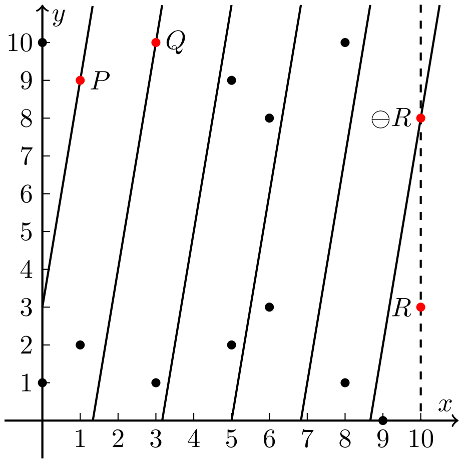


Figure 2.5: Point addition in $E/\mathbb{F}_{11}$: $y^2 = x^3 + 2x + 1$.

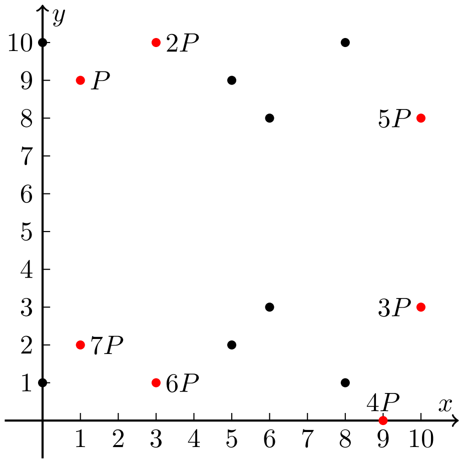


Figure 2.6: Point miultiplication in $E/\mathbb{F}_{11} : y^2 = x^3 + 2x + 1$.

Example 2.38. (Point multiplication) Let E be an elliptic curve defined by $E/\mathbb{F}_{11}$: $y^2 = x^3 + 2x + 1$. For $P = (1, 9) \in E$, one can compute $2P = (1, 9) \oplus (1, 9) = (3, 10)$, $3P = (3, 10) \oplus (1, 9) = (10, 3)$, $4P = (10, 3) \oplus (1, 9) = (9, 0)$, $5P = (9, 0) \oplus (1, 9) = (10, 8)$, $6P = (10, 8) \oplus (1, 9) = (3, 1)$, $7P = (3, 1) \oplus (1, 9) = (1, 2)$, and $8P = (1, 2) \oplus (1, 9) = \mathcal{O}$. Thus, $G = \langle P \rangle$ forms a cyclic group of order 8. Figure 2.6 illustrates that these points excluding $\mathcal{O}$.

From here on, we drop the special symbols $\oplus$ and $\ominus$, and simply write $+$ and $-$ for the group operations on an elliptic curve E.

2.3.4 Point multiplication and Frobenius endomorphisms

Since the rational point group of an elliptic curve E over $\mathbb{F}_q$ is defined, this subsection discusses the endomorphisms from E to E. This subsection provides very elementary endomorphisms that play an important role in E. Note that the endomorphisms are one kind of map which is called an isogeny and is defined in Sect. 2.5.

In this context, the group law on E leads to a map from a point $P \in E$ to a multiplied point $mP \in E$ with certain integer m, which results in the following endomorphism.

Definition 2.39. (Point multiplication endomorphism) Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$ and let m be an integer. The *point multiplication endomorphism* is defined by

$$[m] : E \to E, P \mapsto mP. \tag{2.38}$$

For $m > 0$, the explicit descriptions of the point multiplication endomorphism are usually given in terms of *division polynomials* $\Psi_i \in \mathbb{Z}[x, y]$ defined by

$$\Psi_1 = 1, \tag{2.39}$$

$$\Psi_2 = 2y, \tag{2.40}$$

$$\Psi_3 = 3x^4 + 6ax^2 + 12bx - a^2, \tag{2.41}$$

$$\Psi_4 = 4y(x^6 + 5ax^4 + 20bx^3 - 5a^2x^2 - 4abx - 8b^2 - a^3), \tag{2.42}$$

and then inductively by the formulas

$$\Psi_{2i+1} = \Psi_{i+2}\Psi_i - \Psi_{i-1}\Psi_{i+1}^3, \quad i \geq 2, \tag{2.43}$$

$$2y\Psi_{2i} = \Psi_i(\Psi_{i+2}\Psi_{i-1}^2 - \Psi_{i-2}\Psi_{i+1}^2), \quad i \geq 3. \tag{2.44}$$

In addition to this, let us define polynomials Φ_i and Ω_i:

$$\Phi_i = x\Psi_i^2 - \Psi_{i+1}\Psi_{i-1}, \quad 4y\Omega_i = \Psi_{i+2}\Psi_{i-1}^2 - \Psi_{i-2}\Psi_{i+1}^2. \tag{2.45}$$

Then, the image of $(x, y) \in E$ under $[m]$ is given as follows:

$$[m](x, y) = \left(\frac{\Phi_m(x, y)}{\Psi_m^2(x, y)}, \frac{\Omega_m(x, y)}{\Psi_m^3(x, y)} \right). \tag{2.46}$$

There is the Frobenius endomorphism in $\mathbb{F}_q$ such that $\pi_q : \overline{\mathbb{F}}_q \to \overline{\mathbb{F}}_q, a \to a^q$. Applying the endomorphism for the both side of E in short Weierstrass equation, we have $(y^2)^q = (x^3 + ax + b)^q$ which can be written as $(y^q)^2 = (x^q)^3 + a^q x^q + b^q = (x^q)^3 + ax^q + b$ and thus $(x^q, y^q) \in E$. This leads to the following endomorphism.

Definition 2.40. (Frobenius endomorphism on E) Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$. The *Frobenius endomorphism on E* is defined by

$$\pi_p : E \to E, (x, y) \mapsto (x^q, y^q). \tag{2.47}$$

Similar to the Frobenius endomorphism in $\overline{\mathbb{F}}_q$, there is a property such that $\pi_q(P) = P$ for a point $P = (x, y)$ with $x, y \in \mathbb{F}_q$. This leads to the following representation of $\mathbb{F}_q$-rational point group.

$$E(\mathbb{F}_q) = \{P \in E : \pi_q(P) = P\} \subset E. \tag{2.48}$$

In the following, m-th iterate of the Frobenius endomorphism in E is denoted by π_q^m.

Besides the above, one can also define various endomorphisms from E to E. Let $\mathrm{End}(E)$ be a set of all endomorphisms, $\mathrm{End}(E)$ forms a ring under the following addition and multiplication. For $\varphi, \phi \in \mathrm{End}(E)$, the addition $\varphi + \phi \in \mathrm{End}(E)$ and multiplication

$\varphi \cdot \phi \in \mathrm{End}(E)$ are defined by $(\varphi + \phi)(P) = \varphi(P) + \phi(P)$ and $(\varphi \cdot \phi)(P) = \phi(\varphi(P))$, respectively. Similarly, there is a subtraction $\varphi - \phi \in \mathrm{End}(E)$ by $(\varphi - \phi)(P) = \varphi(P) + (-\phi(P))$. As we used the notation of the m-th iterate of Frobenius endomorphism, a power φ to m is defined by $\varphi^m(P) = \varphi \cdot \varphi \cdot \cdots \cdot \varphi$.

2.3.5 Twisting isomorphisms

This subsection describes the isomorphisms between two elliptic curves. In the following, let E and E' be elliptic curves defined over $\mathbb{F}_q$.

Definition 2.41. (Twist) If there exists such isomorphism $\phi_d : E' \to E$ defined over $\mathbb{F}_{q^d}$ with the minimal integer d, then E' is called a *twist of degree d of E*.

In this thesis, $\phi_d : E' \to E$ is called a *twisting isomorphism*. It is naturally found that there is an inverse isomorphism $\phi_d^{-1} : E \to E'$, which is called an *untwisting isomorphism*. The important fact is that there are only possibilities $d = 1$, 2, 3, 4, and 6 which depend on the j-invariant of E. If $d = 1$, E' is typically not called the twist, E' is also considered as a twist in this thesis. If $d = 2$, 3, 4, and 6, E' is called a *quadratic twist*, *cubic twist*, *quartic twist*, and *sextic twist*, respectively. The explicit formulas of the twist E' of degree d of E and twisting isomorphism $\phi_d : E' \to E$ are summarized below.

- $d = 1, 2$: The twist can be occur for every value of $j(E)$. For $E : y^2 = x^3 + ax + b$, the twist E' of E is given by $y^2 = x^3 + a/\delta^2 x + b/\delta^3$ where δ is quadratic-residue if $d = 1$; quadratic non-residue in $\mathbb{F}_q^*$ if $d = 2$. The twisting isomorphism is written as follows:

$$\phi_d : E' \to E, (x, y) \mapsto (\delta x, \delta^{\frac{1}{2}} y). \tag{2.49}$$

- $d = 4$: The twist occur only the case of $j(E) = 1728$, i.e., E is given by $y^2 = x^3 + ax$. The twist E' of E is given by $y^2 = x^3 + a/\delta x$ where δ is quartic non-residue (4-th non-residue) in $\mathbb{F}_q^*$. The twisting isomorphism is given by

$$\phi_d : E' \to E, (x, y) \mapsto (\delta^{\frac{1}{2}} x, \delta^{\frac{3}{4}} y). \tag{2.50}$$

- $d = 3, 6$: The twist occur only the case of $j(E) = 0$, i.e., E is given by $y^2 = x^3 + b$. The twist E' of E is given by $y^2 = x^3 + b/\delta$ where δ is quadratic residue but cubic non-residue in $\mathbb{F}_q^*$ if $d = 3$; quadratic and cubic non-residue in $\mathbb{F}_q^*$ if $d = 6$. The twisting isomorphism is given as follows:

$$\phi_d : E' \to E, (x, y) \mapsto (\delta^{\frac{1}{3}} x, \delta^{\frac{1}{2}} y). \tag{2.51}$$

The twisting and untwisting isomorphisms are often used for computing the pairings on elliptic curves efficiently. The high degree twists result in speeding up the pairings.

2.3.6 n-torsion subgroups

Since the point multiplication endomorphism is defined, a subset of an elliptic curve can be defined as follows:

Definition 2.42. Let E be an elliptic curve defined over $\mathbb{F}_q$ and let n be a positive integer. An n-torsion subgroup is a subgroup of E which consists of rational points of order n defined by

$$E[n] = \{P \in E : [n]P = \mathcal{O}\}. \tag{2.52}$$

Let $p = \text{char}(\mathbb{F}_q)$ and $i > 0$ be an integer. Then, the structure of $E[n]$ is determined as follows:

$$E[n] \cong \begin{cases} \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z} & \text{if } n \neq p^i, \\ \mathbb{Z}/n\mathbb{Z} \text{ or } \{0\} & \text{if } n = p^i. \end{cases} \tag{2.53}$$

From the above, there are two possibilities for the case of $n = p^i$. The difference of the structure leads to the definition of exceptional elliptic curves described in Sect. 2.3.7.

Our main concern is in the n-torsion subgroup of the structure given by $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$. The details of the structure are described below. The structure $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ indicates that the number of points in $E[n]$ is given by $\#E[n] = n^2$. This implies that $E[n]$ consists of $(n + 1)$ subgroups of order n. This is because that the identity $\mathcal{O}$ overlaps into all subgroups of order n, i.e., $\#E[n]$ is decomposed into $\#E[n] = n^2 = (n + 1)n - n$.

Example 2.43. (2-torsion subgroup) Let E be an elliptic curve over $\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$. It is easily found that the y-coordinate of the point of order 2 is zero. Since E is nonsingular, $x^3 + ax + b = 0$ has three distinct solutions, there are three points of order 2 of the form $(\alpha, 0)$ over $\overline{\mathbb{F}}_q$. Thus, $E[2]$ consists of the three points of order 2 and $\mathcal{O}$. Since a subgroup of order 2 is generated per one point of order 2, it is found that $E[2]$ consists of three subgroups of order 2, which leads to $E[2] \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$.

Let picking up two distinct subgroups G_1 and G_2 from $E[n]$ and let P_1 and P_2 be points in G_1 and G_2, respectively. Then, any point $P \in E[n]$ can be represented by the $\mathbb{Z}/n\mathbb{Z}$-linear combination of P_1 and P_2 as follows:

$$P = m_1P_1 + m_2P_2, \tag{2.54}$$

where $m_1, m_2 \in \mathbb{Z}/n\mathbb{Z}$. Thus, we have $E[n] = \langle P_1, P_2\rangle$ by using generators P_1 and P_2.

Let restrict the endomorphisms $\varphi : E \to E$ to $\varphi_n : E[n] \to E[n]$. Since $P \in E[n]$ is given by vector of the basis (P_1, P_2), one can represent the endomorphism φ_n as a 2×2 matrix $\varphi_n = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ where $a, b, c, d \in \mathbb{Z}/n\mathbb{Z}$ that are determined by

$$\varphi_n(P_1) = aP_1 + bP_2, \tag{2.55}$$

$$\varphi_n(P_2) = cP_1 + dP_2. \tag{2.56}$$

The trace and determinant of φ_n are computed by $\mathrm{tr}(\varphi_n) = a+d$ and $\det(\varphi_n) = ad-bc$ and are the values in $\mathbb{Z}$. Unlike the matrix representation, $\mathrm{tr}(\varphi_n)$ and $\det(\varphi_n)$ are independent of the choice of basis. Assuming $E = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$ is the identity matrix, the characteristic polynomial of φ_n is defined by $\det(\lambda E - \varphi_n) = \lambda^2 - \mathrm{tr}(\varphi_n)\lambda + \det(\varphi_n)$ where $\lambda \in \mathbb{Z}$. According to the Cayley-Hamilton theorem, φ_n satisfies the following.

$$\varphi_n^2 - [\mathrm{tr}(\varphi_n)]_n \cdot \varphi_n + [\det(\varphi_n)]_n = [0]_n, \tag{2.57}$$

where $[m]_n$ is the restricted point multiplication endomorphism defined by $[m]_n : E[n] \to E[n], P \mapsto mP$. It is also possible to eliminate the subscript n from Eq. (2.57) since $\mathrm{tr}(\varphi_n)$ and $\det(\varphi_n)$ are independent on n and $\cup_{n=1}^{\infty} E[n] = E$. Particularly, the characteristic polynomial of the Frobenius endomorphism is explicitly determined by Hasse. This is also related to the estimation of the number of points on a certain rational point group. The details are described in the next subsection.

In the rest of this subsection, the belonging of the n-torsion subgroup is discussed.

Theorem 2.44. Let E be an elliptic curve over $\mathbb{F}_q$, let $n > 0$ be an integer satisfying $\gcd(q-1, n) = 1$[2], and let $k > 0$ be the smallest integer such that there is a multiplicative subgroup μ_n of $\mathbb{F}_{q^k}^*$ of order n. If there exists a point P of order n in the $\mathbb{F}_q$-rational point group $E(\mathbb{F}_q)$, then $E[n] \subset E(\mathbb{F}_{q^k})$.

The quantity k is called the *embedding degree with respect to n*. Since the multiplicative group $\mathbb{F}_{q^k}^*$ is a cyclic group of order $q^k - 1$, it can be said that k is the smallest integer satisfying $n \mid (q^k - 1)$.

2.3.7 Supersingular elliptic curves

As seen in the previous subsection, there are two possibilities of the structure of the p-torsion subgroup $E[p]$ of a prime p. This leads to the exceptional elliptic curves:

Definition 2.45. (Supersingularity) Let E be an elliptic curve over $\mathbb{F}_q$ with $p = \mathrm{char}(\mathbb{F}_q)$. If $E[p]$ has the structure $E[p] \cong \{0\}$, E is said to be *supersingular*; otherwise, E is *non-supersingular* or *ordinary.*

[2] gcd is a function that returns the greatest common divisor of integers.

Note that the term “supersingular” has nothing to do with “singular” curves, and all supersingular elliptic curves are non-singular. Elliptic curves over such fields which are not supersingular are called ordinary and these two classes of elliptic curves behave fundamentally differently in many aspects.

There is a convenient theorem that can determine the supersingularity from the curve equation.

Theorem 2.46. Let E be an elliptic curve over $\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$. Then, E is supersingular if and only if the coefficient of x^{p-1} in $(x^3 + ax + b)^{(p-1)/2}$ is zero.

The following shows the application of the above theorem.

Example 2.47. (Supersingular elliptic curve) Let E be an elliptic curve given by $y^2 = x^3 + x$. We need to compute the coefficient of x^{p-1} in the polynomial $(x^3 + x)^{(p-1)/2}$. Since $x^{(p-1)/2}(x^2 + 1)^{(p-1)/2}$ and $x^{p-1} = x^{(p-1)/2+(p-1)/2}$, this is equivalence to compute the coefficient of $x^{(p-1)/2}$ in the polynomial $(x^2 + 1)^{(p-1)/2}$. If $p \equiv 1 \pmod 4$, the target coefficient can exist in the $(p-1)/2$-th row and $(p-1)/4$-th column of Pascal's triangle, which is computed by ${}_{(p-1)/2}C_{(p-1)/4}$. If $p \equiv 3 \pmod 4$, the coefficient cannot be exist in the triangle, and thus it is zero. Hence, E is supersingular if $p \equiv 3 \pmod 4$ and ordinary if $p \equiv 1 \pmod 4$.

2.3.8 The number of rational points

This section wishes to estimate the number of points in the subset of an elliptic curve E over $\mathbb{F}_q$. The border of the number of $\mathbb{F}_q$-rational point $E(\mathbb{F}_q)$ is conjectured by E. Artin in his thesis, which is proven by Hasse, and is given as follows [Sil09]:

Theorem 2.48. (Hasse) Let E be an elliptic curve defined over $\mathbb{F}_q$.

$$\#E(\mathbb{F}_q) = q + 1 - t, \quad |t| \leq 2\sqrt{q}. \tag{2.58}$$

The quantity t is called the *Frobenius trace* since that plays the trace $\mathrm{tr}(\pi_q)$ of the Frobenius endomorphism π_q in the n-torsion subgroups for $1 \leq n \leq \infty$ that is given by the 2×2 matrix as described in Sect. 2.3.6. Indeed, Hasse also proved the following theorem.

Theorem 2.49. Let E be an elliptic curve over $\mathbb{F}_q$ and let π_q be the Frobenius endomorphism $\pi_q : E \to E, (x, y) \to (x^q, y^q)$. Then, the characteristic polynomial of π_q is $\lambda^2 - t\lambda + q$ and π_q satisfies

$$\pi_q^2 - [t] \cdot \pi + [q] = [0]. \tag{2.59}$$

For an integer $m > 1$, the number of $\mathbb{F}_{q^m}$-rational point group $E(\mathbb{F}_{q^m})$ can also be estimated from the knowledge of q and t.

Theorem 2.50. (Weil) Let E be an elliptic curve and let $t = q + 1 - \#E(\mathbb{F}_q)$. Let α, β be roots of the polynomial $x^2 - tx + q \in \mathbb{C}[x]$. Then, α and β are complex conjectures satisfying $\alpha\beta = q$ and $\alpha + \beta = t$, and for any $m > 0$,

$$\#E(\mathbb{F}_{q^m}) = q^m + 1 - t_m, \quad t_m = \alpha^m + \beta^m. \tag{2.60}$$

If E is supersingular, there is a special number of the points as shown below.

Corollary 2.51. Let E be a supersingular elliptic curve over $\mathbb{F}_p$ with a prime p such that $p > 3$. For $m > 0$,

$$\#E(\mathbb{F}_{p^m}) = \begin{cases} p^m + 1 & \text{if } m \text{ is odd,} \\ (p^{m/2} - (-1)^{m/2})^2 & \text{if } m \text{ is even.} \end{cases} \tag{2.61}$$

Besides, the change of the number of rational points by the twisting isomorphisms are described below. Given elliptic curve E, the number of rational points of a twist E' of E is specifically determined as follows [HSV06]:

Theorem 2.52. Let E be an elliptic curve defined over $\mathbb{F}_q$ such that $\#E(\mathbb{F}_q) = q + 1 - t$. Let E' be a twist of degree d of E. Then, the number of points on $E'(\mathbb{F}_q)$ is given by

$$\#E'(\mathbb{F}_q) = \begin{cases} q + 1 - t & \text{if } d = 1, \\ q + 1 + t & \text{if } d = 2, \\ q + 1 - \frac{\pm 3f - t}{2} & \text{if } d = 3, \\ q + 1 \pm f & \text{if } d = 4, \\ q + 1 - \frac{\pm 3f + t}{2} & \text{if } d = 6, \end{cases} \tag{2.62}$$

where f is an integer satisfying $f^2 = 4q - t^2$ if $d = 4$; $3f^2 = 4q - t^2$ if $d = 3, 6$.

Let D be an integer satisfying $Df^2 = 4q - t^2$. Since D is often used for the complex multiplication (CM) method [AM93] for constructing an elliptic curve with the desirable number of rational points, D is called the *CM discriminant.* Note that the value of D is corresponding to the j-invariant $j(E)$, e.g., $D = 3$ if $j(E) = 0$; $D = 1$ if $j(E) = 1728$.

2.3.9 Computational problems

This subsection presents computational problems related to elliptic curves E over $\mathbb{F}_q$. Similar to the finite field, there are problems related to the ECDH key exchange.

Definition 2.53. (Elliptic curve Diffie-Hellman problem (ECDHP)) Given $P, xP, yP \in E(\mathbb{F}_q)$ with $x, y \in \mathbb{Z}$, compute $xyP \in E(\mathbb{F}_q)$.

Definition 2.54. (Elliptic curve decisional Diffie-Hellman problem (ECDDHP)) Given $P, xP, yP, zP \in E(\mathbb{F}_q)$ with integers $x, y, z \in \mathbb{Z}$, determine if $xyP = zP$ or not.

Definition 2.55. (Elliptic curve discrete logarithm problem (ECDLP)) Given $P, Q \in E(\mathbb{F}_q)$, an integer x such that $Q = xP$.

If there is an efficient algorithm for solving ECDHP, it is trivial to solve ECDDHP since xyP can be computed from P, xP, yP. If there is an efficient algorithm for solving ECDLP, it is also possible to solve ECDHP by computing x from P and xP and then computing $x(yP) = xyP$. The algorithm for solving ECDLP and its complexity are summarized below.

In [Pol78], Pollard introduced an algorithm for solving ECDLP, which is known as Pollard's rho algorithm. An original idea of the algorithm comes from the analogous to Pollard's rho algorithm for factoring a composite number. The algorithm is known as the most efficient solution to ECDLP except for special curves such that supersingular elliptic curves, and it require $O(\sqrt{q})$ steps. There are several revisions of Pollard's rho algorithm, however, there is no dramatic improvement at this time. Thus, the running time is exponential. Unlike the DLP in $\mathbb{F}_q$, since there is no sub-exponential algorithm for solving ECDLP, the size of q can be fixed smaller than that of DLP. Thus, it is expected that the cryptographic systems based on ECDLP are faster and more compact cryptographic systems than that are based on the DLP or factorization problem at the same security level. To achieve the 128-bit security level, it is needed to use 256-bit size of order of $P \in E$.

2.4 Pairings on elliptic curves

Recall that the pairing is a bilinear and non-degenerate map defined by $e : G_1 \times G_2 \to G_T$ where G_1, G_2, and G_T are abelian groups of common order (see Sect. 2.1.3). In cryptography, for practical reasons, it is used the pairings defined on elliptic curves E over $\mathbb{F}_q$. Let fix G_1 and G_2 as two rational point groups of E of prime order r, which are subgroups in r-torsion subgroup $E[r]$ generated by points $P_1, P_2 \in E[r]$, and let define the pairing as follows:

$$e : \langle P_1 \rangle \times \langle P_2 \rangle \to \mu_r, \tag{2.63}$$

where μ_r is a multiplicative subgroup of $\overline{\mathbb{F}}_q$ of order r. This section provides the necessary fundamentals of the pairings on elliptic curves and defines the Weil and Tate pairings, which are the most well-known and are often applied for cryptography based on the pairings. Note that the descriptions are written by referring to the textbook [EMJ17] by El Mrabet and Joye.

2.4.1 Function fields

This subsection describes the field built from an elliptic curve E. For easy understanding, we work on the affine space $\mathbb{A}^2$, however, we actually have to work on the projective space $\mathbb{P}^2$. Let $\mathbb{F}_q[x, y]$ be a set of all polynomials defined over $\mathbb{F}_q$ with two variables x and y, which forms a polynomial ring. The important fact is that a polynomial $f(x, y)$ in $\mathbb{F}_q[x, y]$ is absolutely irreducible over $\mathbb{F}_q$ if $f(x, y)$ is non-singular. Thus, the elliptic curve equation $E(x, y)$ is irreducible in $\mathbb{F}_q[x, y]$. This leads to the following definition of a function field.

Definition 2.56. (Function field) Let E be an elliptic curve defined over $\mathbb{F}_q$ given by the Weierstrass equation $y^2 = x^3 + ax + b$ and let $E(x, y) = y^2 - x^3 - ax - b \in \mathbb{F}_q[x, y]$. Let $\mathbb{F}_q[E]$ be a polynomial ring defined by $\mathbb{F}_q[E] = \mathbb{F}_q[x, y]/(E(x, y))$, which is an integral domain. A *function field* of E is a quotient field of $\mathbb{F}_q[E]$ defined by

$$\mathbb{F}_q(E) = \{f = g/h : g \in \mathbb{F}_q[E], 0 \neq h \in \mathbb{F}_q[E]\}. \tag{2.64}$$

The set of $\mathbb{F}_q[E]$ consists of equivalence classes of polynomials $f(x, y)$ modulo $E(x, y)$. Every element in $\mathbb{F}_q[E]$ can be denoted by the form $f(x, y) = u(x) + v(x)y$ where $u(x)$ and $v(x)$ are polynomials in $\mathbb{F}_q[x]$. Therefore, an element in $\mathbb{F}_q(E)$ can be considered as a rational function.

Let f be a rational function in $\mathbb{F}_q(E)$ given by $f = g/h$ where g and h are elements in $\mathbb{F}_q[E]$ that have no common factors. When evaluating f at a point on E, there are the points that are roots of g and h, and which are called *zeros* and *poles*, respectively. For example, suppose that $g = (u(x) + v(x)y)^m$ and $h = (s(x) + t(x)y)^n$ and P and Q are points on E such that $g(P) = 0$ and $h(Q) = 0$. Then, we say f has zeros at P with multiplicity m and poles at Q with multiplicity n. It can also be considered the poles of multiplicity n to be the zeros of negative multiplicity $-n$. Notice that the zeros are intersection points of $f(x, y) = g(x, y)/h(x, y) = 0$ and $E(x, y) = 0$. The following examples provide the zeros and poles of sloped and vertical line functions in $\mathbb{F}_q(E)$.

Example 2.57. (Zeros and poles of a sloped line) Let E be an elliptic curve over $\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$ and let $l = y - \lambda x - \nu$ be a sloped line function in $\mathbb{F}_q(E)$. As described in Sect. 2.3.3, when substituting $l(x, y) = 0$ into $E(x, y) = 0$, there is an equation $x^3 - \lambda^2 x^2 + (a - 2\lambda\nu)x - \nu^2 + b = 0$, which indicates that there are three intersection points P, Q, and $-(P+Q)$ of l and E. Thus, l has zeros at P, Q, and $-(P+Q)$ with multiplicity 1. Moving the function into $\mathbb{P}^2$, since we have $\frac{X^3 - \lambda^2 X^2 Z + (a - 2\lambda\nu) X Z^2 - (\nu^2 - b) Z^3}{Z^3} = 0$, it is also confirmed that l has a pole at $\mathcal{O} = (0 : 1 : 0)$ with multiplicity 3. Specifically, if $Q = P$, it is confirmed that l has a zero at P with multiplicity 2, zero at $-2P$ with multiplicity 1, and pole at $\mathcal{O}$ with multiplicity 3.

Example 2.58. (Zeros and poles of a vertical line) Let E be an elliptic curve over $\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$ and let $v = x - \mu$ be a vertical line function in $\mathbb{F}_q(E)$. When substituting $v(x, y) = 0$ into $E(x, y) = 0$, there are $y^2 - \mu^3 - a\mu - b = 0$, which indicates there are two intersection points R and $-R$ of v and E. Therefore, v has zeros at R and $-R$ with multiplicity 1. Besides, v also has a pole at $\mathcal{O}$ with multiplicity 2 since $y^2 - \mu^3 - ax - b = 0$ is represented by $\frac{Y^2-\mu^3Z^2-a\mu Z^2-bZ^2}{Z^2} = 0$ in $\mathbb{P}^2$.

2.4.2 Divisors

This subsection describes divisors which are necessary materials to define pairings on elliptic curves E over $\mathbb{F}_q$.

Definition 2.59. (Divisor) A *divisor* D on E is a way to denote a multi-set of all rational points on E and is denoted by

$$D = \sum_i n_i(P_i), \tag{2.65}$$

where P_i is a point on E, $n_i \in \mathbb{Z}$, and $n_i = 0$ for all but finitely many points $P_i \in E$.

Given divisor D, the set of all points P such that $n_i \neq 0$ is said to be a *support* of D and is defined by $\text{supp}(D) = \{P_i \in E : n_i \neq 0\}$. Given two divisors D_1 and D_2, if $\text{supp}(D_1) \cap \text{supp}(D_2)$ is empty set, we say D_1 and D_2 have *disjoint supports*. Let us define the degree of D by $\deg(D) = \sum_i n_i$. The set of all divisors on E is denoted by $\text{Div}(E)$. Then, $\text{Div}(E)$ forms an abelian group under the following addition. For $D_1 = \sum_i n_i(P_i)$ and $D_2 = \sum_i m_i(P_i)$ in $\text{Div}(E)$, an addition is defined by

$$D_1 + D_2 = \sum_i (n_i + m_i)(P_i). \tag{2.66}$$

In the same manner, the subtraction in $\text{Div}(E)$ can be naturally defined by $D_1 - D_2 = \sum_i (n_i - m_i)(P_i)$. The identity of $\text{Div}(E)$ is the divisor of $n_i = 0$ for all $P_i \in E$. An examples of divisors are given in the following.

Example 2.60. Let P, Q, R be points on E. Let D_1 and D_2 are divisors given by $D_1 = 3(P) + (Q) - 2(R)$ and $D_2 = (P) - (Q) + 3(R)$. Then, the degrees of D_1 and D_2 are given by $\deg(D_1) = 2$ and $\deg(D_2) = 3$. Since $D_1, D_2 \in \text{Div}(E)$, the other divisor is obtained by computing $D_1 + D_2 = 3(P) + (Q) - 2(R) + (P) - (Q) + 3(R) = 4(P) + (R) \in \text{Div}(E)$. The supports of D_1 and D_2 are given by $\text{supp}(D_1) = \text{supp}(D_2) = \{P, Q, R\}$, respectively. On the other hand, the support of $D_1 + D_2$ is given by $\text{supp}(D_1 + D_2) = \{P, R\}$. The divisors D_1, D_2, and $D_1 + D_2$ are not disjoint supports.

Associating divisors with a function f in $\mathbb{F}_q(E)$ is a convenient way to write down the intersection points and their multiplicities of f and E defined as follows:

Definition 2.61. (Divisor of function) Let f be a rational function in $\mathbb{F}_q(E)$ such that f has zeros at $P_1, P_2, \ldots \in E$ with multiplicity $n_1, n_2, \ldots \in \mathbb{Z}$, respectively. Note that if $n_i < 0$, then f has a pole at P_i. A *divisor of f* is a way to denote a multi-set of the zeros and is defined as follows:

$$\mathrm{div}(f) = \sum_i n_i(P_i). \tag{2.67}$$

A divisor D is said to be *principal* if there exists a function f such that $D = \mathrm{div}(f)$. Two divisors D_1 and D_2 are said to be *linearly equivalent* if there exists a function f such that $D_1 - D_2 = \mathrm{div}(f)$. The divisor of function has the following properties:

- Given $f, g \in \mathbb{F}_q(E)$, if and only if f is a non-zero constant multiplication by g, then $\mathrm{div}(f) = \mathrm{div}(g)$.
- For all $f \in \mathbb{F}_q(E)$, the degree of $\mathrm{div}(f)$ is zero.
- Given $f \in \mathbb{F}_q(E)$ with a divisor $\mathrm{div}(f) = \sum_i n_i(P_i)$, $\sum_i n_i P_i = \mathcal{O} \in E$, and vice versa.

The properties lead to the fact that the set of all divisors of functions forms a subgroup of $\mathrm{Div}(E)$. Moreover, the multiplication and inversion in $\mathbb{F}_q(E)$ naturally translates across to the addition and subtraction in $\mathrm{Div}(E)$, i.e., for $f, g \in \mathbb{F}_q(E)$, it is obtained $\mathrm{div}(f \cdot g) = \mathrm{div}(f) + \mathrm{div}(g)$ and $\mathrm{div}(f/g) = \mathrm{div}(f) - \mathrm{div}(g)$. Several examples of the divisor of some functions are provided below.

Example 2.62. (Divisor of lines) Let E be an elliptic curve and l and v be rational functions in $\mathbb{F}_q(E)$ as defined in Examples 2.57 and 2.58. Since l has zeros at three points P, Q, and $-(P+Q)$ with multiplicities 1 and pole at $\mathcal{O}$ with multiplicity 3, the divisor of l is given as follows:

$$\mathrm{div}(l) = (P) + (Q) + (-(P+Q)) - 3(\mathcal{O}). \tag{2.68}$$

If $Q = P$, it is computed $\mathrm{div}(l) = (P)+(P)+(-(P+P)) - 3(\mathcal{O}) = 2(P) + (-2P) - 3(\mathcal{O})$. Since v has zeros at two points R and $-R$ with multiplicities 1 and pole at $\mathcal{O}$ with multiplicity 2,

$$\mathrm{div}(v) = (R) + (-R) - 2(\mathcal{O}). \tag{2.69}$$

If v is regarded as l with $P = R$ and $Q = -P$, the divisor of v can also be estimated by $\mathrm{div}(l) = (R)+(-R)+(-(R-R))-3(\mathcal{O}) = (R)+(-R)+(\mathcal{O})-3(\mathcal{O}) = (R)+(-R)-2(\mathcal{O}) = \mathrm{div}(v)$. As seen in the above, the degree of $\mathrm{div}(l)$ and $\mathrm{div}(v)$ are exactly zero. It is also found that $P + Q - (P+Q) - 3\mathcal{O} = \mathcal{O}$ and $R - R - 2(\mathcal{O}) = \mathcal{O}$.

Example 2.63. (Divisor of a function) Let l and v be rational functions in $\mathbb{F}_q(E)$ as defined in Examples 2.57 and 2.58 of which divisors are $\mathrm{div}(l) = (P) + (Q) + (-(P + Q)) - 3(\mathcal{O})$ and $\mathrm{div}(v) = (R) + (-R) - 2(\mathcal{O})$, respectively. Then, the divisor of the function $l \cdot v = (y - \lambda x - \nu) \cdot (x - \mu)$ is computed as follows:

$$\begin{aligned} \mathrm{div}(l \cdot v) &= \mathrm{div}(l) + \mathrm{div}(v) \\ &= (P) + (Q) + (-(P+Q)) - 3(\mathcal{O}) + (R) + (-R) - 2(\mathcal{O}) \\ &= (P) + (Q) + (-(P+Q)) + (R) + (-R) - 5(\mathcal{O}). \end{aligned} \tag{2.70}$$

On the other hand, the divisor of the function $l/v = \frac{y-\lambda x-\nu}{x-\mu}$ is computed as follows:

$$\begin{aligned} \mathrm{div}(f/v) &= \mathrm{div}(l) - \mathrm{div}(v) \\ &= (P) + (Q) + (-(P+Q)) - 3(\mathcal{O}) - (R) + (-R) + 2(\mathcal{O}) \\ &= (P) + (Q) + (-(P+Q)) - (R) - (-R) - (\mathcal{O}). \end{aligned} \tag{2.71}$$

Particularly, if $R = P + Q$, it is found $\mathrm{div}(f) = (P) + (Q) + (-(P + Q)) - (P + Q) - (-(P + Q)) - (\mathcal{O}) = (P) + (Q) - (P + Q) - (\mathcal{O})$. The function l/v plays an important role of pairing computations which are described in the later.

Example 2.64. (Divisor of a function) Let l_1, l_2, and v be line functions in $\mathbb{F}_q(E)$ with divisors $\mathrm{div}(l_1) = (P)+(Q)+(-R)-3(\mathcal{O})$, $\mathrm{div}(l_2) = (P)+(R)+(-(P+R))-3(\mathcal{O})$, and $\mathrm{div}(v) = (R) + (-R) - 2(\mathcal{O})$ where $P, Q \in E$ and $R = P + Q \in E$, respectively. Then, the divisor of the function $l_1 \cdot l_2$ is computed by

$$\begin{aligned} \mathrm{div}(l_1 \cdot l_2) &= \mathrm{div}(l_1) + \mathrm{div}(l_2) \\ &= 2(P) + (Q) + (R) + (-R) + (-(P+R)) - 6(\mathcal{O}). \end{aligned} \tag{2.72}$$

The divisor of the function $l_1 \cdot l_2/v$ is also computed by

$$\begin{aligned} \mathrm{div}((l_1 \cdot l_2)/v) &= \mathrm{div}(l_1) + \mathrm{div}(l_2) - \mathrm{div}(v) \\ &= 2(P) + (Q) + (R) + (-R) + (-(P+R)) - 6(\mathcal{O}) - (R) - (-R) + 2(\mathcal{O}) \\ &= 2(P) + (Q) + (-(P+R)) - 4(\mathcal{O}). \end{aligned} \tag{2.73}$$

Fortunately, one can illustrate the functions l_1, l_2, v, and $(l_1 \cdot l_2)/v$ over $\mathbb{R}$ together with intersections of these functions and E over $\mathbb{R}$ in Figures 2.7 and 2.8, respectively. Figure 2.8 indicates that the function $(l_1 \cdot l_2)/v$ is a parabola. It is also possible to find the correctness of the computed divisor.

The rest of this subsection describes the Weil reciprocity given by André Weil, which is a heat of the pairings. The theorem comes from a requirement of an evaluation of an

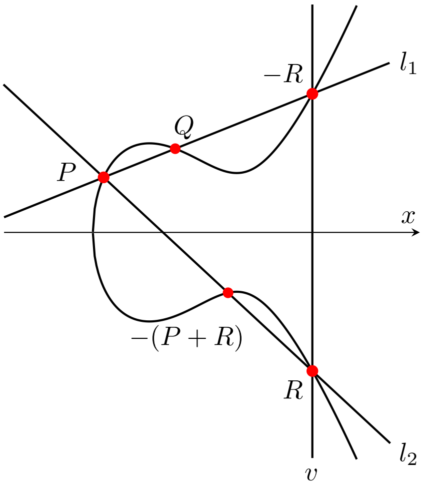


Figure 2.7: The lines l_1, l_2, v, and elliptic curve E over $\mathbb{R}$.

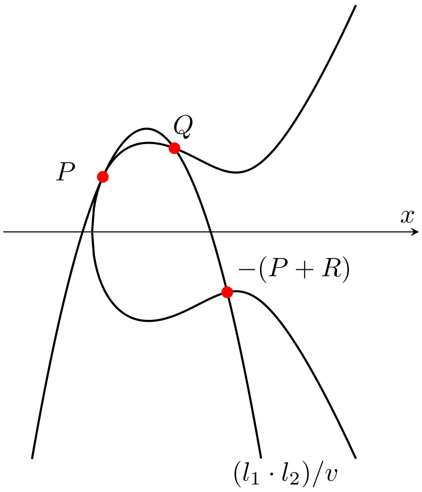


Figure 2.8: The function $(l_1 \cdot l_2)/v$ and E over $\mathbb{R}$.

element $f \in \mathbb{F}_q$ at a divisor $D = \sum_i n_i(P_i)$, where the divisors $\mathrm{div}(f)$ and D have disjoint support. Indeed, there is the following definition of the evaluation.

$$f(D) = \prod_i f(P_i)^{n_i}. \tag{2.74}$$

Then, Weil provided the following theorem.

Theorem 2.65. (Weil reciprocity) Let $f, g \in \mathbb{F}_q(E)$ having disjoint supports. Then,

$$f(\mathrm{div}(g)) = g(\mathrm{div}(f)). \tag{2.75}$$

The following example confirms the correctness of the Weil theorem with the concrete E and functions in $\mathbb{F}_q(E)$.

Example 2.66. Let E be an elliptic curve $E/\mathbb{F}_{103} : y^2 = x^3 + 5$. Let f and g be an element in $\mathbb{F}_q(E)$ given by $f = y - 82x + 19$ and $g = \frac{y-60x+62}{y-91x+69}$, respectively. Then, the divisors of f and g are given by $\mathrm{div}(f) = (28,11) + (102,2) + (2,42) - 3(\mathcal{O})$ and $\mathrm{div}(g) = (36,38) + (95,76) + (70,18) - 3(\mathcal{O}) - (95,27) - (82,80) - (70,18) + 3(\mathcal{O}) = (36,38) + (95,76) - (95,27) - (82,80)$, respectively. Computing $f(\mathrm{div}(g))$ results in

$$f(\mathrm{div}(g)) = \frac{(38 - 82 \cdot 36 + 19) \cdot (76 - 82 \cdot 95 + 19)}{(27 - 82 \cdot 95 + 19) \cdot (80 - 82 \cdot 82 + 19)} = \frac{82}{9} = 32. \tag{2.76}$$

On the other hand, computing $g(\mathrm{div}(f))$ results in

$$g(\mathrm{div}(f)) = \frac{\frac{11-60\cdot28+62}{11-91\cdot28+69} \cdot \frac{2-60\cdot102+62}{2-91\cdot102+69} \cdot \frac{42-60\cdot2+62}{42-91\cdot2+69}}{\left(\frac{1-60\cdot0+62\cdot0}{1-91\cdot0+69\cdot0}\right)^3} = \frac{\frac{41}{4} \cdot \frac{21}{59} \cdot \frac{87}{32}}{1} = \frac{26}{33} = 32. \tag{2.77}$$

Note that $\mathcal{O} = (0 : 1 : 0)$ is substituted in the projective formula of g. As a result, it is confirmed that $f(\mathrm{div}(g)) = g(\mathrm{div}(f))$.

2.4.3 Weil and Tate pairings

This subsection provides basic definitions of the Weil and Tate pairing. Let us assume that E is an elliptic curve over $\mathbb{F}_q$, r is a prime such that $r \neq \mathrm{char}(\mathbb{F}_q)$ and divides $\#E(\mathbb{F}_q)$, and $k > 1$ is the embedding degree with respect to r, which lead to $\mu_r \subset \mathbb{F}_{q^k}$ and $E[r] \subset E(\mathbb{F}_{q^k})$.

Definition 2.67. (The Weil pairing) Let $P, Q \in E[r]$ and let D_P and D_Q be divisors that have disjoint supports and are linearly equivalent to $(P)-(\mathcal{O})$ and $(Q)-(\mathcal{O})$, respectively. Then, there exist functions f_{rD_P} and f_{rD_Q} in $\mathbb{F}_{q^k}(E)$ such that $\mathrm{div}(f_{rD_P}) = rD_P$ and $\mathrm{div}(f_{rD_Q}) = rD_Q$. The *Weil pairing* is a pairing defined as follows:

$$e_{W_r} : E[r] \times E[r] \to \mu_r, \tag{2.78}$$

$$e_{W_r}(P, Q) = \frac{f_{rD_P}(D_Q)}{f_{rD_Q}(D_P)}. \tag{2.79}$$

For constructing the Weil pairing, it is needed to find suitable divisors D_P and D_Q. When taking a point $R \in E(\mathbb{F}_{q^k})$ such that $R \neq P, P-Q, -Q$, the desired divisors are given by $D_P = (P)-(\mathcal{O})$ and $D_Q = (Q+R)-(R)$. Then, the Weil pairing can be defined as follows:

$$e_{W_r}(P, Q) = \frac{f_{rD_P}(Q+R)}{f_{rD_P}(R) \cdot f_{rD_Q}(P)}. \tag{2.80}$$

Note that $f_{rD_Q}(\mathcal{O}) = 1$. Since there are several candidates of D_P and D_Q, note that there exist other constructions of the Weil pairings.

The Weil pairing has the following properties.

- Bilinear: For all $P, Q \in E[r]$, $e_{W_r}(aP, bQ) = e_{W_r}(P, Q)^{ab}$ for all $a, b \in \mathbb{Z}$.
- Alternating: For all $P, Q \in E[r]$, $e_{W_r}(P, Q) = e_{W_r}(Q, P)^{-1}$.
- Non-degenerate: For all $P \in E[r]$, $e_{W_r}(P, Q) = 1$ if and only if $Q = \mathcal{O}$. For all $Q \in E[n]$, $e_{W_r}(P, Q) = 1$ if and only if $P = \mathcal{O}$.
- Endomorphisms: For all $P, Q \in E[r]$, $e_{W_r}(\varphi(P), \varphi(Q)) = e_{W_r}(P, Q)^{\deg(\varphi)}$ for any non-zero endomorphism φ. (The definition of the degree of φ is found in Sect. 2.5.2.)

The proof of the fact that the pairing is well-defined and many facts of the properties are based on the Weil reciprocity.

For the definition of the Tate pairing, we need to define several quotient groups. Let $rE(\mathbb{F}_{q^k})$ be a subgroup of $E(\mathbb{F}_{q^k})$ defined by $rE(\mathbb{F}_{q^k}) = \{[r]P : P \in E(\mathbb{F}_{q^k})\}$ and let $E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ be a quotient group which consists of equivalence classes of points in $E(\mathbb{F}_{q^k})$ with the equivalence relation $P_1 \sim P_2$ if $P_1 - P_2 \in rE(\mathbb{F}_{q^k})$. Let $(\mathbb{F}_{q^k}^*)^r$ be a subgroup of $\mathbb{F}_{q^k}^*$ such that $(\mathbb{F}_{q^k}^*)^r = \{a^r : a \in \mathbb{F}_{q^k}^*\}$ and let $\mathbb{F}_q^*/(\mathbb{F}_{q^k}^*)^r$ be a quotient group which consists of equivalence classes of elements in $\mathbb{F}_{q^k}^*$ with $a_1 \sim a_2$ if $a_1/a_2 \in (\mathbb{F}_{q^k}^*)^r$.

Definition 2.68. (The Tate pairing) Let $P \in E[r]$ and $Q \in E(\mathbb{F}_{q^k})$ be in any equivalence class in $E(\mathbb{F}_q)/rE(\mathbb{F}_{q^k})$. Let D_P and D_Q be divisors that have disjoint supports and are linearly equivalent to $(P)-(\mathcal{O})$ and $(Q)-(\mathcal{O})$, respectively. Then, there exist a functions $f_{rD_P} \in \mathbb{F}_{q^k}(E)$ such that $\mathrm{div}(f_{rD_P}) = rD_P$. The *Tate pairing* is a pairing defined as follows:

$$e_{T_r} : E[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mathbb{F}_q^*/(\mathbb{F}_{q^k}^*)^r, \tag{2.81}$$

$$e_{T_r}(P,Q) = f_{rD_P}(D_Q). \tag{2.82}$$

When taking a point $R \in E(\mathbb{F}_{q^k})$ such that $R \neq P, P-Q, -Q$, the desired divisors are given by $D_P = (P) - (\mathcal{O})$ and $D_Q = (Q+R) - (R)$, the Tate pairing is given as follows:

$$e_{T_r}(P,Q) = \frac{f_{rD_P}(Q+R)}{f_{rD_P}(R)}. \tag{2.83}$$

The Tate pairing has the following properties.

- Bilinear: For all $P \in E[r]$ and $Q \in E(\mathbb{F}_q)/rE(\mathbb{F}_{q^k})$, $e_{T_r}(aP, bQ) = e_{T_r}(P,Q)^{ab}$ for all $a, b \in \mathbb{Z}$.
- Non-degenerate: For all $P \in E[r]$, $e_{T_r}(P,Q) = 1$ if and only if $Q = \mathcal{O}$. For all $Q \in E(\mathbb{F}_q)/rE(\mathbb{F}_{q^k})$, $e_{T_r}(P,Q) = 1$ if and only if $P = \mathcal{O}$.
- Endomorphisms: For all $P \in E[r]$ and $Q \in E(\mathbb{F}_q)/rE(\mathbb{F}_{q^k})$, $e_{T_r}(\varphi(P), \varphi(Q)) = e_{T_r}(P,Q)^{\deg(\varphi)}$ for any non-zero endomorphism φ. (The definition of the degree of φ is found in Sect. 2.5.2.)

Unlike the Weil pairing, the Tate pairing does not have the alternating property but it is not needed in cryptography. However, the Tate pairing has an undesirable property such that the output value lies on an equivalence class in $\mathbb{F}_q^*/(\mathbb{F}_{q^k}^*)^r$. To be suitable in practice, the following modified Tate pairing is typically used.

Definition 2.69. (The reduced Tate pairing) Let P, Q, D_P, D_Q, and f_{rD_P} be as in Definition 2.68. The *reduced Tate pairing* is a pairing defined as follows:

$$\tilde{e}_{T_r} : E[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \to \mu_r, \tag{2.84}$$

$$\tilde{e}_{T_r}(P,Q) = f_{rD_P}(D_Q)^{\frac{q^k-1}{r}}. \tag{2.85}$$

Exponentiating elements in $\mathbb{F}_q^*/(\mathbb{F}_{q^k}^*)^r$ to the power of $(q^k-1)/r$ kills r-th powers and sends the elements to r-th roots of identity in $\mathbb{F}_{q^k}^*$. The additional exponentiation is called the *final exponentiation*. In PBC, the above definition of the reduced Tate pairing is typically adopted for efficiency reasons.

2.4.4 Miller's algorithm

To compute the Weil and Tate pairings, it is required to build functions with the specific divisors, e.g., $f_{rD_P} \in \mathbb{F}_{q^k}(E)$ with the divisor $\mathrm{div}(f_{rD_P}) = r(P) - r(\mathcal{O})$. For constructing the function, let us define a function $f_{m,P} \in \mathbb{F}_q(E)$ with a point $P \in E$ and integer m of which divisor is given as follows:

$$\mathrm{div}(f_{m,P}) = m(P) - (mP) - (m-1)(\mathcal{O}). \tag{2.86}$$

If $m = r$ and $P \in E[r]$, since $\mathrm{div}(f_{r,P}) = r(P) - (rP) - (r-1)(\mathcal{O}) = r(P) - (\mathcal{O}) - (r-1)(\mathcal{O}) = r(P) - r(\mathcal{O}) = \mathrm{div}(f_{rD_P})$, one can define $f_{rD_P} = f_{r,P}$. In [Mil04], Miller gave an algorithm for constructing $f_{m,P}$ with any m, which is called *Miller's algorithm*. The original Miller's algorithm is proposed for computing the Weil pairing and is a double-and-add algorithm governed by a binary representation of m. Currently, an extended algorithm that is managed by a signed binary representation of m is often employed [Beu+10; Ter+13]. The heart of the algorithms is based on the fact that $f_{m,P}$ is possible to build via lines functions as described below.

For $P, Q \in E[r]$, let $l_{P,Q}$ be a line function in $\mathbb{F}_{q^k}(E)$ passing through P and Q and let v_{P+Q} be a vertical line function in $\mathbb{F}_{q^k}(E)$ passing through the point $P+Q$. Let us recall Example 2.62 that shows the line functions have specific divisors given by $\mathrm{div}(l_{P,Q}) = (P) + (Q) + (-(P+Q)) - 3(\mathcal{O})$ and $\mathrm{div}(v_{P,Q}) = (P+Q) + (-(P+Q)) - 2(\mathcal{O})$. Then, one can build a function $l_{P,Q}/v_{P+Q}$ of which divisor is given as follows (see Example 2.63):

$$\mathrm{div}\left(\frac{l_{P,Q}}{v_{P+Q}}\right) = \mathrm{div}(l_{P,Q}) - \mathrm{div}(v_{P+Q}) = (P) + (Q) - (P+Q) - (\mathcal{O}). \tag{2.87}$$

For any integers i, j, considering a function $f_{i,P} \cdot f_{j,P} \cdot l_{iP,jP}/v_{iP+jP}$,

$$\begin{aligned}
&\mathrm{div}\left(f_{i,P} \cdot f_{j,P} \cdot \frac{l_{iP,jP}}{v_{iP+jP}}\right) \\
=&\mathrm{div}(f_{i,P}) + \mathrm{div}(f_{j,P}) + \mathrm{div}\left(\frac{l_{iP,jP}}{v_{iP+jP}}\right) \\
=&i(P) - (iP) - (i-1)(\mathcal{O}) + j(P) - (jP) - (j-1)(\mathcal{O})
\end{aligned}$$

$$\begin{aligned}&+(iP)+(jP)-((i+j)P)-(\mathcal{O})\\=&(i+j)(P)-((i+j)P)-(i+j-1)(\mathcal{O}).\end{aligned} \tag{2.88}$$

This leads to the following definition of the function $f_{i+j,P}$.

$$f_{i+j,P} = f_{i,P} \cdot f_{j,P} \cdot \frac{l_{iP,jP}}{v_{iP+jP}}. \tag{2.89}$$

According to Eq. (2.86), since $\mathrm{div}(f_{1,P}) = 0$, one can set $f_{1,P} = 1$ and $f_{-1,P} = v_P^{-1}$. Then, $f_{2i,P}$, $f_{i+1,P}$, and $f_{i-1,P}$ can be built from the knowledge of $f_{i,P}$ and $R = iP$:

$$f_{2i,P} = f_{i,P}^2 \cdot \frac{l_{R,R}}{v_{R+R}},\ f_{i+1,P} = f_{i,P} \cdot \frac{l_{R,P}}{v_{R+P}},\ f_{i-1,P} = f_{i,P} \cdot \frac{l_{R,-P}}{v_{R-P}} \cdot v_P^{-1}. \tag{2.90}$$

Thus, it is possible to advance from $f_{i,P}$ to either $f_{2i,P}$, $f_{i+1,P}$, or $f_{i-1,P}$, which is corresponding to the doubling, addition, or subtraction operation of i, respectively. This gives rise to a double-and-add/sub algorithm to reach $f_{m,P}$ in $O(\log_2 m)$ steps governed by a signed binary representation of m as shown in Algorithm 2.1. In the algorithm, we look i-th bit m_i of m from the highest bit. After the initializations, we execute the doubling operation with the addition operation if $m_i = 1$; subtraction operation if $m_i = -1$ for each bit.

Note that there are several cases that Miller's algorithm does not correctly work, however, it does not happen in most cases of E used for practical pairings. The details are described in [Ogu+12].

2.4.5 Base-field and trace-zero subgroups

In the Tate and Weil pairings, we always work on an elliptic curve E over $\mathbb{F}_q$ such that there is an r-torsion subgroup $E[r] \subset E(\mathbb{F}_{q^k})$ with a prime $r \neq \mathrm{char}(\mathbb{F}_q)$ and embedding degree k. This subsection introduces two interest subgroups in $E[r]$ for the Tate and Weil pairings.

Definition 2.70. (Base-field subgroup) Let π_q be the Frobenius endomorphism in E. A *base-field subgroup* is a unique subgroup of $E[r]$ defined over $\mathbb{F}_q$, which is defined as follows:

$$\mathcal{G}_1 = \{P \in E[r] : \pi_q(P) = P\}. \tag{2.91}$$

Definition 2.71. (Trace-zero subgroup) Let π_q be as in Definition 2.70. A *trace-zero subgroup* is a unique subgroup of $E[n]$ defined over $\mathbb{F}_{q^k}$, which is defined by

$$\mathcal{G}_2 = \{P \in E[r] : \pi_q(P) = [q]P\}. \tag{2.92}$$

Algorithm 2.1: Miller's algorithm (extended version).

```
Input: $P, Q \in E$, $m = m_l 2^l + m_{l-1} 2^{l-1} + \cdots + m_0 2^0$ where $l+1$ is a bit length
       of $m$ and $m_i \in \{-1, 0, 1\}$ for $0 \leq i \leq l$.
Output: $f_{m,P}(Q)$
If $m_l = 1$ then
  $f \leftarrow 1, R \leftarrow P$;
else if $m_l = -1$ then
  $f \leftarrow v_P^{-1}(Q), R \leftarrow -P$;
endif
For $i$ from $l-1$ downto 0 do
  $f \leftarrow f^2 \cdot \frac{l_{R,R}(Q)}{v_{R+R}(Q)}, R \leftarrow R + R$;          //DBL
  If $m_i = 1$ then
    $f \leftarrow f \cdot \frac{l_{R,P}(Q)}{v_{R+P}(Q)}, R \leftarrow R + P$;          //ADD
  else if $m_i = -1$ then
    $f \leftarrow f \cdot \frac{l_{R,-P}(Q)}{v_{R-P}(Q)} \cdot v_P^{-1}(Q), R \leftarrow R - P$;          //SUB
  endif
endfor
return $f$;
```

The subgroups are more often denoted by $\mathcal{G}_1 = E[r] \cap \ker(\pi_q - [1])$ and $\mathcal{G}_2 = E[r] \cap \ker(\pi_q - [q])$. In fact, $\mathcal{G}_1$ and $\mathcal{G}_2$ are 1- and q-eigenspaces of π_q. It has been described that π_q can be denoted by a 2×2 matrix with the specific trace $\mathrm{tr}(\pi_q) = t = q + 1 - \#E(\mathbb{F}_q)$ and determinant $\det(\pi_q) = q$. This leads to the characteristic polynomial of π_q is given by $\lambda^2 - t\lambda + q$, which can be written by $\lambda^2 - t\lambda + q \equiv \lambda^2 - (q+1)\lambda + q = (x-1)(x-q) \pmod{r}$. Then, the eigenvalues of π_q restrict to $E[r]$ are determined by q and 1. From the definition, it is obvious that $\mathcal{G}_1$ and $\mathcal{G}_2$ are corresponding to 1- and q-eigenspaces of π_q, respectively. The above fact also indicates $\mathcal{G}_1 \times \mathcal{G}_2 = E[r]$.

We need to investigate maps between $\mathcal{G}_1$, $\mathcal{G}_2$, and any subgroup $\mathcal{G}$ of $E[r]$ such that $\mathcal{G} \neq \mathcal{G}_1, \mathcal{G}_2$. There are two endomorphisms that play important roles in $E[r]$.

Definition 2.72. (Trace map) Let π_q be the Frobenius endomorphism in E. A *trace map* is an endomorphism defined as follows:

$$\mathrm{Tr} : E \to E, P + \pi_q(P) + \cdots + \pi_q^{k-1}(P). \tag{2.93}$$

Definition 2.73. (Anti-trace map) Let Tr be as in Definition 2.72. An *anti-trace map* is an endomorphism defined by

$$\mathrm{aTr} : E \to E, P \to [k]P - \mathrm{Tr}(P). \tag{2.94}$$

When restricting the above maps to $E[r] \subset E(\mathbb{F}_{q^k})$, one can see that the trace map acts as $\mathrm{Tr} : \mathcal{G} \to \mathcal{G}_1$, $\mathcal{G}_1 \to \mathcal{G}_1$, and $\mathcal{G}_2 \to \{\mathcal{O}\}$. The fact that the trace map sends all points in $\mathcal{G}_2$ into $\mathcal{O}$ leads to the name of the trace-zero subgroup $\mathcal{G}_2$. In contrast to this, the anti-trace map acts as $\mathrm{aTr} : \mathcal{G} \to \mathcal{G}_2$, $\mathcal{G}_1 \to \{\mathcal{O}\}$, and $\mathcal{G}_2 \to \mathcal{G}_2$.

As for the existence of the practical computable map between $\mathcal{G}_1$ and $\mathcal{G}_2$, it is corresponding to the supersingularity of E. If E is supersingular, there is an isomorphism $\phi : \mathcal{G}_2 \to \mathcal{G}_1$ which is called a *distortion map*. Since ϕ is an isomorphism, there is an inverse map $\phi^{-1} : \mathcal{G}_1 \to \mathcal{G}_2$. However, if E is ordinary, there is no known efficient isomorphism out of $\mathcal{G}_1$ or $\mathcal{G}_2$.

2.4.6 Restricting the pairings to the subgroups

For practical applications, it is more convenient to restrict the pairings to the subgroups, rather than full r-torsion subgroup $E[r]$. In the following, let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the base-field and trace-zero subgroups of $E[r]$ described in the previous section. Let $\mathcal{G}$ be a subgroup of $E[r]$ such that $\mathcal{G} \neq \mathcal{G}_1, \mathcal{G}_2$.

According to Proposition 3.4 in [EMJ17], the Weil and reduced Tate pairings can be generally restricted to $\mathcal{G}_1 \times \mathcal{G}$, $\mathcal{G} \times \mathcal{G}_1$, $\mathcal{G}_2 \times \mathcal{G}$, and $\mathcal{G} \times \mathcal{G}_2$ are non-degenerate. Besides, it is trivial that the Weil pairing restricted to $\mathcal{G}_1 \times \mathcal{G}_2$ or $\mathcal{G}_2 \times \mathcal{G}_1$ is non-degenerate. Although the Tate pairing is not as simple as the Weil pairing, if there are no points r^2-torsion subgroup in $E(\mathbb{F}_{q^k})$, which means that $k > 1$, the Tate pairing restricted to $\mathcal{G}_1 \times \mathcal{G}_2$ or $\mathcal{G}_2 \times \mathcal{G}_1$ is non-degenerate. Particularly, the reduced Tate pairing restricted to $\mathcal{G}_1 \times \mathcal{G}_2$ leads to remove the conditions of the divisors having disjoint supports and allows us the following definition:

$$\tilde{e}_{T_r} : \mathcal{G}_1 \times \mathcal{G}_2 \to \mu_r, \tag{2.95}$$

$$\tilde{e}_{T_r}(P, Q) = f_{r,P}(Q)^{\frac{q^k-1}{r}}, \tag{2.96}$$

where $f_{r,P}$ is a function in $\mathbb{F}_q(E)$ with the divisor $\mathrm{div}(f_{r,P}) = r(P) - r(\mathcal{O})$. Since the original definition of the Tate pairings involves evaluation of $f_{r,P}$ at two points in E, it is expected that the restriction results in reducing the computational complexity of the pairing. In order to reduce the more computational complexity, Hess et al. provided a variant of the above reduced Tate pairings in [HSV06]. They observed that $t - 1 \equiv q \pmod{r}$ where t is the Frobenius trace results in $\pi_q(Q) = [q]Q = [t-1]Q$ for $Q \in \mathbb{G}_2$ and found that it leads to the following definition of the pairing, which is called the ate pairing.

Definition 2.74. (Ate pairing) Let t be the Frobenius trace and let $T = t - 1$. The *ate* *pairing* is a pairing defined as follows:

$$e_{a_T} : \mathcal{G}_2 \times \mathcal{G}_1 \to \mu_r, \tag{2.97}$$

$$e_{a_T}(Q, P) = f_{T,Q}(P)^{\frac{q^k-1}{r}}, \tag{2.98}$$

where $f_{T,Q}$ is a function in $\mathbb{F}_{q^k}(E)$ with the divisor $\mathrm{div}(f_{T,Q}) = T(Q) - (TQ) - (T-1)(\mathcal{O})$.

The important fact is that the ate pairing requires $\log_2 T$ steps of Miller's algorithm for computing $f_{T,Q}(P)$, instead of $\log_2 r$ for $f_{r,P}(Q)$. From Hasse's theorem, $\log_2 T < \log_2 r$ is typically satisfied for the practical pairings. The ate pairing corresponding to $T \equiv q \pmod r$ is one of the special cases of ate-like pairings which are obtained by taking any power $T^i \equiv q^i \pmod r$. More generally, in [Ver09], Vercauteren proposed an ate-like pairing constructed by any linear combination of $\sum_i c_i q^i \equiv 0 \pmod r$ as follows:

Definition 2.75. (Ate-like pairing) Let $k' = \phi(k)$ with Euler's totient function ϕ and let $\lambda = \sum_{i=0}^{k'-1} c_i q^i$ with $c_i \in \mathbb{Z}$ such that $\lambda = mr$ and $mkq^{k-1} \neq \frac{q^k-1}{r} \sum_{i=0}^{k'-1} i c_i q^{i-1} \pmod r$ with some integer m. Then, an *ate-like pairing* is defined as follows:

$$e_{a_{c_i}} : \mathcal{G}_2 \times \mathcal{G}_1 \to \mu_r, \tag{2.99}$$

$$e_{a_{c_i}}(Q, P) = \left(\prod_{i=0}^{k'-1} f_{c_i,Q}(P)^{q^i} \cdot \prod_{i=0}^{k'-2} \frac{l_{s_{i+1}Q, c_i q^i Q}(P)}{v_{s_i Q}(P)} \right)^{\frac{q^k-1}{r}}, \tag{2.100}$$

where $s_i = \sum_{j=i}^{k'-1} c_j q^j$ and $l_{s_{i+1}Q, c_i q^i Q}$ and $v_{s_i Q}(P)$ are line function in $\mathbb{F}_{q^k}(E)$ with the divisors $\mathrm{div}(l_{s_{i+1}Q, c_i q^i Q}) = (s_{i+1}Q) + (c_i q^i Q) + (-(s_{i+1} + c_i q^i)Q) - 3(\mathcal{O})$ and $\mathrm{div}(v_{s_i Q}(P)) = (s_i Q) + (-s_i Q) - 2(\mathcal{O})$, respectively.

Then, one can find $\lambda = \sum_{i=0}^{k'-1} c_i q^i$ which generates the ate-like pairings with one of the smallest numbers of the steps of Miller's algorithm. Indeed, the number of steps can be fixed at least $\log_2 r/\phi(k)$. Such ate-like pairing is especially called the *optimal-ate pairing*.

2.4.7 Use of twists

Let E be an elliptic curve over $\mathbb{F}_q$ such that $\mathcal{G}_1 \times \mathcal{G}_2 = E[r] \subset E(\mathbb{F}_{q^k})$. Since $\mathcal{G}_1$ is defined over $\mathbb{F}_q$, it admits an efficient representation. This subsection describes $\mathcal{G}_2$ which is defined over $\mathbb{F}_{q^k}$ also admits an efficient representation on a twist E' of E.

Let d be a factor of k such that $d = 1$, 2, 3, 4, or 6. One can find a twist E' of degree d of E defined over $\mathbb{F}_{q^{k/d}}$ with an isomorphism $\phi_d : E' \to E$ over $\mathbb{F}_{q^k}$. The more important fact is that one can also find a unique twist E' such that $r \mid \#E(\mathbb{F}_{q^{k/d}})$, which is called the *correct twist* in this thesis. Then, since E' and E are isomorphic over $\mathbb{F}_{q^k}$, there is an r-torsion subgroup $E'[r] \subset E'(\mathbb{F}_{q^k})$. Furthermore, there are subgroups $\mathcal{G}'_1$ and $\mathcal{G}'_2$ of $E'[r]$ which are preimages of $\mathcal{G}_1$ and $\mathcal{G}_2$, i.e., $\mathcal{G}'_1 = \phi_d^{-1}(\mathcal{G}_1)$ and $\mathcal{G}'_2 = \phi_d^{-1}(\mathcal{G}_2)$, respectively. In fact, $\mathcal{G}_2'$ is a unique subgroup of $E'[r]$ such that $\mathcal{G}_2' = E'[r] \cap E(\mathbb{F}_{q^{k/d}})$ which is defined over a subfield $\mathbb{F}_{q^{k/d}}$ of $\mathbb{F}_{q^k}$, however, $\mathcal{G}_1$ is a unique subgroup of $E'[r]$ defined over $\mathbb{F}_{q^k}$. Since $\mathcal{G}_1 \to \mathcal{G}_1'$ and $\mathcal{G}_2 \to \mathcal{G}_2'$ are group isomorphisms, $\mathcal{G}_1'$ and $\mathcal{G}_2'$ have the similar properties of $\mathcal{G}_1$ and $\mathcal{G}_2$ described in Sect. 2.4.5.

The above gives rise to the ate pairing moved entirely on E' given as follows:

$$e_{a_T}' : \mathcal{G}_2' \times \mathcal{G}_1' \to \mu_r, \tag{2.101}$$

$$e_{a_T}'(Q', P') = f_{T,Q'}'(P')^{\frac{p^k-1}{r}}, \tag{2.102}$$

where $f_{T,Q'}'$ is a function in $\mathbb{F}_{q^k}(E')$ with the divisor $\mathrm{div}(f_{T,Q'}') = T(Q') - (TQ') - (T-1)(\mathcal{O}')$ where $\mathcal{O}'$ is the point on E' at infinity. Although the ate pairings on E and E' are typically not distinguished, note that $e_{a_T}(Q,P)$ and $e_{a_T}'(Q',P')$ do not always take the same value even though there are relations $P' = \phi_d^{-1}(P)$ and $Q' = \phi_d^{-1}(Q)$. Indeed, Costello et al. provided the following theorem [CLN10].

Theorem 2.76. Let E be an elliptic curve over $\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$, let E' be a correct twist of degree d of E, and let $\phi_d : E' \to E$. For $P \in \mathcal{G}_1$, $Q \in \mathcal{G}_2$, $P' = \phi_d^{-1}(P) \in \mathcal{G}_1'$, and $Q' = \phi_d^{-1}(Q) \in \mathcal{G}_2'$,

$$e_{a_T}(Q,P)^{\gcd(d,6)} = e_{a_T}'(Q',P')^{\gcd(d,6)}. \tag{2.103}$$

Since the fields in which the groups $\mathcal{G}_1$ and $\mathcal{G}_2'$ are defined are smaller than these of $\mathcal{G}_1'$ and $\mathcal{G}_2'$, respectively, the ate pairings are often regarded as $\mathcal{G}_2' \times \mathcal{G}_1 \to \mu_r$ which is defined by either of the following.

$$e_{a_T}(\phi_d(Q'), P) = f_{T,\phi_d(Q')}(P)^{\frac{p^k-1}{r}}, \tag{2.104}$$

$$e_{a_T}'(Q', \phi_d^{-1}(P)) = f_{T,Q'}'(\phi_d^{-1}(P))^{\frac{p^k-1}{r}}. \tag{2.105}$$

To make the movement of the curves easily and enable efficient arithmetics, it is often used a tower of extension fields constructed by quotient rings by binomial ideals as follows [BS10]:

$$\begin{cases} \mathbb{F}_{p(z)^d} & \cong \mathbb{F}_{p(z)}[x]/(x^d - c) & \cong \mathbb{F}_{p(z)}(\alpha), \\ \mathbb{F}_{p(z)^k} & \cong \mathbb{F}_{p(z)^d}[x]/(x^{k/d} - \alpha) & \cong \mathbb{F}_{p(z)^d}(\beta), \end{cases} \tag{2.106}$$

where α and β are elements in $\mathbb{F}_{p(z)^d}$ and $\mathbb{F}_{p(z)^k}$ such that $\alpha^d = c$ and $\beta^{k/d} = \alpha$, respectively.

Note that there are many optimizations related to the twist corresponding to its degree, e.g., the twist enables the smooth application of the denominator elimination techniques [Bar+02; Lin+08; CLN10; ZL12]. Particularly, if the curve admits the quadratic twist, all values of vertical line functions that appeared in Miller's algorithm can be ignored since the final exponentiation by $(q^k - 1)/r$ brings these values for the identity in $\mathbb{F}^*_{p^k}$.

2.4.8 Types of pairings

The differences in the restrictions give rise to three types of pairings, which are introduced by Galbraith et al. in [GPS08]. Actually, there are four types in the literature; Galbraith et al. originally presented three, but a fourth type was added soon after by Shacham [Sha06], however, the fourth type is not discussed. The details of the types are summarized below.

- Type 1: The pairings $G_1 \times G_2 \to G_T$ with $G_1 = G_2$, i.e., there are $G_1 \to G_2$ and $G_2 \to G_1$; Suppose that E is a supersingular curve such that there is a distortion map $\phi : \mathcal{G}_2 \to \mathcal{G}_1$. Then, the Weil and Tate pairings given by $e_{W_r}(P, \phi(P))$ and $e_{T_r}(P, \phi(P))$ give rise to the pairings restricted to $\mathcal{G}_1 \times \mathcal{G}_1$, which are classified into this type.

- Type 2: The pairings $G_1 \times G_2 \to G_T$ with $G_1 \neq G_2$ but an efficiently computable isomorphism $G_2 \to G_1$ is known, while none is known in the other direction; If E is an ordinary elliptic curve, the Weil and Tate pairings restricted to $\mathcal{G}_1 \times \mathcal{G}$ (or $\mathcal{G}_2 \times \mathcal{G}$) in this type since there is $\mathrm{Tr} : \mathcal{G} \to \mathcal{G}_1$ (or $\mathrm{aTr} : \mathcal{G} \to \mathcal{G}_2$) but there is no known efficient map $\mathcal{G}_1 \to \mathcal{G}$ (or $\mathcal{G}_2 \to \mathcal{G}$).

- Type 3: The pairings $G_1 \times G_2 \to G_T$ with $G_1 \neq G_2$ and no efficiently computable isomorphism is known between G_1 and G_2, in either direction; If E is an ordinary elliptic curve, the Weil and Tate pairings restricted to $\mathcal{G}_1 \times \mathcal{G}_2$ or $\mathcal{G}_2 \times \mathcal{G}_1$ classified into this type since there are known efficient map out of $\mathcal{G}_1$ and $\mathcal{G}_2$ in $E[r]$. The ate pairings restricted to $\mathcal{G}_2 \times \mathcal{G}_1$ or $\mathcal{G}'_2 \times \mathcal{G}'_1$, which is often regarded $\mathcal{G}'_2 \times \mathcal{G}_1$, also classified into this type.

It is known that the properties of the different types of pairings provide subtle differences to protocols and their proofs. The type 1 pairings were used in the early age of pairing-based protocols, they have gradually been discarded in favor of type 3 pairings. In fact, the state-of-the-art implementations of pairings take place on the ordinary curves that assume the type 3 pairings. Moreover, Chatterjee and Menezes [CM11] argued that there are no known protocols and proofs of security that cannot be translated into the type 3 setting. Thus, it is currently recommended to design the protocols with the type 3 pairings.

2.4.9 Computational problems

This subsection provides computational difficult problems related to the pairings. Although the problems related to the pairings are still often discussed with type 1 settings, let us focus on type 2 or 3 settings. Many pairing-based protocols are based on the difficulty of one or both of the following problems.

Definition 2.77. (Bilinear Diffie-Hellman problem (BDHP)) Given $P, xP, yP \in G_1$ and $Q, xQ, zQ \in G_2$ with $x, y, z \in \mathbb{Z}$, compute $e(P, Q)^{xyz} \in G_T$.

Definition 2.78. (Bilinear decisional Diffie-Hellman problem (BDDHP)) Given $P, xP, yP \in G_1$, $Q, xQ, zQ \in G_2$ with $x, y, z \in \mathbb{Z}$, and $g \in G_T$, determine whether or not $g = e(P, Q)^{xyz} \in G_T$.

If the BDHP is solved, the BDDHP can be broken. Moreover, the BDHP is no harder than either the ECDHP in G_1 and G_2 or DHP in G_T; If the ECDHP in G_1 (or G_2) is solved, one can solve the BDHP by computing xyP (or xzQ) and thus $e(xyP, zQ)$ (or $e(yP, xzQ)$) is obtained; If the DHP in G_T is solved, one can also solve the BDHP by computing $g = e(P, Q)$, $g^{xy} = e(yP, xQ)$, and $g^z = e(P, zQ)$ and thus g^{xyz} is obtained. Since the ECDHP and DHP are solved if the ECDLP and DLP are solved, respectively, the security of the pairings depends on the difficulty of solving both the ECDLP in G_1 and G_2, and the DLP in G_T.

Besides, there is a basic calculation problem peculiar to the pairing operations.

Definition 2.79. (Pairing inverse problem) Let $e : G_1 \times G_2 \to G_T$ be a pairing. There are the following problems related to the pairing inversion problems.

1. (The fixed argument pairing inversion 1 problem (FAPI-1P)) Given $Q \in G_2$ and $g \in G_T$, compute $P \in G_1$ such that $e(P, Q) = g$.

2. (The fixed argument pairing inversion 2 problem (FAPI-2P)) Given $P \in G_1$ and $g \in G_T$, compute $Q \in G_2$ such that $e(P, Q) = g$.

3. (The generalized pairing inversion problem (GPIP)) Given $g \in G_T$, compute $P \in G_1$ and $Q \in G_2$ such that $e(P, Q) = g$.

If the FAPI-1 and FAPI-2 problems are solved, one can solve all the ECDHP on G_1 and G_2 and DLP on G_T. Conversely, assuming the difficulties of the ECDHP and DHP problems, the difficulty of the FAPI-1 and FAPI-2 problems are guaranteed. At this time, there are no known special curves that give rise to efficient computation of the GPIP which is typically easier than FAPI-1 and FAPI-2 problems.

2.4.10 Pairing-friendly elliptic curves

For secure and efficient pairings, it is needed to carefully choose an elliptic curve in which the pairing is defined. To guarantee the security of the pairings, the DLPs should be infeasible in both $G_1, G_2 \subset E[r]$ and $G_T \subset \mathbb{F}_{q^k}$ having the common order r. Thus, it is necessary to set the appropriate sizes of the r and q^k that can achieve the certain security level; for the 128-bit security level, it is suggested to fix $\log_2 r \geq 256$ and $\log_2 q^k \geq 5000$ around [Gui20]. As discussed in Sects. 2.2.5 and 2.3.9, G_1 and G_2 currently obtain much greater security per bit than G_T since the best-known attacks for the ECDLP in G_1 and G_2 have exponential complexity, however, that of the DLP in G_T have sub-exponential complexity. Moreover, since the attacks for the DLP in G_T are improved in recent years, we have to pay much more attention to the settings of G_T than that of G_1 and G_2.

In addition to this, to guarantee the efficiency of the pairings, it is important that the order r is a large factor of $\#E(\mathbb{F}_q) = q + 1 - t$. To discuss this idea conveniently, let us define a quantity that indicates the ratio of the sizes of q and r.

$$\rho = \frac{\log_2 q}{\log_2 r}, \tag{2.107}$$

which is called the *ρ-value*. Since it is preferred that $\#E(\mathbb{F}_q) = q+1-t$ involves the large factor r, the ideal case is considered to be $\rho \approx 1$. Note that the size of the embedding degree k with respect to r is entirely determined by ρ and the choice of the bit sizes of r and q^k, since $\log_2 q^k / \log_2 r = \rho k$.

Based on the above facts, Freeman et al. gave the following definition of the elliptic curves suitable for the pairings [FST10]:

Definition 2.80. (Pairing-friendly elliptic curve) An elliptic curve E is *pairing-friendly* if the following conditions are satisfied:

1. The ρ-value satisfies $1 \leq \rho \leq 2$.

2. The embedding degree k with respect to r satisfies $k \leq \log_2 r/8$.

Notice that the pairing-friendly elliptic curves are very special since randomly found E typically have $k \approx q$.

One can heuristically find the pairing-friendly curves with $\rho \approx 2$ by the Cocks-Pinch method of which algorithm is presented in [FST10]. The pairing-friendly curves with $\rho < 2$ can also be found by the Brezing-Weng methods [BW05]. Rather than this method, it is more often used construction methods of the pairing-friendly curves based on the parameterization of q, r, and t by the polynomials making the curves with the favorite properties which are defined as follows [FST10]:

Definition 2.81. (Family of pairing-friendly elliptic curves) Let $q(x)$, $r(x)$, and $t(x)$ be non-zero polynomials in $\mathbb{Q}[x]$, k be a positive integer, and D be a square-free integers. The triple $(q(x), r(x), t(x))$ is referred to as *family of pairing-friendly elliptic curve* with embedding degree k and CM discriminant D if the following conditions are satisfied:

1. $q(x) = p(x)^i$ for some $i \geq 1$ and $p(x)$ that represent primes. (It is typically chosen as $i = 1$.)

2. $r(x)$ is non-constant, irreducible, integer-valued, i.e., $r(x_0) \in \mathbb{Z}$ for all $x_0 \in \mathbb{Z}$, and has positive-leading coefficients.

3. $r(x)$ divides $\Phi_k(t(x) - 1)$ where Φ_k is the k-th cyclotomic polynomial.

4. $r(x)$ divides $q(x) + 1 - t(x)$

5. The equation $Dy^2 = 4q(x) - t(x)^2$ in (x, y) has infinitely many integer solutions.

The family is *ordinary* if $\gcd(t(x), q(x)) = 1$ and is also *complete* if y is denoted by $y(x) \in \mathbb{Q}[x]$. In the case of family, the ρ-value of the family is defined by

$$\rho = \frac{\deg q(x)}{\deg r(x)}, \tag{2.108}$$

instead of the typical definition in Eq. (2.107). There are families of pairing-friendly curves, such that Miyaji-Nakabayashi-Takano (MNT) family [MNT01], Barreto-Lynn-Scott (BLS) family [BLS02], Barreto-Naehrig (BN) family [BN05], Kachisa-Schaefer-Scott (KSS) family [KSS08], and many others [FST10].

Note that it is required to find an integer seed z making $p = p(z)$ and $r = r(z)$ being primes for specifying the curves. Then, there is an elliptic curve $E/\mathbb{F}_{q(z)}$ such that

- The group order is given by $n(z) = \#E(\mathbb{F}_{q(z)}) = q(z) + 1 - t(z)$, which is divisible by $r(z)$.

- The embedding degree with respect to $r(z)$ is k, i.e., k is the minimal integer satisfying $r(z) \mid (q(z)^k - 1)$.

Then, there exist the correct twist $E'/\mathbb{F}_{q(z)^{k/d}}$ of degree d of E such that $r(z) \mid n'(z) = \#E'(\mathbb{F}_{q(z)^{k/d}})$ and twisting isomorphism $\phi_d : E' \to E$ defined over $\mathbb{F}_{q(z)^k}$. It is typically exploited the pairings defined over such E or E' for practical protocol implementation. The following example presents the curve $E/\mathbb{F}_{q(z)}$ which is generated by the BN family.

Example 2.82. (BN curve) Let $p(x)$, $r(x)$, $t(x)$ be the polynomials in $\mathbb{Q}[x]$ defined as follows:

$$\begin{cases} p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1, \\ r(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1, \\ t(x) &= 6x^2 + 1. \end{cases} \tag{2.109}$$

Then, $(p(x), r(x), t(x))$ parametrizes a family of pairing-friendly elliptic curves with $k = 12$, $D = 3$, and $\rho = 1$. A curve in the BN family is called the BN curve. As a toy example, the seed $x_0 = 1$ generates $p(x_0) = 103$, $r(x_0) = 97$, $t(x_0) = 7$, which leads to the BN curve given by $E/\mathbb{F}_{p(x_0)} : y^2 = x^3 + 5$ such that $r(x_0) = 97$ divides $\#E(\mathbb{F}_{p(x_0)}) = p(x_0) + 1 - t(x_0) = 103 + 1 - 7 = 97$ and $\#\mathbb{F}^*_{p(x)^k} = p(x_0)^k - 1 = 97^{12} - 1 = 1425760886846178945447840 = 2^5 \cdot 3^2 \cdot 5 \cdot 7 \cdot 13 \cdot 17 \cdot 19 \cdot 37 \cdot 79 \cdot 97 \cdot 1061 \cdot 3571 \cdot 31357$. Actually, we already have seen this curve in Figure 2.3.

2.5 Isogenies between elliptic curves

This section provides the mathematical preliminaries of the isogenies which leads to the isogeny problems which are hard to compute even though the quantum computers are applied. The definition of the isogenies is referred to [Sil09].

2.5.1 Isgenies

This subsection describes maps between elliptic curves, which are called *isogenies*.

Definition 2.83. (Isogeny) Let E and $\tilde{E}$ be elliptic curves. Let us define that an *isogeny* from E to $\tilde{E}$ is a morphism given as follows:

$$\varphi : E \to \tilde{E} \text{ satisfying } \phi(\mathcal{O}) = \tilde{\mathcal{O}}. \tag{2.110}$$

Note that this thesis considers a zero morphism such that $\varphi(E) = \{\tilde{\mathcal{O}}\}$ as an isogeny. This zero morphism is called the *zero isogeny*; otherwise, *non-zero isogeny*. If there exists a non-zero isogeny from E to $\tilde{E}$, we say E and $\tilde{E}$ are *isogenous*. If there exist non-zero isogenies $\varphi : E \to \tilde{E}$ and $\tilde{\varphi} : \tilde{E} \to E$ such that $\tilde{\varphi}(\varphi(P)) = P$ for any $P \in E$, then E and $\tilde{E}$ are isomorphic, i.e., $j(E) = j(\tilde{E})$. The isogenies φ and $\tilde{\varphi}$ are endomorphisms if $E = \tilde{E}$. Furthermore, if φ is an isomorphism and $E = \tilde{E}$, φ is called an *automorphism*.

The set of all isogenies from E and $\tilde{E}$ are denoted as follows:

$$\mathrm{Hom}(E, \tilde{E}) = \{\text{isogenies } E \to \tilde{E}\}. \tag{2.111}$$

Then, $\mathrm{Hom}(E, \tilde{E})$ form an abelian group under an additive law, i.e., for $\varphi, \psi \in \mathrm{Hom}(E, \tilde{E})$, $\varphi + \psi \in \mathrm{Hom}(E, \tilde{E})$ is defined by $(\varphi + \psi)P = \varphi(P) + \psi(P)$, where the right side of $+$ is the point addition. Note that the zero isogeny $[0] : E \to \tilde{E}, P \mapsto 0P$ plays a role of the identity. Besides, if $E = \tilde{E}$, $\mathrm{End}(E) = \mathrm{Hom}(E, \tilde{E})$ also forms a monoid under a multiplicative law $\cdot$ defined by $(\varphi \cdot \psi)(P) = \psi(\varphi(P))$ with the identity $[1] : E \to E, P \mapsto 1P$. This means that $\mathrm{End}(E)$ form a ring under the above addition and multiplication. The ring is called the *endomorphism ring of E*. A set of invertible elements of $\mathrm{End}(E)$ forms the automorphism group of E, which is denoted by $\mathrm{Aut}(E)$.

2.5.2 A standard form for isogenies

For easily understanding, this subsection considers the isogenies from E and $\tilde{E}$ in the affine space. Note that we actually have to work on the projective space for the point $\mathcal{O}$ that cannot express in the affine space. If E and $\tilde{E}$ are defined over $\mathbb{F}_q$, a non-zero isogeny can be defined by an affine rational map

$$\varphi(x,y) = (r_1(x,y), r_2(x,y)), \tag{2.112}$$

where $r_1(x,y), r_2(x,y)$ are rational functions defined over $\mathbb{F}_q$. If E and $\tilde{E}$ are elliptic curves over $\mathbb{F}_q$ given by short Weierstrass equations, a non-zero isogeny from E to $\tilde{E}$ is defined by an affine map

$$\varphi(x,y) = \left(\frac{u(x)}{v(x)}, \frac{s(x)}{t(x)}y\right), \tag{2.113}$$

where $u(x), v(x), s(x), t(x) \in \mathbb{F}_q[x]$ such that $u(x)$ and $v(x)$ have no common factor over $\mathbb{F}_q$, and $s(x)$ and $t(x)$ are too. The form of the isogeny given by Eq. (2.113) is a *standard form.* Let us define two important invariants of the non-zero isogeny that can be easily determined when it is in this form. For the isogeny from E_1 to E_2 given by the standard form, let us define the *degree of* φ as deg $\varphi = \max(\deg u, \deg v)$. An isogeny of degree l is said to be l-isogeny. If there exists an l-isogeny from E to $\tilde{E}$, we say E and $\tilde{E}$ are l-isogenous. Besides, φ is called *separable* if the derivative of $\frac{u(x)}{v(x)}$ is non-zero; otherwise, it is called *inseparable.* If the isogeny φ is separable, then $\deg\varphi = \#\ker\varphi$.

There are examples of isogenies, which we already have seen in Sects. 2.3.4 and 2.3.5, respectively.

Example 2.84. (Point multiplication endomorphism) Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\overline{\mathbb{F}}_q$. Then, the point multiplication endomorphism $[m] : E \to E, (x,y) \mapsto m(x,y)$ is an isogeny. If $m \neq 0$ and $m \neq \mathrm{char}(\mathbb{F}_q)$, the isogeny $[m]$ is separable and $\deg[m] = m^2$ (see Corollary 6.4. in [Sil09]), which is related to the fact that $\#E[m] = m^2$. For some small m, the standard form, separability, and degree of isogenies are obtained as follows:

- $m = -1$: The image of (x,y) under $[-1]$ is given by $[-1](x,y) = (x,-y)$ in the standard form. It is separable and has degree $\deg[-1] = 1$.
- $m = 2$: The image of (x,y) under $[2]$ is given by $[2](x,y) = (r_1(x,y), r_2(x,y)) = (\lambda^2 - 2x, \lambda(x - r_1(x,y)) - y)$ where $\lambda = \frac{3x^2+a}{2y}$ (see Eq. (2.36)). Then, $r_1(x,y)$ and $r_2(x,y)$ can be modified as follows:

$$r_1(x,y) = \lambda^2 - 2x = \frac{(3x^2+a)^2}{4y^2} - 2x = \frac{9x^4+6ax^2+a^2}{4(x^3+ax+b)} - 2x = \frac{x^4-2ax^2-8bx+a^2}{4(x^3+ax+b)},$$

$$\begin{aligned} r_2(x,y) = \lambda(x - r_1(x,y)) - y &= \frac{(3x^2+a)y}{2y^2}\left(\frac{3x^4+6ax^2+12bx-a^2}{4(x^3+ax+b)}\right) - y \\ &= \frac{x^6+5ax^4+20bx^3-5a^2x^2-4abx-8b^2-a^3}{8(x^3+ax+b)}y. \end{aligned}$$

Thus, the standard form of [2] is given by

$$[2](x,y) = \left(\frac{x^4-2ax^2-8bx+a^2}{4(x^3+ax+b)}, \frac{x^6+5ax^4+20bx^3-5a^2x^2-4abx-8b^2-a^3}{8(x^3+ax+b)}y\right).$$

It is separable since $\frac{d}{dx}(r_1(x,y)) \neq 0$ in $\mathbb{F}_q$ and has $\deg[2] = 4$.

Example 2.85. (Frobenius endomorphism) Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_q$. The Frobenius endomorphism $\pi_q : E \to E, (x,y) \mapsto (x^q, y^q)$ is an isogeny. Since $y^q = y^{q-1}y = (y^2)^{\frac{q-1}{2}}y = (x^3+ax+b)^{\frac{q-1}{2}}y$, the standard form is given by $\pi(x,y) = (x^q, (x^3+ax+b)^{\frac{q-1}{2}}y)$. According to the form, it is found $\deg \pi_q = q$. Besides, π_q is inseparable since $\frac{d}{dx}(x^q) = qx^{q-1} = 0$ in $\mathbb{F}_q$.

Example 2.86. (Twisting isomorphism) Let $E : y^2 = x^3 + b$ be an elliptic curve over $\mathbb{F}_q$. Let E' be a twist of degree 2 of E given by $y^2 = x^3 + a/\delta^2 + b/\delta^3$ where δ is quadratic non-residue in $\mathbb{F}_q^*$. Then, a twisting isomorphism $\phi_2 : E' \to E, (x,y) \mapsto (\delta x, \delta^{1/2}y)$ over $\mathbb{F}_{q^2}$ is an isogeny which already has a standard form. According to the form, it is found $\deg \phi_2 = 1$. Besides, π_q is inseparable since $\frac{d}{dx}(\delta x) = \delta$ in $\mathbb{F}_{q^2}$.

2.5.3 Vélu's formula

For a given curve E and subgroup G, there is a unique separable isogeny $\phi : E \to \tilde{E}$ such that $\tilde{E}$ is isomorphic to a quotient group E/G. Then, ϕ is an $\#G$-isogeny since $\ker(\phi) = G$. In [Vél71], Vélu describes how to explicitly write down equations for the curve $\tilde{E}$ such that $\tilde{E} \cong E/G$ and isogeny $\phi : E \to \tilde{E}$. In the context, an explicit formula for 2-isogenies is given as follows:

Theorem 2.87. (2-isogeny) Let E be an elliptic curve over $\mathbb{F}_q$ given by $y^2 = x^3 + ax + b$. Let G be a group of E of order 2 defined by $G = \langle (x_0, 0) \rangle$ where $x_0 \in \overline{\mathbb{F}}_q$ be a root of $x^3 + ax + b$. Assuming $t = 3x_0^2 + a$ and $w = x_0 t$, the following rational map

$$\varphi(x,y) = \left(\frac{x^2 - x_0x + t}{x - x_0}, \frac{(x-x_0)^2 - t}{(x-x_0)^2}y\right), \tag{2.114}$$

is a separable isogeny from E to $\tilde{E} : y^2 = x^3 + \tilde{a}x + \tilde{b}$ where $\tilde{a} = a - 5t$ and $\tilde{b} = b - 7w$ such that $\tilde{E} = E/G$.

Example 2.88. (2-isogeny) Let $E : y^2 = x^3 - x$ be an elliptic curve over $\mathbb{F}_q$ where $a = -1$ and $b = 0$. Then, roots of $x^3 - x$ are 0, 1, and -1. When taking $x_0 = 1$, it is obtained

$t = 3i^2 + a = 2$, $w = x_0 t = 2$, and 2-isogeny φ given by

$$\varphi(x, y) = \left(\frac{x^2 - x + 2}{x - 1}, \frac{x^2 - 2x - 1}{x^2 - 2x + 1} y \right). \tag{2.115}$$

It is also found $\tilde{a} = a - 5t = -11$ and $\tilde{b} = b - 7w = -14$ and thus the curve equation is $\tilde{E} : y^2 = x^3 - 11x - 14$.

Besides, an explicit formula for l-isogenies with odd l is given as follows:

Theorem 2.89. (l-isogeny with odd number l) Let E be an elliptic curve given by $y^2 = x^3 + ax + b$ defined over $\mathbb{F}_q$. Let G be a subgroup of E of odd order. For a generator $Q = (x_Q, y_Q) \neq \mathcal{O}$ of G, let

$$t_Q = 3x_Q^2 + a, u_Q = 2y_Q^2, w_Q = u_Q + t_Q x_Q, \tag{2.116}$$

$$t = \sum_{Q \in G, Q \neq \mathcal{O}} t_Q, w = \sum_{Q \in G, Q \neq \mathcal{O}} w_Q, \tag{2.117}$$

$$r(x) = x + \sum_{Q \in G, Q \neq \mathcal{O}} \left(\frac{t_Q}{x - x_Q} + \frac{u_Q}{(x - x_Q)^2} \right). \tag{2.118}$$

Then, the rational map

$$\varphi(x, y) = \left(r(x), \frac{d}{dx} r(x) y \right), \tag{2.119}$$

is a separable isogeny from E to $\tilde{E} : y^2 = x^3 + \tilde{a}x + \tilde{b}$ where $\tilde{a} = a - 5t$ and $\tilde{b} = b - 7w$ such that $\tilde{E} = E/G$.

If $l = \#G$ is odd, any point $Q \neq \mathcal{O}$ in G end up to a negation point $-Q = (x_Q, -y_Q)$ in G. Since the formulas of t, w, and $r(x)$ only depend on x-coordinates x_Q, it is enough to sum over the half of the points in G and double the result.

Example 2.90. (3-isogeny) Let $E : y^2 = x^3 + 1$ be an elliptic curve over $\mathbb{F}_q$ where $a = 0$ and $b = 1$. Let G be a subgroup of E of order 3 which consists by $\mathcal{O}$, $(0, 1)$, and $(0, -1)$. When taking $Q = (0, 1)$, $t_Q = 0$, $u_Q = 2$, $w_Q = 2$, $t = 0$, $w = 4$ and $r(x) = x + 4/x^2$ which leads to $\frac{d}{dx} r(x) = 1 - 8/x^3$. Thus, the formula of 3-isogeny φ is given by

$$\varphi(x, y) = \left(x + \frac{4}{x^2}, y - \frac{8y}{x^3} \right). \tag{2.120}$$

Since $\tilde{a} = a - 5t = 0$ and $\tilde{b} = b - 7w = -27$ and thus the curve equation is $\tilde{E} : y^2 = x^3 - 27$.

An isogeny of composite degrees can always be decomposed into a sequence of isogenies of prime degrees. This means that if an isogeny has degree $l_1^{e_1} l_2^{e_2} \cdots l_n^{e_n}$ with primes l_i and positive integers e_i for $1 \leq i \leq n$, the isogeny is decomposed into e_1 l_1-isogenies, e_2 l_2-isogenies, ..., and e_n l_n-isogenies. Particularly, let us consider the case of l^e-isogeny where l and e are positive integers with small l. Let G be a subgroup of E of order l^e and R be a generator of G, i.e., $G = \langle R \rangle$. Then, l^e-isogeny $\varphi : E \to \tilde{E} \cong E/G$ is decomposed into e isogenies of degree l and is computed by initializing $E_0 = E$ and $R_0 = R$ and constructing the curve E_{i+1} and isogeny φ_i for $0 \leq i < e$ as follows [JDF11]:

$$E_{i+1} = E_i/\langle [l^{e-i-1}]R_i \rangle, \varphi_i : E_i \to E_{i+1}, R_{i+1} = \varphi_i(R). \tag{2.121}$$

This results in $\varphi = \varphi_{e-1} \circ \cdots \circ \varphi_1 \circ \varphi_0$ where $\circ$ is a symbol of composite mapping and φ_i for $0 \leq i \leq e-1$ are l-isogenies. Note that the large-degree isogenies can be accelerated by finding an optimal path of a directed acyclic graph as described in Sect. 4.2.2 of [JDF11]. The optimal path is determined by the relative costs of point multiplication by l and l-isogeny evaluation.

2.5.4 Isogeny graphs

An isogeny graph is often used to discuss the security of the protocols based on the isogeny. The isogeny graph has nodes of the j-invariant of isogenous elliptic curves of an elliptic curve E over $\mathbb{F}_q$, which are elements in $\mathbb{F}_q$. If E and $\tilde{E}$ are isogenous over $\mathbb{F}_q$, i.e., if there exists an isogeny $\varphi : E \to \tilde{E}$, the nodes $j(E)$ and $j(\tilde{E})$ are connected by an edge. Since isogenous elliptic curves of E is supersingular if and only if E is supersingular, there appear ordinary and supersingular isogeny graphs. For the l-isogeny graph which is considered the l-isogenous curves of E with a small prime l such that $l \neq p = \mathrm{char}(\mathbb{F}_q)$, there are structural differences between the graphs; the supersingular l-isogeny graph is one of Ramanujan graphs which have attractive properties in cryptography.

2.5.5 Computational problems

This subsection presents basic computational problems of the isogenies. Since the supersingular isogeny DH (SIDH) key exchange is not so simpler than DH/ECDH, this subsection does not provide the problems related to SIDH. Note that the details of the problems are described in Sect. 5.2.3 after the description of the details of the steps of SIDH. The following is a problem that is the template for the whole subject.

Definition 2.91. (General isogeny problem) Given $x, y \in \mathbb{F}_q$, find an isogeny $\varphi : E \to \tilde{E}$ such that $j(E) = x$ and $j(\tilde{E}) = y$.

This is equivalent to finding an isogeny $\varphi : E \to \tilde{E}$ for given E and $\tilde{E}$ over $\mathbb{F}_q$. A variant of this problem is when the degree of φ is given.

Table 2.4: The running time for solving the computational problems.

Problems		Classical	Quantum
DLP		Sub-exponential	Polynomial
ECDLP		Exponential	Polynomial
Isogeny problem	Ordinary	Exponential	Sub-exponential
	Supersingular	Exponential	Exponential

Definition 2.92. (l^e-isogeny problem) Given $x, y \in \mathbb{F}_q$ and positive integer l^e, compute an l^e-isogeny $\varphi : E \to \tilde{E}$ such that $j(E) = x$ and $j(\tilde{E}) = y$.

The isogeny problems are classified into supersingular and ordinary cases, which are corresponding to the differences of the isogeny graphs. Currently, the best generic algorithm for finding a path between two vertices in the isogeny graph is given by Galbraith [Gal99], which is a meet-in-the-middle strategy. In the supersingular case, Delfs and Galbraith improved the algorithm to only use a constant amount of memory in [DG16].

In general, the complexity for executing the algorithms for solving the isogeny problems would be exponential in the input size. In certain special cases that are used for practical applications such as SIDH key exchange, there is a compact description of the path. For such cases of the problems, the algorithm typically requires the number of steps $O(\sqrt{q})$ and $O(\sqrt[4]{q})$ for solving the supersingular and ordinary isogeny problems with a classical computer, respectively. Furthermore, even though the algorithm is executed with a quantum computer, it requires the number of steps $O(\sqrt[4]{q})$ for the supersingular isogeny problems having the exponential running time, however, it is sub-exponential time for the ordinary case. Indeed, using the form of Eq. (2.28), the algorithm takes $L_q(1/2, \sqrt{3}/2)$ for solving the ordinary case.

2.6 Chapter summary

This chapter described the fundamentals of the materials of the finite fields, elliptic curves, pairings, and isogenies used for cryptography. The important facts of the materials are summarized in the following descriptions. Table 2.4 also summarizes the computational problems and running time for solving the problems.

- A finite field is a set that consists of finite elements in which addition + and multiplication are defined. A prime field $\mathbb{F}_p$ is the smallest subgroups of a field $\mathbb{F}_q$. If $q = p^m$ $(m > 1)$, $\mathbb{F}_q$ is an extension field of $\mathbb{F}_p$ and is isomorphic to a quotient ring $\mathbb{F}_p[x]/(f(x))$ with an irreducible polynomial $f(x)$ of degree m in a polynomial ring $\mathbb{F}_p[x]$. There is a computational problem called the discrete logarithm problem (DLP) in a multiplicative group of $\mathbb{F}_q$. For solving the problem, it is required the sub-exponential running time by using the variant of the number field sieve (NFS).

- An elliptic curve defined over $\mathbb{F}_q$ with a characteristic $p > 3$ is given by the short Weierstrass affine equation $y^2 = x^3 + ax + b$. Note that the equation drop out a point $\mathcal{O} = (0 : 1 : 0)$ in projective space into infinity. For the set E of all rational points on the elliptic curve, there is a law $\oplus$ based on the chord-and-tangent rule. Then, E forms a group under $\oplus$. There is a computational problem called the elliptic curve discrete logarithm problem (ECDLP) in an $\mathbb{F}_q$-rational point group $E(\mathbb{F}_q)$. Unlike the DLP in $\mathbb{F}_q$, the most efficient algorithm for solving ECDLP, i.e., Pollard's rho algorithm, requires the exponential running time. This means that there is an advantage to using the elliptic curves for cryptography in terms of the size of the security parameter q.

- A pairing on elliptic curve E is a map $e : G_1 \times G_2 \to G_T$ where G_1 and G_2 are subgroups of r-torsion subgroup of E and G_T is a multiplicative subgroup of $\mathbb{F}_{q^k}^*$ of order r. For $P \in G_1$ and $Q \in G_2$, the ate pairing is defined by $e(Q, P) = f_{T,Q}(P)^{(q^k-1)/r} \in G_T$ where $f_{T,Q}(P)$ is a rational function with the divisor $\mathrm{div}(f_{T,Q}) = TQ - (TQ) - (T-1)(\mathcal{O})$ and $T = t - 1$ with the Frobenius trace t. The value $f_{T,Q}(P)$ is computed by Miller's algorithm and then the final exponentiation by $(q^k - 1)/r$ in $\mathbb{F}_{q^k}^*$ is applied. The security of the pairing-based protocols is based on the difficulties of the ECDLP in G_1 and G_2, and DLP in G_T. To balance the security and efficiency, the families of pairing-friendly elliptic curves, e.g., the BLS, BN, and KSS families, are typically adopted.

- For two elliptic curves E and $\tilde{E}$ over $\mathbb{F}_q$, an isogeny is a morphism $\varphi : E \to \tilde{E}$ such that $\varphi(\mathcal{O}) = \tilde{\mathcal{O}}$. If the curves are Weierstrass form, the isogeny is written by the standard form which gives rise to easy determination of the degree and separability. For E and cyclic subgroup $G \in E$ of order l, there exists an isogeny $\varphi : E \to \tilde{E}$ of degree l such that $\tilde{E} = E/G$ and $\ker(\varphi) = G$. Then, the equations of $\tilde{E}$ and φ are determined by Vélu's formula. It is difficult to find an isogeny $E \to \tilde{E}$ from E and $\tilde{E}$, which is also known as the isogeny problem. Although solving the DLP and ECDLP only require the polynomial time with the quantum computer, solving the ordinary and supersingular isogeny problems requires the sub-exponential and exponential running times by the meet-in-middle attack, respectively. Thus, it is expected that the isogenies are used for post-quantum cryptography.

Chapter 3

Final Exponentiation for Fast Pairings

Pairings on elliptic curves are important tools for realizing innovative protocols such as searchable encryption and attribute-based encryption for secure database systems in cloud service. This chapter presents research for optimizing a computation of pairing, especially, a step of the final exponentiation, which is introduced in Sect. 1.3. In the following, the background and motivation of this research are described.

3.1 Background and motivation

Pairings are typically carried out by two steps, which are the Miller loop and final exponentiation for practical reasons. The final exponentiation is a powering an output of the Miller loop to the specific exponent in a finite field of order q^k to bring the output in an equivalence class to be the unique value. However, there is a problem that the final exponentiation becomes more of a computational bottleneck with a large embedding degree k. To achieve fact final exponentiation, the author tries to optimize that.

Before describing the details, the previous optimizations techniques are briefly described. The techniques are typically based on the p-adic expansion of d or its multiple d' of d that allows us to use the Frobenius endomorphism with low computational complexity. In [Sco+09], Scott et al. gave a systematic method to find short vectorial addition chains to compute the final exponentiation. In [FCKRH11], Fuentes et al. presented a lattice-based method for determining d' which results in an efficient final exponentiation. It is considered that the lattice-based method provides one of the most efficient final exponentiation algorithms for many curves.

The author works on the following topic.

- The author focuses on the BLS family of pairing-friendly elliptic curves with $k = 15$ which is suggested for the pairings at the 128-bit security level in the recent works

[FMP20; BEMG19]. In [FMP20], Fouotsa et al. found one of the best multiple d' by using the lattice-based method and provided the steps for computing the final exponentiation. Thus, the author presents another computation method with a new multiple of the exponent which results in more efficient final exponentiation than the previous method [FMP20]. Indeed, it is obtained by using the property of the polynomial parameterization of q for the BLS family, which is also used for expanding the exponent for the BLS curves with $k = 27$ in [ZL12] by Zhang et al.

After the publication of the first work, in [HHT20], Hayashida et al. showed the generalization of Zhang et al.'s method [ZL12] for any family of curves. Their method exactly gives rise to the same decomposition as the proposed method for the BLS family of curves with $k = 15$. At the same time as the publication of [HHT20], the generalization of the method for the BLS curves with any k is published by the author, which is also described together with the first work. The author also works on the following two topics.

- For the pairing at the 128-bit security level, in [Gui20], Guillevic provided a shortlist of the curves with $k = 10, 11, 12, 13, 14,$ and 16 that have a resistance to the STNFS. Since it had been considered that the pairings on curves with k of multiple of 4 or 6 are the best choices of the pairings, there is not enough research of the pairings on curves with a prime k or k of multiple of 2 or 3. Particularly, for the curves with $k =$ 10, 11, 13, and 14, the algorithms for computing the final exponentiations have not been provided. Thus, the author provides them by using the lattice-based method [FCKRH11]. The author also applies the latest work [HHT20] for the curves and compares the calculation costs of the final exponentiations between the two methods.
- Although the lattice-based method [FCKRH11] might produce one of the most efficient algorithms for computing the final exponentiation, it involves several heuristic processes and thus it is complicated. Contrary, the generalized method [HHT20] can generate the algorithm without not so much effort, however, it is not effective for the majority of the families of curves having the property $\deg t > 1$. Thus, the author establishes similar methods that are especially effective for such families of curves. Since the importance of curves with a prime k has been notably increased for STNFS-secure pairing, the author focuses on the specific family of curves with any prime k of $k \equiv 1 \pmod 6$ and proposes the decomposition of the multiple d' for that family.

Notation. The calculation costs of the exponentiation by s, multiplication, squaring, cubing, inversion, and p^i-th power Frobenius endomorphism in $\mathbb{F}_{p^k}^*$ are denoted as u_k^s, m_k, s_k, c_k, i_k, and f_k^i, respectively. The calculation costs of the inversion, squaring, and cubing in a subgroup of $\mathbb{F}_{p^k}^*$ of order $\Phi_k(p)$ where Φ_k is the cyclotomic polynomial are denoted as i_{ck}, s_{ck}, and c_{ck}, respectively.

Organization. Sect. 3.2 reviews the basic facts of the final exponentiation. Sects. 3.3, 3.4, and 3.5 describe the first, second, and third works of the final exponentiation, respectively. Finally, the contributions are summarized in Sect. 3.6.

3.2 Review of the final exponentiation

This section describes the cyclotomic polynomial and reviews the basic structure of the final exponentiation. Particularly, this section reviews two related methods for constructing the algorithm for computing the final exponentiation by [FCKRH11; HHT20].

3.2.1 Cyclotomic polynomial

This subsection introduces cyclotomic polynomials which play an important role in the final exponentiation. Before providing the description, Euler's totient function is defined as follows:

Definition 3.1. (Euler's totient function) For any positive integer n, *Euler's totient function* ϕ is given as follows:

$$\phi(n) = \#\{i \in 1, 2, \ldots, n-1 : \gcd(i, n) = 1\}. \tag{3.1}$$

Definition 3.2. (Cyclotomic polynomial) For any positive integer n, the *n-th cyclotomic polynomial* is defined by

$$\Phi_n(x) = \prod_{\substack{1 \le i \le n \\ \gcd(i,n)=1}} (x - e^{2\pi i k/n}). \tag{3.2}$$

When enumerating the cyclotomic polynomials from the smallest order n, we have the following.

$$\begin{array}{ll} \Phi_1(x) = x - 1, & \Phi_2(x) = x + 1, \\ \Phi_3(x) = x^3 + x + 1, & \Phi_4(x) = x^2 + 1, \\ \Phi_5(x) = x^4 + x^3 + x^2 + x + 1, & \Phi_6(x) = x^2 - x + 1, \ldots \end{array}$$

As seen above, the cyclotomic polynomial can be defined by a polynomial with one variable and integer coefficients that is the minimal polynomial over the field of the rational numbers of a primitive n-th root of unity. The degree of Φ_n is given by $\phi(n)$. A fundamental relation involving cyclotomic polynomials is

$$\prod_{i|n} \Phi_i(x) = x^n - 1. \tag{3.3}$$

The following shows a new concept of homogeneous cyclotomic polynomial built from the cyclotomic polynomial, which is introduced in [HHT20] and is used for the final exponentiation technique.

Definition 3.3. (Homogeneous cyclotomic polynomial) For any positive integer n, the *n-th homogeneous cyclotomic polynomial* is defined as follows:

$$\Psi_n(x, y) = \begin{cases} \Phi_n(x/y)y^{\phi(n)} & \text{if } n > 1, \\ 1 & \text{if } n = 1. \end{cases} \tag{3.4}$$

When enumerating the homogeneous cyclotomic polynomials from the smallest order n, we have the following.

$$\begin{aligned} &\Psi_1(x, y) = 1, && \Psi_2(x, y) = x + y, \\ &\Psi_3(x, y) = x^2 + yx + y^2, && \Psi_4(x, y) = x^2 + y^2, \\ &\Psi_5(x, y) = x^4 + x^3y + x^2y^2 + xy^3 + y^4, && \Psi_6(x, y) = x^2 - xy + y^2, \ldots \end{aligned}$$

The homogeneous cyclotomic polynomial can also be defined by a polynomial with two variables and integer coefficients. For $n > 2$, a fundamental relation involving cyclotomic polynomials is given as follows:

$$\prod_{i|n} \Psi_i(x, y) = \sum_{j=0}^{n-1} x^{n-1-j}y^j. \tag{3.5}$$

3.2.2 Decomposition of the final exponentiation

The pairings such that the reduced Tate pairing and its variants are typically computed by two steps, i.e., the Miller loop and final exponentiation. The final exponentiation step is given as a powering $(q^k - 1)/r$ in the finite field of order q^k. For easy description, let us assume that q is not a power of a prime p but p, which is adopted for many cases of the settings of the pairings. To achieve fast computation, the exponent is typically broken into two parts as follows [KM05]:

$$\frac{p^k - 1}{r} = \left(\frac{p^k - 1}{\Phi_k(p)}\right) \cdot \left(\frac{\Phi_k(p)}{r}\right). \tag{3.6}$$

Then, the final exponentiation is performed as follows:

$$\begin{aligned} &\mathbb{F}_{p^k}^* \to G_{\Phi_k(p)} \to \mu_r, \\ &f_0 \mapsto f_1 = f_0^{\frac{p^k-1}{\Phi_k(p)}} \mapsto f_1^{\frac{\Phi_k(p)}{r}}, \end{aligned} \tag{3.7}$$

where $G_{\Phi_k(p)}$ and μ_r is a multiplicative subgroup of $\mathbb{F}_{p^k}^*$ of order $\Phi_k(p)$ and r, respectively. Note that $G_{\Phi_k(p)}$ is especially called a *cyclotomic subgroup.* The first part can be denoted as $(p^k-1)/\Phi_k(p) = \sum_i c_i p_i$ with a small integer c_i. Since the computation is clearly inexpensive by using the Frobenius endomorphisms, the first part is called the *easy part.* However, the second part, i.e, $d = \Phi_k(p)/r$, is more difficult to compute than the easy part and is called the *hard part.* The usual continuation is to express d to the base p; This is because the p-th powering in $\mathbb{F}_{p^k}$ is computed by the Frobenius endomorphism. Indeed, let us denote d as $d = d_0 + d_1 p + \cdots + d_{k'-1}p^{k'-1}$ where $k' = \phi(k)$ and d_i for $0 \le i \le k'-1$ are integers such that $0 \le d_i < p$. Assuming f is an element after raising to the power of the easy part, the hard part f^d of the final exponentiation can be computed as $f^d = f^{d_0} \cdot (f^{d_1})^p \cdot \cdots \cdot (f^{d_{k'-1}})^{p^{k'-1}}$. Then, one can construct a multi-exponentiation algorithm for computing f^{d_0}, f^{d_1}, ..., $f^{d_{k'-1}}$.

Since we can work on $G_{\Phi_k(p)}$ after raising to the easy part, several efficient operations, which are called *cyclotomic operations*, can be used during the hard part computation. It is trivial that there is an efficient inversion for any case of k. For curves with even k, several efficient arithmetic operations are also available $G_{\Phi_k(p)}$ as described in [SL02; GS10; Kar13]. It is also mentioned that there is an efficient cubing for curves with k of multiple of 3 in [GS10]. Since there is no explicit formula of this cubing, the author provides that in App. A.

3.2.3 Related works for constructing the algorithm

For families of pairing-friendly elliptic curves that have polynomial parameters $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$, there are several construction methods of the algorithms for computing the hard part of the final exponentiation [Sco+09; FCKRH11; ZL12; HHT20]. A curve in a family is specified by finding an integer z making $p(z)$ and $r(z)$ being primes and $t(z)$ being an integer. Thus, it is possible to consider the polynomials $p(z)$, $r(z)$, and $t(z)$ with the integer variable z. Then, the hard part can be expressed by a polynomial $d(z) = \Phi_k(p(z))/r(z) = d_0(z) + d_1(z)p(z) + \cdots + d_{k'-1}(z)p(z)^{k'-1}$ where $d_i(z)$ for $0 \le i \le k'-1$ are polynomials of degree $0 \le \deg d_i < \deg p$. This subsection describes the two state-of-the-art methods given by Fuentes et al. in [FCKRH11] and Hayashida et al. in [HHT20], which are referred to as the *lattice-based method* and *generalized method*, respectively.

The lattice-based method

In [Sco+09], Scott et al. proposed to construct the algorithm by using an addition-chain method for the integer coefficients of $d_0(z), d_1(z), \ldots, d_{k'-1}(z)$. Although Scott et al. straightforwardly decompose $d(z)$, it is also possible to use a multiple $d'(z)$ of $d(z)$ such that $r(z) \nmid d'(z)$ instead of $d(z)$ for the hard part since this change does not affect to the non-degenerate and bilinear of the pairings. In [FCKRH11], Fuentes et al. focused on this fact and presented a lattice-based method for determining $d'(z)$ such that $f \mapsto f^{d'(z)}$ can be computed at least as efficiently as $f \mapsto f^{d(z)}$ applied [Sco+09].

In this context, an efficient $d'(z)$ can be found by constructing a rational matrix M' with dimensions $k' \times (k' \deg p(z))$ with $k' = \phi(k)$ given as follows:

$$\begin{bmatrix} d(z) \\ xd(z) \\ \vdots \\ x^{k'-1}d(z) \end{bmatrix} = M' \left(\begin{bmatrix} 1 \\ p(z) \\ \vdots \\ p(z)^{k'-1} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{\deg p-1} \end{bmatrix} \right), \tag{3.8}$$

where $\otimes$ is a Kronecker product. Note that a i-th row and j-th column of M' consists of integer coefficients of $z^i p(z)^j$ with the basis $\{1, z, \ldots, z^{\deg p-1}\} \times \{1, p(z), \ldots, p(z)^{k'-1}\}$. Then, let us consider the integer matrix M constructed from M' as the unique matrix whose rows are multiples of the rows of M' such that the entries of M are integers, and the greatest common divisor of the set of entries is 1. Applying the LLL algorithm [LLL82] to M, a matrix with small entries can be obtained. Then, small integer linear combinations of the basis elements of the matrix are examined with the hope of finding attractive $d'(z)$.

As an example, the author refers to the application of the lattice-based method to the BN family of pairing-friendly elliptic curves with $k = 12$ in [FCKRH11] and describes the details of the derivation.

Example 3.4. (The hard part of the BN curves with $k = 12$) The BN family of curves has the following parameters.

$$\begin{cases} p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1, \\ r(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1, \\ t(x) &= 6x^2 + 1. \end{cases} \tag{3.9}$$

For an integer z making $p(z)$ and $r(z)$ being primes and $t(z)$ be an integer, the exponent of the final exponentiation is expressed as follows:

$$\frac{p(z)^{12} - 1}{r(z)} = \left((p(z)^6 - 1) \cdot (p(z)^2 + 1) \right) \cdot \frac{p(z)^4 - p(z)^2 + 1}{r(z)}, \tag{3.10}$$

where $d(z) = (p(z)^4 - p(z)^2 + 1)/r(z)$ is the hard part. In order to derive one of the best multiple $d'(z)$ of $d(z)$, let us construct a matrix M such that

$$\begin{bmatrix} d(z) \\ xd(z) \\ 6x^2d(z) \\ 6x^3d(z) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(z) \\ p(z)^2 \\ p(z)^3 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ z^2 \\ z^3 \end{bmatrix} \right). \tag{3.11}$$

Note that one can choose $6z^2d(z)$ and $6z^3d(z)$ which are the smallest multiples of $z^2d(z)$ and $z^3d(z)$ that do not involve the denominators, respectively. Indeed, M is represented as follows:

$$M = \begin{pmatrix} -2 & -18 & -30 & -36 & 1 & -12 & -18 & -36 & 1 & 0 & 6 & 0 & 1 & 0 & 0 & 0 \\ 1 & 4 & 6 & 6 & 0 & 7 & 12 & 18 & -1 & 1 & 0 & 6 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -2 & -18 & -30 & -36 & 2 & -12 & -18 & -36 & 1 & 0 & 6 & 0 \\ 0 & -1 & 0 & 0 & 1 & 4 & 6 & 6 & 0 & 8 & 12 & 18 & -1 & 1 & 0 & 6 \end{pmatrix}.$$

Applying the LLL algorithm to M, we have

$$\mathrm{LLL}(M) = \begin{pmatrix} 1 & -3 & -6 & -12 & 0 & 2 & 6 & 6 & -1 & 0 & 0 & 6 & 1 & 5 & 6 & 6 \\ 0 & -2 & -6 & -6 & 0 & 3 & 6 & 6 & -1 & -5 & -6 & -6 & 1 & -3 & -6 & -12 \\ 1 & 0 & 0 & -6 & -1 & -5 & -6 & -6 & 0 & -3 & -6 & -6 & 1 & 7 & 12 & 12 \\ 1 & 7 & 12 & 12 & -1 & 0 & 0 & 6 & 0 & -2 & -6 & -6 & 0 & 3 & 6 & 6 \end{pmatrix}.$$

When considering the linear combinations of the i-th row of $\mathrm{LLL}(M)$, there is one of the simplest sequences given as follows:

$$\begin{aligned} &\begin{pmatrix} 1 & -3 & -6 & -12 & 0 & 2 & 6 & 6 & -1 & 0 & 0 & 6 & 1 & 5 & 6 & 6 \end{pmatrix} \\ -&\begin{pmatrix} 0 & -2 & -6 & -6 & 0 & 3 & 6 & 6 & -1 & -5 & -6 & -6 & 1 & -3 & -6 & -12 \end{pmatrix} \\ -&\begin{pmatrix} 1 & 0 & 0 & -6 & -1 & -5 & -6 & -6 & 0 & -3 & -6 & -6 & 1 & 7 & 12 & 12 \end{pmatrix} \\ +&\begin{pmatrix} 1 & 7 & 12 & 12 & -1 & 0 & 0 & 6 & 0 & -2 & -6 & -6 & 0 & 3 & 6 & 6 \end{pmatrix} \\ =&\begin{pmatrix} 1 & 6 & 12 & 12 & 0 & 4 & 6 & 12 & 0 & 6 & 6 & 12 & -1 & 4 & 6 & 12 \end{pmatrix}. \end{aligned}$$

This corresponds to the multiple $d'(z) = (12z^3 + 6z^2 + 2z) \cdot d(z) = d'_0(z) + d'_1(z)p(z) + d'_2(z)p(z)^2 + d'_3(z)p(z)^3$ where $d'_i(z)$ for $0 \leq i \leq 3$ are the following polynomials.

$$d'_0(z) = 12z^3 + 12z^2 + 6z + 1, \tag{3.12a}$$

$$d'_1(z) = 12z^3 + 6z^2 + 4z, \tag{3.12b}$$

$$d'_2(z) = 12z^3 + 6z^2 + 6z, \tag{3.12c}$$

$$d'_3(z) = 12z^3 + 6z^2 + 4z - 1. \tag{3.12d}$$

Then, the polynomials verify the relation such that

$$\begin{aligned} d'_2(z) &= 12z^3 + 6z^2 + 6z, & d'_1(z) &= d'_2(z) - 2z, \\ d'_3(z) &= d'_1(z) - 1, & d'_0(z) &= d'_2(z) + 6z^2 + 1. \end{aligned}$$

Let f be an element in $G_{\Phi_{12}(p(z))}$ after computing the easy part. Then, the hard part can be computed by $f^{d'(z)} = \mu_0 \cdot \mu_1^{p(z)} \cdot \mu_2^{p(z)^2} \cdot \mu_3^{p(z)^3}$ where $\mu_i = f^{d'_i(z)}$ for $0 \leq i \leq 3$ are computed by the following sequence of operations.

$$\begin{aligned} &t_0 = f^z, t_1 = t_0^2, t_2 = t_1^3, t_3 = t_2^z, t_4 = t_3^2, t_5 = t_4^z, \\ &\mu_2 = t_5 \cdot t_3 \cdot t_2, \mu_1 = \mu_2 \cdot t_1^{-1}, \mu_3 = \mu_1 \cdot f^{-1}, \mu_0 = \mu_2 \cdot t_3 \cdot f. \end{aligned} \tag{3.13}$$

where t_i for $0 \leq i \leq 5$ are variables. Then, the hard part requires 3 exponentiations by z, 9 multiplications, 2 cyclotomic squarings, 1 $p(z)$, $p(z)^2$, $p(z)^3$-th power Frobenius endomorphisms in $\mathbb{F}_{p(z)^{12}}$, 1 cyclotomic cubing, and 2 cyclotomic inversion in $G_{\Phi_{12}(p(z))}$, i.e., $3u_{12}^z + 9m_{12} + 2s_{c12} + c_{c12} + 2i_{c12} + f_{12}^1 + f_{12}^2 + f_{12}^3$.

It is considered that the lattice-based method can produce one of the most efficient algorithms for the majority of families of curves. However, as seen in the example, the method involves several heuristic processes and requires complicated works for each family of curves.

The generalized method

The hard part can also be decomposed by using the relation between $p(x)$ and $r(x)$ as Zhang et al. used for the BLS curves with $k = 27$ in [ZL12]. As the latest work, in [HHT20], Hayashida et al. generalized Zhang et al.'s method and provided a fixed expansion of the hard part for any families of curves. The heart of their method is the fact that one can express $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$ parameterizing a family of pairing-friendly elliptic curves by the following form of the polynomials:

$$\begin{cases} p(x) &= h_1(z) \cdot r(z) + T(z), \\ r(x) &= \Phi_k(T(z))/h_2(z), \\ t(x) &= T(x) + 1, \end{cases} \tag{3.14}$$

where $T(x)$, $h_1(x)$, and $h_2(x)$ are certain polynomials in $\mathbb{Q}[x]$. When taking an integer z making $p(z)$ and $r(z)$ being primes, the hard part $d(z) = \Phi_k(p(z))/r(z)$ of the final exponentiation can be automatically written by a polynomial in base $p(z)$ by using $T(z)$, $h_1(z)$, and $h_2(z)$ as follows:

Theorem 3.5. Let $k' = \phi(k)$ and let c_i for $0 \leq i \leq k'$ be integers such that $\Phi_k(X) = \sum_{i=0}^{k'} c_i X^i$. Then,

$$d(z) = h_1(z) \left(\sum_{i=0}^{k'-1} \lambda_i(z) p(z)^i \right) + h_2(z), \tag{3.15}$$

where $\lambda_{k'-1}(z) = c_{k'}$ and $\lambda_i(z) = T(z)\lambda_{i+1}(z) + c_{i+1}$ for $0 \leq i < k' - 1$.

Although the hard part is typically decomposed by using a one-variable polynomial, they also proposed to factorize the hard part as a two-variable polynomial. The factorization can be obtained by using homogeneous cyclotomic polynomials.

Theorem 3.6. For any positive integers m and n, the following is true.

1. If $k = 2^m$,

$$d(z) = h_1(z) \left(\prod_{i | \frac{k}{2}} \Psi_i(T(z), p(z)) \right) + h_2(z). \tag{3.16}$$

2. If $k = 3^n$,

$$d(z) = h_1(z) \left(\prod_{i | \frac{k}{3}} \Psi_i(T(z), p(z)) \right) (T(z)^{\frac{k}{3}} + p(z)^{\frac{k}{3}} + 1) + h_2(z). \tag{3.17}$$

3. If $k = 2^m \cdot 3^n$,

$$d(z) = h_1(z) \left(\prod_{i | \frac{k}{6}} \Psi_i(T(z), p(z)) \right) (T(z)^{\frac{k}{6}} + p(z)^{\frac{k}{6}} + 1) + h_2(z). \tag{3.18}$$

As an example, we refer to the application of Theorems 3.5 and 3.6 to the BLS family of pairing-friendly elliptic curves with $k = 12$ in [HHT20].

Example 3.7. (The hard part of the BLS curves with $k = 12$) The BLS family of curves with $k = 12$ has the parameterization

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot r(x) + x, \\ r(x) &= \Phi_{12}(x) = x^4 - x^2 + 1, \\ t(x) &= x + 1. \end{cases} \tag{3.19}$$

Then, we have $h_1(x) = \frac{1}{3}(x-1)^2$, $h_2(x) = 1$, and $T(x) = x$. For an integer z making $p(z)$ and $r(z)$ being primes, the exponent of the final exponentiation is given by the same equation as Eq. (3.10), where $d(z) = (p(z)^4 - p(z)^2 + 1)/r(z)$ is the hard part.

Applying Theorem 3.5, it is obtained that $d(z) = \frac{1}{3}(z-1)^2(\sum_{i=0}^{3} \lambda_i(z)p(z)^i) + 1$ where $\lambda_3(z) = 1$, $\lambda_2(z) = z\lambda_3(z)$, $\lambda_1(z) = z\lambda_2(z) - \lambda_3(z)$, and $\lambda_0(z) = z\lambda_1(z) + 1$. When taking the smallest multiple $d'(z) = 3d(z)$ and assuming $d'(z) = d'_0(z) + d'_1(z)p(z) + d'_2(z)p(z)^2 + d'_3(z)p(z)^3$, there are following relations between the coefficients.

$$\begin{aligned} d'_3(z) &= (z-1)^2, & d'_2(z) &= zd'_3(z), \\ d'_1(z) &= zd'_2(z) - d'_3(z), & d'_0(z) &= zd'_1(z) + 3. \end{aligned}$$

Assuming f is an element in $G_{\Phi_{12}(p(z))}$ after computing the easy part, the hard part can be computed by $f^{d'(z)} = \mu_0 \cdot \mu_1^{p(z)} \cdot \mu_2^{p(z)^2} \cdot \mu_3^{p(z)^3}$ where $\mu_i = f^{d'_i(z)}$ for $0 \le i \le 3$ are computed by the following sequence of operations.

$$\mu_3 = (f^{z-1})^{z-1}, \mu_2 = \mu_3^z, \mu_1 = \mu_2^z \cdot \mu_3^{-1}, \mu_0 = \mu_1^z \cdot f^3. \tag{3.20}$$

Then, the hard part requires 2 exponentiations by $(z-1)$, 3 exponentiation by z, 5 multiplications, 1 $p(z)$, $p(z)^2$, $p(z)^3$-th power Frobenius endomorphism in $\mathbb{F}_{p(z)^{12}}$, 1 cyclotomic cubing, and 1 cyclotomic inversion in $G_{\Phi_{12}(p(z))}$, i.e., $2u_{12}^{z-1} + 3u_{12}^z + 5m_{12} + c_{c12} + i_{c12} + f_{12}^1 + f_{12}^2 + f_{12}^3$.

On the other hand, applying Theorem 3.6, it is obtained that $d(z) = \frac{1}{3}(z-1)^2 \cdot (z + p(z)) \cdot (z^2 + p(z)^2 - 1) + 1$. When taking the smallest multiple $d'(z) = 3d(z)$, the hard part $\mu = f^{d'(z)}$ can be computed by

$$t_0 = (f^{z-1})^{z-1}, t_1 = t_0^z \cdot t_0^{p(z)}, t_2 = (t_1^z)^z \cdot t_1^{p(z)^2} \cdot t_1^{-1}, \mu = t_2 \cdot f^3, \tag{3.21}$$

where t_i for $0 \le i \le 2$ are variables. Then, the hard part takes 2 exponentiations by $(z-1)$, 3 exponentiation by z, 4 multiplications, 1 $p(z)$, $p(z)^2$-th power Frobenius endomorphism in $\mathbb{F}_{p(z)^{12}}$, 1 cyclotomic cubing, and 1 cyclotomic inversion in $G_{\Phi_{12}(p(z))}$ i.e., $2u_{12}^{z-1} + 3u_{12}^z + 4m_{12} + c_{c12} + i_{c12} + f_{12}^1 + f_{12}^2$. If $2 \mid z$, the following modification is available.

$$\begin{aligned} &t_0 = f^2, t_1 = t_0^z, t_2 = t_0^w, t_1 = t_2 \cdot t_1^{-1} \cdot f, \\ &t_1 = t_0^z \cdot t_0^{p(z)}, t_2 = (t_1^z)^z \cdot t_1^{p(z)^2} \cdot t_1^{-1}, \mu = t_2 \cdot t_0 \cdot f. \end{aligned} \tag{3.22}$$

where $w = z/2$. This hard part takes 4 exponentiations by z, 1 exponentiation by w, 7 multiplications, 1 $p(z)$, $p(z)^2$-th power Frobenius endomorphism in $\mathbb{F}_{p(z)^{12}}$, 1 cyclotomic squaring, and 2 cyclotomic inversions in $G_{\Phi_{12}(p(z))}$, i.e., $4u_{12}^z + u_{12}^w + 7m_{12} + s_{c12} + 2i_{c12} + f_{12}^1 + f_{12}^2$.

Unlike the lattice-based method, the generalized method does not require complicated works for constructing an algorithm. In fact, assuming s is the smallest integer making $sh_1(z)$ and $sh_2(z)$ to be integers, an algorithm for computing the hard part $f \to f^{sd(z)}$

Algorithm 3.1: Hard part computation [HHT20].

```
Input: $h_1(z)$, $h_2(z)$, $T(z)$, $k'$, $c_i$ for $0 \le i \le k'$, and $s$, $f \in G_{\Phi_k(p(z))}$
Output: $f^{sd(z)} \in \mu_{r(z)}$
$t \leftarrow f^{sh_2(z)}, u \leftarrow f^{sh_1(z)}$;                    //INIT
$v_{n-1} = u^{c_{k'}}$;                                                //EVAL_INIT
for $i = k' - 2$ downto $0$ do
    $v_i \leftarrow v_{i+1}^{T(z)} \cdot u^{c_{i+1}}$;                   //EVAL
endfor
$w \leftarrow v_0 \cdot t$;                                            //PROD_INIT
for $i = 1$ to $k' - 1$ do
    $w \leftarrow w \cdot v_i^{p(z)^i}$;                                  //PROD
endfor
return $w = f^{sd(z)}$;
```

is constructed for any family of curves as seen in Algorithm 3.1. Moreover, especially for the families of curves with $\deg t = 1$, e.g., the BLS family, the generalized method might provide more efficient algorithms for computing the hard part than the lattice-based method. However, for the families of curves with $\deg t > 1$, it is considered that the lattice-based method still provides an efficient algorithm than the generalized method.

3.3 Improvement of the final exponentiation for the BLS curves with $k = 15$

This section proposes a new decomposition of the hard part of the final exponentiation for the BLS curves with $k = 15$. This section also compares the operation counts of the final exponentiation for the pairing at the 128-bit security level with the previous work. The proposed decomposition is also generalized for the BLS curves with any k.

3.3.1 BLS family of pairing-friendly curves with $k = 15$

Let us recall the parameterizations $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$ of the BLS family of pairing-friendly elliptic curves with the CM discriminant $D = 3$ and embedding degree k of a composite number generated by $2^m, 3^n$, and l^o where m, n, o are positive integers $l > 3$ is a prime [BLS02]:

- $k = 2^m \cdot 3$

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot r(x) + x, \\ r(x) &= \Phi_k(x), \\ t(x) &= x + 1. \end{cases} \tag{3.23}$$

- $k = 3^n$

$$\begin{cases} p(x) &= (x-1)^2 \cdot r(x) + x, \\ r(x) &= \frac{1}{3}\Phi_k(x), \\ t(x) &= x+1. \end{cases} \tag{3.24}$$

- $k = 3^n \cdot l^o$

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot \Phi_{3^n}(x^{l^{o-1}}) \cdot r(x) + x, \\ r(x) &= \Phi_k(x), \\ t(x) &= x+1. \end{cases} \tag{3.25}$$

- $k = 2^m \cdot 3^n \cdot l^o$

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot \Phi_{2^m \cdot 3}(x^{3^{n-1}l^{o-1}}) \cdot r(x) + x, \\ r(x) &= \Phi_k(x), \\ t(x) &= x+1. \end{cases} \tag{3.26}$$

This section mainly focuses on the BLS family of curves with $k = 15$, $D = 3$, and $\rho = 1.5$ having the following parameters.

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot (x^2+x+1) \cdot \Phi_{15}(x) + x, \\ r(x) &= \Phi_{15}(x) = x^8 - x^7 + x^5 - x^4 + x^3 - x + 1, \\ t(x) &= x+1. \end{cases} \tag{3.27}$$

The above parameterization is also found by Duan et al. in [DCC05]. For constructing the curve, we need to find an integer z making $p(z)$ and $r(z)$ being primes. One can find z by applying the restriction $z \equiv 1 \pmod 3$.

3.3.2 Previous final exponentiation

In [FMP20], Fouotsa et al. proposed to decompose the exponent as follows:

$$\frac{p(z)^{15}-1}{r(z)} = \big(p(z)^5 - 1\big) \cdot \left(\frac{\Phi_3(p(z)^5)}{r(z)}\right). \tag{3.28}$$

Note that they dared to use the above decomposition, however, the exponent is typically decomposed as shown in Eq. (3.6). The first and second parts are referred to as the easy and hard parts, respectively.

For the hard part $\tilde{d}(z) = \Phi_3(p(z)^5)/r(z)$, they found one of the best multiple $\tilde{d}'(z)$ of $\tilde{d}(z)$ by the lattice-based method [FCKRH11]. In the context, they found $\tilde{d}'(z) = 3z^3 \cdot \tilde{d}(z)$ which is represented as a polynomial in base $p(z)$ given as $\tilde{d}'(z) = \tilde{d}'_0(z) + \tilde{d}'_1(z)p(z) +$

$\cdots + \tilde{d}'_9(z)p(z)^9$ where $\tilde{d}'_i(z)$ for $0 \le i \le 9$ are polynomials given as follows:

$$\begin{cases} \tilde{d}'_0(z) &= -z^6 + z^5 + z^3 - z^2, \\ \tilde{d}'_1(z) &= -z^5 + z^4 + z^2 - z, \\ \tilde{d}'_2(z) &= -z^4 + z^3 + z - 1, \\ \tilde{d}'_3(z) &= z^{11} - 2z^{10} + z^9 + z^6 - 2z^5 + z^4 - z^3 + z^2 + z + 2, \\ \tilde{d}'_4(z) &= z^{11} - z^{10} - z^9 + z^8 + z^6 - z^5 - z^4 + z^3 - z^2 + 2z + 2, \\ \tilde{d}'_5(z) &= z^{11} - z^{10} - z^8 + z^7 + 3, \\ \tilde{d}'_6(z) &= z^{10} - z^9 - z^7 + z^6, \\ \tilde{d}'_7(z) &= z^9 - z^8 - z^6 + z^5, \\ \tilde{d}'_8(z) &= z^8 - z^7 - z^5 + z^4, \\ \tilde{d}'_9(z) &= z^7 - z^6 - z^4 + z^3. \end{cases} \tag{3.29}$$

These polynomials verify the following relations.

$$\begin{aligned} &\tilde{d}'_2(z) = -(z-1)^2 \cdot (z^2 + z + 1), && \tilde{d}'_1(z) = z\tilde{d}'_2(z), \\ &\tilde{d}'_0(z) = z\tilde{d}'_1(z), && \tilde{d}'_9(z) = -z\tilde{d}'_0(z), \\ &\tilde{d}'_8(z) = z\tilde{d}'_9(z), && \tilde{d}'_7(z) = z\tilde{d}'_8(z), \\ &\tilde{d}'_6(z) = z\tilde{d}'_7(z), && \tilde{d}'_5(z) = z\tilde{d}'_6(z) + 3, \\ &\tilde{d}'_4(z) = v(z) - (\tilde{d}'_1(z) + \tilde{d}'_7(z)), && \tilde{d}'_3(z) = v(z) - (\tilde{d}'_0(z) + \tilde{d}'_6(z) + \tilde{d}'_9(z)), \end{aligned}$$

where $v(z) = \tilde{d}'_2(z) + \tilde{d}'_5(z) + \tilde{d}'_8(z)$.

For an element $\tilde{f}$ in $G_{\Phi_3(p(z)^5)}$ after raising to the power of the easy part $(p(z)^5 - 1)$, the exponentiation by the hard part $\tilde{f} \mapsto \tilde{f}^{\tilde{d}'(z)}$ is given by $\tilde{f}^{\tilde{d}'(z)} = \mu_0 \cdot \mu_1^{p(z)} \cdot \mu_2^{p(z)^2} \cdot \mu_3^{p(z)^3} \cdot \mu_4^{p(z)^4} \cdot \mu_5^{p(z)^5} \cdot \mu_6^{p(z)^6} \cdot \mu_7^{p(z)^7} \cdot \mu_8^{p(z)^8} \cdot \mu_9^{p(z)^9}$ where $\mu_i = \tilde{f}^{\tilde{d}'_i(z)}$ for $0 \le i \le 9$ are computed by the following sequence of operations.

$$\begin{aligned} &t_0 = (\tilde{f}^{z-1})^{z-1}, t_1 = t_0^z, t_2 = t_1^z, \mu_2 = (t_0 \cdot t_1 \cdot t_2)^{-1}, \\ &\mu_1 = \mu_2^z, \mu_0 = \mu_1^z, \mu_9 = (\mu_0^z)^{-1}, \mu_8 = \mu_9^z, \\ &\mu_7 = \mu_8^z, \mu_6 = \mu_7^z, \mu_5 = \mu_6^z \cdot \tilde{f}^3, t_3 = \mu_2 \cdot \mu_5 \cdot \mu_8, \\ &\mu_4 = t_3 \cdot (\mu_1 \cdot \mu_7)^{-1}, \mu_3 = t_3 \cdot (\mu_0 \cdot \mu_6 \cdot \mu_9)^{-1}, \end{aligned} \tag{3.30}$$

where t_i for $0 \le i \le 3$ are variables.

Applying the above method, the calculation cost of powering the easy part is 1 $p(z)^5$-Frobenius endomorphism, 1 inversion, and 1 multiplication in $\mathbb{F}_{p(z)^{15}}$. Besides, the calculation cost of powering the hard part is 2 exponentiations by $(z-1)$, 9 exponentiations by z, 19 multiplications, 1 $p(z)$, $p(z)^2$, $p(z)^3$, $p(z)^4$, $p(z)^5$, $p(z)^6$, $p(z)^7$, $p(z)^8$, $p(z)^9$-Frobenius endomorphisms in $\mathbb{F}_{p(z)^{15}}$, 1 cubing and 4 inversions in $G_{\Phi_3(p(z)^5)}$. Thus, the calculation cost of the final exponentiation is given by $2u_{15}^{z-1} + 9u_{15}^{z} + 20m_{15} + c_{c15} + 4i_{c15} + i_{15} + \sum_{i=0}^{9} f_{15}^{i} + f_{15}^{5}$.

3.3.3 Proposed final exponentiation

Unlike Fouotsa et al.'s method [FMP20], the author decomposes the exponent according to Eq. (3.6) as follows:

$$\frac{p(z)^{15}-1}{r} = \left((p(z)^5-1)\cdot(p(z)^2+p(z)+1)\right)\cdot\left(\frac{\Phi_{15}(p(z))}{r(z)}\right), \tag{3.31}$$

where the first and second parts are easy and hard parts of the final exponentiation, respectively. With the above decomposition, the author proposes to represent a multiple of $d(z) = \Phi_{15}(p(z))/r(z)$ as a polynomial in base $p(z)$ which are derived by the following process.

Let us define an extra parameter $m(z)$ such that

$$m(z) = \tfrac{1}{3}(z-1)^2\cdot(z^2+z+1). \tag{3.32}$$

Then, $p(z)$ is denoted by $p(z) = m(z)\cdot r(z)+z$ and the hard part $d(z)$ is represented as a polynomial in base $r(z)$ such that $d(z) = \Phi_{15}(m(z)\cdot r(z)+z)/r(z)$. Since the constant term of numerator of $d(z)$ in base $r(z)$ is $\Phi_{15}(z) = r(z)$, the denominator of $d(z)$ is easily canceled. Then, the polynomial $d(z)$ in base $r(z)$ can be converted to a polynomial in base $p(z)$ by replacing r with $(p(z)-z)/m(z)$ in a straightforward way. Note that in [ZL12], Zhang et al. also expanded the polynomial of the hard part for the BLS curves with $k = 27$ by using the property of $p(z) = m(z)\cdot r(z)+z$ which leads to a recursion relation $p(z)^{i+1} = m(z)\cdot r(z)\cdot p(z)^i + zp(z)^i$ where i is a positive integer.

As a result, it is found that $d(z) = d_0(z)+d_1(z)p(z)+\cdots+d_7(z)p(z)^7$ where $d_i(z)$ for $0 \le i \ \le 7$ are polynomials given as follows:

$$\begin{cases} d_0(z) &= m(z)\cdot(z^7-z^6+z^4-z^3+z^2-1)+1,\\ d_1(z) &= m(z)\cdot(z^6-z^5+z^3-z^2+z),\\ d_2(z) &= m(z)\cdot(z^5-z^4+z^2-z+1),\\ d_3(z) &= m(z)\cdot(z^4-z^3+z-1),\\ d_4(z) &= m(z)\cdot(z^3-z^2+1),\\ d_5(z) &= m(z)\cdot(z^2-z),\\ d_6(z) &= m(z)\cdot(z-1),\\ d_7(z) &= m(z). \end{cases} \tag{3.33}$$

Then, it is observed that the above polynomials already have the following simple relations before applying the lattice-based method [FCKRH11].

$$d_7(z) = m(z), \qquad d_6(z) = (z-1)\cdot d_7(z),$$
$$d_5(z) = zd_6(z), \qquad d_4(z) = zd_5(z)+d_7(z),$$

$$d_3(z) = zd_4(z) - d_7(z), \qquad d_2(z) = zd_3(z) + d_7(z),$$
$$d_1(z) = zd_2(z), \qquad d_0(z) = zd_1(z) - d_7(z) + 1,$$

which implies that the relations can provide one of the efficient computations for the final exponentiation. Indeed, since there exists a denominator 3 of $d_7(z) = m(z) = \frac{1}{3}(z-1)^2 \cdot (z^2+z+1)$ which leads to a cube root computation, the author proposes to use a minimum multiple $d'(z) = 3d(z)$ for a practical final exponentiation. Assuming $d'(z) = d'_0(z) + d'_1(z)p(z) + \cdots + d'_7(z)p(z)^7$ where $d'_i(z) = 3d_i(z)$ for $0 \le i \ \le 7$, the polynomials clearly verify the following simpler relations than that of the previous method [FMP20].

$$d'_7(z) = (z-1)^2 \cdot (z^2+z+1), \qquad d'_6(z) = (z-1) \cdot d'_7(z),$$
$$d'_5(z) = zd'_6(z), \qquad d'_4(z) = zd'_5(z) + d'_7(z),$$
$$d'_3(z) = zd'_4(z) - d'_7(z), \qquad d'_2(z) = zd'_3(z) + d'_7(z),$$
$$d'_1(z) = zd'_2(z), \qquad d'_0(z) = zd'_1(z) - d'_7(z) + 3.$$

Note that above decomposition is the exactly same as the current state-of-the-art given by Theorem 3.5 by [HHT20].

For an element f in $G_{\Phi_{15}(p(z))}$ after raising to the power of the easy part given as $(p(z)^5-1)\cdot(p(z)^2+p(z)+1)$, the exponentiation by the hard part $f \mapsto f^{d'(z)}$ is computed as $f^{d'(z)} = \nu_0 \cdot \nu_1^{p(z)} \cdot \nu_2^{p(z)^2} \cdot \nu_3^{p(z)^3} \cdot \nu_4^{p(z)^4} \cdot \nu_5^{p(z)^5} \cdot \nu_6^{p(z)^6} \cdot \nu_7^{p(z)^7}$ where $\nu_i = f^{d'_i(z)}$ for $0 \le i \le 7$ are computed by the following sequence of operations.

$$\begin{aligned}
&t_0 = (f^{z-1})^{z-1}, t_1 = t_0^z, t_2 = t_1^z, \nu_7 = t_0 \cdot t_1 \cdot t_2,\\
&\nu_6 = \nu_7^{z-1}, \nu_5 = \nu_6^z, \nu_4 = \nu_5^z \cdot \nu_7, t_3 = \nu_7^{-1}, \nu_3 = \nu_4^z \cdot t_3\\
&\nu_2 = \nu_3^z \cdot \nu_7, \nu_1 = \nu_2^z, \nu_0 = \nu_1^z \cdot t_3 \cdot f^3,
\end{aligned} \tag{3.34}$$

where t_i for $0 \le i \le 3$ are variables.

As a result, the calculation cost of powering the easy part is 1 $p(z)$, $p(z)^2$, $p(z)^5$-Frobenius endomorphisms, 1 inversion, and 3 multiplications in $\mathbb{F}_{p(z)^{15}}$. The calculation cost of powering the hard part is 3 exponentiations by $(z-1)$, 8 exponentiations by z, 14 multiplications, 1 $p(z)$, $p(z)^2$, $p(z)^3$, $p(z)^4$, $p(z)^5$, $p(z)^6$, $p(z)^7$-Frobenius endomorphisms in $\mathbb{F}_{p(z)^{15}}$, and 1 cubing and 1 inversion in $G_{\Phi_{15}(p(z))} \subset G_{\Phi_3(p(z)^5)}$. Thus, the calculation cost of the final exponentiation is given by $3u_{15}^{z-1} + 8u_{15}^z + 17m_{15} + c_{c15} + i_{c15} + i_{15} + \sum_{i=0}^{7} f_{15}^i + f_{15}^1 + f_{15}^2 + f_{15}^5$. Comparing the previous and proposed methods, the proposed method results in reducing $u_{15}^{z-1} + 3m_{15} + 3i_{c15} + f_{15}^8 + f_{15}^9$ and increasing $u_{15}^z + f_{15}^1 + f_{15}^2$.

Table 3.1: The number of operations in $\mathbb{F}_{p(z)^{15}}$ for computing single final exponentiation of the pairing at the 128-bit security level.

Method	m_{15}	s_{15}	i_{15}	c_{c15}	i_{c15}	Frobenius end.								
						p	p^2	p^3	p^4	p^5	p^6	p^7	p^8	p^9
Fouotsa et al. [FMP20]	55	341	1	1	6	1	1	1	1	2	1	1	1	1
This work	53	341	1	1	4	2	2	1	1	2	1	1	0	0

Table 3.2: The calculation cost of arithmetic operations in $\mathbb{F}_{p(z)^{15}}$.

m_{15}	s_{15}	i_{15}	c_{c15}	i_{c15}	f_{15}^{i}
$78m_1$	$78m_1$	$229m_1$	$117m_1$	$78m_1$	$14m_1$

Table 3.3: The number of operations in $\mathbb{F}_{p(z)}$ for computing single final exponentiation of the pairing at the 128-bit security level.

Methods	Calculation costs
Fouotsa et al. [FMP20]	$31842m_1$
This work	$31530m_1$

3.3.4 Calculation cost estimations

The author estimates the calculation costs of the previous and proposed final exponentiations of the pairings on the BLS with $k = 15$ at the 128-bit security level. In [FMP20], Fouotsa et al. provided an integer seed z for the pairings on the BLS curves with $k = 15$ at the 128-bit security level given as follows:

$$z = 2^{31} + 2^{19} + 2^5 + 2^2. \tag{3.35}$$

The above parameter can generate primes $p(z)$ and $r(z)$ with 383-bit and 249-bit lengths, which is closed to the 256-bit as required to have 128-bit security on elliptic curves. With the square-and-multiply algorithm, the exponentiation by z in $\mathbb{F}_{p(z)^{15}}$ takes $u_{15}^{z} = 3m_{15}+31s_{15}$. The exponentiation by $(z-1)$ in $\mathbb{F}_{p(z)^{15}}$ also takes $u_{15}^{z-1} = 4m_{15}+31s_{15}+i_{c15}$.

Substituting the calculation costs of u_{15}^{z} and u_{15}^{z-1}, the number of operations in $\mathbb{F}_{p(z)^{15}}$ for the previous and proposed final exponentiations are obtained as in Table 3.1. Comparing the costs, the proposed method results in reducing $2m_{15}+2i_{c15}+f_{15}^{8}+f_{15}^{9}$ from the previous cost of the final exponentiation. Although the proposal also results in increasing $f_{15}^{1} + f_{15}^{2}$, the reduced calculation costs are still larger than the increased ones.

The calculation costs for the arithmetic operations in $\mathbb{F}_{p^{15}}$ are denoted by the calculation cost m_1 of multiplication in $\mathbb{F}_{p(z)}$ as in Table 3.2, which is derived by referring to [GMT20]. Replacing the costs of operations in $\mathbb{F}_{p(z)^k}$ with that of $\mathbb{F}_{p(z)}$, the costs of the proposed and previous final exponentiations are denoted as in Table 3.3. This indicates that the proposed method results in reducing $312m_1$ from the previous ones. Thus, it is concluded that the proposed method is clearly more effective than the previous method by Fouotsa et al. in [FMP20].

3.3.5 Generalization for any k

The author briefly describes the generalization of the proposed method of the BLS curve with $k = 15$ for the BLS curves with any k. It is observed that one can find a polynomial $m(z)$ such that $p(z)$ by $p(z) = m(z) \cdot r(z) + z$ for the case of any k. Indeed, $m(z)$ and $r(z)$ are given as follows:

- $k = 2^m \cdot 3$

$$\begin{cases} m(z) &= \frac{1}{3}(z-1)^2, \\ r(z) &= \Phi_k(z). \end{cases} \tag{3.36}$$

- $k = 3^n$

$$\begin{cases} m(z) &= (z-1)^2, \\ r(z) &= \frac{1}{3}\Phi_k(z). \end{cases} \tag{3.37}$$

- $k = 3^n \cdot l^o$

$$\begin{cases} m(z) &= \frac{1}{3}(z-1)^2 \cdot \Phi_{3^n}(z^{l^{o-1}}), \\ r(z) &= \Phi_k(z). \end{cases} \tag{3.38}$$

- $k = 2^m \cdot 3^n \cdot l^o$

$$\begin{cases} m(z) &= \frac{1}{3}(z-1)^2 \cdot \Phi_{2^m.3}(z^{3^{n-1}l^{o-1}}), \\ r(z) &= \Phi_k(z). \end{cases} \tag{3.39}$$

Thus, the derivation of the decomposition of the hard part $d(z) = \Phi_k(p(z))/r(z)$ of the final exponentiation for the case of $k = 15$ described in Sect. 3.3.2 can be extended for the case of any k.

Theorem 3.8. Let $k' = \phi(k)$ and let c_i for $0 \le i \le k'$ be integers such that $\Phi_k(X) = \sum_{i=0}^{k'} c_i X^i$. Then, the polynomials $d_i(z)$ for $0 \le i \le k'-1$ such that $d(z) = \sum_{i=0}^{k'-1} d_i(z)p(z)^i$ are generated by Algorithm 3.2.

Before providing proof of Theorem 3.8, the following lemma is provided.

Lemma 3.9. Let $m'(z)$ be any polynomial and let $p(z)$ be a polynomial such that $p(z) = m'(z) \cdot \Phi_k(z) + z$. Let $\mu_i(z)$ for $0 \le i \le k' - 1$ be polynomials defined as follows:

$$\mu_{k'-1}(z) = c_{k'} m'(z),$$

Algorithm 3.2: Generation of the coefficients of the hard part in base p for the BLS curves with any k.

```
Input: $k$, $k'$, $c_i$ for $0 \leq i \leq k'$, $m(z)$
Output: $d_i(z)$ for $0 \leq i \leq k' - 1$
$d_{k'-1}(z) \leftarrow c_{k'} m(z)$;
For $i$ from $k' - 2$ downto $0$ do
    $d_i(z) \leftarrow z d_{i+1}(z) + c_{i+1} m(z)$;
endfor
If $k = 3^n$ then
    $d_0(z) \leftarrow d_0(z) + 3c_0$;
else
    $d_0(z) \leftarrow d_0(z) + c_0$;
endif
return $d_i(z)$ for $0 \leq i \leq k' - 1$
```

$$\begin{aligned}
\mu_{k'-2}(z) &= z\mu_{k'-1}(z) + c_{k'-1}m'(z),\\
\mu_{k'-3}(z) &= z\mu_{k'-2}(z) + c_{k'-2}m'(z),\\
&\vdots\\
\mu_1(z) &= z\mu_2(z) + c_2 m'(z),\\
\mu_0(z) &= z\mu_1(z) + c_1 m'(z) + c_0.
\end{aligned}$$

Then, the following is true.

$$\frac{\Phi_k(p(z))}{\Phi_k(z)} = \sum_{i=0}^{k'-1} \mu_i(z) p(z)^i. \tag{3.40}$$

Proof of Lemma 3.9. Let α be one of roots of $\Phi_k(z)$. Then, since $\Phi_k(\alpha) = 0$, we have

$$\Phi_k(p(\alpha)) = \Phi_k(m(\alpha) \cdot \Phi_k(\alpha) + \alpha) = \Phi_k(\alpha) = 0. \tag{3.41}$$

This means that $\Phi_k(p(z))$ has a factor $\Phi_k(z)$ and thus there exits a polynomial $\Phi_k(p(z))/\Phi_k(z)$. Since $\deg(\Phi_k(p(z))/\Phi_k(z)) = \deg(\Phi_k(p(z))) - \deg(\Phi_k(z)) = k'(\deg p - 1)$, one can find $\nu_i(z)$ for $0 \leq i \leq k'-1$ of degree $\deg \nu_i = \deg p - 1$ such that $\Phi_k(p(z))/\Phi_k(z) = \sum_{i=0}^{k'-1} \nu_i(z) p(z)^i$. From the definition of $p(z)$, we have $\Phi_k(z) = (p(z) - z)/m'(z)$, the above equation can be modified as follows:

$$m'(z) \cdot \Phi_k(p(z)) = (p(z) - z) \cdot \sum_{i=0}^{k'-1} \nu_i(z) p(z)^i. \tag{3.42}$$

Expanding the left and right sides of the equation, we have

$$\begin{aligned}
&m'(z) \cdot (c_{k'}p(z)^{k'} + c_{k'-1}p(z)^{k'-1} + \cdots + c_1p(z) + c_0) \\
=&\nu_{k'-1}(z)p(z)^{k'} \\
&+ (\nu_{k'-2}(z) - z\nu_{k'-1}(z))p(z)^{k'-1} \\
&+ (\nu_{k'-3}(z) - z\nu_{k'-2}(z))p(z)^{k'-2} \\
&+ \cdots \\
&+ (\nu_1(z) - z\nu_2(z))p(z)^2 \\
&+ (\nu_0(z) - z\nu_1(z))p(z) \\
&- z\nu_0(z).
\end{aligned} \tag{3.43}$$

The equation is regarded as a polynomial in base $p(z)$ and the coefficients of $p(z)^i$ are compared for $2 \leq i \leq k'$. Then, one can determine the polynomials $\nu_i(z)$ for $0 \leq i \leq k' - 1$ as follows:

$$\begin{aligned}
\nu_{k'-1}(z) &= c_{k'}m'(z), \\
\nu_{k'-2}(z) &= z\nu_{k'-1} + c_{k'-1}m'(z) && \text{from } \nu_{k'-2}(z) - z\nu_{k'-1}(z) = c_{k'-1}m'(z), \\
\nu_{k'-3}(z) &= z\nu_{k'-2} + c_{k'-2}m'(z) && \text{from } \nu_{k'-3}(z) - z\nu_{k'-2}(z) = c_{k'-2}m'(z), \\
&\vdots \\
\nu_1(z) &= \nu_2(z) + c_2m'(z) && \text{from } \nu_1(z) - z\nu_2 = c_2m'(z).
\end{aligned}$$

The construction results in the relation $\nu_1(z) = m'(z)\cdot(c_{k'}z^{k'-2}+c_{k'-1}z^{k'-3}+\cdots+c_3z+c_2)$. The remaining $\nu_0(z)$ needs to satisfy $m'(z)\cdot(c_1p(z)+c_0) = (\nu_0-\nu_1)p(z)-\nu_0z$ which leads to $\nu_0(z) = c_1\nu_1(z) + c_0$. Since $\nu_i = \mu_i$ for $0 \leq i \leq k' - 1$, the lemma is true. □

Proof of Theorem 3.8. If $k \neq 3^n$, $r(z) = \Phi_k(z)$ and $p(z) = m(z) \cdot \Phi_k(z) + z$. Application of Lemma 3.9 straightforwardly leads to $d(z) = \Phi_k(p(z))/\Phi_k(z) = \sum_{i=0}^{k'-1} \mu_i(z)p(z)^i$, which indicates $d_i(z) = \mu_i(z)$ for $0 \leq i \leq k' - 1$. On the other hand, if $k = 3^n$, $r(z) = \frac{1}{3}\Phi_k(z)$ and $p(z) = \frac{1}{3}m(z) \cdot \Phi_k(z) + z$. Application of Lemma 3.9 also indicates $d(z) = 3\Phi_k(p(z))/\Phi_k(z) = 3\sum_{i=0}^{k'-1} \mu_i(z)p(z)^i$. Since the multiple 3 is canceled by $m'(z) = m(z)/3$, this indicates $d_i(z) = \mu_i(z)$ for $1 \leq i \leq k' - 1$ and $d_0(z) = z\mu_1(z) + c_1m(z) + 3c_0$. It can be easy confirmed that Algorithm 3.2 generates such $d_i(z)$ for $0 \leq i \leq k' - 1$. □

Note that Theorem 3.8 exactly provides the same of the hard part $d(z) = \Phi_k(z)/r(z)$ as Theorem 3.5 by Hayashida et al. in [HHT20]. For the case of $k = 2^m \cdot 3^n$ with any positive integers m and n, Theorem 3.6 might give rise to a slightly simpler decomposition than Theorem 3.8.

3.4 Efficient final exponentiation for the curves resistant to STNFS

This section provides the final exponentiation computations for the families of pairing-friendly elliptic curves with $k = 10$, 11, 13, and 14 that are suggested for the STNFS-secure pairings at the 128-bit security level. Both the lattice-based and generalized methods are applied for estimations of the calculation costs of the final exponentiation.

3.4.1 Cyclotomic families of pairing-friendly curves with $k = 10$, 11, 13, and 14

There are five families of pairing-friendly elliptic curves with $k = 10$, 11, 13, and 14 and which are the cyclotomic families introduced in [FST10] and its variants and are suggested for the STNFS-secure pairings in [Gui20]. To distinguish the families, the CM discriminant D and ρ-value are also presented. The following shows the polynomials $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$ parameterizing the families of curves.

(i) $k = 10$, $D = 15$, and $\rho = 1.75$

$$\begin{cases} p(x) &= \frac{1}{15}(4x^{14} + 4x^{13} + x^{12} - 12x^{11} - 12x^{10} - 7x^9 + 11x^8 \\ & \quad +17x^7 + 15x^6 - 3x^5 - 11x^4 + x^3 - 2x^2 + 3x + 6), \\ r(x) &= \Phi_{30}(x) = x^8 + x^7 - x^5 - x^4 - x^3 + x + 1, \\ t(x) &= x^3 + 1. \end{cases} \tag{3.44}$$

(ii) $k = 11$, $D = 3$, and $\rho = 1.30$

$$\begin{cases} p(x) &= \frac{1}{3}(x^{26} + x^{24} + x^{22} + x^{15} - 2x^{13} + x^{11} + x^4 - 2x^2 + 1), \\ r(x) &= \Phi_{33}(x) = x^{20} - x^{19} + x^{17} - x^{16} + x^{14} \\ & \quad -x^{13} + x^{11} - x^{10} + x^9 - x^7 + x^6 - x^4 + x^3 - x + 1, \\ t(x) &= -x^{13} - x^2 + 1. \end{cases} \tag{3.45}$$

(iii) $k = 11$, $D = 11$, and $\rho = 1.60$

$$\begin{cases} p(x) &= \frac{1}{11}(x^{16} + 2x^{15} + x^{14} - x^{12} - 3x^{11} - x^5 + 9x^4 - x^3 + x + 3), \\ r(x) &= \Phi_{11}(x) = x^{10} + x^9 + x^8 + x^7 \\ & \quad +x^6 + x^5 + x^4 + x^3 + x^2 + x + 1, \\ t(x) &= x^4 + 1. \end{cases} \tag{3.46}$$

(iv) $k = 13$, $D = 3$, and $\rho = 1.17$

$$\begin{cases} p(x) &= \frac{1}{3}(x^{28} + x^{27} + x^{26} + x^{15} - 2x^{14} + x^{13} + x^2 - 2x + 1), \\ r(x) &= \Phi_{39}(x) = x^{24} - x^{23} + x^{21} - x^{20} + x^{18} - x^{17} + x^{15} \\ & \qquad -x^{14} + x^{12} - x^{10} + x^9 - x^7 + x^6 - x^4 + x^3 - x + 1, \\ t(x) &= -x^{14} - x + 1. \end{cases} \tag{3.47}$$

(v) $k = 14$, $D = 3$, and $\rho = 1.33$

$$\begin{cases} p(x) &= \frac{1}{3}(x^{16} + x^{15} + x^{14} - x^9 + 2x^8 - x^7 + x^2 - 2x + 1), \\ r(x) &= \Phi_{42}(x) = x^{12} + x^{11} - x^9 - x^8 + x^6 - x^4 - x^3 + x + 1, \\ t(x) &= x^8 - x + 1. \end{cases} \tag{3.48}$$

In the following, for corresponding parameterizations, let z be an integer making $p(z)$ and $r(z)$ being primes.

3.4.2 Final exponentiations by the lattice-based method

The author provides the algorithm for computing the hard part by applying the lattice-based method [FCKRH11] for the families of curves with $k = 10, 11, 13$, and 14 described in the previous section.

(i) The cyclotomic family of curves with $k = 10$, $D = 15$, and $\rho = 1.75$

The exponent of the final exponentiation is expressed by

$$\frac{p(z)^{10} - 1}{r(z)} = \big((p(z)^5 - 1) \cdot (p(z) + 1)\big) \cdot \left(\frac{\Phi_{10}(p(z))}{r(z)}\right), \tag{3.49}$$

where $d(z) = \Phi_{10}(p(z))/r(z) = (p(z)^4 - p(z)^3 + p(z)^2 - p(z) + 1)/r(z)$ is the hard part. Let us refer to the lattice-based method and derive a multiple $d'(z)$ of $d(z)$ by constructing the matrix M such that

$$\begin{bmatrix} 15d(z) \\ 15zd(z) \\ 15z^2d(z) \\ 15z^3d(z) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(z) \\ p(z)^2 \\ p(z)^3 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{13} \end{bmatrix} \right). \tag{3.50}$$

As a result of the application the LLL algorithm to M, one of the rows of M indicates the multiple $d'(z) = 15z(z+1) \cdot d(z) = \sum_{i=0}^{3} d'_i(z)p(z)^i$ where $d'_i(z)$ for $0 \leq i \leq 3$ are

Table 3.4: The hard part computation for the cyclotomic family of curves with $k = 10$, $D = 15$, and $\rho = 1.75$.

Steps	Computed	Cost
Input: $f \in G_{\Phi_{10}(p(z))}$ Output: $f^{d'(z)} \in \mu_{r(z)}$		
$t_0 \leftarrow (((f^2)^2)^2)^2 \cdot f^{-1}$	f^{15}	$m_{10} + 4s_{c10} + i_{c10}$
$v_0 \leftarrow f^{z-1}$, $v_0 \leftarrow v_0^{z-1}$, $t_1 \leftarrow v_0^z$, $t_2 \leftarrow t_1^z$, $v_0 \leftarrow (t_2 \cdot v_0)^2 \cdot t_1$, $t_1 \leftarrow v_0^z$, $t_2 \leftarrow t_1^z$, $t_1 \leftarrow t_1 \cdot v_0$, $t_2 \leftarrow t_1 \cdot t_2$, $v_0 \leftarrow t_1 \cdot t_2^2$,	 $f^{m(z)}$	$2u_{10}^{z-1}$ $2u_{10}^z + 2m_{10} + s_{c10}$ $2u_{10}^z + m_{10}$ $2m_{10} + s_{c10}$
$v_1 \leftarrow v_0^z$, $v_2 \leftarrow v_1^z$, $g_3 \leftarrow v_1 \cdot v_2$ $g_1 \leftarrow g_3^z \cdot v_0^{-1}$, $t_1 \leftarrow v_1^{-1}$, $g_2 \leftarrow g_1^{z^2} \cdot t_1 \cdot t_0$, $g_0 \leftarrow g_2^z \cdot t_1$,	$f^{d'_3(z)}$ $f^{d'_1(z)}$ $f^{d'_2(z)}$ $f^{d'_0(z)}$	$2u_{10}^z + m_{10}$ $u_{10}^z + m_{10} + i_{c10}$ $2u_{10}^z + 2m_{10} + i_{c10}$ $u_{10}^z + m_{10}$
$g \leftarrow g_0 \cdot g_1^{p(z)} \cdot g_2^{p(z)^2} \cdot g_3^{p(z)^3}$, Returen g;	$f^{d'(z)}$	$3m_{10} + \sum_{i=1}^3 f_{10}^i$

polynomials given as follows:

$$\begin{cases} d'_0(z) &= 4z^{12} + 4z^{11} + z^{10} - 12z^9 - 12z^8 - 7z^7 \\ & \quad +11z^6 + 13z^5 + 11z^4 - 4z^3 - 3z^2 + 9z, \\ d'_1(z) &= 4z^9 + 4z^8 + z^7 - 12z^6 - 8z^5 - 3z^4 + 12z^3 + 5z^2 + 3z - 6, \\ d'_2(z) &= 4z^{11} + 4z^{10} + z^9 - 12z^8 - 12z^7 - 3z^6 \\ & \quad +11z^5 + 14z^4 + 2z^3 - 3z^2 - 6z + 15, \\ d'_3(z) &= 4z^8 + 4z^7 + z^6 - 8z^5 - 8z^4 - 2z^3 + 3z^2 + 6z. \end{cases} \tag{3.51}$$

Assuming $m(z) = (z-1)^2 \cdot (2z^2 + z + 2) \cdot (2z^2 + 3z + 3)$, there are the following relations:

$$\begin{aligned} d'_3(z) &= z^2 m(z) + zm(z), & d'_1(z) &= zd'_3(z) - m(z), \\ d'_2(z) &= z^2 d'_1(z) - zm + 15, & d'_0(z) &= zd'_2(z) - zm(z), \end{aligned}$$

which leads to the computation of the hard part given in Table 3.4.

As a result, the hard part computations requires $2u_{10}^{z-1} + 10u_{10}^z + 14m_{10} + 6s_{c10} + 3i_{c10} + \sum_{i=1}^3 f_{10}^i$. Since the calculation cost of the easy part is given by $i_{10} + 2m_{10} + f_{10}^1 + f_{10}^5$, the cost of the final exponentiation is $2u_{10}^{z-1} + 10u_{10}^z + i_{10} + 16m_{10} + 6s_{c10} + 3i_{c10} + 2f_{10}^1 + f_{10}^2 + f_{10}^3 + f_{10}^5$.

(ii) The cyclotomic family of curves with $k = 11$, $D = 3$, and $\rho = 1.30$

The exponent of the final exponentiation is expressed as follows:

$$\frac{p(z)^{11}-1}{r(z)} = (p(z)-1)\cdot\left(\frac{\Phi_{11}(p(z))}{r(z)}\right), \tag{3.52}$$

where $d(z) = \Phi_{11}(p(z))/r(z) = \sum_{i=0}^{10} p(z)^i/r(z)$ is the hard part. Following the lattice-based method, let us construct the matrix M such that

$$\begin{bmatrix} 3d(z) \\ 3zd(z) \\ \vdots \\ 3z^9d(z) \end{bmatrix} = M\left(\begin{bmatrix} 1 \\ p(z) \\ \vdots \\ p(z)^9 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{25} \end{bmatrix}\right). \tag{3.53}$$

Application of the LLL algorithm to M leads to a multiple of the hard part given by $d'(z) = 3(z-1)\cdot(z^6+z^3+1)\cdot d(z) = \sum_{i=0}^{9} d'_i(z)p(z)^i$ where $d'_i(z)$ for $0 \le i \le 9$ are polynomials defined as follows:

$$\begin{cases} d'_0(z) &= -z^{22} - z^{20} - z^{18} + z^{13} + z^{11} + 4z^9 - 3, \\ d'_1(z) &= z^{20} + z^{18} + z^{16} + z^{13} + z^{11} + 2z^9 - 2z^7 + z^5 - 3, \\ d'_2(z) &= z^{13} + z^{11} + z^9 - z^7 - z^5 - z^3 - 3, \\ d'_3(z) &= -z^{16} - z^{14} + z^{13} - z^{12} + z^{11} + z^9 + 3z^3 - 3, \\ d'_4(z) &= -z^{25} - z^{23} - z^{21} + z^{13} + 3z^{12} + z^{11} + z^9 - 3, \\ d'_5(z) &= z^{23} + z^{21} + z^{19} + z^{13} + z^{12} + z^{11} - 2z^{10} + z^9 + z^8 - 3, \\ d'_6(z) &= z^{13} + z^{11} - z^{10} + z^9 - z^8 - z^6 - 3, \\ d'_7(z) &= -z^{19} - z^{17} - z^{15} + z^{13} + z^{11} + z^9 + 3z^6 - 3, \\ d'_8(z) &= z^{17} + z^{15} + 2z^{13} + z^{11} + z^9 + z^6 - 2z^4 + z^2 - 3, \\ d'_9(z) &= z^{13} + z^{11} + z^9 - z^4 - z^2 - 4. \end{cases} \tag{3.54}$$

Assuming $m(z) = (z^2+z+1)\cdot(z^4+z^2+1)$, we have the following.

$$\begin{aligned} d'_6(z) &= (z^7 - z^6)m(z) - 3, & d'_2(z) &= d'_6(z) + (z^4 - z^3)m(z), \\ d'_9(z) &= d'_2(z) + (z-1)m(z), & d'_3(z) &= -z^3d'_2(z) + d'_6(z), \\ d'_8(z) &= z(d'_2(z) - d'_3(z)) + d'_2(z) + z^2m(z), & d'_7(z) &= z^2(d'_2(z) - d'_8(z)) + d'_6(z) + z^4m(z), \\ d'_1(z) &= z(d'_2(z) - d'_7(z)) + d'_6(z) + z^4m(z), & d'_0(z) &= z^2(d'_6(z) - d'_1(z)) + d'_6(z) + z^6m(z), \\ d'_5(z) &= z(d'_6(z) - d'_0(z)) + d'_6(z) + z^6m(z), & d'_4(z) &= -z^3(d'_6(z) - d'_0(z)) - 3. \end{aligned}$$

The above formulas lead to the hard part computation given in Table 3.5.

The algorithm requires the calculation cost $25u_{11}^z + 35m_{11} + s_{11} + 10i_{c11} + \sum_{i=1}^{9} f_{11}^i$. Since the calculation cost of the easy part is given by $i_{11} + m_{11} + f_{11}^1$, the cost of the final

Table 3.5: The hard part computation for the cyclotomic family of curves with $k = 11$, $D = 3$, and $\rho = 1.30$.

Steps	Computed	Cost
Input: $f \in G_{\Phi_{11}(p(z))}$ Output: $f^{d'(z)} \in \mu_{r(z)}$		
$t_0 \leftarrow f^2 \cdot f$,	f^3	$m_{11} + s_{11}$
$t_1 \leftarrow f^z$, $t_2 \leftarrow t_1^z$, $v_0 \leftarrow f \cdot t_1 \cdot t_2$,		$2u_{11}^z + 2m_{11}$
$t_1 \leftarrow v_0^{z^2}$, $t_2 \leftarrow t_1^{z^2}$, $v_0 \leftarrow v_0 \cdot t_1 \cdot t_2$	$f^{m(z)}$	$4u_{11}^z + 2m_{11}$
$v_1 \leftarrow v_0^z$, $v_2 \leftarrow v_1^z$, $v_3 \leftarrow v_2^z$, $v_4 \leftarrow v_3^z$,		$4u_{11}^z$
$v_5 \leftarrow v_4^z$, $v_6 \leftarrow v_5^z$, $v_7 \leftarrow v_6^z$,		$3u_{11}^z$
$g_6 \leftarrow v_6 \cdot t_0$, $g_6 \leftarrow g_6^{-1}$, $g_6 \leftarrow g_6 \cdot v_7$,	$f^{d'_6(z)}$	$2m_{11} + i_{c11}$
$g_2 \leftarrow v_3^{-1} \cdot v_4$, $g_2 \leftarrow g_2 \cdot g_6$,	$f^{d'_2(z)}$	$2m_{11} + i_{c11}$
$g_9 \leftarrow v_0^{-1} \cdot v_1$, $g_9 \leftarrow g_9 \cdot g_2$,	$f^{d'_9(z)}$	$2m_{11} + i_{c11}$
$g_3 \leftarrow g_2^{z^3}$, $g_3 \leftarrow g_3^{-1} \cdot g_6$,	$f^{d'_3(z)}$	$3u_{11}^z + m_{11} + i_{c11}$
$g_8 \leftarrow g_3^{-1} \cdot g_2$, $g_8 \leftarrow g_8^z$, $g_8 \leftarrow g_8 \cdot g_2 \cdot v_2$,	$f^{d'_8(z)}$	$u_{11}^z + 3m_{11} + i_{c11}$
$g_7 \leftarrow g_8^{-1} \cdot g_2$, $g_7 \leftarrow g_7^{z^2}$,		$2u_{11}^z + m_{11} + i_{c11}$
$t_1 \leftarrow g_6 \cdot v_4$, $g_7 \leftarrow g_7 \cdot t_1$,	$f^{d'_7(z)}$	$2m_{11}$
$g_1 \leftarrow g_7^{-1} \cdot g_2$, $g_1 \leftarrow g_1^z$, $g_1 \leftarrow g_1 \cdot t_1$,	$f^{d'_1(z)}$	$u_{11}^z + 2m_{11} + i_{c11}$
$g_0 \leftarrow g_1^{-1} \cdot g_6$, $g_0 \leftarrow g_0^{z^2}$,		$2u_{11}^z + m_{11} + i_{c11}$
$t_1 \leftarrow g_6 \cdot v_6$, $g_0 \leftarrow g_0 \cdot t_1$,	$f^{d'_0(z)}$	$2m_{11}$
$t_2 \leftarrow g_0^{-1} \cdot g_6$, $t_2 \leftarrow t_2^z$, $g_5 \leftarrow t_1 \cdot t_2$,	$f^{d'_5(z)}$	$u_{11}^z + 2m_{11} + i_{c11}$
$g_4 \leftarrow t_2^{z^2}$, $g_4 \leftarrow g_4 \cdot t_0$, $g_4 \leftarrow g_4^{-1}$	$f^{d'_4(z)}$	$2u_{11}^z + m_{11} + i_{c11}$
$g \leftarrow g_0 \cdot g_1^{p(z)} \cdot g_2^{p(z)^2} \cdot g_3^{p(z)^3} \cdot g_4^{p(z)^4}$,		$4m_{11} + \sum_{i=1}^4 f_{11}^i$
$g \leftarrow g \cdot g_5^{p(z)^5} \cdot g_6^{p(z)^6} \cdot g_7^{p(z)^7} \cdot g_8^{p(z)^8} \cdot g_9^{p(z)^9}$,	$f^{d'(z)}$	$5m_{11} + \sum_{i=5}^9 f_{11}^i$
Returen g;		

exponentiation is $25u_{11}^z + i_{11} + 36m_{11} + 1s_{11} + 10i_{c11} + 2f_{11}^1 + \sum_{i=2}^9 f_{11}^i$.

(iii) The cyclotomic family of curves with $k = 11$, $D = 11$, and $\rho = 1.60$

The exponent of the final exponentiation is represented by the same equation as Eq. (3.52) where $d(z) = \Phi_{11}(p(z))/r(z) = \sum_{i=0}^{10} p(z)^i/r(z)$ is the hard part. Then, let us construct the matrix M such that

$$\begin{bmatrix} 11d(z) \\ 11zd(z) \\ \vdots \\ 11z^9 d(z) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(z) \\ \vdots \\ p(z)^9 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{15} \end{bmatrix} \right). \tag{3.55}$$

As a result, one of the best multiples of the hard part is given by $d'(z) = 11z^5 \cdot (z+1) \cdot$

Table 3.6: The hard part computation for the cyclotomic family of curves with $k = 11$, $D = 11$, and $\rho = 1.60$.

Steps	Computed	Cost
Input: $f \in G_{\Phi_{11}(p(z))}$		
Output: $f^{d'(z)} \in \mu_{r(z)}$		
$t_1 \leftarrow f^2$, $t_0 \leftarrow t_1 \cdot f$, $t_1 \leftarrow (t_0 \cdot t_1)^2 \cdot f$	f^3, f^{11}	$3m_{11} + 2s_{11}$
$v_0 \leftarrow f^z$, $t_2 \leftarrow v_0^2$, $v_0 \leftarrow v_0^z \cdot t_2$,		$2u_{11}^z + m_{11} + s_{11}$
$t_2 \leftarrow v_0^2$, $t_3 \leftarrow v_0^z$, $v_0 \leftarrow t_3^z$,		$2u_{11}^z + s_{11}$
$v_0 \leftarrow t_2 \cdot t_3 \cdot t_0$,		$2m_{11}$
$v_0 \leftarrow v_0^{z-1}$, $v_0 \leftarrow v_0^{z-1}$, $v_0 \leftarrow v_0^{-1}$,	$f^{m(z)}$	$2u_{11}^{z-1} + i_{c11}$
$v_1 \leftarrow v_0^z$, $v_2 \leftarrow v_1^z$, $v_3 \leftarrow v_2^z$,		$3u_{11}^z$
$g_2 \leftarrow v_3^z$, $g_5 \leftarrow g_2 \cdot v_3$	$f^{d'_2(z)}$, $f^{d'_5(z)}$	$u_{11}^z + m_{11}$
$g_8 \leftarrow g_5 \cdot v_2$, $g_0 \leftarrow g_8 \cdot v_1$	$f^{d'_8(z)}$, $f^{d'_0(z)}$	$2m_{11}$
$g_3 \leftarrow g_0 \cdot v_0$, $g_7 \leftarrow g_2^z \cdot g_7^{-1}$,	$f^{d'_3(z)}$, $f^{d'_7(z)}$	$u_{11}^z + 2m_{11} + i_{c11}$
$g_4 \leftarrow g_7^{z+1} \cdot t_3$, $g_1 \leftarrow g_4^z \cdot g_7$,	$f^{d'_4(z)}$, $f^{d'_1(z)}$	$u_{11}^{z+1} + u_{11}^z + 2m_{11}$
$g_9 \leftarrow g_1^z \cdot g_7$, $g_6 \leftarrow g_9^z \cdot g_7$,	$f^{d'_9(z)}$, $f^{d'_6(z)}$	$2u_{11}^z + 2m_{11}$
$g \leftarrow g_0 \cdot g_1^{p(z)} \cdot g_2^{p(z)^2} \cdot g_3^{p(z)^3} \cdot g_4^{p(z)^4}$,		$4m_{11} + \sum_{i=1}^4 f_{11}^i$
$g \leftarrow g \cdot g_5^{p(z)^5} \cdot g_6^{p(z)^6} \cdot g_7^{p(z)^7} \cdot g_8^{p(z)^8} \cdot g_9^{p(z)^9}$,	$f^{d'(z)}$	$5m_{11} + \sum_{i=5}^9 f_{11}^i$
Returen g;		

$(z^2+1) \cdot d(z) = \sum_{i=0}^{9} d'_i(z)p(z)^i$ where $d'_i(z)$ for $0 \le i \le 9$ are polynomials defined by

$$\begin{cases} d'_0(z) &= -z^{10} - 2z^9 - z^8 + 2z^6 + 5z^5 + z^4 - z^2 - 3z, \\ d'_1(z) &= z^{13} + 2z^{12} + z^{11} - z^{10} - 3z^9 - 4z^8 + z^6 + 3z^5 + 11z, \\ d'_2(z) &= -z^{10} - z^9 + z^8 + z^7 + z^6 + 2z^5 - 3z^4, \\ d'_3(z) &= -z^{10} - 2z^9 - z^8 + z^6 + 4z^5 + 2z^4 + z^3 - z - 3, \\ d'_4(z) &= z^{12} + 2z^{11} - 2z^9 - 2z^8 - 3z^7 + z^6 + 3z^5 + 11, \\ d'_5(z) &= -z^{10} - 2z^9 + 2z^7 + 2z^6 + 3z^5 - z^4 - 3z^3, \\ d'_6(z) &= z^{15} + 2z^{14} + z^{13} - z^{11} - 4z^{10} - 2z^9 - z^8 + z^6 + 3z^5 + 11z^3, \\ d'_7(z) &= z^{11} + z^{10} - z^9 - z^8 - z^7 - 2z^6 + 3z^5, \\ d'_8(z) &= -z^{10} - 2z^9 - z^8 + z^7 + 3z^6 + 4z^5 - z^3 - 3z^2, \\ d'_9(z) &= z^{14} + 2z^{13} + z^{12} - 2z^{10} - 5z^9 - z^8 + z^6 + 3z^5 + 11z^2. \end{cases} \tag{3.56}$$

Assuming $m(z) = -(z-1)^2 \cdot (z^4 + 3z^3 + 4z^2 + 4z + 3)$, there are the following relations.

$$\begin{aligned} d'_2(z) &= z^4 m(z), & d'_5(z) &= d'_2(z) + z^3 m(z), \\ d'_8(z) &= d'_5(z) + z^2 m(z), & d'_0(z) &= d'_8(z) + z m(z), \\ d'_3(z) &= d'_0(z) + m(z), & d'_7(z) &= -z d'_2(z), \\ d'_4(z) &= (z+1) d'_7(z) + 11, & d'_1(z) &= z d'_4(z) + d'_7(z), \\ d'_9(z) &= z d'_1(z) + d'_7(z), & d'_6(z) &= z d'_9(z) + d'_7(z). \end{aligned}$$

The above formulas lead to the hard part computation given in Table 3.6.

The calculation cost of the hard part is given by $12u_{11}^{z}+u_{11}^{z+1}+2u_{11}^{z-1}+24m_{11}+4s_{11}+2i_{c11}+\sum_{i=1}^{9}f_{11}^{i}$. Adding the cost of computing the easy part $i_{11}+m_{11}+f_{11}^{1}$, the cost of the hard part are obtained by $12u_{11}^{z}+u_{11}^{z+1}+2u_{11}^{z-1}+i_{11}+25m_{11}+4s_{11}+2i_{c11}+2f_{11}^{1}+\sum_{i=2}^{9}f_{11}^{i}$.

(iv) The cyclotomic family of curves with $k = 13$, $D = 3$, and $\rho = 1.17$

The exponent of the final exponentiation is given by

$$\frac{p(z)^{13}-1}{r(z)} = (p(z)-1)\cdot\left(\frac{\Phi_{13}(p(z))}{r(z)}\right), \tag{3.57}$$

where $d(z) = \Phi_{13}(p(z))/r(z)$ is the pard part. Let us refer to the lattice-based method and construct the matrix M such that

$$\begin{bmatrix} 3d(z) \\ 3zd(z) \\ \vdots \\ 3z^{11}d(z) \end{bmatrix} = M\left(\begin{bmatrix} 1 \\ p(z) \\ \vdots \\ p(z)^{11} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{27} \end{bmatrix}\right). \tag{3.58}$$

Applying the LLL algorithm to M, one can obtain one of the best multiples of $d(z)$ by $d'(z) = 3(z-1)\cdot(z+1)\cdot(z^2+1)\cdot(z^2-z+1)\cdot(z^4-z^2+1)\cdot d(z) = \sum_{i=0}^{11} d_i'(z)\cdot p(z)^i$, where $d_i(z)$ for $0 \le i \le 11$ are polynomials in $\mathbb{Q}[z]$ is defined as follows:

$$\begin{cases} d_0'(z) &=& -z^{26}-z^{25}-z^{24}+z^{14}+z^{13}+4z^{12}-3, \\ d_1'(z) &=& z^{25}+z^{24}+z^{23}+z^{14}+z^{13}+2z^{12}-2z^{11}+z^{10}-3, \\ d_2'(z) &=& z^{14}+z^{13}+z^{12}-z^{11}-z^{10}-z^{9}-3, \\ d_3'(z) &=& -z^{23}-z^{22}-z^{21}+z^{14}+z^{13}+z^{12}+3z^{9}-3, \\ d_4'(z) &=& z^{22}+z^{21}+z^{20}+z^{14}+z^{13}+z^{12}+z^{9}-2z^{8}+z^{7}-3, \\ d_5'(z) &=& z^{14}+z^{13}+z^{12}-z^{8}-z^{7}-z^{6}-3, \\ d_6'(z) &=& -z^{20}-z^{19}-z^{18}+z^{14}+z^{13}+z^{12}+3z^{6}-3, \\ d_7'(z) &=& z^{19}+z^{18}+z^{17}+z^{14}+z^{13}+z^{12}+z^{6}-2z^{5}+z^{4}-3, \\ d_8'(z) &=& z^{14}+z^{13}+z^{12}-z^{5}-z^{4}-z^{3}-3, \\ d_9'(z) &=& -z^{17}-z^{16}-z^{15}+z^{14}+z^{13}+z^{12}+3z^{3}-3, \\ d_{10}'(z) &=& z^{16}+z^{15}+2z^{14}+z^{13}+z^{12}+z^{3}-2z^{2}+z-3, \\ d_{11}'(z) &=& z^{14}+z^{13}+z^{12}-z^{2}-z-4. \end{cases} \tag{3.59}$$

Assuming $m(z) = (z^2+z+1)^2$, there are the following relations:

$d_2'(z) = (z^{10}-z^{9})m(z)-3,$ $\quad$ $d_5'(z) = d_2'(z)+(z^{7}-z^{6})m(z),$

$d_8'(z) = d_5'(z)+(z^{4}-z^{3})m(z),$ $\quad$ $d_{11}'(z) = d_8'(z)+(z-1)m(z),$

Table 3.7: The hard part computation for the cyclotomic family of curves with $k = 13$, $D = 3$, and $\rho = 1.17$.

Steps	Computed	Cost
Input: $f \in G_{\Phi_{13}(p(z))}$ Output: $f^{d'(z)} \in \mu_{r(z)}$		
$t_0 \leftarrow f^2 \cdot f$,	f^3	$m_{13} + s_{13}$
$t_1 \leftarrow f^z$, $t_2 \leftarrow t_1^z$, $v_0 \leftarrow f \cdot t_1 \cdot t_2$,		$2u_{13}^z + 2m_{13}$
$t_1 \leftarrow v_0^z$, $t_2 \leftarrow t_1^z$, $v_0 \leftarrow v_0 \cdot t_1 \cdot t_2$,	$f^{m(z)}$	$2u_{13}^z + 2m_{13}$
$v_1 \leftarrow v_0^z$, $v_2 \leftarrow v_1^z$, $v_3 \leftarrow v_2^z$, $v_4 \leftarrow v_3^z$,		$4u_{13}^z$
$v_5 \leftarrow v_4^z$, $v_6 \leftarrow v_5^z$, $v_7 \leftarrow v_6^z$, $v_8 \leftarrow v_7^z$,		$4u_{13}^z$
$v_9 \leftarrow v_8^z$, $v_{10} \leftarrow v_9^z$,		$2u_{13}^z$
$g_2 \leftarrow v_9 \cdot t_0$, $g_2 \leftarrow g_2^{-1} \cdot v_{10}$,	$f^{d'_2(z)}$	$2m_{13} + i_{c13}$
$g_5 \leftarrow v_6^{-1} \cdot v_7$, $g_5 \leftarrow g_5 \cdot g_2$,	$f^{d'_5(z)}$	$2m_{13} + i_{c13}$
$g_8 \leftarrow v_3^{-1} \cdot v_4$, $g_8 \leftarrow g_8 \cdot g_5$,	$f^{d'_8(z)}$	$2m_{13} + i_{c13}$
$g_{11} \leftarrow v_0^{-1} \cdot v_1$, $g_{11} \leftarrow g_{11} \cdot g_8$,	$f^{d'_{11}(z)}$	$2m_{13} + i_{c13}$
$t_1 \leftarrow g_{11} \cdot v_0$, $g_{10} \leftarrow g_{11}^{z^2}$, $g_{10} \leftarrow g_{10} \cdot t_1$,	$f^{d'_{10}(z)}$	$2u_{13}^z + 2m_{13}$
$g_9 \leftarrow g_{10}^{-1} \cdot g_{11}$, $g_9 \leftarrow g_9^z$, $g_9 \leftarrow g_9 \cdot t_1$,	$f^{d'_9(z)}$	$u_{13}^z + 2m_{13} + i_{c13}$
$t_1 \leftarrow g_5 \cdot v_4$, $g_7 \leftarrow g_9^{-1} \cdot g_8$,		$2m_{13} + i_{c13}$
$g_7 \leftarrow g_7^{z^2}$, $g_7 \leftarrow g_7 \cdot t_1$,	$f^{d'_7(z)}$	$2u_{13}^z + m_{13}$
$g_6 \leftarrow g_7^{-1} \cdot g_8$, $g_6 \leftarrow g_6^z$, $g_6 \leftarrow g_6 \cdot t_1$,	$f^{d'_6(z)}$	$u_{13}^z + 2m_{13} + i_{c13}$
$t_1 \leftarrow g_5 \cdot v_6$, $g_4 \leftarrow g_6^{-1} \cdot g_5$,		$2m_{13} + i_{c13}$
$g_4 \leftarrow g_4^{z^2}$, $g_4 \leftarrow g_4 \cdot t_1$,	$f^{d'_4(z)}$	$2u_{13}^z + m_{13}$
$g_3 \leftarrow g_4^{-1} \cdot g_5$, $g_3 \leftarrow g_3^z$, $g_3 \leftarrow g_3 \cdot t_1$,	$f^{d'_3(z)}$	$u_{13}^z + 2m_{13} + i_{c13}$
$t_1 \leftarrow g_2 \cdot v_9$, $g_1 \leftarrow g_3^{-1} \cdot g_2$,		$2m_{13} + i_{c13}$
$g_1 \leftarrow g_1^{z^2}$, $g_1 \leftarrow g_1 \cdot t_1$,	$f^{d'_1(z)}$	$2u_{13}^z + m_{13}$
$g_0 \leftarrow g_1^{-1} \cdot g_2$, $g_0 \leftarrow g_0^z$, $g_0 \leftarrow g_0 \cdot t_1$,	$f^{d'_0(z)}$	$u_{13}^z + 2m_{13} + i_{c13}$
$g \leftarrow g_0 \cdot g_1^{p(z)} \cdot g_2^{p(z)^2} \cdot g_3^{p(z)^3} \cdot g_4^{p(z)^4} \cdot g_5^{p(z)^5}$		$5m_{13} + \sum_{i=1}^{5} f_{13}^i$
$g \leftarrow g \cdot g_6^{p(z)^6} \cdot g_7^{p(z)^7} \cdot g_8^{p(z)^8} \cdot g_9^{p(z)^9} \cdot g_{10}^{p(z)^{10}} \cdot g_{11}^{p(z)^{11}}$,	$f^{d'(z)}$	$6m_{13} + \sum_{i=6}^{11} f_{13}^i$
Returen g;		

$$
\begin{aligned}
d'_{10}(z) &= z^2 d'_{11}(z) + d'_{11}(z) + m(z), & d'_9(z) &= -z(d'_{10}(z) - d'_{11}(z)) + d'_{11}(z) + m(z),\\
d'_7(z) &= z^2(d'_8(z) - d'_9(z)) + d'_5(z) + z^4 m(z), & d'_6(z) &= z(d'_8(z) - d'_7(z)) + d'_5(z) + z^4 m(z),\\
d'_4(z) &= z^2(d'_5(z) - d'_6(z)) + d'_5(z) + z^6 m(z), & d'_3(z) &= z(d'_5(z) - d'_4(z)) + d'_5(z) + z^6 m(z),\\
d'_1(z) &= z^2(d'_2(z) - d'_3(z)) + d'_2(z) + z^9 m(z), & d'_0(z) &= z(d'_2(z) - d'_1(z)) + d'_2(z) + z^9 m(z).
\end{aligned}
$$

The above formulas lead to the hard part computation given in Table 3.7.

As a result, the calculation costs of the easy and hard parts given by $i_{13} + m_{13} + f_{13}^1$ and $26u_{13}^z + 43m_{13} + s_{13} + 11i_{c13} + \sum_{i=1}^{11} f_{13}^i$, respectively. The calculation cost of the final exponentiation is $26u_{13}^z + i_{13} + 44m_{13} + s_{13} + 11i_{c13} + 2f_{13}^1 + \sum_{i=2}^{11} f_{13}^i$.

Table 3.8: The hard part computation for the cyclotomic family of curves with $k = 14$, $D = 3$, and $\rho = 1.33$.

Steps	Computed	Cost
Input: $f \in G_{\Phi_{14}(p(z))}$ Output: $f^{d'(z)} \in \mu_{r(z)}$		
$t_0 \leftarrow f^2 \cdot f$,	f^3	$m_{14} + s_{G_{\Phi_{14}(p)}}$
$t_1 \leftarrow f^{z^2}$, $t_2 \leftarrow t_1^{z^2}$, $v_0 \leftarrow t_1 \cdot t_2 \cdot f$,	$f^{m(z)}$	$4u_{14}^z + 2m_{14}$
$v_1 \leftarrow v_0^z$, $v_2 \leftarrow v_1^z$, $v_3 \leftarrow v_2^z$, $v_4 \leftarrow v_3^z$,		$4u_{14}^z$
$g_2 \leftarrow v_4 \cdot v_3$, $g_2 \leftarrow g_2 \cdot t_0$, $g_2 \leftarrow g_2^{-1}$,	$f^{d'_2(z)}$	$2m_{14} + i_{c14}$
$g_5 \leftarrow v_1 \cdot v_0$, $g_5 \leftarrow g_5 \cdot g_2$, $g_5 \leftarrow g_5^{-1}$,	$f^{d'_5(z)}$	$2m_{14} + i_{c14}$
$g_4 \leftarrow g_5^{z^2}$, $g_4 \leftarrow g_4^{-1}$,		$2u_{14}^z + i_{c14}$
$t_1 \leftarrow g_2 \cdot v_1$, $g_4 \leftarrow g_4 \cdot t_1$,	$f^{d'_4(z)}$	$2m_{14}$
$g_3 \leftarrow g_4 \cdot g_5$, $g_3 \leftarrow g_3^z$,		$u_{14}^z + m_{14}$
$g_3 \leftarrow g_3 \cdot t_1$, $g_3 \leftarrow g_3^{-1}$,	$f^{d'_3(z)}$	$m_{14} + i_{c14}$
$g_1 \leftarrow g_3 \cdot g_2$, $g_1 \leftarrow g_1^{z^2}$,		$2u_{14}^z + m_{14}$
$t_1 \leftarrow g_2 \cdot v_3$, $g_1 \leftarrow g_1 \cdot t_1$, $g_1 \leftarrow g_1^{-1}$,	$f^{d'_1(z)}$	$2m_{14} + i_{c14}$
$g_0 \leftarrow g_1 \cdot g_2$, $g_0 \leftarrow g_0^z$,		$u_{14}^z + m_{14}$
$g_0 \leftarrow g_0^{-1}$, $g_0 \leftarrow g_0 \cdot t_1$,	$f^{d'_0(z)}$	$m_{14} + i_{c14}$
$g \leftarrow g_0 \cdot g_1^{p(z)} \cdot g_2^{p(z)^2} \cdot g_3^{p(z)^3} \cdot g_4^{p(z)^4} \cdot g_5^{p(z)^5}$, Returen g;	$f^{d'(z)}$	$5m_{14} + \sum_{i=1}^5 f_{14}^i$

(v) The cyclotomic family of curves with $k = 14$, $D = 3$, and $\rho = 1.33$

The exponent of the final exponentiation is expressed as

$$\frac{p(z)^{14} - 1}{r(z)} = \big((p(z)^7 - 1) \cdot (p(z) + 1)\big) \cdot \left(\frac{\Phi_{14}(p(z))}{r(z)}\right), \tag{3.60}$$

where $d(z) = \Phi_{14}(p(z))/r(z)$ is the hard part. Let us refer to the lattice-based method and construct the matrix M such that

$$\begin{bmatrix} 3d(z) \\ 3zd(z) \\ \vdots \\ 3z^5 d(z) \end{bmatrix} = M \left(\begin{bmatrix} 1 \\ p(z) \\ \vdots \\ p(z)^5 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ z \\ \vdots \\ z^{15} \end{bmatrix} \right). \tag{3.61}$$

As a result, one of the best multiples of $d(z)$ is given by $d'(z) = 3(z-1) \cdot (z+1) \cdot (z^2 + z + 1) \cdot d(z) = \sum_{i=0}^{5} d'_i(z) p(z)^i$, where $d'_i(z)$ for $0 \le i \le 5$ is defined as follows:

$$\begin{cases} d'_0(z) &=& z^{14}+z^{13}+z^{12}-z^8-z^7+2z^6-3, \\ d'_1(z) &=& -z^{13}-z^{12}-z^{11}+z^8+z^7+2z^6-2z^5+z^4+3, \\ d'_2(z) &=& -z^8-z^7-z^6-z^5-z^4-z^3-3, \\ d'_3(z) &=& z^{11}+z^{10}+z^9+z^8+z^7+z^6+3z^3+3, \\ d'_4(z) &=& -z^{10}-z^9-2z^8-z^7-z^6+z^3-2z^2+z-3, \\ d'_5(z) &=& z^8+z^7+z^6-z^2-z+2. \end{cases} \tag{3.62}$$

Assuming $m(z)=z^4+z^2+1$, there are the following relations:

$$\begin{aligned} &d'_2(z)=(-z^4-z^3)m(z)-3, && d'_5(z)=-d'_2(z)-(z+1)m(z), \\ &d'_4(z)=-z^2d'_5(z)+d'_2(z)+zm(z), && d'_3=-z(d'_4(z)+d'_5(z))-d'_2(z)-zm(z), \\ &d'_1(z)=-z^2(d'_3(z)+d'_2(z))-d'_2(z)-z^3m(z), && d'_0=-z(d'_1(z)+d'_2(z))+d'_2(z)+z^3m(z). \end{aligned}$$

The above formulas lead to the hard part computation given in Table 3.8.

As a result, the calculation costs of the easy and hard parts given by $i_{14}+2m_{14}+f^1_{14}+f^7_{14}$ and $14u^z_{14}+21m_{14}+s_{c14}+6i_{c14}+\sum_{i=1}^{5}f^i_{14}$, respectively. Thus, the calculation cost of the final exponentiation is given by $14u^z_{14}+i_{14}+23m_{14}+s_{c14}+6i_{c14}+2f^1_{14}+\sum_{i=2}^{5}f^i_{14}+f^7_{14}$.

3.4.3 Final exponentiations by the generalized method

The final exponentiation by applying the generalized method is also considered. The author provides the input of $h_1(z), h_2(z), T(z)$, k', c_i for $0\le i\le k'$, and s of Algorithm 3.1 for the families of curves with $k=10$, 11, 13, and 14. Note that one can obtain $T(x)=t(x)-1$, $h_1(x)=(p(x)-T(x))/r(x)$, and $h_2(x)=\Phi_k(T(x))/r(x)$ in $\mathbb{Q}[x]$.

(i) The cyclotomic family of curves with $k=10$, $D=15$, and $\rho=1.75$

The polynomials $h_1(x)$, $h_2(x)$, and $T(x)$ are given as follows:

$$\begin{cases} h_1(x) &=& \frac{1}{15}(2x^2+3x+3)\cdot(2x^2+x+2)\cdot(x-1)^2, \\ h_2(x) &=& x^4-x^3+x^2-x+1, \\ T(x) &=& x^3. \end{cases} \tag{3.63}$$

For $h_1(z)$, $h_2(z)$, and $T(z)$ with an integer seed z, it is found that $s=15$ makes $sh_1(z)$ and $sh_2(z)$ being integers. Since $\Phi_{10}(X)=X^4-X^3+X^2-X+1$, it is obtained $k'=4$, $c_4=1$, $c_3=-1$, $c_2=1$, $c_1=-1$, and $c_0=1$.

According to Algorithm 3.1, the calculation cost of the `INIT` step is approximately given by $\approx 6u^z_{10}$. Besides, the calculation costs of the `EVAL_INIT`, `EVAL`, and `PROD_INIT`, `PROD` steps are given by $3(3u^z_{10}+m_{10})+i_{c10}$ and $4m_{10}+\sum_{i=1}^{3}f^i_{10}$, respectively. Thus, the hard part requires $15u^z_{10}+7m_{10}+i_{c10}+\sum_{i=1}^{3}f^i_{10}$. When adding the calculation cost of the easy part $i_{10} + 2m_{10} + f^1_{10} + f^5_{10}$, it is found that the final exponentiation requires $15u^z_{10} + 9m_{10} + i_{10} + i_{c10} + 2f^1_{10} + f^2_{10} + f^3_{10} + f^5_{10}$.

(ii) The cyclotomic family of curves with $k = 11$, $D = 3$, and $\rho = 1.30$

The polynomials $h_1(x)$, $h_2(x)$, and $T(x)$ in $\mathbb{Q}[x]$, are given by

$$\begin{cases} \begin{aligned} h_1(x) &= \tfrac{1}{3}(x^2 - x + 1) \cdot (x^2 + x + 1)^2, \\ h_2(x) &= x^{110} + x^{109} + x^{108} + 9x^{99} + 9x^{98} + 8x^{97} - x^{96} - x^{95} \\ &\quad +35x^{88} + 35x^{87} + 27x^{86} - 8x^{85} - 7x^{84} + x^{83} + x^{82} \\ &\quad +76x^{77} + 76x^{76} + 49x^{75} - 27x^{74} - 20x^{73} + 7x^{72} \\ &\quad +6x^{71} - x^{70} - x^{69} + 99x^{66} + 99x^{65} + 50x^{64} - 49x^{63} \\ &\quad -29x^{62} + 20x^{61} + 14x^{60} - 6x^{59} - 5x^{58} + x^{57} + x^{56} \\ &\quad +77x^{55} + 77x^{54} + 27x^{53} - 50x^{52} - 21x^{51} + 29x^{50} \\ &\quad +15x^{49} - 14x^{48} - 9x^{47} + 5x^{46} + 4x^{45} + 33x^{44} + 33x^{43} \\ &\quad +7x^{42} - 27x^{41} - 6x^{40} + 21x^{39} + 6x^{38} - 15x^{37} - 6x^{36} \\ &\quad +9x^{35} + 5x^{34} + 5x^{33} + 6x^{32} + 3x^{31} - 6x^{30} - x^{29} \\ &\quad +6x^{28} - 6x^{26} + 6x^{24} + x^{23} - 3x^{22} + 3x^{20} - 2x^{18} + x^{16} \\ &\quad -x^{12} - 2x^{11} + x^9 - x^7 + x^5 - x^3 + x + 1, \\ T(x) &= -x^{13} - x^2. \end{aligned} \end{cases} \tag{3.64}$$

For $h_1(z)$, $h_2(z)$, and $T(z)$ with an integer z, it is found that $s = 3$ makes $sh_1(z)$ and $sh_2(z)$ being integers. Since $\Phi_{11}(X) = X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$, $k' = 10$ and $c_i = 1$ for $0 \leq i \leq k'$.

Then, the calculation cost of the `INIT` step, `EVAL_INIT` and `EVAL` steps, and `PROD_INIT` and `PROD` steps in Algorithm 3.1 for computing the hard part are given by $\approx 110u^z_{11}$, $9(13u^z_{11} + 2m_{11} + i_{c11})$, and $10m_{11} + \sum_{i=1}^{9} f^i_{11}$, respectively. Thus, the calculation cost of the hard part is given by $15u^z_{10} + 7m_{10} + i_{c10} + \sum_{i=1}^{3} f^i_{10}$. Since the easy part requires $m_{11} + i_{11} + f^1_{11}$, the calculation cost of the final exponentiation is given by $227u^z_{11} + 29m_{11} + i_{11} + 9i_{c11} + 2f^1_{11} + \sum_{i=2}^{9} f^i_{11}$.

(iii) The cyclotomic family of curves with $k = 11$, $D = 11$, and $\rho = 1.60$

The polynomials $h_1(x)$, $h_2(x)$, and $T(x)$ in $\mathbb{Q}[x]$ are given as follows:

$$\begin{cases} \begin{aligned} h_1(x) &= \tfrac{1}{11}(x - 1)^2 \cdot (x^4 + 3x^3 + 4x^2 + 4x + 3), \\ h_2(x) &= (x^{10} - x^9 + x^8 - x^7 + x^6 - x^5 + x^4 - x^3 + x^2 - x + 1) \\ &\quad \cdot(x^{20} - x^{18} + x^{16} - x^{14} + x^{12} - x^{10} + x^8 - x^6 + x^4 - x^2 + 1), \\ T(x) &= x^4. \end{aligned} \end{cases} \tag{3.65}$$

For $h_1(z)$, $h_2(z)$, and $T(z)$ with an integer seed z, it is found that $s = 11$ makes $sh_1(z)$ and $sh_2(z)$ being integers. It is also found that $k' = 10$ and $c_i = 1$ for $0 \leq i \leq k'$.

As a result, the calculation cost of the `INIT` step, `EVAL_INIT` and `EVAL` steps, and `PROD_INIT`, `PROD` steps in Algorithm 3.1 for computing the hard part are given by $\approx 30u_{11}^{z}$, $9(4u_{11}^{z} + 1m_{11})$, and $10m_{11} + \sum_{i=1}^{9} f_{11}^{i}$, respectively. Therefore, the calculation cost of the final exponentiation is given by $66u_{11}^{z} + 19m_{11} + \sum_{i=1}^{9} f_{11}^{i}$. Since the easy part requires $m_{11} + i_{11} + f_{11}^{1}$, the calculation cost of the final exponentiation can be obtained as $66u_{11}^{z} + 20m_{11} + i_{11} + 2f_{11}^{1} + \sum_{i=2}^{9} f_{11}^{i}$.

(iv) The cyclotomic family of curves with $k = 13$, $D = 3$, and $\rho = 1.17$

The polynomials $h_1(x)$, $h_2(x)$, and $T(x)$ are given by

$$\left\{ \begin{array}{rcl} h_1(x) & = & \frac{1}{3}(x^2 + x + 1)^2, \\ h_2(x) & = & x^{144} + x^{143} + x^{142} + 11x^{131} + 10x^{130} + 10x^{129} - x^{128} \\ & & +54x^{118} + 44x^{117} + 45x^{116} - 9x^{115} + x^{114} + 155x^{105} \\ & & +111x^{104} + 120x^{103} - 36x^{102} + 8x^{101} - x^{100} + 286x^{92} \\ & & +175x^{91} + 210x^{90} - 84x^{89} + 28x^{88} - 7x^{87} + x^{86} + 351x^{79} \\ & & +176x^{78} + 252x^{77} - 126x^{76} + 56x^{75} - 21x^{74} + 6x^{73} - x^{72} \\ & & +287x^{66} + 111x^{65} + 210x^{64} - 126x^{63} + 70x^{62} - 35x^{61} \\ & & +15x^{60} - 5x^{59} + x^{58} + 154x^{53} + 43x^{52} + 120x^{51} - 84x^{50} \\ & & +56x^{49} - 35x^{48} + 20x^{47} - 10x^{46} + 4x^{45} - x^{44} + 54x^{40} \\ & & +11x^{39} + 45x^{38} - 36x^{37} + 28x^{36} - 21x^{35} + 15x^{34} - 10x^{33} \\ & & +6x^{32} - 3x^{31} + x^{30} + 12x^{27} + x^{26} + 10x^{25} - 9x^{24} + 8x^{23} \\ & & -7x^{22} + 6x^{21} - 5x^{20} + 4x^{19} - 3x^{18} + 2x^{17} - x^{16} - x^{13} \\ & & +x^{12} - x^{11} + x^{10} - x^{9} + x^{8} - x^{7} + x^{6} - x^{5} + x^{4} - x^{3} \\ & & +x^{2} + 1, \\ T(x) & = & -x^{14} - x. \end{array} \right. \tag{3.66}$$

For $h_1(z)$, $h_2(z)$, and $T(z)$ with an integer seed z, one can find the smallest integer $s = 3$ such that $sh_1(z)$ and $sh_2(z)$ being integers. Since $\Phi_{13}(X) = X^{12} + X^{11} + X^{10} + X^9 + X^8 + X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$, it is found that $k' = 12$ and $c_i = 1$ for $0 \leq i \leq k'$.

From the above, the calculation costs of the `INIT` step, `EVAL_INIT` and `EVAL` steps, and `PROD_INIT`, `PROD` steps in Algorithm 3.1 for computing the hard part are estimated by $\approx 144u_{13}^{z}$, $11(14u_{13}^{z} + 2m_{13} + 1i_{c13}))$, and $12m_{13} + \sum_{i=1}^{11} f_{13}^{i}$. Since the costs of the easy part and hard part are $m_{13} + i_{13} + f_{13}^{1}$ and $298u_{13}^{z} + 34m_{13} + 11i_{c13} + \sum_{i=1}^{11} f_{13}^{i}$, respectively, it is found that the final exponentiation requires $298u_{13}^{z} + 35m_{13} + i_{13} + 11i_{c13} + 2f_{13}^{1} + \sum_{i=2}^{11} f_{13}^{i}$.

Table 3.9: The curves for the STNFS-secure pairings at the 128-bit security level.

Families (k, D, ρ)	Seeds z	Bit sizes		
		$p(z)$	$p(z)^k$	$r(z)$
(i) $(10, 15, 1.75)$	$2^{32} - 2^{26} - 2^{17} + 2^{10} - 1$	446	4460	256
(ii) $(11, 3, 1.30)$	$-2^{13} + 2^{10} - 2^8 - 2^5 - 2^3 - 2$	333	3663	258
(iii) $(11, 11, 1.60)$	$-2^{26} + 2^{21} + 2^{19} - 2^{11} - 2^9 - 1$	412	4522	256
(iv) $(13, 3, 1.17)$	$2^{11} + 2^8 - 2^6 - 2^4$	310	4027	267
(v) $(14, 3, 1.33)$	$2^{21} + 2^{19} + 2^{10} - 2^6$	340	4755	256

(v) The cyclotomic family of curves with $k = 14$, $D = 3$, and $\rho = 1.33$

The polynomials $h_1(x)$, $h_2(x)$, and $T(x)$ in $\mathbb{Q}[x]$ given as follows:

$$\begin{cases} h_1(x) &= \frac{1}{3}(x^2 - x + 1) \cdot (x^2 + x + 1), \\ h_2(x) &= x^{36} - x^{35} + x^{34} - 5x^{29} + 4x^{28} - 4x^{27} - x^{26} + 9x^{22} - 5x^{21} \\ & \quad +6x^{20} + 3x^{19} + x^{18} - 6x^{15} + x^{14} - 4x^{13} - 3x^{12} - 2x^{11} \\ & \quad -x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + 1, \\ T(x) &= x^8 - x. \end{cases} \tag{3.67}$$

Then, $s = 3$ is the smallest integer such that $sh_1(x)$ and $sh_2(x)$ being integers. Since $\Phi_{14}(X) = X^6 - X^5 + X^4 - X^3 + X^2 - X + 1$, it is found that $k' = 6$, $c_6 = 1$, $c_5 = -1$, $c_4 = 1$, $c_3 = -1$, $c_2 = 1$, $c_1 = -1$, and $c_0 = 1$.

Since the calculation costs of the `INIT` step, `EVAL_INIT` and `EVAL` steps, and `PROD_INIT`, `PROD` steps in Algorithm 3.1 for computing the hard part are given by $\approx 36u_{14}^z$, $5(8u_{14}^z + 2m_{14} + 1i_{c14}) + 1i_{\Phi_{14}(p)}$, and $6m_{14} + \sum_{i=1}^{5} f_{14}^i$, respectively, It is found that the hard part requires $76u_{14}^z + 16m_{14} + 6i_{\Phi_{14}(p)} + \sum_{i=1}^{5} f_{14}^i$. In addition, since the calculation cost of the easy part is given by $2m_{14} + i_{14} + f_{14}^1 + f_{14}^7$, the final exponentiation takes $76u_{14}^z + 18m_{14} + i_{14} + 6i_{\Phi_{14}(p)} + 2f_{14}^1 + \sum_{i=2}^{5} f_{14}^i + f_{14}^7$.

3.4.4 Calculation cost estimations

In this subsection, the author estimates the calculation costs of the final exponentiations for the STNFS-secure pairings at the 128-bit security level. Indeed, in [Gui20], Guillevic suggested using the integer seeds z for the families of curves with $k = 10$, 11, 13, and 14 in Table 3.9. Then, the calculation costs u_k^z, u_k^{z-1}, and u_k^{z+1} of the exponentiation by z, $z - 1$, and $z + 1$ performed by the square-and-multiply algorithm are specifically given as Table 3.10. Substituting the calculation costs u_k^z, u_k^{z-1}, and u_k^{z+1} to the costs of the final exponentiations for the families of curves with $k = 10$, 11, 13, and 14, it is obtained the operation counts given in Table 3.11. For $k = 10, 11, 13$, and 14, the calculation costs of the arithmetic operations in $\mathbb{F}_{p^k}$ can be replaced with the cost m_1 as in Table 3.12,

Table 3.10: The calculation costs of the exponentiations by z, $z-1$, and $z+1$ in $\mathbb{F}_{p^k}$.

Families (k, D, ρ)	u_k^z	u_k^{z-1}	u_k^{z+1}
(i) $(10, 15, 1.75)$	$4m_{10} + 32s_{c10} + i_{c10}$	$4m_{10} + 32s_{c10} + i_{c10}$	-
(ii) $(11, 3, 1.30)$	$5m_{11} + 13s_{11} + i_{c11}$	-	-
(iii) $(11, 11, 1.60)$	$5m_{11} + 26s_{11} + i_{c11}$	$5m_{11} + 26s_{11} + i_{c11}$	$4m_{11} + 26s_{11} + i_{c11}$
(iv) $(13, 3, 1.17)$	$3m_{13} + 11s_{13} + i_{c13}$	-	-
(v) $(14, 3, 1.33)$	$3m_{14} + 21s_{c14} + i_{c14}$	-	-

Table 3.11: The number of operations in $\mathbb{F}_{p^k}$ for computing the final exponentiation of the pairings at the 128-bit security level.

Families (k, D, ρ)	Methods	m_k	$s_k/$ s_{ck}	i_k	i_{ck}	f_k^1	f_k^2	f_k^3	f_k^4	f_k^5	f_k^6	f_k^7	f_k^8	f_k^9	f_k^{10}	f_k^{11}
(i) $(10, 15, 1.75)$	Lattice-based	64	390	1	15	2	1	1	0	1	0	0	0	0	0	0
	Generalized	69	480	1	16	2	1	1	0	1	0	0	0	0	0	0
(ii) $(11, 3, 1.30)$	Lattice-based	161	326	1	35	2	1	1	1	1	1	1	1	1	0	0
	Generalized	1164	2951	1	236	2	1	1	1	1	1	1	1	1	0	0
(iii) $(11, 11, 1.60)$	Lattice-based	98	395	1	17	2	1	1	1	1	1	1	1	1	0	0
	Generalized	350	1716	1	66	2	1	1	1	1	1	1	1	1	0	0
(iv) $(13, 3, 1.17)$	Lattice-based	122	287	1	37	2	1	1	1	1	1	1	1	1	1	1
	Generalized	929	3278	1	309	2	1	1	1	1	1	1	1	1	1	1
(v) $(14, 3, 1.33)$	Lattice-based	65	295	1	20	2	1	1	1	1	0	1	0	0	0	0
	Generalized	246	1596	1	82	2	1	1	1	1	0	1	0	0	0	0

which is derived from [Gui20; GMT20]. Replacing the costs in Table 3.11 with m_1, the calculation costs of the final exponentiations are finally estimated as in Table 3.13.

According to Table 3.13, it is found that the lattice-based method [FCKRH11] is a better choice than the generalized method [HHT20] for constructing the algorithm for computing the final exponentiation for the target family of curves. In fact, the final exponentiations given by lattice-based method successfully reduce *16.4%*, *87.2%*, *75.5%*, *89.3%*, and *80.0%* calculation costs from that of the generalized method for the families of curves with (k, D, ρ) such that $(10, 15, 1.75)$, $(11, 3, 1.30)$, $(11, 11, 1.60)$, $(13, 3, 1.17)$, and $(14, 3, 1.33)$, respectively. This is because not only the high degree $T(x)$ but also the complicated polynomial representations of $h_1(z)$ and $h_2(z)$ result in increasing the calculation costs for computing Algorithm 3.1. For such families of curves, it is considered that the `INIT` step in Algorithm 3.1 should be optimized by exploiting $p(z)$-adic representations of $h_1(z)$ and $h_2(z)$.

3.4.5 Performance comparison of the STNFS-secure pairings

The author compares the performance of the STNFS-secure pairings on the candidate curves at the 128-bit security level in [Gui20]. Table 3.14 shows the calculation costs of

Table 3.12: The calculation costs of the arithmetic operations in $\mathbb{F}_{p^k}$ with $k = 10$, 11, 13, and 14.

k	m_k	s_k/s_{ck}	i_k	i_{ck}	f_k^i
10	$39m_1$	$26m_1$	$125m_1$	$8m_1$	$8m_1$
11	$45m_1$	$45m_1$	$332m_1$	$285m_1$	$10m_1$
13	$59m_1$	$59m_1$	$489m_1$	$438m_1$	$12m_1$
14	$66m_1$	$44m_1$	$217m_1$	$12m_1$	$12m_1$

Table 3.13: The calculation costs for computing the final exponentiation of the pairings at the 128-bit security level.

Families (k, D, ρ)	Methods	Costs
(i) $(10, 15, 1.75)$	Lattice-based Generalized	$12921m_1$ $\approx 15464m_1$
(ii) $(11, 3, 1.30)$	Lattice-based Generalized	$32322m_1$ $\approx 252867m_1$
(iii) $(11, 11, 1.60)$	Lattice-based Generalized	$27462m_1$ $\approx 112212m_1$
(iv) $(13, 3, 1.17)$	Lattice-based Generalized	$40970m_1$ $\approx 384188m_1$
(v) $(14, 3, 1.33)$	Lattice-based Generalized	$17811m_1$ $\approx 87745m_1$

the Miller loop, final exponentiation, and total pairing with the time estimation. The calculation costs of the pairings on the Cocks-Pinch curves with $k = 6$ and 8 [GMT20], BN curves with $k = 12$ [BN05], BLS curves with $k = 12$ [BN05], FK curves with $k = 12$ (FK12) [FM19], and KSS curves with $k = 16$ [KSS08] are given by [GMT20; FM19]. For the curves with $k = 10, 11, 13$, and 14, the calculation costs of the Miller loop are given by [Gui20] and these of the final exponentiations are given by this work. Note that the inversions involved in the costs of the Miller loop are replaced with m_1 by this work. For the Cocks-Pinch curve with $k = 6$ and BLS curve with $k = 12$, the calculation costs of the final exponentiation are also reproduced in App. B. The timing is estimated from the $\mathbb{F}_p$-multiplication timing for RELIC [Ara13] on a Intel Core i7-8700 CPU, 3.20GHz with TurboBoost disabled, i.e., $\mathbb{F}_p$-multiplication can be performed in 65ns for $320 < \log_2 p \leq 384$, 85ns for $384 < \log_2 p \leq 448$, 129ns for $512 < \log_2 p \leq 576$, and 181ns for $640 < \log_2 p \leq 704$ (see Table 9 in [GMT20]). Although there is no data for $\log_2 p = 310$, the author assumes that the timing is 65ns.

According to Table 3.14, it is found that the BLS curve with $k = 12$ might be the best choice for the pairings at the 128-bit security level. The second-best candidates are the Cocks-Pinch curve with $k = 6$ and the FK12 curve with $k = 12$. Although the author

Table 3.14: The calculation costs and time estimations for computing the pairings of Miller's algorithm (ML) and final exponentiation (FE) with the curves resistant to the STNFS at the 128-bit security level.

Curves, (k, D, ρ)	$\log_2 p$	ML	FE	Total	Time est.
Cocks-Pinch, $(6, 3, 2.63)$	672	$4601m_1$	$2977m_1$	$7578m_1$	$\approx$ 1.37ms
Cocks-Pinch, $(8, 4, 2.13)$	544	$4502m_1$	$7056m_1$	$11558m_1$	$\approx$ 1.49ms
(i) Cyclo, $(10, 15, 1.75)$	446	$15982m_1$	$12921m_1$	$28903m_1$	$\approx$ 2.46ms
(ii) Cyclo, $(11, 3, 1.30)$	333	$29851m_1$	$32322m_1$	$62173m_1$	$\approx$ 4.04ms
(iii) Cyclo, $(11, 11, 1.60)$	412	$25485m_1$	$27462m_1$	$52947m_1$	$\approx$ 4.50ms
BN, $(12, 3, 1.00)$	446	$11620m_1$	$5349m_1$	$16969m_1$	$\approx$ 1.44ms
BLS, $(12, 3, 1.50)$	446	$7805m_1$	$6161m_1$	$13966m_1$	$\approx$ 1.19ms
FK, $(12, 3, 1.50)$	446	$7853m_1$	$8002m_1$	$15855m_1$	$\approx$ 1.35ms
(iv) Cyclo, $(13, 3, 1.17)$	310	$30897m_1$	$40970m_1$	$71867m_1$	$\approx$ 4.67ms
(v) Cyclo, $(14, 3, 1.33)$	340	$16546m_1$	$17811m_1$	$34357m_1$	$\approx$ 2.23ms
KSS, $(16, 1, 1.25)$	339	$7691m_1$	$18235m_1$	$25926m_1$	$\approx$ 1.69ms

improves the final exponentiation for the curves with $k = 10, 11, 13$, and 14, these curves can not give rise to efficient pairings compared with the other curves. This is because these curves do not have high-degree twists for the efficient Miller loop and also do not have efficient squaring in the cyclotomic subgroups of $\mathbb{F}_{p^k}$ that contribute to speeding up the final exponentiation. To prepare further improvements of the STNFS, it is considered that more optimization techniques for these curves are required.

3.5 A new construction method of the final exponentiation

This section provides a new construction method of the algorithm for computing the hard part of the final exponentiation for the cyclotomic family of pairing-friendly elliptic curves with prime embedding degree k of $k \equiv 1 \pmod 6$.

3.5.1 Cyclotomic family of pairing-friendly curves of k of $k \equiv 1 \pmod 6$

In [FST10], Freeman et al. introduced the cyclotomic families of pairing-friendly elliptic curves with any embedding degree k except for $18 \nmid k$. For $k \equiv 1 \pmod 6$, there is the following parameterization of $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$.

$$\begin{cases} p(x) &=& \frac{1}{3}(x+1)^2(x^{2k}-x^k+1)-x^{2k+1}, \\ r(x) &=& \Phi_{6k}(x), \\ t(x) &=& -x^{k+1}+x+1. \end{cases} \tag{3.68}$$

The above allows us to construct pairing-friendly curves with a prime k that have the advantage in terms of the resistance of the STNFS. In the following, let z be an integer making $p(z)$ and $r(z)$ being primes.

3.5.2 Proposed final exponentiation

For the curves in the cyclotomic family with a prime k of $k \equiv 1 \pmod 6$, the exponent of the final exponentiation is given as follows:

$$\frac{p(z)^k-1}{r(z)} = (p(z)-1)\cdot\left(\frac{\Phi_k(p(z))}{r(z)}\right), \tag{3.69}$$

where $d(z) = \Phi_k(p(z))/r(z) = \sum_{i=0}^{k-1} p(z)^i/r(z)$ the hard part. The author proposes to decompose the hard part $d(z)$ as shown below.

Theorem 3.10. Let n be an integer defined by $n = (k-1)/6$ and $c(z)$ be a polynomial defined as follows:

$$c(z) = 3\Phi_1(z)\Phi_2(z)\Phi_3(z)\sum_{i=0}^{n-1} z^{6i}. \tag{3.70}$$

Then, $d'(z) = c(z)\cdot d(z)$ is represented as follows:

$$d'(z) = \sum_{i=0}^{6n-1}\left(z^{6n}\Phi_6(z) - 3 + \mu_{6n-1-i}(z)\right)\cdot p(z)^i, \tag{3.71}$$

where $\mu_s(z)$ with an integer s such that $0 \le s < 6n$ is a polynomial defined as follows:

$$\mu_s(z) = \begin{cases} -z^s\Phi_6(z) & \text{if } s \equiv 0 \pmod 6, \\ z^{6n+1+s}\Phi_6(z) - z^s\Phi_6(z) - 3z^{s+1} & \text{if } s \equiv 1 \pmod 6, \\ z^{6n+1+s}\Phi_6(z) - 3z^{s+1} & \text{if } s \equiv 2 \pmod 6, \\ z^s\Phi_6(z) & \text{if } s \equiv 3 \pmod 6, \\ -z^{6n+1+s}\Phi_6(z) + z^s\Phi_6(z) + 3z^{s+1} & \text{if } s \equiv 4 \pmod 6, \\ -z^{6n+1+s}\Phi_6(z) + 3z^{s+1} & \text{if } s \equiv 5 \pmod 6. \end{cases} \tag{3.72}$$

Proof of Theorem 3.10. Let us start to modify $d'(z) = c(z)d(z)$ by using the expansion of

$d(z) = \Phi_k(p(z))/r(z)$ by Theorem 3.5 in [HHT20]. In this case of the cyclotomic family for a prime $k = 6n + 1$, since $\phi(k) = 6n$ and $\Phi_k(X)$ is the all one polynomial of degree $6n$, $d'(z)$ can be denoted as

$$d'(z) = \underbrace{c(z)h_1(z)\left(\sum_{i=0}^{6n-1}\sum_{j=0}^{6n-1-i} T(z)^j \cdot p(z)^i\right)}_{=A(z)} + \underbrace{c(z)h_2(z)}_{=B(z)}, \tag{3.73}$$

where $h_1(z) = \frac{1}{3}\Phi_6(z)^2$, $h_2(z) = \sum_{i=0}^{6n} T(z)^i/r(z)$, and $T(z) = -z^{6n+2}+z$. In the following, the first and second terms of $d'(z)$ are referred to $A(z)$ and $B(z)$, respectively.

Modification of $A(z)$

Firstly, the polynomial $A(z)$ of $d'(z)$ are modified as follows: The coefficient of $p(z)^i$ of $A(z)$ is denoted as $c(z)h_1(z)\sum_{j=0}^{s} T(z)^j$ where $s = 6n - 1 - i$. If $s = 0$, the coefficient can be easily obtained by $c(z)h_1(z) = (z^{6n} - 1)\Phi_6(z)$. If $s > 0$, the coefficients can be denoted as

$$c(z)h_1(z)\sum_{j=0}^{s} T(z)^j = -3\sum_{i=0}^{s-1} T(z)^i p(z) + (z^{6n} - T(z)^s)\Phi_6(z) + 3\sum_{i=1}^{s} T(z)^i, \tag{3.74}$$

which can be proven by the injection of s.

Applying the equations to the polynomial $A(z)$,

$$\begin{aligned}
A(z) &= c(z)h_1(z)\sum_{i=0}^{6n-1}\sum_{j=0}^{6n-1-i} T(z)^j p(z)^i \\
&= c(z)h_1(z)p(z)^{6n-1} \\
&\quad + c(z)h_1(z)(T(z)+1)p(z)^{6n-2} \\
&\quad + c(z)h_1(z)(T(z)^2+T(z)+1)p(z)^{6n-3} + \cdots \\
&\quad + c(z)h_1(z)\sum_{j=0}^{6n-1} T(z)^j p(z)^0 \\
&= (z^{6n}\Phi_6(z) - \Phi_6(z))p(z)^{6n-1} \\
&\quad + (-3p(z) - \Phi_6(z)T(z) + z^{6n}\Phi_6(z) + 3T(z))p(z)^{6n-2} \\
&\quad + (-3(T(z)+1)p(z) - \Phi_6(z)T(z)^2 + z^{6n}\Phi_6(z) + 3(T(z)^2+T(z)))p(z)^{6n-3} + \cdots \\
&\quad + \left(-3\sum_{i=0}^{6n-2} T(z)^i p(z) - \Phi_6(z)T(z)^{6n-1}z^{6n}\Phi_6(z) + 3\sum_{i=1}^{6n-1} T(z)^i\right)p(z)^0 \\
&= (z^{6n}\Phi_6(z) - 3 - \Phi_6(z))p(z)^{6n-1} \\
&\quad + (z^{6n}\Phi_6(z) - 3 - \Phi_6(z)T(z))p(z)^{6n-2}
\end{aligned}$$

$$
\begin{aligned}
&+ (z^{6n}\Phi_6(z) - 3 - \Phi_6(z)T(z)^2)p(z)^{6n-3} + \cdots \\
&+ (z^{6n}\Phi_6(z) - 3 - \Phi_6(z)T(z)^{6n-1})p(z)^0 + 3\sum_{i=0}^{6n-1} T(z)^i \\
= &\underbrace{\left(\sum_{i=0}^{6n-1}(z^{6n}\Phi_6(z) - 3)p(z)^i\right)}_{=A_1(z)} + \underbrace{\left(-\Phi_6(z)\sum_{i=0}^{6n-1} T(z)^{6n-1-i}p(z)^i\right)}_{=A_2(z)} + \underbrace{\left(3\sum_{i=0}^{6n-1} T(z)^i\right)}_{=A_3(z)}.
\end{aligned}
\tag{3.75}
$$

The first, second, and third terms of $A(z)$ are referred to as $A_1(z), A_2(z)$, and $A_3(z)$, respectively.

Modification of $A_2(z)$

Then, let us modify $A_2(z)$. The coefficient of $p(z)^i$ of $A_2(z)$ is denoted as $-\Phi_6(z)T(z)^s$ where $s = 6n - 1 - i$. For $s > 0$, $\Phi_6(z)T(z)^s$ can be denoted as follows:

$$
\Phi_6(z)T(z)^s = \alpha_s(z)(p(z) - T(z)) + \beta_s(z), \tag{3.76}
$$

where $\alpha_s(z)$ and $\beta_s(z)$ are polynomials in $\mathbb{Q}[z]$ defined with $\gamma_s(z) \in \mathbb{Q}[z]$ as follows:

$$
\alpha_1(z) = 0, \alpha_s(z) = \alpha_{s-1}(z)T(z) + \gamma_s(z), \tag{3.77}
$$

$$
\beta_s(z) = \begin{cases}
z^s\Phi_6(z) & \text{if } s \equiv 0 \pmod 6, \\
-z^{6n+1+s}\Phi_6(z) + z^s\Phi_6(z) & \text{if } s \equiv 1 \pmod 6, \\
-z^{6n+1+s}\Phi_6(z) & \text{if } s \equiv 2 \pmod 6, \\
-z^s\Phi_6(z) & \text{if } s \equiv 3 \pmod 6, \\
z^{6n+1+s}\Phi_6(z) - z^s\Phi_6(z) & \text{if } s \equiv 4 \pmod 6, \\
z^{6n+1+s}\Phi_6(z) & \text{if } s \equiv 5 \pmod 6,
\end{cases} \tag{3.78}
$$

$$
\gamma_s(z) = \begin{cases}
0 & \text{if } s \equiv 1, 4 \pmod 6, \\
3z^s & \text{if } s \equiv 2, 3 \pmod 6, \\
-3z^s & \text{if } s \equiv 0, 5 \pmod 6.
\end{cases} \tag{3.79}
$$

The correctness of the above equation can be proven by induction on $s' \geq 0$ such that $s = 6s' + i > 0$ for $i \in \{1, 2, 3, 4, 5, 6\}$, however, the details are omitted in this thesis.

Applying the above to $A_2(z)$, there is the following modification.

$$
\begin{aligned}
A_2(z) &= -\Phi_6(z)\sum_{i=0}^{6n-1} T(z)^{6n-1-i}p(z)^i \\
&= -\Phi_6(z)p(z)^{6n-1} \\
&\quad - \beta_1(z)p(z)^{6n-2}
\end{aligned}
$$

$$
\begin{aligned}
&- (\alpha_2(z)(p(z) - T(z)) + \beta_2(z))p(z)^{6n-3}\\
&- (\alpha_3(z)(p(z) - T(z)) + \beta_3(z))p(z)^{6n-4} - \cdots\\
&- (\alpha_{6n-1}(z)(p(z) - T(z)) + \beta_{6n-1}(z))p(z)^{0}\\
= &-\Phi_6(z)p(z)^{6n-1}\\
&- (\beta_1(z) + \alpha_2(z))p(z)^{6n-2}\\
&- (-\alpha_2(z)T(z) + \beta_2(z) + \alpha_3(z))p(z)^{6n-3}\\
&- (-\alpha_3(z)T(z) + \beta_3(z) + \alpha_4(z))p(z)^{6n-4} - \cdots\\
&- (-\alpha_{6n-1}(z)T(z) + \beta_{6n-1}(z))p(z)^{0}\\
= &-\Phi_6(z)p(z)^{6n-1}\\
&- (\beta_1(z) + \alpha_1(z)T(z) + \gamma_2(z))p(z)^{6n-2}\\
&- (-\alpha_2(z)T(z) + \beta_2(z) + \alpha_2(z)T(z) + \gamma_3(z))p(z)^{6n-3}\\
&- (-\alpha_3(z)T(z) + \beta_3(z) + \alpha_3(z)T(z) + \gamma_4(z))p(z)^{6n-4} - \cdots\\
&- (-\alpha_{6n-1}(z)T(z) + \beta_{6n-1}(z))p(z)^{0}\\
= &-\Phi_6(z)p(z)^{6n-1}\\
&- (\beta_1(z) + \gamma_2(z))p(z)^{6n-2}\\
&- (\beta_2(z) + \gamma_3(z))p(z)^{6n-3}\\
&- (\beta_3(z) + \gamma_4(z))p(z)^{6n-4} - \cdots\\
&- (\beta_{6n-1}(z) + \gamma_{6n}(z))p(z)^{0}\\
&+ (\alpha_{6n-1}(z)T(z) + \gamma_{6n}(z))\\
= &\underbrace{\left(\sum_{i=0}^{6n-1} -(\beta_{6n-1-i}(z) + \gamma_{6n-i}(z))p(z)^{i}\right)}_{=A_{21}(z)} + \underbrace{(\alpha_{6n-1}(z)T(z) + \gamma_{6n}(z))}_{=A_{22}(z)},
\end{aligned} \tag{3.80}
$$

where the first and second terms of $A_2(z)$ are referred to $A_{21}(z)$ and $A_{22}(z)$, respectively. From the definition, it is easily found $-(\beta_s(z)+\gamma_{s+1}(z)) = \mu_s(z)$ for $s = 6n-1-i$. Thus, $A_{21}(z)$ can be denoted as follows:

$$A_{21}(z) = \sum_{i=0}^{6n-1} \mu_{6n-1-i}(z)p(z)^{i}. \tag{3.81}$$

Besides, $A_{22}(z)$ can also be denoted as follows:

$$
\begin{aligned}
A_{22}(z) =&\alpha_{6n-1}(z)T(z) + \gamma_{6n}(z) = \sum_{i=0}^{6n-1} \gamma_{6n-i}(z)T(z)^{i}\\
=&3\sum_{i=0}^{n-1} \left(-z^{6n-6i}T(z)^{6i} - z^{6n-(6i+1)}T(z)^{6i+1}\right.
\end{aligned}
$$

$$+z^{6n-(6i+3)}(z)T(z)^{6i+3}+z^{6n-(6i+4)}T(z)^{6i+4}\big). \tag{3.82}$$

Modification of $B(x)$.

The polynomial $B(z)$ can be modified as follows:

$$\begin{aligned}
B(z) &= c(z)h_2(z)\\
&= \frac{c(z)T(z)\sum_{i=0}^{6n-1}T(z)^i+c(z)}{r(z)}\\
&= \frac{(-3r(z)+z^2c(z)+3)\sum_{i=0}^{6n-1}T(z)^i+c(z)}{r(z)}\\
&= \underbrace{-3\sum_{i=0}^{6n-1}T(z)^i}_{=B_1(z)}+\underbrace{\frac{(z^2c(z)+3)\sum_{i=0}^{6n-1}T(z)^i+c(z)}{r(z)}}_{=B_2(z)}.
\end{aligned}\tag{3.83}$$

The first and second terms are referred to as $B_1(z)$ and $B_2(z)$, respectively. From the modifications, it is found $B_1(z)+A_3(z)=0$.

In the following, proof of $B_2(z)+A_{22}(z)=0$ is provided. Since $B_2(z)$ involves denominator $r(z)=\Phi_6(z^{6n+1})/\Phi_6(z)$, it is enough to show $t_1(z)=\Phi_6(z)r(z)B_2(z)$ and $t_2(z)=-\Phi_6(z^{6n+1})A_{22}(z)$ are the same as shown in the below.

$$\begin{aligned}
t_1(z) &= \Phi_6(z)\left((z^2c(z)+3)\sum_{i=0}^{6n-1}T(z)^i+c(z)\right)\\
&= 3(z^{6n+1}-z+1)\sum_{i=0}^{6n-1}T(z)^i+3(z^{6n}-1)\\
&= 3(-T(z)+1)\sum_{i=0}^{6n-1}T(z)^i+3(z^{6n}-1)\\
&= 3(-T^{6n}+1)+3(z^{6n}-1)\\
&= 3(-T^{6n}+z^{6n}).
\end{aligned}\tag{3.84}$$

$$\begin{aligned}
t_2(z) &= 3\Phi_6(z^{6n+1})\sum_{i=0}^{n-1}\big(z^{6n-6i}T(z)^{6i}+z^{6n-(6i+1)}T(z)^{6i+1}\\
&\qquad -z^{6n-(6i+3)}T(z)^{6i+3}-z^{6n-(6i+4)}T(z)^{6i+4}\big)\\
&= 3(T(z)^2-zT(z)+z^2)\\
&\quad\cdot\sum_{i=0}^{n-1}\big(z^{6n-6i-2}T(z)^{6i}+z^{6n-(6i+1)-2}T(z)^{6i+1}\\
&\qquad -z^{6n-(6i+3)-2}(z)T(z)^{6i+3}-z^{6n-(6i+4)-2}T(z)^{6i+4}\big)
\end{aligned}$$

$$\begin{aligned}
&= 3\left(\sum_{i=0}^{n-1} z^{6n-6i-2}T(z)^{6i+2} + \sum_{i=0}^{n-1} z^{6n-6i-3}T(z)^{6i+3}\right.\\
&\qquad - \sum_{i=0}^{n-1} z^{6n-6i-5}(z)T(z)^{6i+5} - \sum_{i=0}^{n-1} z^{6n-6i-6}T(z)^{6i+6}\\
&\qquad - \sum_{i=0}^{n-1} z^{6n-6i-1}T(z)^{6i+1} - \sum_{i=0}^{n-1} z^{6n-6i-2}T(z)^{6i+2}\\
&\qquad + \sum_{i=0}^{n-1} z^{6n-6i-4}(z)T(z)^{6i+4} + \sum_{i=0}^{n-1} z^{6n-6i-5}T(z)^{6i+5}\\
&\qquad + \sum_{i=0}^{n-1} z^{6n-6i}T(z)^{6i} + \sum_{i=0}^{n-1} z^{6n-6i-1}T(z)^{6i+1}\\
&\qquad \left. - \sum_{i=0}^{n-1} z^{6n-6i-3}(z)T(z)^{6i+3} - \sum_{i=0}^{n-1} z^{6n-6i-4}T(z)^{6i+4}\right)\\
&= 3\left(\sum_{i=0}^{n-1} z^{6n-6i}T(z)^{6i} - \sum_{i=0}^{n-1} z^{6n-6i-6}T(z)^{6i+6}\right)\\
&= 3(-T^{6n} + z^{6n}). \qquad (3.85)
\end{aligned}$$

Since $t_1(z) = t_2(z)$, it is obtained that $B_2(z) + A_{22}(z) = 0$.

As a result of the modifications, it is obtained $d'(z) = A(z)+B(z) = A_1(z)+(A_{21}(z)+A_{22}(z))+A_3(z)+B_1(z)+B_2(z)$ with the relations $B_1(z)+A_3(z) = 0$ and $B_2(z)+A_{22}(z) = 0$, i.e., $d'(z) = A_1(z) + A_{21}(z) = \sum_{i=0}^{6n-1}(z^{6n}\Phi_6(z) - 3 + \mu_{6n-1-i}(z))p(z)^i$. □

3.5.3 Application

In this subsection, the author applies Theorem 3.10 and presents the representation of the hard part for the curves with several embedding degrees, e.g., $k = 7, 13$, and 19.

Example 3.11. ($k = 7$) The cyclotomic family of curves with $k = 7$ has the following parameters.

$$\left\{\begin{array}{rcl} p(x) & = & \frac{1}{3}(x+1)^2(x^{14} - x^7 + 1) - x^{15},\\ r(x) & = & \Phi_{42}(x),\\ t(x) & = & -x^8 + x + 1. \end{array}\right. \qquad (3.86)$$

For an integer seed z making $p(z)$ and $r(z)$ being primes, the hard part of the final exponentiation is $d(z) = \Phi_7(p(z))/r(z)$. Applying Theorem 1, it is obtained $n = 1$, $c(z) = 3\Phi_1(z)\Phi_2(z)\Phi_3(z)$, and $d'(z) = \sum_{i=0}^{5} d'_i(z)p(z)^i$ where $d'_i(z) = z^6\Phi_6(z)-3+\mu_{6n-1-i}$

are given as follows:

$$\begin{cases} d'_5(z) &=& z^6\Phi_6(z) - 3 - \Phi_6(z), \\ d'_4(z) &=& z^6\Phi_6(z) - 3 + z^8\Phi_6(z) - z\Phi_6(z) - 3z^2, \\ d'_3(z) &=& z^6\Phi_6(z) - 3 + z^9\Phi_6(z) - 3z^3, \\ d'_2(z) &=& z^6\Phi_6(z) - 3 + z^3\Phi_6(z), \\ d'_1(z) &=& z^6\Phi_6(z) - 3 - z^{11}\Phi_6(z) + z^4\Phi_6(z) + 3z^5, \\ d'_0(z) &=& z^6\Phi_6(z) - 3 - z^{12}\Phi_6(z) + 3z^6. \end{cases} \tag{3.87}$$

Example 3.12. ($k = 13$) The cyclotomic family of curves with $k = 13$ has the following parameters.

$$\begin{cases} p(x) &=& \frac{1}{3}(x+1)^2(x^{26} - x^{13} + 1) - x^{27}, \\ r(x) &=& \Phi_{78}(x), \\ t(x) &=& -x^{14} + x + 1. \end{cases} \tag{3.88}$$

Then, for an integer seed z making $p(z)$ and $r(z)$ being primes, the exponent of the hard part is $d(z) = \Phi_{13}(p(z))/r(z)$. Applying Theorem 1, it is obtained $n = 2$, $c(z) = 3\Phi_1(z)\Phi_2(z)\Phi_3(z)(z^6+1)$, and $d'(z) = \sum_{i=0}^{11} d'_i(z)p(z)^i$ where $d'_i(z) = z^{12}\Phi_6(z) - 3 + \mu_{6n-1-i}$ are given as follows:

$$\begin{cases} d'_{11}(z) &=& z^{12}\Phi_6(z) - 3 - \Phi_6(z), \\ d'_{10}(z) &=& z^{12}\Phi_6(z) - 3 + z^{14}\Phi_6(z) - z\Phi_6(z) - 3z^2, \\ d'_9(z) &=& z^{12}\Phi_6(z) - 3 + z^{15}\Phi_6(z) - 3z^3, \\ d'_8(z) &=& z^{12}\Phi_6(z) - 3 + z^3\Phi_6(z), \\ d'_7(z) &=& z^{12}\Phi_6(z) - 3 - z^{17}\Phi_6(z) + z^4\Phi_6(z) + 3z^5, \\ d'_6(z) &=& z^{12}\Phi_6(z) - 3 - z^{18}\Phi_6(z) + 3z^6, \\ d'_5(z) &=& z^{12}\Phi_6(z) - 3 - z^6\Phi_6(z), \\ d'_4(z) &=& z^{12}\Phi_6(z) - 3 + z^{20}\Phi_6(z) - z^7\Phi_6(z) - 3z^8, \\ d'_3(z) &=& z^{12}\Phi_6(z) - 3 + z^{21}\Phi_6(z) - 3z^9, \\ d'_2(z) &=& z^{12}\Phi_6(z) - 3 + z^9\Phi_6(z), \\ d'_1(z) &=& z^{12}\Phi_6(z) - 3 - z^{23}\Phi_6(z) + z^{10}\Phi_6(z) + 3z^{11}, \\ d'_0(z) &=& z^{12}\Phi_6(z) - 3 - z^{24}\Phi_6(z) + 3z^{12}. \end{cases} \tag{3.89}$$

Example 3.13. ($k = 19$) The cyclotomic family of curves with $k = 19$ has the following parameters.

$$\begin{cases} p(x) &=& \frac{1}{3}(x+1)^2(x^{38} - x^{19} + 1) - x^{39}, \\ r(x) &=& \Phi_{114}(x), \\ t(x) &=& -x^{20} + x + 1. \end{cases} \tag{3.90}$$

In this case, for an integer seed z making $p(z)$ and $r(z)$ being primes, the exponent of the hard part is $d(z) = \Phi_{19}(p(z))/r(z)$. Applying Theorem 1, it is obtained $n = 3$, $c(x) = 3\Phi_1(x)\Phi_2(x)\Phi_3(x)(x^{12} + x^6 + 1)$, and $d'(x) = \sum_{i=0}^{19}(x^{18}\Phi_6(x) - 3 + \mu_{6n-1-i})p(x)^i$ where μ_{6n-1-i} are given as follows: Applying Theorem 1, it is obtained $n = 3$, $c(z) = 3\Phi_1(z)\Phi_2(z)\Phi_3(z)(z^{12} + z^6 + 1)$, and $d'(z) = \sum_{i=0}^{19}(z^{18}\Phi_6(z) - 3 + \mu_{6n-1-i})p(z)^i$ where μ_{6n-1-i} are given as follows:

$$\begin{cases} d'_{17}(z) &= z^{18}\Phi_6(z) - 3 - \Phi_6(z), \\ d'_{16}(z) &= z^{18}\Phi_6(z) - 3 + z^{20}\Phi_6(z) - z\Phi_6(z) - 3z^2, \\ d'_{15}(z) &= z^{18}\Phi_6(z) - 3 + z^{21}\Phi_6(z) - 3z^3, \\ d'_{14}(z) &= z^{18}\Phi_6(z) - 3 + z^3\Phi_6(z), \\ d'_{13}(z) &= z^{18}\Phi_6(z) - 3 - z^{23}\Phi_6(z) + z^4\Phi_6(z) + 3z^5, \\ d'_{12}(z) &= z^{18}\Phi_6(z) - 3 - z^{24}\Phi_6(z) + 3z^6, \\ d'_{11}(z) &= z^{18}\Phi_6(z) - 3 - z^6\Phi_6(z), \\ d'_{10}(z) &= z^{18}\Phi_6(z) - 3 + z^{26}\Phi_6(z) - z^7\Phi_6(z) - 3z^8, \\ d'_9(z) &= z^{18}\Phi_6(z) - 3 + z^{27}\Phi_6(z) - 3z^9, \\ d'_8(z) &= z^{18}\Phi_6(z) - 3 + z^9\Phi_6(z), \\ d'_7(z) &= z^{18}\Phi_6(z) - 3 - z^{29}\Phi_6(z) + z^{10}\Phi_6(z) + 3z^{11}, \\ d'_6(z) &= z^{18}\Phi_6(z) - 3 - z^{30}\Phi_6(z) + 3z^{12}, \\ d'_5(z) &= z^{18}\Phi_6(z) - 3 - z^{12}\Phi_6(z), \\ d'_4(z) &= z^{18}\Phi_6(z) - 3 + z^{32}\Phi_6(z) - z^{13}\Phi_6(z) - 3z^{14}, \\ d'_3(z) &= z^{18}\Phi_6(z) - 3 + z^{33}\Phi_6(z) - 3z^{15}, \\ d'_2(z) &= z^{18}\Phi_6(z) - 3 + z^{15}\Phi_6(z), \\ d'_1(z) &= z^{18}\Phi_6(z) - 3 - z^{35}\Phi_6(z) + z^{16}\Phi_6(z) + 3z^{17}, \\ d'_0(z) &= z^{18}\Phi_6(z) - 3 - z^{36}\Phi_6(z) + 3z^{18}. \end{cases} \tag{3.91}$$

For the curves with $k = 7$, 13, and 19, the multiple $d'(z)$ and its decomposition are exactly one of the same representations given by the lattice-based method [FCKRH11]. For curves with arbitrary prime k of $k \equiv 1 \pmod 6$, there is a possibility that the proposed method gives rise to one of the same results as [FCKRH11]. Besides, since the decomposition of $d'(z)$ has systematic relations between the coefficients $d'_i(z)$ which consists of f^{z^i} for $0 \leq i \leq 6n$ and $f^{z^i\Phi_6(z)}$ for $0 \leq i \leq 12n$, one can construct an efficient algorithm for computing $f \mapsto f^{d'(z)}$ in Algorithm 3.3. The details of the steps in Algorithm 3.3 are summarized below.

- Steps 1–3 compute $f_i = f^{z^i}$ for $1 \leq i \leq 6n$, which take $6nu_k^z$.
- Steps 4–8 compute $g_i = f^{z^i\Phi_6(z)}$ for $0 \leq i \leq 6n - 2$ from f_i, which require $n(6m_k + 4i_{ck}$.
- Steps 9–11 also compute $g_i = f^{z^i\Phi_6(z)}$ for $6n - 1 \leq i \leq 12n$, which take $(6n + 2)u_k^z$.
- Steps 12–19 compute $v_{6i+j} = f^{z^{6n}\Phi_6(z) - 3 + \mu_{6n-1-(6i+j)}(z)}$ for $0 \leq i \leq n-1$ and $0 \leq j \leq 5$

from the knowledge of the f_i and g_i, which take $i_{G_{\Phi_k(p)}} + c_k + n(18m_k + 4c_k + 6i_{ck}$.

- Steps 20–22 compute $w = \sum_{i=0}^{6n-1} v_i^{p(z)^i}$, which require $(6n-1)m_k + \sum_{i=1}^{6n-1} f_k^i$.

Algorithm 3.3: Proposed hard part computation for the family of curves with prime k of $k \equiv 1 \pmod 6$.

```
   Input: $f \in G_{\Phi_k(p(z))}$
   Output: $f^{d'(z)} \in \mu_{r(z)}$
 1 $f_0 \leftarrow f$;
 2 for $i = 1$ to $6n$ do
 3     $f_i \leftarrow f_{i-1}^z$;
 4 endfor
 5 For $i = 0$ to $n-1$ do
 6     $t_1 \leftarrow f_{6i+5} \cdot f_{6i+4}^{-1}$, $t_2 \leftarrow t_1^{-1}$;
 7     $g_{6i+3} \leftarrow t_1 \cdot f_{6i+3}$, $g_{6i+4} \leftarrow t_2 \cdot f_{6i+6}$;
 8     $t_1 \leftarrow f_{6i+2} \cdot f_{6i+1}^{-1}$, $t_2 \leftarrow t_1^{-1}$;
 9     $g_{6i} \leftarrow t_1 \cdot f_{6i}$, $g_{6i+1} \leftarrow t_2 \cdot f_{6i+3}$;
10 endfor
11 $g_{6n} \leftarrow g_{6n-2}^{z^2}$;
12 for $i = 1$ to $6n$ do
13     $g_{6n+i} \leftarrow g_{6n+i-1}^z$;
14 endfor
15 $t_1 \leftarrow f^{-1}$, $t_2 \leftarrow t_1^3$
16 for $i = 0$ to $n-1$ do
17     $v_{6i+5} \leftarrow g_{6n} \cdot g_{6n-1-(6i+5)}^{-1} \cdot t_2$;
18     $v_{6i+4} \leftarrow g_{6n} \cdot g_{12n-(6i+4)} \cdot g_{6n-1-(6i+4)}^{-1} \cdot (f_{6n-(6i+4)} \cdot f)^{-3}$;
19     $v_{6i+3} \leftarrow g_{6n} \cdot g_{12n-(6i+3)} \cdot (f_{6n-(6i+3)} \cdot f)^{-3}$;
20     $v_{6i+2} \leftarrow g_{6n} \cdot g_{6n-1-(6i+2)} \cdot t_2$;
21     $v_{6i+1} \leftarrow g_{6n} \cdot g_{12n-(6i+1)}^{-1} \cdot g_{6n-1-(6i+1)} \cdot (f_{6n-(6i+1)} \cdot t_1)^3$;
22     $v_{6i} \leftarrow g_{6n} \cdot g_{12n-6i}^{-1} \cdot (f_{6n-6i} \cdot t_1)^3$;
23 endfor
24 $w \leftarrow v_0$;
25 for $i = 1$ to $6n-1$
26     $w \leftarrow w \cdot v_i^{p(z)^i}$;
27 endfor
   return $w$;
```

Thus, the total of the calculation costs for executing the algorithm is given by $(12n+2)u_k^z + (30n-1)m_k + (4n+1)c_k + (10n+1)i_{ck} + \sum_{i=1}^{6n-1} f_k^i$ in this case. In the target family of curves, the proposed algorithm is considered to be more efficient than that of the generalized method [HHT20]; This is because Algorithm 3.1 requires at least $(k'-1)\deg Tu_k^z = (6n-1)(6n+2)u_k^z = (36n^2 + 6n - 2)u_k^z$ in the `EVAL` step.

3.6 Summary of contributions

This section presents the works related to optimization of the final exponentiation of the pairings. The major contributions are summarized as follows:

- The author improves the final exponentiation for the pairing on the BLS curves with $k = 15$ by using the property of the characteristic of the BLS family, which is also used by Zhang et al. in [ZL12]. For the pairing at the 128-bit security level, the proposed method contributes to reducing the calculation cost $312m_1$ from the previous final exponentiation given by Fouotsa et al. in [FMP20]. It is also found that the decomposition method can be extended for the BLS curves with any k. At the same time as this publication, Hayashida et al. generalized Zhang et al.'s method for any family of curves in [HHT20], however, the result is still in the state-of-the-art for the BLS curves with $k = 15$.

- For the families of curves with $k = 10, 11, 13$, and 14 resistant to the STNFS, the author presents the final exponentiation computations that are constructed by applying the lattice-based method [FCKRH11] and generalized method [HHT20]. Comparing the calculation costs of the final exponentiations between two methods, it is found that the lattice-based method results in notable reducing the calculation costs. However, comparing the calculation costs of the STNFS-secure pairings between the shortlist curves in [Gui20], it is found that the curves with $k = 10, 11, 13$, and 14 are not efficient choices for the pairings at the 128-bit security level. As one of the future works, the author would like to achieve more optimizations for these curves, e.g., optimizations of the arithmetic operations in the cyclotomic subgroup of the full extension field for these curves.

- The author proposes a new decomposition method of the hard part for the cyclotomic family of pairing-friendly curves with any prime k of $k \equiv 1 \pmod 6$. It is found that the proposed method results in one of the same state-of-the-art algorithms for computing the hard part given by the lattice-based method for the cases of $k = 7$, 13, and 19. Unlike the lattice-based method, the proposed method can easily reach the same result. Moreover, the proposed hard part takes approximately $(12n + 2)u_k^z$, however, that of the generalized method [HHT20] requires at least $(36n^2 + 6n - 2)u_k^z$. As one of the future works, the author would like to obtain similar results for the other families of curves.

Chapter 4

Attractive Subfamilies of Pairing-friendly Curves for Fast Pairings

The algorithms for computing the final exponentiation of the pairings are optimized in Chapter 3. To achieve more efficient pairings, it is necessary to consider the efficiency of elliptic curves and finite fields in which the pairings are defined. This chapter describes research for generating curves and finite fields that have advantages for the pairings, which is introduced as the second work in Sect. 1.3. This chapter starts to review the background and motivation.

4.1 Background and motivation

The family of pairing-friendly curves with fixed embedding degree k are parameterized by polynomials $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$. The pairings with the family require several initial settings such that finding an integer parameter z and constructing a field, curve, and its correct twist corresponding to z. In the settings, it is needed to consider not only the security of pairing but also the efficiency of pairing computation since it strongly depends on the field construction, curve equation, and twisting or untwisting isomorphism. However, since it is typically complicated to handle the favorite constructions, it is desired to establish some convenient ways for the settings which have advantages for the pairings. The author would like to overcome this problem.

There are several related works that focused on the specific families of curves which offer advantages with respect to some aspects of a pairing computation [Shi10; Per+11; CLN11; Cos12; YTS15]. Particularly, in [Per+11], Pereira et al. firstly consider the problem for the BN family. In [CLN11], Costello et al. were motivated by [Per+11] and provided attractive subfamilies of the BLS family with $k = 24$ which guarantee the implementation-friendly field and curve equations. The important fact is that the subfamilies are simply generated by finding z satisfying certain restrictions. In short, once finding z, we can automatically have favorite constructions of field and curve. After this publication, Costello treated the other eight stand-out candidates' families of curves with $8 \leq k < 50$ and point out attractive subfamilies of each in [Cos12].

This work is summarized below.

- Although there are previous works for the BN family [Shi10; Per+11; YTS15], it is still ambiguous about the generation of the BN subfamilies. Thus, the author refers to [CLN11] and explicitly provides restrictions of z for generating the attractive subfamilies of the BN family. The author shows sample seeds z for generating concrete curves and confirms the performance of the pairings by an implementation. The efficiency of the untwist isomorphisms for the pairings is also observed.

- According to recent works, the BLS family is often used for the pairings at the various security levels rather than the BN family. Since the BLS family has high flexibility of k and can strongly support optimizing the pairings, it will be regularly adopted for the pairings even if there is progress in the security analyses in the future. Thus, the author extends [CLN11] and provides restrictions for finding z that can generate the specific BLS subfamilies with more generalized embedding degrees $k = 2^m \cdot 3$ and 3^n for any integers $m, n > 0$. For the BLS family of curves with $k = 9, 12, 24,$ and 27, the author provides sample seeds z for generating concrete curves for the pairings at the 128- and 192-bit security levels. The pairings with the proposed curves are also evaluated by an implementation.

Organization. Sect. 4.2 reviews the previous work and provides mathematical descriptions. The main results of the attractive subfamilies of the BN and BLS families with $k = 2^m \cdot 3$ and 3^n are proposed in Sects. 4.3 and 4.4, respectively. Sect. 4.5 summarizes the major contributions in this chapter.

4.2 Related works and mathematical materials

This section reviews the related works for the BLS family of curves with $k = 24$ given by Costello et al. in [CLN11]. This section also provides the mathematical preliminaries for generating attractive subfamilies of curves which are referred to in [CLN11].

4.2.1 Related works

There are several related works for the determination of the construction of the tower of extension fields and curve equations in [Shi10; Per+11; CLN11; Cos12; YTS15]. In

[CLN11], Costello et al. proposed the restrictions of z for generating specific subfamilies of the BLS family of curves with $k = 24$, $D = 3$, and $\rho = 1.25$, which facilitate efficient instantiations of the pairings. In this context, the BLS family of curves with $k = 24$ has the following parameterizations.

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot r(x) + x, \\ r(x) &= \Phi_{24}(x) = x^8 - x^4 + 1, \\ t(x) &= x + 1. \end{cases} \tag{4.1}$$

Let us find an integer seed z making $p(z)$ and $r(z)$ being primes and $t(z)$ being an integer satisfying the condition given as follows:

$$z \equiv 7, 16, 31, 64 \pmod{72}. \tag{4.2}$$

Then, we have the specific subfamilies of the BLS family with attractive options, i.e.,

(i) A fixed tower of extension fields with one of the best performing arithmetics is always available;

(ii) The BLS curve $E/\mathbb{F}_{p(z)}$ is immediately determined;

(iii) The correct twist $E'/\mathbb{F}_{p(z)^4}$ of degree 6 of E is also immediately determined.

The details of the options are found in the following theorems.

Theorem 4.1. If z satisfies Eq. (4.2), the following tower of extension fields is always available.

$$\begin{cases} \mathbb{F}_{p(z)^2} & \cong \mathbb{F}_{p(z)}[x]/(x^2+1) & \cong \mathbb{F}_{p(z)}(\alpha), \\ \mathbb{F}_{p(z)^4} & \cong \mathbb{F}_{p(z)^2}[x]/(x^2+(\alpha+1)) & \cong \mathbb{F}_{p(z)^2}(\beta), \\ \mathbb{F}_{p(z)^{24}} & \cong \mathbb{F}_{p(z)^4}[x]/(x^6+\beta) & \cong \mathbb{F}_{p(z)^4}(\gamma), \end{cases} \tag{4.3}$$

where α, β, and γ are elements in $\mathbb{F}_{p(z)^2}$, $\mathbb{F}_{p(z)^4}$, and $\mathbb{F}_{p(z)^{24}}$ such that $\alpha^2 = -1$, $\beta^2 = -(\alpha+1)$, and $\gamma^6 = -\beta$, respectively.

Theorem 4.2. If z satisfies Eq. (4.2), the BLS curve $E/\mathbb{F}_{p(z)}$ is determined by

$$E/\mathbb{F}_{p(z)} : \begin{cases} y^2 = x^3 + 1, & \text{if } z \equiv 7, 31 \pmod{72}, \\ y^2 = x^3 + 4, & \text{if } z \equiv 16 \pmod{72}, \\ y^2 = x^3 - 2, & \text{if } z \equiv 64 \pmod{72}. \end{cases} \tag{4.4}$$

Theorem 4.3. Suppose that the tower of extension fields of degree $k = 24$ and BLS curve $E/\mathbb{F}_{p(z)}$ are constructed as in Theorems 4.1 and 4.2 with z satisfying Eq. (4.2). Then, the correct twist $E'/\mathbb{F}_{p(z)^4}$ of degree 6 of E is determined by

$$E'/\mathbb{F}_{p(z)^4} : \begin{cases} y^2 = x^3 \pm 1/\beta, & \text{if } z \equiv 7 \pmod{72}, \\ y^2 = x^3 \pm 4\beta, & \text{if } z \equiv 16 \pmod{72}, \\ y^2 = x^3 \pm \beta, & \text{if } z \equiv 31 \pmod{72}, \\ y^2 = x^3 \pm 2/\beta, & \text{if } z \equiv 64 \pmod{72}, \end{cases} \tag{4.5}$$

where β is an element in $\mathbb{F}_{p(z)^4}$ such that $\beta^2 = -(\alpha + 1)$.

Proof of Theorems 4.1, 4.2, and 4.3. Please refer to [CLN11]. □

The field and curve options can reduce the time-consuming pre-computation of the curve constructions. Moreover, the fixed constructions give rise to the flexibility of scaling the size of the parameters without changing any of the implementations for the field and curve arithmetics. In [Cos12], Costello also treated the other eight stand-out candidates such that the Brezing-Weng family with $k = 8$, the BLS families with $k = 12$, 27, and 48, and the KSS families with $k = 16$, 18, 32, and 36 for pairing implementations and point out highly attractive subfamilies of each.

4.2.2 Mathematical materials

This subsection briefly describes the construction method of extension fields and the determination method of curve equations used for proof of the theorems in [CLN11].

The construction method of the tower of the extension field

Let p be a prime and $q = p^n$ with an integer $n > 0$. To admit an extension field $\mathbb{F}_{q^m}$ of degree m of $\mathbb{F}_q$ defined as $\mathbb{F}_{q^m} = \mathbb{F}_q[x]/(x^m - \zeta)$ with $\zeta \in \mathbb{F}_q$, it is known that the binomial $x^m - \zeta$ must be irreducible in $\mathbb{F}_q[x]$. According to [BS10], the irreducibility of the binomial can be verified as follows:

Lemma 4.4. The binomial $x^m - \zeta$ is irreducible in $\mathbb{F}_q[x]$ if the following two conditions are satisfied.

(a) Each prime factor d of m divides $(p-1)$ and the Norm of ζ, i.e., $N_{\mathbb{F}_q/\mathbb{F}_p}(\zeta)$, is d-th non-residue in $\mathbb{F}_p^*$.

(b) If $m \equiv 0 \pmod 4$, then $q \equiv 1 \pmod 4$.

Proof of Lemma 4.4. Please refer to [BS10]. □

In [BS10], Benger and Scott described that a condition of p for constructing a fixed extension field of degree $k = 2^m \cdot 3^n$ for $n, m > 0$ can be easily obtained by applying Lemma 4.4 since the quadratic and cubic residue properties of the specific element in $\mathbb{F}_p^*$

can be obtained by Lemmas 2.25 and 2.28. As examples, they provided conditions for constructing some implementation-friendly towers of extension fields for the BN and KSS families of curves with $k = 12$ and 18, respectively. With the same strategy, Costello et al. reached the condition of the integer parameter z for constructing the tower of extension fields as shown in Theorem 4.1.

Determination method of the curve equations

Let p be a prime such that $p \equiv 1 \pmod 6$ and let $q = p^n$ with an integer $n > 0$. Let $E/\mathbb{F}_q$ be an ordinary elliptic curve defined over $\mathbb{F}_q$ given by $y^2 = x^3+b$. Then, all possible number $\#E(\mathbb{F}_q)$ of rational points of $E(\mathbb{F}_q)$ can be obtained by taking $b \in \{1, g, g^2, g^3, g^4, g^5\}$ where g is quadratic and cubic non-residue in $\mathbb{F}_q^*$. In fact, there are only six possibilities of $\#E(\mathbb{F}_q) = n_i$ for $0 \leq i \leq 5$:

$$\begin{cases} n_0 & = q + 1 - t, \\ n_1 & = q + 1 - \frac{t-3V}{2}, \\ n_2 & = q + 1 - \frac{-t-3V}{2}, \\ n_3 & = q + 1 + t, \\ n_4 & = q + 1 - \frac{-t+3V}{2}, \\ n_5 & = q + 1 - \frac{t+3V}{2}, \end{cases} \tag{4.6}$$

where t and V are integers satisfying $3V^2 = 4q - t^2$. Therefore, the curve E with the specific order can be obtained by a randomly chosen b with a probability of 1/6.

Let $E'/\mathbb{F}_q$ be a twist of degree d of E. Since $j(E) = 0$, there are only the possibilities $d \in \{1, 2, 3, 6\}$. The curve equation of $E'/\mathbb{F}_q$ is explicitly given as $y^2 = x^3 + b/\delta$ where δ is an element in $\mathbb{F}_q^*$ having the specific properties:

$$\delta \text{ is} \begin{cases} \text{quadratic and cubic residue in } \mathbb{F}_q^* & \text{if } d = 1, \\ \text{quadratic non-residue and cubic residue in } \mathbb{F}_q^* & \text{if } d = 2, \\ \text{quadratic residue and cubic non-residue in } \mathbb{F}_q^* & \text{if } d = 3, \\ \text{quadratic and cubic non-residue in } \mathbb{F}_q^* & \text{if } d = 6. \end{cases} \tag{4.7}$$

Thus, once E is determined, the possibilities of finding the twist E' of degree $d = \{1, 2\}$ and $\{3, 6\}$ of E are 1 and 1/2, respectively. According to Theorem 2.52 given by [HSV06], if $\#E(\mathbb{F}_q) = n_0$, the possible group orders $\#E'(\mathbb{F}_q)$ are also determined by

$$\#E'(\mathbb{F}_q) = \begin{cases} n_0 & \text{if } d = 1, \\ n_3 & \text{if } d = 2, \\ n_2, n_4 & \text{if } d = 3, \\ n_1, n_5 & \text{if } d = 6. \end{cases} \tag{4.8}$$

The curve equations can be determined or narrowed down by checking the small cofactors of $\#E(\mathbb{F}_q)$ by using the following Lemma 4.5. Note that (a) and (b) in Lemma 4.5 are found by [CLN11] (similar lemmas can also be found in [Per+11]), and (c) is found by the author. The following shows the complete proof of Lemma 4.5.

Lemma 4.5. Let E be an ordinary elliptic curve with $D = 3$ defined over $\mathbb{F}_q$, where $q = p^n$ with an integer $n > 0$ and p is an odd prime such that $p \equiv 1 \pmod 6$. Then, the following is true.

(a) If and only if $2 \mid \#E(\mathbb{F}_q)$, b is cubic residue in $\mathbb{F}_q^*$.

(b) If and only if $3 \mid \#E(\mathbb{F}_q)$ and $9 \nmid \#E(\mathbb{F}_q)$, b is quadratic residue in $\mathbb{F}_q^*$ and $4b$ is cubic non-residue in $\mathbb{F}_q^*$.

(c) If and only if $9 \mid \#E(\mathbb{F}_q)$, b is quadratic residue in $\mathbb{F}_q^*$ and $4b$ is cubic residue in $\mathbb{F}_q^*$.

Proof of Lemma 4.5. (a): If $2 \mid \#E(\mathbb{F}_q)$, $E(\mathbb{F}_q)$ involves points of order 2 given as $P_2 = (-\sqrt[3]{b}, 0)$, which is not equal to $\mathcal{O}$. Thus, b is cubic residue in $\mathbb{F}_q^*$.

(b): If $3 \mid \#E(\mathbb{F}_q)$, $E(\mathbb{F}_q)$ involves a subgroup or subgroups of $E(\mathbb{F}_q)$ of order 3, i.e., there exists a group structure given as $E(\mathbb{F}_q)[3] \cong \mathbb{Z}/3\mathbb{Z}$ or $\mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$, which consists points of order 3 given as $P_3 = (0, \sqrt{b})$ or both P_3 and $P_3' = (-\sqrt[3]{4b}, \sqrt{-3} \cdot \sqrt{b})$. Note that $\sqrt{-3} \in \mathbb{F}_q^*$ from (c) in Lemma 2.25. If $3 \mid \#E(\mathbb{F}_q)$ and $9 \nmid \#E(\mathbb{F}_q)$, then $E(\mathbb{F}_q)$ has a group structure of $E(\mathbb{F}_q)[3] \cong \mathbb{Z}/3\mathbb{Z}$ but does not have $\mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$. This means P_3 is in $E(\mathbb{F}_q)$ but P_3' is not in $E(\mathbb{F}_q)$. Therefore, it is found that b is quadratic residue in $\mathbb{F}_q^*$ and $4b$ is cubic non-residue in $\mathbb{F}_q^*$.

(c): If $9 \mid \#E(\mathbb{F}_q)$, $E(\mathbb{F}_q)$ involves either $E(\mathbb{F}_q)[9] \cong \mathbb{Z}/9\mathbb{Z}$ or $E(\mathbb{F}_q)[3] \cong \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$. Indeed, $E(\mathbb{F}_q)$ does not have $E(\mathbb{F}_q)[9] \cong \mathbb{Z}/9\mathbb{Z}$ but has $E(\mathbb{F}_q)[3] \cong \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ for the following reasons.

(i) In this case of q, 9 does not divide the possible group orders except for $\#E(\mathbb{F}_q)$. This can be easily found by checking the values of the possible group orders modulo 9 with the possible q, t, and V satisfying $3V^2 = 4q - t^2$.

(ii) There exists an ordinary elliptic curve given as $\tilde{E}/\mathbb{F}_q : y^2 = x^3 + \tilde{b}$ defined over $\mathbb{F}_q$ having a group order of multiple of 9 with the group structure $\tilde{E}(\mathbb{F}_q)[3] \cong \mathbb{Z}/3\mathbb{Z} \times \mathbb{Z}/3\mathbb{Z}$ since there exactly exists $\tilde{b}$ in $\mathbb{F}_q^*$ which gives rise to points of order 3 denoted as $\tilde{P}_3 = (0, \sqrt{\tilde{b}})$ and $\tilde{P}_3' = (-\sqrt[3]{4\tilde{b}}, \sqrt{-3} \cdot \sqrt{\tilde{b}})$ in $\tilde{E}(\mathbb{F}_q)$.

The above means that E is isomorphic to $\tilde{E}$ over $\mathbb{F}_q$. Thus, there exist points P_3 and P_3' in $E(\mathbb{F}_q)$ and b is quadratic residue in $\mathbb{F}_q^*$ and $4b$ is cubic residue in $\mathbb{F}_q^*$. □

In [CLN11], Costello et al. applied (a) and (b) of Lemma 4.5 for the BLS family of curves with $k = 24$ and completely determined the curve equations as found in Theorems 4.2 and 4.3. There is a possibility that this strategy is also available for the BLS family of curves with the other k.

4.3 Proposed BN subfamilies

In this section, the author provides the proposed subfamilies of the BN family of curves which gives rise to a fixed tower of extension field and curve equation. The mathematical proof to reach the results are also described.

4.3.1 Review of the pairings with the BN family

The BN family of curves with $k = 12$, $D = 3$, and $\rho = 1$ has the specific parameterization of $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$. Since the BN family is complete, there is a polynomial $V(x)$ in $\mathbb{Q}[x]$ such that $3V(x)^2 = 4p(x) - t(x)^2$ The polynomials $p(x)$, $r(x)$, $t(x)$, and $V(x)$ are given as follows [BN05]:

$$\begin{cases} p(x) &= 36x^4 + 36x^3 + 24x^2 + 6x + 1, \\ r(x) &= 36x^4 + 36x^3 + 18x^2 + 6x + 1, \\ t(x) &= 6x^2 + 1, \\ V(x) &= 6x^2 + 4x + 1. \end{cases} \tag{4.9}$$

Let z be an integer seed making $p(z)$ and $r(z)$ being primes and $t(z)$ and $V(z)$ being integers. One can find an elliptic curve $E/\mathbb{F}_{p(z)} : y^2 = x^3 + b$ such that the group order is given by $n(z) = \#E(\mathbb{F}_{p(z)}) = p(z) + 1 - t(z) = r(z)$, which we say E is the BN curve. Besides, one can also find a correct twist $E'/\mathbb{F}_{p(z)^2} : y^2 = x^3 + b'$ of degree 6 of E such that $r(z) \mid n'(z) = \#E'(\mathbb{F}_{p(z)^2})$ which results in a twisting isomorphism $\phi_6 : E' \to E$ defined over $\mathbb{F}_{p(z)^{12}}$.

Let $\mu_{r(z)}$ be the multiplicative subgroup of $\mathbb{F}_{p(z)^k}$ of order $r(z)$ consisting of the $r(z)$-th root of identity. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the base-field and trace-zero groups of $r(z)$-torsion subgroup $E[r(z)]$, respectively. Since $T(z) = t(z) - 1 = 6z^2$, the standard ate pairing on the BN curve is defined as follows:

$$\begin{aligned} e_{a_T} &: \mathcal{G}_2 \times \mathcal{G}_1 \to \mu_{r(z)}, \\ e_{a_T}(Q, P) &= f_{6z^2,Q}(P)^{\frac{p(z)^{12}-1}{r(z)}}. \end{aligned} \tag{4.10}$$

The ate pairing requires $\log_2 6z^2$ iterations of Miller's algorithm for computing $f_{6z^2,Q}(P)$. To reduce the number of iterations, it is often adopted an ate-like pairing by taking

$p(z)^3 - p(z)^2 + p(z) + 6z + 2 \equiv 0 \pmod{r(z)}$ as follows [Ver09]:

$$\begin{aligned} e_{o_{c_i}} &: \mathcal{G}_2 \times \mathcal{G}_1 \to \mu_r, \\ e_{o_{c_i}}(Q,P) &= \Big(f_{6z+2,Q}(P) \cdot f_{1,Q}(P)^{p(z)} \cdot f_{-1,Q}(P)^{p(z)^2} \cdot f_{1,Q}(P)^{p(z)^3} \\ &\cdot \frac{l_{(p(z)^3-p(z)^2+p(z))Q,(6z+2)Q}(P)}{v_{p(z)^3-p(z)^2+p(z)+6z+2}(P)} \cdot \frac{l_{(p(z)^3-p(z)^2)Q,p(z)Q}(P)}{v_{(p(z)^3-p(z)^2+p(z))Q}(P)} \cdot \frac{l_{p(z)^3Q,-p(z)^2Q}(P)}{v_{(p(z)^3-p(z)^2)Q}(P)} \Big)^{\frac{p(z)^{12}-1}{r(z)}}. \end{aligned}$$

When discarding the elements which disappear in the final exponentiation and modifying the line functions, we have

$$\begin{aligned} e_{o_{c_i}}(Q,P) = \Big(& f_{6z+2,Q}(P) \cdot l_{(6z+2)Q,\pi_{p(z)}(Q)}(P) \\ & \cdot l_{(6z+2)Q+\pi_{p(z)}(Q),-\pi^2_{p(z)}(Q)}(P) \Big)^{\frac{p(z)^{12}-1}{r(z)}}. \end{aligned} \tag{4.11}$$

This pairing requires one of the shortest $\log_2(6z+2)$ iterations of Miller's algorithm for computing $f_{6z+2,Q}(P)$ since $\log_2 r(z)/\phi(12) = \log_2 r(z)/4 \approx \log_2 x$. In [Nog+08], Nogami et al. also provided another ate-like pairing defined by

$$\begin{aligned} e_{o_{c_i}}(Q,P) = \bigg(& \Big(f_{z,Q}(P)^{p^3+1} \cdot l_{zQ,\pi^3_{p(z)}(zQ)}(P) \Big)^{p(z)^{10}+1} \\ & \cdot l_{zQ+\pi^3_{p(z)}(zQ),\pi^{10}_p(zQ+\pi^3_{p(z)}(zQ))}(P) \bigg)^{\frac{p(z)^{12}-1}{r(z)}}, \end{aligned} \tag{4.12}$$

which leads to slightly faster pairing than Eq. (4.11). The pairings can be moved entirely on E', which we denote an optimal ate pairing defined on E' as $e'_{o_{c_i}}$. Assuming $\mathcal{G}'_2$ is a preimage of $\mathcal{G}_2$ under ϕ_6 and letting $P \in \mathcal{G}_1$ and $Q' \in \mathcal{G}'_2$, the ate pairing is computed by either $e_{o_{c_i}}(\phi_6(Q'), P)$ or $e'_{o_{c_i}}(Q', \phi_6^{-1}(P))$.

Not only Miller's algorithm but also the final exponentiation can be optimized by using the decomposition of $(p(z)^k - 1)/r(z)$ in base $p(z)$. The state-of-the-art algorithm for computing the hard part is given by the lattice-based method [FCKRH11] where the calculation step is described in Example 3.4 in Sect. 3.2.3.

4.3.2 Proposed BN subfamilies of curves with $k = 12$

For an integer seed z for specifying the curves in the BN family, the author proposes to restrict z as follows:

$$z \equiv 7, 11 \pmod{12}. \tag{4.13}$$

Once finding z satisfies the condition, the specific BN subfamilies with the following options are obtained.

Table 4.1: The field and curve options for the proposed BN subfamilies of curves with $k = 12$.

z (mod 12)	Tower *(see Theorem 4.7)*	BN curve $E/\mathbb{F}_{p(z)}$ *(see Theorem 4.9)*	Twist $E'/\mathbb{F}_{p(z)^2}$ *(see Theorem 4.10)*
7	$\mathbb{F}_{p(z)^2} \cong \mathbb{F}_{p(z)}(\alpha)$	$y^2 = x^3 + 2^{6n-1}$	$y^2 = x^3 + 2^{6n-1}\zeta$
11	$\mathbb{F}_{p(z)^{12}} \cong \mathbb{F}_{p(z)^2}(\beta)$	$y^2 = x^3 + 2^{6n+1}$	$y^2 = x^3 + 2^{6n+1}/\zeta$

(i) A fixed tower of extension fields with one of the best performing arithmetics is always available;

(ii) The BLS curve $E/\mathbb{F}_{p(z)}$ is immediately determined;

(iii) The correct twist $E'/\mathbb{F}_{p(z)^2}$ is also immediately determined.

These constructions also enable one of the simplest twist isomorphisms. The details of the field and curve options (i), (ii), and (iii) are summarized in Table 4.1, where $n \in \mathbb{Z}$, α and β are elements in $\mathbb{F}_{p(z)^2}$ and $\mathbb{F}_{p(z)^{12}}$ such that $\alpha^2 = -1$ and $\beta^6 = \alpha + 1$, respectively, and $\zeta = \alpha + 1 \in \mathbb{F}_{p(z)^2}$. The correctness of Table 4.1 is found in the following theorems with proof. Before describing that, the author refers to [Shi10] and presents the knowledge of the quadratic and cubic residue properties in $\mathbb{F}^*_{p(z)}$.

Lemma 4.6. For the symbols $(\frac{\cdot}{p(z)})$ and $(\frac{\cdot}{p(z)})_3$, the following is true.

$$\left(\frac{-1}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 0 \pmod 2, \\ -1 & \text{if } z \equiv 1 \pmod 2. \end{cases} \tag{4.14}$$

$$\left(\frac{2}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 0, 1 \pmod 4, \\ -1 & \text{if } z \equiv 2, 3 \pmod 4. \end{cases} \tag{4.15}$$

$$\left(\frac{2}{p(z)}\right)_3 \begin{cases} = 1 & \text{if } z \equiv 0 \pmod 3, \\ \neq 1 & \text{if } z \equiv 1, 2 \pmod 3. \end{cases} \tag{4.16}$$

Proof of Lemma 4.6. Please refer to [Shi10]. □

In what follows, the author provides the theorem which shows the construction of the tower of extension field of degree 12.

Theorem 4.7. If z satisfies $z \equiv 7, 11 \pmod{12}$, the following tower of extension field is always available.

$$\begin{cases} \mathbb{F}_{p(z)^2} & \cong \mathbb{F}_{p(z)}[x]/(x^2+1) & \cong \mathbb{F}_{p(z)}(\alpha), \\ \mathbb{F}_{p(z)^{12}} & \cong \mathbb{F}_{p(z)^2}[x]/(x^6-(\alpha+1)) & \cong \mathbb{F}_{p(z)^2}(\beta), \end{cases} \tag{4.17}$$

where α and β are elements in $\mathbb{F}_{p(z)^2}$ and $\mathbb{F}_{p(z)^{12}}$ such that $\alpha^2 = -1$ and $\beta^6 = \alpha + 1$, respectively.

Proof of Theorem 4.7. To admit the tower of extension field given in Eq. (4.17), the binomial x^2+1 and $x^6-(\alpha+1)$ must be irreducible in $\mathbb{F}_{p(z)}[x]$ and $\mathbb{F}_{p(z)^2}[x]$, respectively. According to Theorem 4.4, x^2+1 is irreducible in $\mathbb{F}_{p(z)}[x]$ if -1 is quadratic non-residue in $\mathbb{F}^*_{p(z)}$; $x^6-(\alpha+1)$ is irreducible in $\mathbb{F}_{p(z)^2}[x]$ if the norm of $\alpha+1$, i.e., $N_{\mathbb{F}_{p(z)^2}/\mathbb{F}_{p(z)}}(\alpha+1) = (\alpha+1)\cdot(\alpha+1)^{p(z)} = (\alpha+1)\cdot(-\alpha+1) = -\alpha^2+1 = 2$, is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)}$. As seen in Lemma 4.6, if $z \equiv 7, 11 \pmod{12}$, we have $(\frac{-1}{p(z)}) = -1$, $(\frac{2}{p(z)}) = -1$, and $(\frac{2}{p(z)})_3 \neq 1$. □

The author shows how uniquely the coefficients of the BN curves and its twisted curves can be determined in $E/\mathbb{F}_{p(z)}$ and $E'/\mathbb{F}_{p(z)^2}$, respectively. For the reference in the proof, the lemma given by [Shi10] is presented below.

Lemma 4.8. Let $n_i(z)$ for $0 \leq i \leq 5$ be polynomial defined as follows:

$$\begin{aligned} n_0(z) &= 12z^2(3z^2+3z+1), & n_1(z) &= 36z^4+36z^3+18z^2+1, \\ n_2(z) &= 3(12z^4+12z^3+10z^2+2z+1), & n_3(z) &= 4(9z^4+9z^3+9z^2+3z+1), \\ n_4(z) &= 3(12z^4+12z^3+10z^2+4z+1), & n_5(z) &= 36z^4+36z^3+18z^2+6z+1. \end{aligned}$$

Then, the group orders of $E_2/\mathbb{F}_{p(z)} : y^2 = x^3+2$ are determined as follows:

$$\#E_2(\mathbb{F}_{p(z)}) = \begin{cases} n_0(z) & \text{if } z \equiv 0, 9 \pmod{12}, \\ n_1(z) & \text{if } z \equiv 7, 10 \pmod{12}, \\ n_2(z) & \text{if } z \equiv 5, 8 \pmod{12}, \\ n_3(z) & \text{if } z \equiv 3, 6 \pmod{12}, \\ n_4(z) & \text{if } z \equiv 1, 4 \pmod{12}, \\ n_5(z) & \text{if } z \equiv 2, 11 \pmod{12}. \end{cases} \tag{4.18}$$

Proof of Lemma 4.8. Please refer to [Shi10]. □

Theorem 4.9. If z satisfies $z \equiv 7, 11 \pmod{12}$, the BN curve is determined as follows:

$$E/\mathbb{F}_{p(z)} : \begin{cases} y^2 = x^3 + 2^{6n-1} & \text{if } z \equiv 7 \pmod{12}, \\ y^2 = x^3 + 2^{6n+1} & \text{if } z \equiv 11 \pmod{12}, \end{cases} \tag{4.19}$$

where n is any integer.

Proof of Theorem 4.9. From the definition, an elliptic curve $E/\mathbb{F}_{p(z)}$ such that $\#E(\mathbb{F}_{p(z)}) = p(z)+1-t(z) = r(z) = 36z^4+36z^3+18z^2+6z+1$ is the BN curve. According to Lemma 4.8, if $z \equiv 11(\bmod\ 12)$, $E_2/\mathbb{F}_{p(z)} : y^2 = x^3+2$ is the BN curve since $\#E(\mathbb{F}_{p(z)}) = n_5(z) = r(z)$.

Then, it is easily found that an elliptic curve $E_{6n+1}/\mathbb{F}_{p(z)} : y^2 = x^3 + 2^{6n+1}$ is a twist of degree 1 of E_2. Since isomorphic two curves have the same group order, E_{6n+1} is also being the BN curve.

On the other hand, if $z \equiv 7 \pmod{12}$, it is found that $\#E_2(\mathbb{F}_{p(z)}) = n_1(z) = 36z^4 + 36z^3 + 18z^2 + 1$. Then, an elliptic curve $E_{2^{6n-1}}/\mathbb{F}_{p(z)} : y^2 = x^3 + 2^{6n-1}$ is a twist of degree 3 of E_{6n+1} since $\delta = 2^2$ of $2^{6n+1}/\delta = 2^{6n-1}$ is quadratic residue and cubic non-residue in $\mathbb{F}^*_{p(z)}$ under the condition. The author refer to Eq. (4.8) and find that $E_{2^{6n-1}}$ has only two possible numbers $n_5(z) = 36z^4+36z^3+18z^2+6z+1$ and $n_3(z) = 4(9z^4+9z^3+9z^2+3z+1)$. According to (a) in Lemma 4.5, if the group order of elliptic curves can be divisible by 2, coefficients of the curve have to be cubic residue in $\mathbb{F}^*_{p(z)}$. Here, $\#E_{2^{6n-1}}(\mathbb{F}_{p(z)})$ cannot have 2 as a factor since the curve coefficient of $E_{2^{6n-1}}$ has cubic non-residue in $\mathbb{F}^*_{p(z)}$. Since $\#E_{2^{6n-1}}(\mathbb{F}_{p(z)}) = n_5(z)$, E_{6n-1} ends up to the BN curve for the respective conditions of z. □

Theorem 4.10. Suppose that the tower of extension fields is constructed as in Theorem 4.7 and $E/\mathbb{F}_{p(z)}$ be the BLS curve determined as in Theorem 4.9. If z satisfies the condition given in Eq. (4.13), the correct twist BN curve is determined as follows:

$$E'/\mathbb{F}_{p(z)^2} : \begin{cases} y^2 = x^3 + 2^{6n-1}(\alpha+1) & \text{if } z \equiv 7 \pmod{12}, \\ y^2 = x^3 + 2^{6n+1}/(\alpha+1) & \text{if } z \equiv 11 \pmod{12}, \end{cases} \tag{4.20}$$

where α is an element in $\mathbb{F}_{p(z)^2}$ such that $\alpha^2 = -1$.

Proof of Theorem 4.10. Since there exist two candidates of the twists of E with the degree 6, E' has only two possible group orders given as $p(z)^2 + 1 - (t_2(z) - 3V_2(z))/2$ or $p(z)^2 + 1 - (t_2(z) + 3V_2(z))/2$ where $t_2(z) = p^2 + 1 - \#E(\mathbb{F}_{p(z)^2})$ and $V_2(z)$ is an integer such that $4p(z)^2 = t_2(z)^2 + 3V_2(z)^2$. In the context of the BN curve, $t_2(z)$ and $V_2(z)$ are given by $t_2(z) = -36z^4 - 72z^3 - 36z^2 - 12z - 1$ and $V_2(z) = (6z^2+1)(6z^2+4z+1)$, respectively. Thus, the possible group orders can be denoted by either $4(324z^8+648z^7+756z^6+540z^5+288z^4+108z^3+30z^2+6z+1)$ or $(36z^4+36z^3+18z^2+6z+1)(36z^4+36z^3+30z^2+6z+1)$. Since the correct twist $E'/\mathbb{F}_{p(z)^2}$ has a group of order $r(z)$, i.e., $r(z) \mid \#E'(\mathbb{F}_{p(z)^2})$, we can guess $\#E'(\mathbb{F}_{p(z)^2}) = (36z^4 + 36z^3 + 18z^2 + 6z + 1)(36z^4 + 36z^3 + 30z^2 + 6z + 1)$. Since it is found that $E'(\mathbb{F}_{p(z)^2})$ is not divisible by 2, the twisted curve E' coefficients should be a cubic non-residue in $\mathbb{F}^*_{p(z)^2}$. Now, in the case of the BN curve denoted by $y^2 = x^3 + 2^{6n+1}$, twisted curves can be denoted as $y^2 = x^3 + 2^{6n+1}(\alpha+1)$ or $y^2 = x^3 + 2^{6n+1}/(\alpha+1)$ since $(\alpha+1)$ and $1/(\alpha+1)$ are quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)^2}$. Then, the cubic residue properties of each curve coefficients are denoted as follows:

$$\left(2^{6n+1}(\alpha+1)\right)^{\frac{p^2-1}{3}} = \left(\left(2^{6n+1}(\alpha+1)\right)^{p+1}\right)^{\frac{p-1}{3}} = \left((2^{6n+1})^2 \cdot 2\right)^{\frac{p-1}{3}} = 1,$$

Table 4.2: The twisting and untwisting isomorphisms for the proposed BN subfamilies of curves with $k = 12$.

z (mod 12)	Twisting isomorphism $\phi_6 : E' \to E$	Untwisting isomorphism $\phi_6^{-1} : E \to E'$
7	$(x, y) \mapsto (\zeta^{-1}x\beta^4, \zeta^{-1}y\beta^3)$	$(x, y) \mapsto (x\beta^2, y\beta^3)$
11	$(x, y) \mapsto (x\beta^2, y\beta^3)$	$(x, y) \mapsto (\zeta^{-1}x\beta^4, \zeta^{-1}y\beta^3)$

$$\left(2^{6n+1}(\alpha+1)^{-1}\right)^{\frac{p^2-1}{3}} = \left(\left(2^{6n+1}(\alpha+1)^{-1}\right)^{p+1}\right)^{\frac{p-1}{3}} = \left((2^{6n+1})^2 \cdot 2^{-1}\right)^{\frac{p-1}{3}} \neq 1.$$

Since the coefficient of E' needs to be a cubic non-residue in $\mathbb{F}_{p(z)^2}^*$, the twisted curve is determined as $y^2 = x^3 + 2^{6n+1}/(\alpha+1)$. In the case of $y^2 = x^3 + 2^{6n-1}$, its twisted curves are also derived in the same way. □

Since the equations of E and E' are determined, it is easily obtained the twisting and untwisting isomorphisms as in Table 4.2. For $z \equiv 11 \pmod{12}$, since E and E' are given by $y^2 = x^3 + 2^{6n+1}$ and $y^2 = x^3 + 2^{6n+1}/\zeta$, the twisting isomorphism is given by $\phi_6 : E' \to E, (x, y) \mapsto (\delta^{1/3}x, \delta^{1/2}y)$. Since there is a relation $\delta^{1/3} = \beta^2$ and $\delta^{1/2} = \beta^3$, the image of (x, y) under ϕ_6 is modified as $(x\beta^2, y\beta^3)$. On the other hand, the untwisting isomorphism is $\phi_6 : E' \to E, (x, y) \mapsto (\delta^{-1/3}x, \delta^{-1/2}y)$, which is the image is simplified as $(\delta^{-1/3}x, \delta^{-1/2}y) = (\delta^{-1}\delta^{2/3}x, \delta^{-1}\delta^{1/2}y) = (\delta^{-1}x\beta^4, \delta^{-1}y\beta^3)$. For the other case $z \equiv 7 \pmod{12}$, the author obtains the result in the same manner. The important fact is that the twisting and untwisting isomorphisms are low complexity since $\{1, \beta, \ldots, \beta^5\}$ is a basis of the 6-th dimensional vector space of $\mathbb{F}_{p(z)^2}$.

4.3.3 Sample parameters and evaluation

Applying the restrictions, the author obtains several seeds for generating the BN curves for the pairings at the 128-bit security level shown in Table 4.3. For the search of z, the author refers to security analyses [Gui20] and tries to find z which gives rise to $r(z)$ with $\log_2 r(z) \geq 256$ and $\log_2 p(z) \geq 5376$ to ensure the pairings at the 128-bit security level. For the efficiency reason of Miller's algorithm, the author also finds z with low-Hamming weight.

The author evaluates the seeds for the pairings on the BN curves given in Table 4.3 by an implementation. For the implementation, the author adopts the optimal ate pairing given in Eq. (4.12) and efficient projective formulas for computing Miller's algorithm given by Costello et al. in [CLN10]. Note that the optimal ate pairing $e'_{o_{c_i}}(Q', \phi_6^{-1}(P))$ on E' with the untwisting isomorphism $\phi_6^{-1} : E \to E'$ is employed. The author also adopts the state-of-the-art algorithm for computing the final exponentiation by Fuentes et al. in [HHT20] (see Example 3.4).

Table 4.3: Sample parameters for the attractive BN subfamilies of curves with $k = 12$ at the 128-bit security level.

No.	z (mod 12)	Seed z	HW	Bit size		
				$p(z)$	$p(z)^k$	$r(z)$
1	7	$+2^{114} + 2^{101} - 2^{14} - 2^{0}$ [BD19]	4	462	5535	462
2	7	$-2^{114} + 2^{88} - 2^{78} - 2^{0}$	4	462	5535	462
3	7	$-2^{113} - 2^{42} + 2^{11} - 2^{0}$	4	458	5487	458
4	7	$+2^{113} - 2^{63} - 2^{50} - 2^{0}$	4	458	5487	458
5	7	$+2^{113} + 2^{78} + 2^{53} - 2^{0}$	4	458	5487	458
6	7	$+2^{113} - 2^{26} + 2^{4} - 2^{0}$	4	458	5487	458
7	7	$+2^{113} - 2^{67} - 2^{58} - 2^{0}$	4	458	5487	458
8	7	$-2^{113} - 2^{38} + 2^{13} - 2^{0}$	4	458	5487	458
9	11	$-2^{114} - 2^{62} - 2^{30} - 2^{0}$	4	462	5535	462
10	11	$+2^{114} + 2^{84} - 2^{53} - 2^{0}$	4	462	5535	462
11	11	$+2^{113} - 2^{86} + 2^{63} - 2^{0}$	4	458	5487	458
12	11	$-2^{113} + 2^{62} + 2^{16} - 2^{0}$	4	458	5487	458

With the optimizations, the author implements the software for executing the pairings by C language. The big integer arithmetics are implemented by using `mp_limb_t` data type of the GMP library in [tea15]. The software is compiled with GCC 8.3.0 with the option `-O2 -march=native` and is executed by 3.50GHz Intel(R) Core(TM) i7-7567U CPU running macOS Big Sur version 11.6. To evaluate the parameters, the average execution times of 100,000 trials of Miller's algorithm and final exponentiation are measured. Note that the measurement is performed by repeating the functions for 1,000 random inputs 100 times.

Table 4.4 shows the results of the average execution time of Miller's algorithm and final exponentiation. The author could not find a significant difference between the timings of the candidates of the curves, however, there are differences between the equations of ϕ_6^{-1}. As seen in Table 4.2, since the equation is simpler than that of $z \equiv 11 \pmod{12}$, it is theoretically better to use z satisfying $z \equiv 7 \pmod{12}$, however, it is considered that the effect is very small in this environment.

4.4 Proposed BLS subfamilies of curves with $k = 2^m \cdot 3$ and 3^n for any $m, n > 0$

The author extends Costello et al.'s work for the BLS family of curves with $k = 24$ and provides the attractive subfamilies of the BLS family of curves with $k = 2^m \cdot 3$ and 3^n for any integers $m, n > 0$.

Table 4.4: Average execution times for computing Miller's algorithm (ML) and final exponentiation (FE) for the pairings on the BN curves with $k = 12$ at the 128-bit security level.

No.	z	Seed z	HW	Word	ML	FE	Total
	(mod 12)			size	[ms]	[ms]	[ms]
1	7	$+2^{114} + 2^{101} - 2^{14} - 2^0$ [BD19]	4	8	2.37	1.41	3.78
2	7	$-2^{114} + 2^{88} - 2^{78} - 2^0$	4	8	2.37	1.41	3.78
3	7	$-2^{113} - 2^{42} + 2^{11} - 2^0$	4	8	2.34	1.40	3.73
4	7	$+2^{113} - 2^{63} - 2^{50} - 2^0$	4	8	2.35	1.40	3.75
5	7	$+2^{113} + 2^{78} + 2^{53} - 2^0$	4	8	2.34	1.39	3.74
6	7	$+2^{113} - 2^{26} + 2^4 - 2^0$	4	8	2.35	1.40	3.75
7	7	$+2^{113} - 2^{67} - 2^{58} - 2^0$	4	8	2.34	1.40	3.74
8	7	$-2^{113} - 2^{38} + 2^{13} - 2^0$	4	8	2.34	1.40	3.74
9	11	$-2^{114} - 2^{62} - 2^{30} - 2^0$	4	8	2.37	1.40	3.77
10	11	$+2^{114} + 2^{84} - 2^{53} - 2^0$	4	8	2.44	1.44	3.88
11	11	$+2^{113} - 2^{86} + 2^{63} - 2^0$	4	8	2.40	1.41	3.81
12	11	$-2^{113} + 2^{62} + 2^{16} - 2^0$	4	8	2.39	1.41	3.80

4.4.1 Review of the pairings with the BLS family

The BLS family is a family of curves E with the CM discriminant $D = 3$, i.e., $j(E) = 0$, and the embedding degree k of multiple of 3 except for $k = 18$. The parameterizations of the BLS family are given by triples of $p(x)$, $r(x)$, and $t(x)$ in $\mathbb{Q}[x]$. Since the BLS family is complete, one can find a polynomial $V(x) \in \mathbb{Q}[x]$ such that $3V(x)^2 = 4p(x) - t(x)^2$. For the case of $k = 2^m \cdot 3$ and 3^n with any integers $m, n > 0$, the polynomial parameters $p(x)$, $r(x)$, $t(x)$, and $V(x)$ in $\mathbb{Q}[x]$ are given as follows [BLS02]:

- $k = 2^m \cdot 3$

$$\begin{cases} p(x) &= \frac{1}{3}(x-1)^2 \cdot r(x) + x, \\ r(x) &= \Phi_k(x) = x^{2^m} - x^{2^{m-1}} + 1, \\ t(x) &= x + 1, \\ V(x) &= \frac{1}{3}(x-1) \cdot (2x^{2^{m-1}} - 1). \end{cases} \tag{4.21}$$

- $k = 3^n$

$$\begin{cases} p(x) &= (x-1)^2 \cdot r(x) + x, \\ r(x) &= \frac{1}{3}\Phi_k(x) = \frac{1}{3}(x^{2\cdot 3^{n-1}} + x^{3^{n-1}} + 1), \\ t(x) &= x + 1, \\ V(x) &= \frac{1}{3}(x-1) \cdot (2x^{3^{n-1}} + 1). \end{cases} \tag{4.22}$$

Let z be an integer making $p(z)$ and $r(z)$ being primes and $t(z)$ and $V(z)$ being integers where the condition $z \equiv 1 \pmod 3$ leads to all involved parameters being integers. One can find an elliptic curve $E/\mathbb{F}_{p(z)} : y^2 = x^3 + b$ such that $n(z) = \#E(\mathbb{F}_{p(z)}) = p(z) + 1 - t(z)$ with the prime divisor $r(z)$, which we say E is the BLS curve. Let $d = 6$ and 3 for $k = 2^m \cdot 3$ and 3^n, respectively. Then, one can also find a correct twist $E'/\mathbb{F}_{p(z)^{k/d}} : y^2 = x^3 + b'$ of degree d of E such that $r(z) \mid n'(z) = \#E'(\mathbb{F}_{p(z)^{k/d}})$ and twisting isomorphism $\phi_d : E' \to E$ defined over $\mathbb{F}_{p(z)^k}$.

Let $\mu_{r(z)}$ be the multiplicative subgroup of $\mathbb{F}_{p(z)^k}$ of order $r(z)$ consisting of the $r(z)$-th root of identity. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be the base-field and trace-zero groups of $r(z)$-torsion subgroup $E[r(z)]$, respectively. Since $T(z) = t(z) - 1 = z$, the standard ate pairing is defined as follows:

$$\begin{aligned} e_{a_T} &: \mathcal{G}_2 \times \mathcal{G}_1 \to \mu_{r(z)}, \\ e_{a_T}(Q, P) &= f_{z,Q'}(P)^{\frac{p(z)^k - 1}{r(z)}}. \end{aligned} \tag{4.23}$$

Since $\log_2 r(z)/\phi(k) \approx \log_2 z$, the ate pairing can be computed by Miller's algorithm with one of the shortest iterations. This means that the ate pairing is exactly one of the optimal ate pairings given by [Ver09]. It is possible to define the pairing on E' by using the preimages $\mathcal{G}_2'$ and $\mathcal{G}_1'$ of $\mathcal{G}_2$ and $\mathcal{G}_1$ under ϕ_d, respectively. Assuming e'_{a_T} is an ate pairing on E' and $P \in \mathcal{G}_1$ and $Q' \in \mathcal{G}_2'$, one can efficiently compute the ate pairing either $e_{a_T}(\phi_d(Q), P)$ or $e'_{a_T}(Q, \phi_d^{-1}(P))$.

As the other optimization, the exponent $(p(z)^k - 1)/r(z)$ of the final exponentiation can be decomposed by using Theorem 3.6 by Hayashida et al. in [HHT20]. This immediately provides one of the efficient algorithms for computing the final exponentiation by using the Frobenius endomorphisms.

4.4.2 Determination of the number of rational points on the correct twists

To determine the twist equation E' by using Lemma 4.5, the knowledge of the number of the rational points on E' is required. This subsection shows the knowledge for the BLS family of curves for $k = 2^m \cdot 3$ and 3^n with any integers $m, n > 0$, respectively.

(i) The case of $k = 2^m \cdot 3$ for any $m > 0$

Let $p(x)$, $r(x)$, $t(x)$, and $V(x)$ be the polynomials fixed as Eq. (4.21) for the BLS family of curves with $k = 2^m \cdot 3$ for any integer $m > 0$. For an integer z making $p(z)$ and $r(z)$ being primes and $t(z)$ and $V(z)$ being integers, let $E/\mathbb{F}_{p(z)}$ and $E'/\mathbb{F}_{p(z)^{2^{m-1}}}$ be the BLS curve and correct twist of degree 6 of E. For any integer $s > 0$, let $t_s(z) = p(z)^s + 1 - \#E(\mathbb{F}_{p(z)^s})$ be a trace of E defined over $\mathbb{F}_{p(z)^s}$ and $V_s(z)$ be a parameter such that $3V_s(z)^2 = 4p(z)^s - t_s(z)^2$.

Then, the group order of the correct twist is specifically represented as follows:

Theorem 4.11. For $k = 2^m \cdot 3$ with any $m > 0$, the group order of the correct twist $E'/\mathbb{F}_{p(z)^{2^{m-1}}}$ of degree 6 of E is uniquely given as

$$\#E'(\mathbb{F}_{p(z)^{2^{m-1}}}) = p(z)^{2^{m-1}} + 1 - \frac{t_{2^{m-1}}(z) - 3V_{2^{m-1}}(z)}{2}. \tag{4.24}$$

To prove Theorem 4.11, the author provides the following Lemmas 4.12, 4.13, and 4.14.

Lemma 4.12. For any integer $l \geq 0$, $t_{2^{l+1}}(z)$ and $V_{2^{l+1}}(z)$ can be built from the knowledge of $t_{2^l}(z)$ and $V_{2^l}(z)$ as follows:

$$t_{2^{l+1}}(z) = t_{2^l}(z)^2 - 2p(z)^{2^l}, \tag{4.25}$$

$$V_{2^{l+1}}(z) = t_{2^l}(z) \cdot V_{2^l}(z). \tag{4.26}$$

Proof of Lemma 4.12. According to Theorem 2.50, for any $l > 0$, the trace $t_{2^l}(z) = p^{2^l} + 1 - \#E(\mathbb{F}_{p(z)^{2^l}})$ can be written as $t_{2^l}(z) = \alpha^{2^l} + \beta^{2^l}$ where α and β are roots of the polynomial $X^2 - t(z) \cdot X + p(z)$, i.e., $\alpha \cdot \beta = p(z)$ and $\alpha + \beta = t(z)$. Thus, $t_{2^{l+1}}(z)$ can be represented as

$$t_{2^{l+1}}(z) = \alpha^{2^{l+1}} + \beta^{2^{l+1}} = (\alpha^{2^l} + \beta^{2^l})^2 - 2(\alpha \cdot \beta)^{2^l} = t_{2^l}(z)^2 - 2p(z)^{2^l}. \tag{4.27}$$

Moreover, with the above, the following is also obtained.

$$\begin{aligned} 3V_{2^{l+1}}(z)^2 &= 4p(z)^{2^{l+1}} - t_{2^{l+1}}(z)^2 \\ &= 4p(z)^{2^{l+1}} - (t_{2^l}(z)^2 - 2p(z)^{2^l})^2 \\ &= 4p(z)^{2^{l+1}} - t_{2^l}(z)^4 + 4t_{2^l}(z)^2 \cdot p(z)^{2^l} - 4p(z)^{2^{l+1}} \\ &= t_{2^l}(z)^2 \cdot (4 \cdot p(z)^{2^l} - t_{2^l}(z)^2) \\ &= t_{2^l}(z)^2 \cdot 3V_{2^l}(z)^2, \end{aligned} \tag{4.28}$$

which leads to $V_{2^{l+1}}(z) = t_{2^l}(z) \cdot V_{2^l}(z)$. □

Lemma 4.13. For any integer $l \geq 0$, the following holds.

$$t_{2^l}(z) \equiv z^{2^l} + 1 \pmod{r(z)}. \tag{4.29}$$

Proof of Lemma 4.13. The lemma can be proven by induction on l.

(i) For $l = 0$, it is obvious that $t_{2^0}(z) = t(z) \equiv z + 1 \pmod{r(z)}$.

(ii) For $l = s$ with an integer $s > 0$, suppose that $t_{2^s}(z) \equiv z^{2^s}+1 \pmod{r(z)}$. According to Lemma 4.12 and $p(z) = (z-1)^2/3 \cdot r(z) + z \equiv z \pmod{r(z)}$,

$$\begin{aligned} t_{2^{s+1}}(z) &= t_{2^s}(z)^2 - 2p(z)^{2^s} \\ &\equiv t_{2^s}(z)^2 - 2z^{2^s} \equiv (z^{2^s}+1)^2 - 2z^{2^s} \equiv z^{2^{s+1}} + 1 \pmod{r(z)}. \end{aligned} \tag{4.30}$$

Thus, $t_{2^{s+1}}(z) \equiv z^{2^{s+1}} + 1 \pmod{r(z)}$ is also true for $l = s+1$.

Since both the base case (i) and the inductive step (ii) have been proven, $t_{2^l}(z) \equiv z^{2^l} + 1 \pmod{r(z)}$ holds for any $l \geq 0$. □

Lemma 4.14. For any integer $l > 0$, the following holds.

$$\frac{t_{2^l}(z) \pm 3V_{2^l}(z)}{2} \equiv \sum_{i=0}^{2^l-1} z^i \cdot \frac{t(z) \pm 3V(z)}{2} - \sum_{i=1}^{2^l-1} z^i \pmod{r(z)}. \tag{4.31}$$

Proof of Lemma 4.14. The lemma can be proven by induction on l.

(i) For $l = 1$, from Lemmas 4.12, 4.13, and $p(z) \equiv z \pmod{r(z)}$, the following can be obtained.

$$\begin{aligned} \frac{t_2(z) \pm 3V_2(z)}{2} &= \frac{(t(z)^2 - 2p(z)) \pm (t(z) \cdot 3V(z))}{2} \\ &= t(z) \cdot \frac{t(z) \pm 3V(z)}{2} - p(z) \\ &\equiv (z+1) \cdot \frac{t(z) \pm 3V(z)}{2} - z \pmod{r(z)}. \end{aligned} \tag{4.32}$$

Thus, the lemma is true for $l = 1$.

(ii) For $l = s$ with an integer $s > 1$, suppose that the lemma is true. Then, the following is obtained.

$$\begin{aligned} &\frac{t_{2^{s+1}}(z) \pm 3V_{2^{s+1}}(z)}{2} \\ =&\frac{(t_{2^s}(z)^2 - 2p(z)^{2^s}) \pm (t_{2^s}(z) \cdot 3V_{2^s}(z))}{2} && \pmod{r(z)} \\ =&t_{2^s}(z) \cdot \frac{t_{2^s}(z) \pm 3V_{2^s}(z)}{2} - p(z)^{2^s} \\ \equiv&(z^{2^s}+1) \cdot \left(\sum_{i=0}^{2^s-1} z^i \cdot \frac{t(z) \pm 3V(z)}{2} - \sum_{i=1}^{2^s-1} z^i \right) - z^{2^s} && \pmod{r(z)} \\ \equiv&\left(\sum_{i=2^s}^{2^{s+1}-1} z^i + \sum_{i=0}^{2^s-1} z^i \right) \cdot \frac{t(z) \pm 3V(z)}{2} - \sum_{i=2^s+1}^{2^{s+1}-1} z^i - \sum_{i=1}^{2^s-1} z^i - z^{2^s} && \pmod{r(z)} \end{aligned}$$

$$\equiv \sum_{i=0}^{2^{s+1}-1} z^i \cdot \frac{t(z) \pm 3V(z)}{2} - \sum_{i=1}^{2^{s+1}-1} z^i \qquad \pmod{r(z)}. \tag{4.33}$$

Thus, the lemma is also true for $l = s + 1$.

Since both the base case (i) and the inductive step (ii) have been proven, it is clear that the lemma is true for any $l > 0$. □

Then, the author provides the proof of Theorem 4.11 by using the above lemmas.

Proof of Theorem 4.11. According to Eq. (4.8), the group order of the twist of $E/\mathbb{F}_{p(z)}$ can be determined corresponding to the twist degree d. In this case, since $d = 6$, it is found that $\#E'(\mathbb{F}_{p(z)^{2^{m-1}}})$ is given by one of the following.

$$n'_0(z) = p(z)^{2^{m-1}} + 1 - \frac{t_{2^{m-1}}(z) + 3V_{2^{m-1}}(z)}{2}, \tag{4.34}$$

$$n'_1(z) = p(z)^{2^{m-1}} + 1 - \frac{t_{2^{m-1}}(z) - 3V_{2^{m-1}}(z)}{2}. \tag{4.35}$$

Besides, from the definition, the group order of the correct twist is divisible by $r(z)$. Thus, to prove the theorem, it is enough to show that $r(z)$ divides $n'_1(z)$ but does not divide $n'_0(z)$, i.e., $n'_0(z) \not\equiv 0 \pmod{r(z)}$ and $n'_1(z) \equiv 0 \pmod{r(z)}$. Note that $r(z) = z^{2^m} - z^{2^{m-1}} + 1 \equiv 0 \pmod{r(z)}$ in this case.

Applying Lemma 4.14, the possible group orders $n'_0(z)$ modulo $r(z)$ can be denoted as follows:

$$\begin{aligned}
n'_0(z) &\equiv z^{2^{m-1}} + 1 - \left(\sum_{i=0}^{2^{m-1}-1} z^i \cdot \frac{t(z) + 3V(z)}{2} - \sum_{i=1}^{2^{m-1}-1} z^i \right) && \pmod{r(z)} \\
&\equiv z^{2^{m-1}} + 1 - \sum_{i=0}^{2^{m-1}-1} z^i \cdot ((z-1) \cdot x^{2^{m-1}} + 1) + \sum_{i=1}^{2^{m-1}-1} z^i && \pmod{r(z)} \\
&\equiv z^{2^{m-1}} + 1 - (z^{2^{m-1}} - 1) \cdot z^{2^{m-1}} - \sum_{i=0}^{2^{m-1}-1} z^i + \sum_{i=1}^{2^{m-1}-1} z^i && \pmod{r(z)} \\
&\equiv z^{2^{m-1}} + 1 - z^{2^m} + z^{2^{m-1}} - 1 && \pmod{r(z)} \\
&\equiv -z^{2^m} + 2z^{2^{m-1}} && \pmod{r(z)}.
\end{aligned} \tag{4.36}$$

On the other hand, for $n'_1(z)$ modulo $r(z)$, the following is obtained.

$$n'_1(z) \equiv z^{2^{m-1}} + 1 - \left(\sum_{i=0}^{2^{m-1}-1} z^i \cdot \frac{t(z) - 3V(z)}{2} - \sum_{i=1}^{2^{m-1}-1} z^i \right) \qquad \pmod{r(z)}$$

$$\begin{aligned}
&\equiv z^{2^{m-1}} + 1 - \sum_{i=0}^{2^{m-1}-1} z^i \cdot (-(z-1) \cdot x^{2^{m-1}} + z) + \sum_{i=1}^{2^{m-1}-1} z^i && (\mathrm{mod}\ r(z)) \\
&\equiv z^{2^{m-1}} + 1 + (z^{2^{m-1}} - 1) \cdot z^{2^{m-1}} - \sum_{i=1}^{2^{m-1}} z^i + \sum_{i=1}^{2^{m-1}-1} z^i && (\mathrm{mod}\ r(z)) \\
&\equiv z^{2^{m-1}} + 1 + z^{2^m} - z^{2^{m-1}} - z^{2^{m-1}} && (\mathrm{mod}\ r(z)) \\
&\equiv z^{2^m} - z^{2^{m-1}} + 1 && (\mathrm{mod}\ r(z)) \\
&\equiv 0 && (\mathrm{mod}\ r(z)). \qquad (4.37)
\end{aligned}$$

Thus, Theorem 4.11 is true. □

(ii) The case of $k = 3^n$ for any $n > 0$

Let $p(x)$, $r(x)$, $t(x)$, and $V(x)$ be the polynomials fixed as Eq. (4.22) for the BLS family of curves with $k = 3^n$ for any $n > 0$. For an integer z making $p(z)$ and $r(z)$ being primes and $t(z)$ and $V(z)$ being integers, let $E/\mathbb{F}_{p(z)}$ and $E'/\mathbb{F}_{p(z)^{3^{n-1}}}$ be the BLS curve and correct twist of degree 3 of E. For any integer $s > 0$, let $t_s(z) = p(z)^s + 1 - \#E(\mathbb{F}_{p(z)^s})$ be a trace of E defined over $\mathbb{F}_{p(z)^s}$ and $V_s(z)$ be an integer such that $3V_s(z)^2 = 4p(z)^s - t_s(z)^2$. Then, the group order of the correct twist can be represented as shown in the below.

Theorem 4.15. For $k = 3^n$ with any $n > 0$, the group order of the correct twist $E'/\mathbb{F}_{p(z)^{3^{n-1}}}$ of degree 3 of E is uniquely given as the following.

$$\#E'(\mathbb{F}_{p(z)^{3^{n-1}}}) = p(z)^{3^{n-1}} + 1 - \frac{-t_{3^{n-1}}(z) - 3V_{3^{n-1}}(z)}{2}. \qquad (4.38)$$

Theorem 4.15 can be proven with the following Lemmas 4.16, 4.17, and 4.18.

Lemma 4.16. For any integer $l \geq 0$, $t_{3^{l+1}}(z)$ and $V_{3^{l+1}}(z)$ can be built from the knowledge of $t_{3^l}(z)$ and $V_{3^l}(z)$ as follows:

$$t_{3^{l+1}}(z) = t_{3^l}(z)^3 - 3p(z)^{3^l} \cdot t_{3^l}(z), \qquad (4.39)$$

$$V_{3^{l+1}}(z) = V_{3^l}(z) \cdot (t_{3^l}(z)^2 - p(z)^{3^l}). \qquad (4.40)$$

Proof of Lemma 4.16. Similar to proof of Lemma 4.12, for any $l > 0$, the trace $t_{3^l}(z) = p^{3^l} + 1 - \#E(\mathbb{F}_{p(z)^{3^l}})$ can be written as $t_{3^l}(z) = \alpha^{3^l} + \beta^{3^l}$ where α and β are roots of the polynomial $X^2 - t(z) \cdot X + p(z)$, i.e., $\alpha \cdot \beta = p(z)$ and $\alpha + \beta = t(z)$ (see [Sil09]). Thus, $t_{3^{l+1}}(z)$ can be denoted as follows:

$$\begin{aligned}
t_{3^{l+1}}(z) = \alpha^{3^{l+1}} + \beta^{3^{l+1}} &= (\alpha^{3^l} + \beta^{3^l})^3 - 3(\alpha \cdot \beta)^{3^l} \cdot (\alpha^{3^l} + \beta^{3^l}) \\
&= t_{3^l}(z)^3 - 3p(z)^{3^l} \cdot t_{3^l}(z). \qquad (4.41)
\end{aligned}$$

Besides, it is also denoted as follows:

$$\begin{aligned}
3V_{3^{l+1}}(z)^2 &= 4p(z)^{3^{l+1}} - t_{3^{l+1}}(z)^2 \\
&= 4p(z)^{3^{l+1}} - (t_{3^l}(z)^3 - 3p(z)^{3^l} \cdot t_{3^l}(z))^2 \\
&= 4p(z)^{3^{l+1}} - t_{3^l}(z)^6 + 6p(z)^{3^l} \cdot t_{3^l}(z)^4 - 9p(z)^{2\cdot 3^l} \cdot t_{3^l}(z)^2 \\
&= (4p(z)^{3^l} - t_{3^l}(z)^2) \cdot (t_{3^l}(z)^2 - p(z)^{3^l})^2 \\
&= 3V_{3^l}(z)^2 \cdot (t_{3^l}(z)^2 - p(z)^{3^l})^2, \qquad (4.42)
\end{aligned}$$

which leads to $V_{3^{l+1}}(z) = V_{3^l}(z) \cdot (t_{3^l}(z)^2 - p(z)^{3^l})$. □

Lemma 4.17. For any integer $l \geq 0$, the following holds.

$$t_{3^l}(z) \equiv z^{3^l} + 1 \pmod{r(z)}. \qquad (4.43)$$

Proof of Lemma 4.17. The lemma can be proven by induction on l.

(i) For $l = 0$, it is clear that $t_{3^0}(z) = t(z) \equiv z + 1 \pmod{r(z)}$.

(ii) For $l = s$ with an integer $s > 0$, let $t_{3^s}(z) \equiv z^{3^s} + 1 \pmod{r(z)}$ be ture. Then, according to Lemma 4.12 and $p(z) \equiv z \pmod{r(z)}$, the case of $l = s + 1$ can be obtained as follows:

$$\begin{aligned}
t_{3^{s+1}}(z) &= t_{3^l}(z)^3 - 3p(z)^{3^s} \cdot t_{3^s}(z) \\
&\equiv (z^{3^s} + 1)^3 - 3z^{3^s} \cdot (z^{3^s} + 1) &&\pmod{r(z)} \\
&\equiv z^{3^{s+1}} + 3z^{2\cdot 3^s} + 3z^{3^s} + 1 - 3z^{2\cdot 3^s} - 3z^{3^s} &&\pmod{r(z)} \\
&\equiv z^{3^{s+1}} + 1 &&\pmod{r(z)}. \qquad (4.44)
\end{aligned}$$

Thus, $t_{3^{s+1}}(z) \equiv z^{3^{s+1}} + 1 \pmod{r(z)}$ is also held for $l = s + 1$.

Since both the base case (i) and inductive step (ii) have been proven, $t_{3^l}(z) \equiv z^{3^l} + 1 \pmod{r(z)}$ is true for any $l > 0$. □

Lemma 4.18. For any integer $l > 0$, the following holds.

$$\frac{-t_{3^l}(z) \pm 3V_{3^l}(z)}{2} \equiv \sum_{i=0}^{3^l-1} z^i \cdot \frac{-t(z) \pm 3V(z)}{2} + \sum_{i=1}^{3^l-1} z^i \pmod{r(z)}. \qquad (4.45)$$

Proof of Lemma 4.18. The lemma can be proven by induction on l.

(i) For $l = 1$, from Lemmas 4.16, 4.17, and $p(z) \equiv z \pmod{r(z)}$, it is found that

$$\frac{-t_3(z) \pm 3V_3(z)}{2}$$

$$
\begin{aligned}
&= \frac{-(t(z)^3 - 3p(z)\cdot t(z)) \pm 3V(z)\cdot(t(z)^2 - p(z))}{2} \\
&= \frac{-t(z)\cdot(t(z)^2 - p(z)) + 2p(z)\cdot t(z) \pm 3V(z)\cdot(t(z)^2 - p(z))}{2} \\
&= (t(z)^2 - p(z))\cdot\frac{-t(z) \pm 3V(z)}{2} + p(z)\cdot t(z).
\end{aligned} \tag{4.46}
$$

Then, taking modulo $r(z)$,

$$
\frac{-t_3(z) \pm 3V_3(z)}{2} \equiv (z^2+z+1)\cdot\frac{-t(z) \pm 3V(z)}{2} + (z^2+z) \pmod{r(z)}. \tag{4.47}
$$

The above shows that the lemma is true for $l = 1$.

(ii) For $l = s$ with an integer $s > 1$, suppose that the lemma is true. With the assumption, for $l = s+1$, the following can be obtained.

$$
\begin{aligned}
&\frac{-t_{3^{s+1}}(z) \pm 3V_{3^{s+1}}(z)}{2} \\
&= \frac{-(t_{3^s}(z)^3 - 3p(z)^{3^s}\cdot t_{3^s}(z)) \pm 3V_{3^s}(z)\cdot(t_{3^s}(z)^2 - p(z)^{3^s})}{2} \\
&= \frac{-t_{3^s}(z)\cdot(t_{3^s}(z)^2 - p(z)^{3^s}) + 2p(z)^{3^s}\cdot t_{3^s}(z) \pm 3V_{3^s}(z)\cdot(t_{3^s}(z)^2 - p(z)^{3^s})}{2} \\
&= (t_{3^s}(z)^2 - p(z)^{3^s})\cdot\frac{-t_{3^s}(z) \pm 3V_{3^s}(z)}{2} + p(z)^{3^s}\cdot t_{3^s}(z).
\end{aligned} \tag{4.48}
$$

Similarly, taking modulo $r(z)$, the following is obtained.

$$
\begin{aligned}
&\frac{-t_{3^{s+1}}(z) \pm 3V_{3^{s+1}}(z)}{2} \\
&\equiv (z^{2\cdot 3^s} + z^{3^s} + 1)\cdot\left(\sum_{i=0}^{3^s-1} z^i\cdot\frac{-t(z) \pm 3V(z)}{2} + \sum_{i=1}^{3^s-1} z^i\right) \\
&\qquad + (z^{2\cdot 3^s} + z^{3^s}) \qquad \pmod{r(z)} \\
&\equiv \left(\sum_{i=2\cdot 3^s}^{3^{s+1}-1} z^i + \sum_{i=3^s}^{2\cdot 3^s-1} z^i + \sum_{i=0}^{3^s-1} z^i\right)\cdot\frac{-t(z) \pm 3V(z)}{2} \qquad \pmod{r(z)} \\
&\qquad + \sum_{i=2\cdot 3^s+1}^{3^{s+1}-1} z^i + \sum_{i=3^s+1}^{2\cdot 3^s-1} z^i + \sum_{i=1}^{3^s-1} z^i + (z^{2\cdot 3^s} + z^{3^s}) \qquad \pmod{r(z)} \\
&\equiv \sum_{i=0}^{3^{s+1}-1} z^i\cdot\frac{-t(z) \pm 3V(z)}{2} + \sum_{i=1}^{3^{s+1}-1} z^i \qquad \pmod{r(z)}.
\end{aligned} \tag{4.49}
$$

Thus, the lemma is also true for $l = s+1$.

Since both the base case (i) and the inductive step (ii) have been proven, the lemma is true for any $l > 0$. □

In the following, the author provides the proof of Theorem 4.15 by using the above lemmas.

Proof of Theorem 4.15. According to Eq. (4.8), the group order $\#E(\mathbb{F}_{p(z)^{3^{n-1}}})$ of twist of degree 3 of $E/\mathbb{F}_{p(z)}$ is given by one of the following.

$$n'_0(z) = p(z)^{3^{n-1}} + 1 - \frac{-t_{3^{n-1}}(z) + 3V_{3^{n-1}}(z)}{2}, \tag{4.50}$$

$$n'_1(z) = p(z)^{3^{n-1}} + 1 - \frac{-t_{3^{n-1}}(z) - 3V_{3^{n-1}}(z)}{2}. \tag{4.51}$$

Since the group order is divisible by $r(z)$, it is enough to show that $r(z)$ divides $n'_1(z)$ but does not divide $n'_0(z)$, i.e., $n'_0(z) \not\equiv 0 \pmod{r(z)}$ and $n'_1(z) \equiv 0 \pmod{r(z)}$. Applying Lemma 4.18, the possible group order $n'_0(z)$ modulo $r(z) = \frac{1}{3}(z^{2\cdot 3^{n-1}} + z^{3^{n-1}} + 1)$ can be written as follows:

$$\begin{aligned}
n'_0(z) &\equiv z^{3^{n-1}} + 1 - \left(\sum_{i=0}^{3^{n-1}-1} z^i \cdot \frac{-t(z) + 3V(z)}{2} + \sum_{i=1}^{3^{n-1}-1} z^i \right) && \pmod{r(z)} \\
&\equiv z^{3^{n-1}} + 1 - \sum_{i=0}^{3^{n-1}-1} z^i \cdot ((z-1) \cdot z^{3^{n-1}} - 1) - \sum_{i=1}^{3^{n-1}-1} z^i && \pmod{r(z)} \\
&\equiv z^{3^{n-1}} + 1 - (z^{3^{n-1}} - 1) \cdot z^{3^{n-1}} + \sum_{i=0}^{3^{n-1}-1} z^i - \sum_{i=1}^{3^{n-1}-1} z^i && \pmod{r(z)} \\
&\equiv z^{3^{n-1}} + 1 - z^{2\cdot 3^{n-1}} + z^{3^{n-1}} + 1 && \pmod{r(z)} \\
&\equiv -z^{2\cdot 3^{n-1}} + 2z^{3^{n-1}} + 2 && \pmod{r(z)}.
\end{aligned} \tag{4.52}$$

For the case of $n'_1(z)$ modulo $r(z)$,

$$\begin{aligned}
n'_0(z) &\equiv z^{3^{n-1}} + 1 - \left(\sum_{i=0}^{3^{n-1}-1} z^i \cdot \frac{-t(z) - 3V(z)}{2} + \sum_{i=1}^{3^{n-1}-1} z^i \right) && \pmod{r(z)} \\
&\equiv z^{3^{n-1}} + 1 - \sum_{i=0}^{3^{n-1}-1} z^i \cdot (-(z-1) \cdot z^{3^{n-1}} - z) - \sum_{i=1}^{3^{n-1}-1} z^i && \pmod{r(z)} \\
&\equiv z^{3^{n-1}} + 1 + (z^{3^{n-1}} - 1) \cdot z^{3^{n-1}} + \sum_{i=1}^{3^{n-1}} z^i - \sum_{i=1}^{3^{n-1}-1} z^i && \pmod{r(z)} \\
&\equiv z^{3^{n-1}} + 1 + z^{2\cdot 3^{n-1}} - z^{3^{n-1}} + z^{3^{n-1}} && \pmod{r(z)} \\
&\equiv z^{2\cdot 3^{n-1}} + z^{3^{n-1}} + 1 && \pmod{r(z)} \\
&\equiv 0 && \pmod{r(z)}.
\end{aligned} \tag{4.53}$$

From the above, Theorem 4.15 is true. □

4.4.3 Proposed restriction of integer parameters

The author extends Costello et al.'s work [CLN11] and provides the restrictions of integer parameters for the BLS subfamilies of curves with $k = 2^m \cdot 3$ and 3^n with any $m, n > 0$. The details of the proposals for the cases of $k = 2^m \cdot 3$ and 3^n are described in the following.

(i) The case of $k = 2^m \cdot 3$ for any $m > 0$

Let z be an integer parameter for the BLS family of curves with $k = 2^m \cdot 3$ where $m > 0$ is an arbitrary integer. The author proposes to restrict z as follows:

$$z \equiv \begin{cases} 7, 10, 16, 28, 31, 34 \pmod{36} & \text{if } m = 1, \\ 7, 16, 31, 64 \pmod{72} & \text{if } m > 1. \end{cases} \tag{4.54}$$

Once finding z under the above restrictions, the specific subfamilies of the BLS family with the options are obtained.

(i) A fixed tower of extension fields with one of the best performing arithmetics is always available;

(ii) The BLS curve $E/\mathbb{F}_{p(z)}$ is immediately determined;

(iii) The correct twist $E'/\mathbb{F}_{p(z)^{2^{m-1}}}$ is also immediately determined.

The constructions also enable one of the simplest twist isomorphisms. The details of the field and curve options (i), (ii), and (iii) are summarized in Table 4.5, where α and β are elements in $\mathbb{F}_{p(z)^2}$ and $\mathbb{F}_{p(z)^{2^m \cdot 3}}$ such that $\alpha^2 = -1$ and $\beta^{2^{m-1} \cdot 3} = \alpha + 1$, respectively, and where $\zeta = \beta^6 \in \mathbb{F}_{p(z)^{2^{m-1}}}$. Note that the case of $m = 3$ can provide almost the same results of [CLN11] described in Sect. 4.2.1. The correctness of Table 4.5 is provided in the following theorem. Before describing the theorems, the author presents the knowledge of the quadratic and cubic residue properties in $\mathbb{F}^*_{p(z)}$ in the following Lemma 4.19.

Lemma 4.19. For the symbols $(\frac{\cdot}{p(z)})$ and $(\frac{\cdot}{p(z)})_3$, the following is true.

(a) For $m = 1$,

$$\left(\frac{-1}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 1 \pmod{12}, \\ -1 & \text{if } z \equiv 4, 7, 10 \pmod{12}. \end{cases} \tag{4.55}$$

For $m > 1$,

$$\left(\frac{-1}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 1, 10 \pmod{12}, \\ -1 & \text{if } z \equiv 4, 7 \pmod{12}. \end{cases} \tag{4.56}$$

Table 4.5: The field and curve options for the proposed BLS subfamilies of curves with $k = 2^m \cdot 3$ for any $m > 0$.

(a) $m = 1$

z	Tower	BLS curve $E/\mathbb{F}_{p(z)}$	Twist $E'/\mathbb{F}_{p(z)^{2^{m-1}}}$
(mod 36)	*(see Theorem 4.20)*	*(see Theorem 4.21)*	*(see Theorem 4.22)*
7		$y^2 = x^3 + 1$	$y^2 = x^3 - 4$
$16, 34$	$\mathbb{F}_{p(z)^2} \cong \mathbb{F}_{p(z)}(\alpha)$	$y^2 = x^3 + 4$	$y^2 = x^3 - 1$
31	$\mathbb{F}_{p(z)^6} \cong \mathbb{F}_{p(z)^2}(\beta)$	$y^2 = x^3 + 1$	$y^2 = x^3 - 1/4$
$10, 28$		$y^2 = x^3 + 16$	$y^2 = x^3 - 1$

(b) $m > 1$

z	Tower	BLS curve $E/\mathbb{F}_{p(z)}$	Twist $E'/\mathbb{F}_{p(z)^{2^{m-1}}}$
(mod 72)	*(see Theorem 4.20)*	*(see Theorem 4.21)*	*(see Theorem 4.22)*
7		$y^2 = x^3 + 1$	$y^2 = x^3 + 1/\zeta$
16	$\mathbb{F}_{p(z)^2} \cong \mathbb{F}_{p(z)}(\alpha)$	$y^2 = x^3 + 4$	$y^2 = x^3 + 4\zeta$
31	$\mathbb{F}_{p(z)^{2^m \cdot 3}} \cong \mathbb{F}_{p(z)^2}(\beta)$	$y^2 = x^3 + 1$	$y^2 = x^3 + \zeta$
64		$y^2 = x^3 - 2$	$y^2 = x^3 - 2/\zeta$

(b) For $m = 1$,

$$\left(\frac{2}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 1, 19 \pmod{24}, \\ -1 & \text{if } z \equiv 4, 7, 10, 13, 16, 22 \ (\mathrm{mod} 24). \end{cases} \tag{4.57}$$

For $m = 2$,

$$\left(\frac{2}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 1, 4, 10, 19 \pmod{24}, \\ -1 & \text{if } z \equiv 7, 13, 16, 22 \pmod{24}. \end{cases} \tag{4.58}$$

For $m > 2$,

$$\left(\frac{2}{p(z)}\right) = \begin{cases} 1 & \text{if } z \equiv 1, 4, 19, 22 \pmod{24}, \\ -1 & \text{if } z \equiv 7, 10, 13, 16 \pmod{24}. \end{cases} \tag{4.59}$$

(c) For $m > 0$,

$$\left(\frac{2}{p(z)}\right)_3 \begin{cases} = 1 & \text{if } z \equiv 1, 4 \pmod{18}, \\ \neq 1 & \text{if } z \equiv 7, 10, 13, 16 \pmod{18}. \end{cases} \tag{4.60}$$

Proof of Lemma 4.19. (a) and (b): The author refers to Lemma 2.25 and verifies the value of $p(z)$ modulo 4 and 8. As a result, (a) and (b) are obtained.

(c): The author refers to Euler's conjecture given in Lemma 2.28. In the following, the author classifies z satisfying $z \equiv 1 \pmod 3$ into two cases, i.e., $z \equiv 1 \pmod 6$ and $z \equiv 4 \pmod 6$.

If $z \equiv 1 \pmod 6$, $p(z)$ can be modified as follows:

$$\begin{aligned} p(z) &= \left(\frac{t(z)}{2}\right)^2 + 3\left(\frac{V(z)}{2}\right)^2 \\ &= \left(\frac{z+1}{2}\right)^2 + 3\left(\frac{z-1}{6} \cdot (2z^{2^{m-1}} - 1)\right)^2. \end{aligned} \tag{4.61}$$

For $b(z) = (z-1)/6 \cdot (2z^{2^{m-1}} - 1)$, if $z \equiv 1 \pmod{18}$ then 3 divides $b(z)$; if $z \equiv 7, 13 \pmod{18}$ then 3 does not divides $b(z)$. Thus, according to (b) in Lemma 2.28, if $z \equiv 1 \pmod{18}$ then $(\frac{2}{p})_3 = 1$; if $z \equiv 7, 13 \pmod{18}$ then $(\frac{2}{p})_3 \neq 1$.

If $z \equiv 4 \pmod 6$, $p(z)$ can be represented as follows:

$$\begin{aligned} p(z) &= \left(\frac{t(z) - 3V(z)}{4}\right)^2 + 3\left(\frac{t(z) + V(z)}{4}\right)^2 \\ &= \left(\frac{-(z-1) \cdot z^{2^{m-1}} + z}{2}\right)^2 + 3\left(\frac{(z-1) \cdot z^{2^{m-1}} + z + 2}{6}\right)^2. \end{aligned} \tag{4.62}$$

For $b(z) = ((z-1) \cdot z^{2^{m-1}} + z + 2)/6$, if $z \equiv 4 \pmod{18}$ then 3 divides $b(z)$; if $z \equiv 10, 16 \pmod{18}$ then 3 does not divide $b(z)$. In the same manner, it is obtained that if $z \equiv 4 \pmod{18}$ then $(\frac{2}{p(z)})_3 = 1$; if $z \equiv 10, 16 \pmod{18}$ then $(\frac{2}{p(z)})_3 \neq 1$. □

Then, the author provides Theorems 4.20, 4.21, and 4.22 associated with the construction of the tower of extension fields, the BLS curve with $k = 2^m \cdot 3$, and its correct twist.

Theorem 4.20. If z satisfies the condition Eq. (4.54), the following tower of extension fields is always available. For $m = 1$,

$$\begin{cases} \mathbb{F}_{p(z)^2} & \cong \mathbb{F}_{p(z)}[x]/(x^2+1) & \cong \mathbb{F}_{p(z)}(\alpha), \\ \mathbb{F}_{p(z)^6} & \cong \mathbb{F}_{p(z)^2}[x]/(x^3-2) & \cong \mathbb{F}_{p(z)^2}(\beta), \end{cases} \tag{4.63}$$

where α and β are elements in $\mathbb{F}_{p(z)^2}$ and $\mathbb{F}_{p(z)^6}$ such that $\alpha^2 = -1$ and $\beta^3 = 2$, respectively. For $m > 1$,

$$\begin{cases} \mathbb{F}_{p(z)^2} & \cong \mathbb{F}_{p(z)}[x]/(x^2+1) & \cong \mathbb{F}_{p(z)}(\alpha), \\ \mathbb{F}_{p(z)^{2^m \cdot 3}} & \cong \mathbb{F}_{p(z)^2}[x]/(x^{2^{m-1} \cdot 3} - (\alpha+1)) & \cong \mathbb{F}_{p(z)^2}(\beta), \end{cases} \tag{4.64}$$

where α and β are elements in $\mathbb{F}_{p(z)^2}$ and $\mathbb{F}_{p(z)^{2^m \cdot 3}}$ such that $\alpha^2 = -1$ and $\beta^{2^{m-1} \cdot 3} = \alpha + 1$, respectively.

Proof of Theorem 4.20. For $m = 1$, to admit the tower of extension fields, the binomials

$x^2 + 1$ and $x^3 - 2$ must be irreducible in $\mathbb{F}_{p(z)}[x]$ and $\mathbb{F}_{p(z)^2}[x]$, respectively. According to (a) in Lemma 4.4, the binomial $x^2 + 1$ is irreducible in $\mathbb{F}_{p(z)}[x]$ if -1 is quadratic non-residue in $\mathbb{F}^*_{p(z)}$. The binomial $x^3 - 2$ is irreducible in $\mathbb{F}_{p(z)^2}[x]$ if the norm of 2, which is computed as $N_{\mathbb{F}_{p(z)^2}/\mathbb{F}_{p(z)}}(2) = 2 \cdot 2^{p(z)} = 2^2 = 4$, is cubic non-residue in $\mathbb{F}^*_{p(z)}$. Note that (b) in Lemma 4.4 is satisfied for both cases. Since it is found that if z satisfies Eq. (4.54), $(\frac{-1}{p(z)}) = -1$ and $(\frac{2}{p(z)})_3 \neq 1$ which results in $(\frac{4}{p(z)})_3 \neq 1$ from Lemma 4.19, the tower is available.

Similarly, for $m > 1$, to admit the tower of extension fields, the binomials $x^2 + 1$ and $x^{2^{m-1}\cdot 3} - (\alpha + 1)$ must be irreducible in $\mathbb{F}_{p(z)}[x]$ and $\mathbb{F}_{p(z)^2}[x]$, respectively. According to (a) in Lemma 4.4, the binomial $x^2 + 1$ is irreducible in $\mathbb{F}_{p(z)}[x]$ if -1 is quadratic non-residue in $\mathbb{F}^*_{p(z)}$. The binomial $x^{2^{m-1}\cdot 3} - (\alpha + 1)$ is irreducible in $\mathbb{F}_{p(z)^2}[x]$ if the the norm of $\alpha + 1$, which is computed by $N_{\mathbb{F}_{p(z)^2}/\mathbb{F}_{p(z)}}(\alpha + 1) = (\alpha + 1) \cdot (\alpha + 1)^{p(z)} = (\alpha + 1) \cdot (-\alpha + 1) = -\alpha^2 + 1 = 2$, is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)}$. Besides, (b) in Lemma 4.4 is satisfies for both cases. Since it is found that if z satisfies Eq. (4.54), $(\frac{-1}{p(z)}) = -1$, $(\frac{2}{p(z)}) = -1$, and $(\frac{2}{p(z)})_3 \neq 1$ from Lemma 4.19, the tower is available. □

Theorem 4.21. Under the same assumptions as in Theorem 4.20, the BLS curve $E/\mathbb{F}_{p(z)}$ can be determined as follows: For $m = 1$,

$$E/\mathbb{F}_{p(z)} : \begin{cases} y^2 = x^3 + 1 & \text{if } z \equiv 7, 31 \pmod{36}, \\ y^2 = x^3 + 4 & \text{if } z \equiv 16, 34 \pmod{36}, \\ y^2 = x^3 + 16 & \text{if } z \equiv 10, 28 \pmod{36}. \end{cases} \tag{4.65}$$

For $m > 1$,

$$E/\mathbb{F}_{p(z)} : \begin{cases} y^2 = x^3 + 1 & \text{if } z \equiv 7, 31 \pmod{72}, \\ y^2 = x^3 + 4 & \text{if } z \equiv 16 \pmod{72}, \\ y^2 = x^3 - 2 & \text{if } z \equiv 64 \pmod{72}. \end{cases} \tag{4.66}$$

Proof of Theorem 4.21. The author verifies the cofactors of the possible group orders to determine the coefficient b of the BLS curve by using Lemma 4.5. From the definition, the curve with the group order $n(z) = p(z) + 1 - t(z)$ is the BLS curve.

If $z \equiv 7, 31 \pmod{36}$ for $m = 1$; $z \equiv 7, 31 \pmod{72}$ for $m > 1$, then $n(z)$ is divisible by 6 but the other group orders are not divisible by 6. According to (a) and (b) in Lemma 4.5, the coefficient b of the BLS curve is quadratic and cubic residue element b in $\mathbb{F}^*_{p(z)}$. Such the coefficient can be chosen as $b = 1$ since it is obvious that $(\frac{1}{p(z)}) = 1$ and $(\frac{1}{p(z)})_3 = 1$.

Similarly, if $z \equiv 16, 34 \pmod{36}$ for $m = 1$; $z \equiv 16 \pmod{72}$ for $m > 1$, $n(z)$ is always divisible by 3 but is not divisible by 2 and 9, however, the other group orders do not have such the properties of cofactors. Thus, according to (a) and (b) in Lemma 4.5,

b is quadratic residue and cubic non-residue in $\mathbb{F}^*_{p(z)}$ and $4b$ is cubic non-residue in $\mathbb{F}^*_{p(z)}$. Then, the coefficient b of the BLS curve can be explicitly chosen as $b = 4$ since $(\frac{2}{p(z)})_3 \neq 1$ from Lemma 4.19.

Finally, if $z \equiv 10, 28 \pmod{36}$ for $m = 1$; $z \equiv 64 \pmod{72}$ for $m > 1$, 9 always divides $n(z)$ but 2 does not divide $n(z)$ and the other group orders are not divisible by 9. According to (a) and (c) in Lemma 4.5, it is found that b is quadratic residue and cubic non-residue in $\mathbb{F}^*_{p(z)}$, and $4b$ is cubic residue in $\mathbb{F}^*_{p(z)}$. Such the coefficient b of the BLS curve can be chosen as $b = 16$ since $(\frac{2}{p(z)})_3 \neq 1$ from Lemma 4.19. For $m > 1$, since the quadratic and cubic residue properties of -2 and 16 are exactly the same, $b = -2$ can also be chosen for the BLS curve. □

Theorem 4.22. Suppose that the tower of extension fields is constructed as in Theorem 4.20 and $E/\mathbb{F}_{p(z)}$ be the BLS curve determined as in Theorem 4.21. Then, the correct twist $E'/\mathbb{F}_{p(z)^{2^{m-1}}}$ of degree 6 of E can be determined as follows: For $m = 1$,

$$E'/\mathbb{F}_{p(z)^{2^{m-1}}} : \begin{cases} y^2 = x^3 - 4 & \text{if } z \equiv 7 \pmod{36}, \\ y^2 = x^3 - 1/4 & \text{if } z \equiv 31 \pmod{36}, \\ y^2 = x^3 - 1 & \text{if } z \equiv 10, 16, 28, 34 \pmod{36}. \end{cases} \tag{4.67}$$

For $m > 1$, letting $\zeta = \beta^6 \in \mathbb{F}_{p(z)^{2^{m-1}}}$ with $\beta \in \mathbb{F}_{p(z)^{2^m \cdot 3}}$ such that $\beta^{2^{m-1} \cdot 3} = \alpha + 1$,

$$E'/\mathbb{F}_{p(z)^{2^{m-1}}} : \begin{cases} y^2 = x^3 + 1/\zeta & \text{if } z \equiv 7 \pmod{72}, \\ y^2 = x^3 + 4\zeta & \text{if } z \equiv 16 \pmod{72}, \\ y^2 = x^3 + \zeta & \text{if } z \equiv 31 \pmod{72}, \\ y^2 = x^3 - 2/\zeta & \text{if } z \equiv 64 \pmod{72}. \end{cases} \tag{4.68}$$

Proof of Theorem 4.22. The author verifies the cofactors of the group order $n'(z)$ of the correct twist $E'/\mathbb{F}_{p(z)^{2^{m-1}}} : y^2 = x^3 + b'$ to determine b' by using Lemma 4.5. Then, b' can be represented as $b' = b/\delta$, where b is the coefficient of the BLS curve and δ is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)^{2^{m-1}}}$. The author also verifies the cofactors of the group order $n''(z)$ of the twist $E''/\mathbb{F}_{p(z)^{2^{m-1}}} : y^2 = x^3 + b''$ of degree 2 of E', where $b'' = b'/\delta^3 = b \cdot \delta^4$. Note that $n'(z)$ is derived as in Theorem 4.11 and $n''(z) = 2p(z)^{2^{m-1}} + 2 - n'(z)$ from Eq. (4.8).

For $m = 1$, if $z \equiv 7 \pmod{36}$, it is found that $n'(z)$ is not divisible by 2, 3, and 9. It is also found that $n''(z)$ is divisible by 3, but is not divisible by 2 and 9. Thus, according to Lemma 4.5, the following information is obtained.

(a) b' is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)}$.

(b) b'' is quadratic residue and cubic non-residue in $\mathbb{F}^*_{p(z)}$.

(c) $4b''$ is cubic non-residue in $\mathbb{F}^*_{p(z)}$.

In this condition, the coefficient b of the BLS curve is determined as $b = 1$ and -4 is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)}$. Thus, the coefficient b' of the correct twist $E'/\mathbb{F}^*_{p(z)}$ can be denoted as either $b' = -1/4$ or -4. In addition, the coefficient b'' of the twist $E''/\mathbb{F}^*_{p(z)}$ of degree 2 of E' can also be denoted as either $b'' = 1/4^4$ or 4^4. From the above, it is found that both candidates of b' and b'' satisfy (a) and (b), however, (c) is satisfied if $b'' = (-4)^4$, which leads to $b' = -4$. Thus, $b' = -4$ is obtained. In the same manner, the other cases of $z \equiv 10, 16, 28, 31, 34 \pmod{36}$ can also be obtained.

For $m > 1$, if $z \equiv 7 \pmod{72}$, $n'(z)$ is not divisible by 2, 3, and 9. Besides, if m is even, $n''(z)$ is divisible by 9 but is not divisible by 2, otherwise, $n''(z)$ is divisible by 3, but is not divisible by 2 and 9. Thus, the following information is obtained from Lemma 4.5.

(a) b' is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)^{2^{m-1}}}$.

(b) b'' is quadratic residue and cubic non-residue in $\mathbb{F}^*_{p(z)^{2^{m-1}}}$.

(c) If m is even, $4b''$ is cubic residue in $\mathbb{F}^*_{p(z)^{2^{m-1}}}$, otherwise, $4b''$ is cubic non-residue in $\mathbb{F}^*_{p(z)^{2^{m-1}}}$.

Under this condition, the coefficient b of the BLS curve is determined as $b = 1$. Besides, $\zeta = \beta^6$ is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)^{2^{m-1}}}$ since the norm of ζ, which is computed as follows, is quadratic and cubic non-residue in $\mathbb{F}^*_{p(z)}$.

$$\begin{aligned}
N_{\mathbb{F}_{p(z)^{2^{m-1}}}/\mathbb{F}_{p(z)}}(\zeta) &= \zeta^{\sum_{i=0}^{2^{m-1}-1} p(z)^i} = \zeta^{(p^{2^{m-2}}+1)\cdot\sum_{i=0}^{2^{m-2}-1} p(z)^i} \\
&= (-\zeta^2)^{\sum_{i=0}^{2^{m-2}-1} p(z)^i} = (-\zeta^{2^2})^{\sum_{i=0}^{2^{m-3}-1} p(z)^i} \\
&= \cdots \\
&= (-\zeta^{2^{m-2}})^{p(z)+1} = (-\beta^{2^{m-1}\cdot 3})^{p(z)+1} = (-(\alpha+1))^{p(z)+1} \\
&= -(\alpha+1)\cdot(\alpha-1) = 2. \qquad (4.69)
\end{aligned}$$

Thus, the coefficient b' of the correct twist $E'/\mathbb{F}^*_{p(z)^{2^{m-1}}}$ can be denoted as either $b' = 1/\zeta$ or ζ. Besides, the coefficient b'' of the twist $E''/\mathbb{F}^*_{p(z)^{2^{m-1}}}$ of degree 2 of E' can also be denoted as either $b'' = 1/\zeta^4$ or ζ^4. From the above, it is found that both candidates of b' and b'' satisfy (a) and (b). As for (c), since the norm of $4/\zeta^4$ and $4\zeta^4$ are computed as $N_{\mathbb{F}_{p(z)^{2^{m-1}}}/\mathbb{F}_{p(z)}}(4/\zeta^4) = 2^{2\cdot 2^{m-1}-4}$ and $N_{\mathbb{F}_{p(z)^{2^{m-1}}}/\mathbb{F}_{p(z)}}(4\zeta^4) = 2^{2\cdot 2^{m-1}+4}$ in the same manner as the computation of the norm of ζ, respectively, it is found that (c) is satisfied if $b'' = 1/\zeta^4$, which leads to $b' = 1/\zeta$. Thus, $b' = 1/\zeta$ is obtained. The other cases of $z \equiv 16, 31, 64 \pmod{72}$ can also be determined. □

From the above theorems, the equations of E and E' are determined corresponding to z. This gives rise to the twisting and untwisting isomorphisms as in Table 4.6. For $m > 1$,

Table 4.6: Twisting and untwisting isomorphisms for the proposed BLS subfamilies with $k = 2^m \cdot 3$.

(a) $m = 1$

z	Twisting isomorphism	Untwisting isomorphism
(mod 36)	$\phi_6 : E' \to E$	$\phi_6 : E \to E'$
7	$(x, y) \mapsto (-2^{-1}x\beta, -2^{-1}y\alpha)$	$(x, y) \mapsto (-x\beta^2, 2y\alpha)$
$16, 34$	$(x, y) \mapsto (-x\beta^2, 2y\alpha)$	$(x, y) \mapsto (-2^{-1}x\beta, -2^{-1}y\alpha)$
31	$(x, y) \mapsto (-x\beta^2, 2y\alpha)$	$(x, y) \mapsto (-2^{-1}x\beta, -2^{-1}y\alpha)$
$10, 28$	$(x, y) \mapsto (-2x\beta, 4y\alpha)$	$(x, y) \mapsto (-4^{-1}x\beta^2, -4^{-1}y\alpha)$

(b) $m > 1$

z	Twisting isomorphism	Untwisting isomorphism
(mod 72)	$\phi_6 : E' \to E$	$\phi_6 : E \to E'$
7	$(x, y) \mapsto (x\beta^2, y\beta^3)$	$(x, y) \mapsto (\zeta^{-1}x\beta^4, \zeta^{-1}y\beta^3)$
16	$(x, y) \mapsto (\zeta^{-1}x\beta^4, \zeta^{-1}y\beta^3)$	$(x, y) \mapsto (x\beta^2, y\beta^3)$
31	$(x, y) \mapsto (\zeta^{-1}x\beta^4, \zeta^{-1}y\beta^3)$	$(x, y) \mapsto (x\beta^2, y\beta^3)$
64	$(x, y) \mapsto (x\beta^2, y\beta^3)$	$(x, y) \mapsto (\zeta^{-1}x\beta^4, \zeta^{-1}y\beta^3)$

the twisting and untwisting isomorphisms constructions are similar to the case of the BN subfamilies of curves. Since $\{1, \beta, \ldots, \beta^5\}$ is a basis of the 6-th dimensional vector space of $\mathbb{F}_{p(z)^{2^{m-1}}}$, the isomorphisms are low complexity. For $m = 1$, the isomorphisms are also determined by using the relations $\alpha^2 = -1$ and $\beta^3 = 2$. For example, if $z \equiv 7 \pmod{36}$, it is obtained $\phi_6 : E' \to E, (x, y) \mapsto ((-4)^{-1/3}x, (-4)^{-1/2}y)$ where the image of (x, y) under ϕ_6 is given by $((-4)^{-1/3}x, (-4)^{-1/2}y) = (-2^{-1}2^{1/3}x, -(-1)^{1/2}2^{-1}y) = (-2^{-1}x\beta, -2^{-1}y\alpha)$; $\phi_6^{-1} : E \to E', (x, y) \mapsto ((-4)^{1/3}x, (-4)^{1/2}y)$ where $((-4)^{1/3}x, (-4)^{1/2}y) = (-2^{2/3}x, (-1)^{1/2}2y) = (-x\beta^2, 2y\alpha)$. For the other cases of $z \equiv 10, 16, 28, 31, 34 \pmod{36}$, the formulas of isomorphisms are explicitly obtained. Note that the isomorphisms are also efficiently computable since $\{1, \alpha\} \times \{1, \beta, \beta^2\}$ is a basis of the 6-th dimensional vector space of $\mathbb{F}_{p(z)}$.

(ii) The case of $k = 3^n$ for any $n > 0$

Let z be an integer parameter for the BLS family of curves with $k = 3^n$ where $n > 0$ is an arbitrary integer. The author proposes to restrict z by

$$z \equiv 4 \pmod{6}. \tag{4.70}$$

Once finding z under the above restriction, the specific BLS subfamily with the options are obtained.

(i) A fixed tower of extension fields with one of the best performing arithmetics is always available;

Table 4.7: The field and curve options for the proposed BLS subfamily of curves with $k = 3^n$ for any $n > 0$.

z (mod 6)	Tower *(see Theorem 4.24)*	BLS curve $E/\mathbb{F}_{p(z)}$ *(see Theorem 4.25)*	Twist $E'/\mathbb{F}_{p(z)^{3^{n-1}}}$ *(see Conjecture 4.26)*
4	$\mathbb{F}_{p(z)^{3^n}} \cong \mathbb{F}_{p(z)}(\alpha)$	$y^2 = x^3 + 16$	$y^2 = x^3 + 16\zeta^2$

(ii) The BLS curve $E/\mathbb{F}_{p(z)}$ is immediately determined;

In addition to this, the BLS subfamily might have the option (iii) the correct twist $E'/\mathbb{F}_{p(z)^{3^{n-1}}}$ is also immediately determined. If that is true, these constructions also enable one of the simplest twist isomorphisms. The details of the field and curve options are found in Table 4.7, where α is an element in $\mathbb{F}_{p(z)^{3^n}}$ such that $\alpha^{3^n} = 2$ and $\zeta = \alpha^3$. The author also provides Theorems 4.24 and 4.25 which show the correctness that the proposed BLS subfamily has the options (i) and (ii), respectively. Although it is required another theorem for the discussion, unfortunately, the author does not complete proof, yet. Therefore, the author shows Conjecture 4.26 about the options (iii). Before providing the theorems and conjecture, the knowledge of the quadratic and cubic residue properties in $\mathbb{F}^*_{p(z)}$ is provided in the following Lemma 4.23.

Lemma 4.23. For any $n > 0$, the following is true.

$$\left(\frac{2}{p(z)}\right)_3 \begin{cases} = 1 & \text{if } z \equiv 1 \pmod 6, \\ \neq 1 & \text{if } z \equiv 4 \pmod 6. \end{cases} \tag{4.71}$$

Proof of Lemma 4.23. The author classifies z into $z \equiv 1 \pmod 6$ and $z \equiv 4 \pmod 6$.

If $z \equiv 1 \pmod 6$, $p(z)$ can be modified as follows:

$$\begin{aligned} p(z) &= \left(\frac{t(z)}{2}\right)^2 + 3\left(\frac{V(z)}{2}\right)^2 \\ &= \left(\frac{z+1}{2}\right)^2 + 3\left(\frac{z-1}{6} \cdot (2z^{3^{n-1}} + 1)\right)^2. \end{aligned} \tag{4.72}$$

For $b(z) = (z-1)/6 \cdot (2z^{3^{n-1}} + 1)$, 3 divides $b(z)$. According to Lemma 2.28, 2 is cubic residue in $\mathbb{F}^*_{p(z)}$ under this condition.

Similarly, if $z \equiv 4 \pmod 6$, $p(z)$ can be modified as follows:

$$\begin{aligned} p(z) &= \left(\frac{t(z) + 3V(z)}{4}\right)^2 + 3\left(\frac{t(z) - V(z)}{4}\right)^2 \\ &= \left(\frac{(z-1)\cdot z^{3^{n-1}} + z}{2}\right)^2 + 3\left(\frac{-(z-1)\cdot z^{3^{n-1}} + z + 2}{6}\right)^2. \end{aligned} \tag{4.73}$$

For $b(z) = (-(z-1) \cdot z^{3^{n-1}} + z + 2)/6$, 3 does not divide $b(z)$. Thus, it is obtained that 2 is cubic non-residue in $\mathbb{F}^*_{p(z)}$ from Lemma 2.28. □

Then, the author provides Theorems 4.24 and 4.25 associated with the construction of the tower of extension fields and the BLS curve.

Theorem 4.24. If z satisfies Eq. (4.70), the following tower of extension fields is always available.

$$\mathbb{F}_{p(z)^{3^n}} \cong \mathbb{F}_{p(z)}[x]/(x^{3^n} - 2) \cong \mathbb{F}_{p(z)}(\alpha), \tag{4.74}$$

where α is an element in $\mathbb{F}_{p(z)^{3^n}}$ such that $\alpha^{3^n} = 2$.

Proof of Theorem 4.24. To adopt the tower of extension fields given in Eq. (4.74), the binomial $x^{3^n} - 2$ has to be irreducible in $\mathbb{F}_{p(z)}[x]$, i.e., $3 \mid (p(z) - 1)$ and 2 is cubic non-residue in $\mathbb{F}^*_{p(z)}$ from Lemma 4.4. The former requirement is satisfied for any z. If $z \equiv 4 \pmod{6}$, the latter requirement is also satisfied since $(\frac{2}{p(z)})_3 \neq 1$ under this condition as found in Lemma 4.23. □

Theorem 4.25. Under the same assumptions as Theorem 4.24, the BLS curve with $k = 3^n$ is immediately determined as $E/\mathbb{F}_{p(z)} : y^2 = x^3 + 16$ for any $n > 0$.

Proof of Theorem 4.25. The author verifies the cofactors of the possible group orders, which $n(z) = p(z) + 1 - t(z)$ is the group order of the BLS curve. If $z \equiv 4 \pmod 6$, 9 always divides $n(z)$, however, 2 does not divide that. Note that the other group orders cannot be divisible by 9. According to (a) and (c) in Lemma 4.5, the coefficient b of the BLS curve is quadratic residue and cubic non-residue in $\mathbb{F}^*_{p(z)}$ and $4b$ is cubic residue in $\mathbb{F}^*_{p(z)}$. From Lemma 4.23, such the coefficient can be chosen as $b = 16$. □

Unfortunately, it could not determine the correct twist $E'/\mathbb{F}_{p(z)^{3^{n-1}}}$ by using Lemma 4.5 since the field $\mathbb{F}_{p(z)^{3^{n-1}}}$ in which twist is defined always makes the coefficient b of the BLS curves $E/\mathbb{F}_{p(z)}$ being cubic residue in $\mathbb{F}^*_{p(z)^{3^{n-1}}}$. However, the author makes the following prediction from the experimental results of the determination of the twist equation with some small n.

Conjecture 4.26. With z satisfying Eq. (4.70), suppose that the tower of extension fields is constructed as in Theorem 4.24 and $E/\mathbb{F}_{p(z)}$ be the BLS curve determined as in Theorem 4.25. The correct twist of degree 3 of E can be determined as $E'/\mathbb{F}_{p(z)^{3^{n-1}}}$: $y^2 = x^3 + 16\zeta^2$ where $\zeta = \alpha^3 \in \mathbb{F}_{p(z)^{3^{n-1}}}$ with $\alpha \in \mathbb{F}_{p(z)^{3^n}}$ such that $\alpha^{3^n} = 2$.

Note that there is a possibility that Conjecture 4.26 can be proven by using another twist determination technique given by Yasuda et al. in [YTS15], however, their technique is not so simpler than Costello et al.'s one [CLN11] and require the knowledge of number

Table 4.8: The twisting and untwisting isomorphisms for the proposed BLS subfamily of curves with $k = 3^n$.

z (mod 6)	Twisting isomorphism $\phi_3 : E' \to E$	Untwisting isomorphsim $\phi_3 : E \to E'$
4	$(x, y) \mapsto (\zeta^{-1}x\alpha, \zeta^{-1}y)$	$(x, y) \mapsto (x\alpha^2, \zeta y)$

theory. According to [YTS15], the author just finds that if the following Conjecture 4.27 is true, Conjecture 4.26 is true.

Conjecture 4.27. Let ϵ be a primitive cube root of the identity in $\mathbb{F}^*_{p(z)}$ which is represented as $\epsilon \equiv -(1 + t(z) \cdot V(z)^{-1})/2 \pmod{p(z)}$. If $z \equiv 4 \pmod 6$, the following is always true.

$$\epsilon \cdot 2^{\frac{p(z)-1}{3}} \equiv 1 \pmod{p(z)}. \tag{4.75}$$

If the conjectures are true, there are efficient performing twisting and untwisting isomorphisms shown in Table 4.8. Since there is a relation $\zeta = \alpha^3$, the twisting isomorphism is given by $\phi_3 : E' \to E, (x, y) \mapsto (\zeta^{-2/3}x, \zeta^{-2/2}y) = (\zeta^{-1}\zeta^{1/3}x, \zeta^{-1}y) = (\zeta^{-1}x\alpha, \zeta^{-1}y)$. The untwisting isomorphism is also given by $\phi_3 : E' \to E, (x, y) \mapsto (\zeta^{2/3}x, \zeta^{2/2}y) = (x\alpha^2, \zeta y)$.

4.4.4 Sample parameters and evaluation

The author applies the proposal and obtains sample parameters z for generating the proposed BLS subfamilies of curves with $k = 2^m \cdot 3$ and 3^n for $m, n \in \{2, 3\}$, i.e., $k = 9$, 12, 24, and 27. For $k = 24$, although Costello et al. provided many candidates of z in [CLN11], the author reproduces the parameters based on the latest security analysis [Gui20]. According to the suggestions of [Gui20], the curves with $k \in \{9, 12\}$ and $\{24, 27\}$ are adopted for the pairings at the 128 and 192-bit security levels, respectively. For the pairings at the 128-bit security, the author searches z which gives rise to $r(z)$ with $\log_2 r(z) \geq 256$ and $p(z)$ with $\log_2 p(z)^k \geq 5472$ for $k = 9$ and $\log_2 p(z)^k \geq 5376$ for $k = 12$. For the pairings at the 192-bit security, the author also searches z which gives rise to $r(z)$ with $\log_2 r(z) \geq 384$ and $p(z)$ with $\log_2 p(z)^k \geq 12202$ for $k = 24$ and $\log_2 p(z)^k \geq 11496$ for $k = 27$. The parameters z having the low-Hamming weight are found for efficiency reasons of the pairings. For $k = 3^n$ such that $2 \nmid k$, it is effective to choose z with the specific binary representations such that $z = \sum_{i=0}^{\log_2 z - 1} 2^i t_i$ where $t_i \in \{0, 1\}$ or $\{-1, 0\}$ for searching z. The details of the fact are described in App. C.

Tables 4.9, 4.10, 4.11, and 4.12 show the sample parameters z for the pairings with the BLS family of curves with $k = 9, 12, 24$, and 27, respectively. Note that all the seeds for

Table 4.9: Sample seeds z for the attractive BLS subfamily of curves with $k = 9$ for the pairings at the 128-bit security level.

No.	z (mod 6)	Seed z	HW	Bit size		
				$p(z)$	$p(z)^k$	$r(z)$
1	4	$-2^{77} - 2^{62} + 2^{20}$	3	615	5530	461
2	4	$-2^{77} - 2^{19} + 2^{9}$	3	615	5530	461
3	4	$-2^{77} - 2^{75} - 2^{32}$	3	617	5553	463
4	4	$+2^{77} + 2^{62} + 2^{35} + 2^{25}$	4	615	5530	461
5	4	$+2^{76} + 2^{74} + 2^{46} + 2^{22}$	4	609	5481	457
6	4	$-2^{76} - 2^{75} - 2^{70} - 2^{25} - 2^{1}$	5	612	5501	459
7	4	$-2^{76} - 2^{74} - 2^{65} - 2^{63} - 2^{19}$	5	609	5481	457
8	4	$-2^{76} - 2^{75} - 2^{57} - 2^{51} - 2^{18}$	5	612	5500	458
9	4	$-2^{76} - 2^{74} - 2^{54} - 2^{34} - 2^{28}$	5	609	5481	457
10	4	$+2^{76} + 2^{74} + 2^{42} + 2^{31} + 2^{27}$	5	609	5481	457
11	4	$+2^{76} + 2^{75} + 2^{74} + 2^{60} + 2^{19}$	5	613	5516	460
12	4	$+2^{76} + 2^{74} + 2^{65} + 2^{54} + 2^{11}$	5	609	5481	457

Table 4.10: Sample seeds z for the attractive BLS subfamilies of curves with $k = 12$ for the pairings at the 128-bit security levels.

No.	z (mod 72)	Seed z	HW	Bit size		
				$p(z)$	$p(z)^k$	$r(z)$
1	7	$-2^{76} - 2^{28} - 2^{23} - 2^{0}$	4	455	5453	305
2	7	$+2^{75} - 2^{61} + 2^{31} - 2^{0}$	4	449	5381	300
3	7	$-2^{75} + 2^{52} + 2^{40} + 2^{7} - 2^{0}$	5	449	5381	300
4	7	$-2^{75} + 2^{54} - 2^{36} + 2^{4} - 2^{0}$	5	449	5381	300
5	7	$-2^{75} + 2^{70} + 2^{50} - 2^{44} - 2^{0}$	5	449	5378	300
6	16	$-2^{77} - 2^{59} + 2^{9}$ [BD19]	3	461	5525	309
7	16	$-2^{77} + 2^{50} + 2^{33}$ [BD19]	3	461	5525	308
8	16	$+2^{75} + 2^{65} - 2^{45} - 2^{10}$	4	449	5382	301
9	16	$-2^{75} - 2^{26} + 2^{21} - 2^{10}$	4	449	5381	301
10	16	$+2^{75} - 2^{60} + 2^{45} + 2^{24}$	4	449	5381	300
11	31	$+2^{76} - 2^{72} - 2^{12} - 2^{0}$	4	454	5447	304
12	31	$+2^{75} + 2^{40} - 2^{36} - 2^{0}$	4	449	5381	301
13	31	$+2^{75} - 2^{70} - 2^{5} - 2^{0}$	4	449	5378	300
14	31	$-2^{75} - 2^{55} - 2^{42} + 2^{40} - 2^{0}$	5	449	5381	301
15	31	$-2^{75} - 2^{51} + 2^{40} - 2^{14} - 2^{0}$	5	449	5381	301
16	64	$+2^{75} + 2^{54} - 2^{27}$	3	449	5381	301
17	64	$+2^{76} - 2^{70} + 2^{66}$	3	455	5452	304
18	64	$+2^{75} + 2^{69} + 2^{64} + 2^{35}$	4	449	5383	301
19	64	$+2^{75} + 2^{55} - 2^{54} - 2^{27}$	4	449	5381	301
20	64	$-2^{75} + 2^{45} + 2^{43} - 2^{6}$	4	449	5381	300

Table 4.11: Sample seeds z for the attractive BLS subfamilies of curves with $k = 24$ for the pairings at the 192-bit security levels.

No.	z (mod 72)	Seed z	HW	Bit size		
				$p(z)$	$p(z)^k$	$r(z)$
1	7	$-2^{51} - 2^{28} + 2^{11} - 2^{0}$ [CLN11]	4	509	12202	409
2	7	$+2^{51} - 2^{32} - 2^{20} + 2^{3} - 2^{0}$	5	509	12202	408
3	7	$-2^{51} - 2^{34} + 2^{24} + 2^{14} - 2^{0}$	5	509	12202	409
4	7	$-2^{51} + 2^{30} - 2^{24} - 2^{13} - 2^{0}$	5	509	12202	408
5	7	$-2^{51} - 2^{48} - 2^{21} - 2^{13} - 2^{0}$	5	511	12243	410
6	16	$+2^{51} + 2^{41} + 2^{34} + 2^{11}$	4	509	12203	409
7	16	$-2^{51} - 2^{48} + 2^{45} + 2^{39}$ [CLN11]	4	510	12238	410
8	16	$+2^{51} + 2^{41} - 2^{36} - 2^{5}$	4	509	12203	409
9	16	$+2^{52} - 2^{49} + 2^{20} + 2^{10}$	4	517	12396	415
10	16	$+2^{52} - 2^{48} - 2^{46} + 2^{15}$	4	518	12414	416
11	31	$+2^{51} - 2^{15} - 2^{8} - 2^{0}$ [CLN11]	4	509	12202	408
12	31	$-2^{52} - 2^{28} + 2^{18} - 2^{0}$ [CLN11]	4	519	12442	417
13	31	$-2^{51} + 2^{30} - 2^{19} + 2^{11} - 2^{0}$	5	509	12202	408
14	31	$+2^{51} + 2^{27} - 2^{12} + 2^{3} - 2^{0}$	5	509	12202	409
15	31	$-2^{51} + 2^{38} - 2^{10} + 2^{4} - 2^{0}$	5	509	12202	408
16	64	$-2^{51} + 2^{34} - 2^{4}$	3	509	12202	408
17	64	$-2^{52} - 2^{39} + 2^{16}$ [BD19]	3	519	12443	417
18	64	$-2^{51} + 2^{35} - 2^{34} - 2^{4}$	4	509	12202	408
19	64	$+2^{51} + 2^{27} + 2^{17} + 2^{4}$	4	509	12202	409
20	64	$+2^{51} - 2^{39} + 2^{33} - 2^{10}$	4	509	12202	408

Table 4.12: Sample seeds z for the attractive BLS subfamily of curves with $k = 27$ for the pairings at the 192-bit security level.

No.	z (mod 6)	Seed z	HW	Bit size		
				$p(z)$	$p(z)^k$	$r(z)$
1	4	$-2^{22} - 2^{12} + 2^{8} - 2^{6}$	4	439	11838	395
2	4	$+2^{23} - 2^{18} + 2^{14} - 2^{10}$	4	458	12354	412
3	4	$-2^{23} - 2^{17} + 2^{8} - 2^{1}$	4	459	12390	413
4	4	$+2^{22} + 2^{18} + 2^{13} + 2^{4} + 2^{1}$	5	441	11886	397
5	4	$-2^{22} - 2^{21} - 2^{19} - 2^{6} - 2^{1}$	5	453	12216	408
6	4	$-2^{23} - 2^{17} - 2^{11} - 2^{10} - 2^{8}$	5	459	12390	413
7	4	$-2^{23} - 2^{18} - 2^{8} - 2^{7} - 2^{3}$	5	460	12402	414
8	4	$+2^{22} + 2^{21} + 2^{19} + 2^{14} + 2^{9} + 2^{7}$	6	453	12218	408
9	4	$+2^{22} + 2^{20} + 2^{14} + 2^{9} + 2^{4} + 2^{2}$	6	445	12014	401
10	4	$+2^{22} + 2^{14} + 2^{11} + 2^{8} + 2^{4} + 2^{2}$	6	439	11841	395
11	4	$+2^{22} + 2^{17} + 2^{9} + 2^{7} + 2^{5} + 2^{4}$	6	440	11862	396
12	4	$-2^{22} - 2^{21} - 2^{15} - 2^{13} - 2^{11} - 2^{9}$	6	451	12159	406
13	4	$-2^{22} - 2^{11} - 2^{10} - 2^{9} - 2^{8} - 2^{6}$	6	439	11838	395
14	4	$-2^{22} - 2^{11} - 2^{10} - 2^{9} - 2^{6} - 2^{4}$	6	439	11838	395
15	4	$-2^{22} - 2^{21} - 2^{17} - 2^{12} - 2^{10} - 2^{8}$	6	451	12170	406

the cases $k = 9$ and 27 can provide the correct twist in Table 4.7. The author evaluates the seeds for the pairings on BLS curves by an implementation. The implementation adopts the ate pairing $e'_{a_T}(Q', \phi_3^{-1}(P))$ on E' with efficient formulas for computing Miller's algorithm given by Costello et al. in [CLN10]. The projective and affine formulas are adopted for the pairings at the 128- and 192-bit security levels, respectively. For the case of the curves with $k = 9$ and 27, the revised version of Miller's algorithm in App. C is adopted as appropriate according to z of the loop parameter. For the final exponentiation algorithm, it is adopted the state-of-the-art algorithm given by Hayashida et al. in [HHT20]. For the curves with $k = 12$ and 24, it is also adopted the compressed squaring in the cyclotomic subgroup in the full extension field given by Karabina in [Kar13], which is available during the computation of the hard part of the final exponentiation. Unfortunately, the curves with $k = 9$ and 27 cannot have such efficient squaring in the final exponentiation.

With the above optimizations, the author implements the software for executing the pairings by C language. The big integer arithmetics are implemented by using `mp_limb_t` data type of the GMP library [tea15]. The software is compiled with GCC 8.3.0 with the option `-O2 -march=native` and is executed by 3.50GHz Intel(R) Core(TM) i7-7567U CPU running macOS Big Sur version 11.2.3. To evaluate the parameters, the average execution times of 100,000 trials of Miller's algorithm and final exponentiation are measured. Note that the measurement is performed by repeating the functions for 1,000 random inputs 100 times.

Tables 4.13, 4.14, 4.15, and 4.16 show the results of the average execution time of Miller's algorithm and final exponentiation for the pairings on the BLS curves with $k = 9$, 12, 24, and 27, respectively. The results are analyzed as follows:

- Comparing the results between the same curves, the execution times of the pairings on the curves with small HW(z) are typically faster than that of the curves with large HW(z) since the performance of the pairing depends on the signed binary representation of z. Although some results do not follow this trend, the author considers that it might come from the effects of cache and parallel processing. Rather than that, the execution times more strongly depend on the word size of $p(z)$. For example, for the curves with $k = 24$, the parameters of No. 18 could not result in the best performing pairing due to the word size of $p(z)$ even though that has the smallest Hamming weight. As for the curves with $k = 12$ and 24, although there is a difference in the untwisting isomorphisms between the congruence classes of z as in Table 4.8, the author could not find the difference between the congruence classes of z. Note that it is theoretically better to choose z satisfying $z \equiv 16, 31 \pmod{72}$ under this assumption. The author considers that this effect might be small enough to ignore in this environment.
- Comparing the results between the same security levels, it is clear that the curves

with $k = 12$ and $k = 24$ result in higher performance of the pairings at the 128 and 192-bit security levels compared with the curves with $k = 9$ and 27, respectively. This cause of that the curves with $k = 9$ and 27 have low degree twists which can have disadvantages for computing Miller's algorithm. Besides, these curves cannot result in an efficient squaring in the cyclotomic multiplicative subgroup of the full extension field for computing the final exponentiation.

As a result, among the candidates shown in this paper, the author suggests the curves with $k = 12$ with the parameters of No. 1, 6, 7, 11, and 17 for the pairing at the 128-bit security level. The author also suggests the curves with $k = 24$ with the parameters of No. 16 for the pairing at the 192-bit security level.

4.5 Summary of contributions

In this chapter, the author proposes specific restrictions of integer parameter z for generating curves in the BN and BLS subfamilies that have the advantage for the pairings-based cryptography by extending Costello et al.'s work [CLN11]. The proposed subfamilies give rise to the fixed field and curve constructions, which allow us to reduce the initial settings of the pairings. In addition to this, since all z in the certain restriction have the common field and curve constructions, the results can also support to change of z smoothly. For example, if there exists an implementation of the pairing with a certain z satisfying the restriction, z can be updated without changing the implementation of the field and curve arithmetics as long as z is chosen from the same restriction. Thus, if there is progress in the security analyses, the results also allow us to flexibly respond to the update of z without changing implementations as far as possible. Moreover, since the results are available for the BLS curves with $k = 2^m \cdot 3$ and 3^n with any integers $m, n > 0$, the proposed method will be useful for the researcher and implementer of the pairings for a long time.

Table 4.13: Average execution times for computing Miller's algorithm (ML) and final exponentiation (FE) for the pairings on BLS curves with $k = 9$ at the 128-bit security level.

No.	z (mod 6)	Seed z	HW	Word size	ML [ms]	FE [ms]	Total [ms]
1	4	$-2^{77} - 2^{62} + 2^{20}$	3	10	2.38	3.41	5.79
2	4	$-2^{77} - 2^{19} + 2^{9}$	3	10	2.37	3.39	5.76
3	4	$-2^{77} - 2^{75} - 2^{32}$	3	10	2.33	3.38	5.71
4	4	$+2^{77} + 2^{62} + 2^{35} + 2^{25}$	4	10	2.35	3.36	5.71
5	4	$+2^{76} + 2^{74} + 2^{46} + 2^{22}$	4	10	2.34	3.34	5.69
6	4	$-2^{76} - 2^{75} - 2^{70} - 2^{25} - 2^{1}$	5	10	2.38	3.46	5.84
7	4	$-2^{76} - 2^{74} - 2^{65} - 2^{63} - 2^{19}$	5	10	2.41	3.51	5.92
8	4	$-2^{76} - 2^{75} - 2^{57} - 2^{51} - 2^{18}$	5	10	2.38	3.48	5.86
9	4	$-2^{76} - 2^{74} - 2^{54} - 2^{34} - 2^{28}$	5	10	2.39	3.49	5.89
10	4	$+2^{76} + 2^{74} + 2^{42} + 2^{31} + 2^{27}$	5	10	2.37	3.40	5.77
11	4	$+2^{76} + 2^{75} + 2^{74} + 2^{60} + 2^{19}$	5	10	2.34	3.35	5.69
12	4	$+2^{76} + 2^{74} + 2^{65} + 2^{54} + 2^{11}$	5	10	2.37	3.41	5.77

Table 4.14: Average execution times for computing Miller's algorithm (ML) and final exponentiation (FE) for the pairings on BLS curves with $k = 12$ at the 128-bit security level.

No.	z (mod 72)	Seed z	HW	Word size	ML [ms]	FE [ms]	Total [ms]
1	7	$-2^{76} - 2^{28} - 2^{23} - 2^{0}$	4	8	1.54	1.54	3.08
2	7	$+2^{75} - 2^{61} + 2^{31} - 2^{0}$	4	8	1.59	1.60	3.20
3	7	$-2^{75} + 2^{52} + 2^{40} + 2^{7} - 2^{0}$	5	8	1.62	1.69	3.31
4	7	$-2^{75} + 2^{54} - 2^{36} + 2^{4} - 2^{0}$	5	8	1.62	1.69	3.30
5	7	$-2^{75} + 2^{70} + 2^{50} - 2^{44} - 2^{0}$	5	8	1.57	1.64	3.21
6	16	$-2^{77} - 2^{59} + 2^{9}$ [BD19]	3	8	1.53	1.52	3.05
7	16	$-2^{77} + 2^{50} + 2^{33}$ [BD19]	3	8	1.54	1.52	3.06
8	16	$+2^{75} + 2^{65} - 2^{45} - 2^{10}$	4	8	1.59	1.66	3.25
9	16	$-2^{75} - 2^{26} + 2^{21} - 2^{10}$	4	8	1.59	1.66	3.25
10	16	$+2^{75} - 2^{60} + 2^{45} + 2^{24}$	4	8	1.59	1.66	3.25
11	31	$+2^{76} - 2^{72} - 2^{12} - 2^{0}$	4	8	1.51	1.52	3.03
12	31	$+2^{75} + 2^{40} - 2^{36} - 2^{0}$	4	8	1.59	1.61	3.20
13	31	$+2^{75} - 2^{70} - 2^{5} - 2^{0}$	4	8	1.54	1.57	3.11
14	31	$-2^{75} - 2^{55} - 2^{42} + 2^{40} - 2^{0}$	5	8	1.61	1.70	3.30
15	31	$-2^{75} - 2^{51} + 2^{40} - 2^{14} - 2^{0}$	5	8	1.61	1.70	3.30
16	64	$+2^{75} + 2^{54} - 2^{27}$	3	8	1.59	1.58	3.17
17	64	$+2^{76} - 2^{70} + 2^{66}$	3	8	1.52	1.51	3.03
18	64	$+2^{75} + 2^{69} + 2^{64} + 2^{35}$	4	8	1.62	1.67	3.28
19	64	$+2^{75} + 2^{55} - 2^{54} - 2^{27}$	4	8	1.60	1.66	3.27
20	64	$-2^{75} + 2^{45} + 2^{43} - 2^{6}$	4	8	1.60	1.65	3.25

Table 4.15: Average execution times for computing Miller's algorithm (ML) and final exponentiation (FE) for the pairings on BLS curves with $k = 24$ at the 192-bit security level.

No.	z (mod 72)	Seed z	HW	Word size	ML [ms]	FE [ms]	Total [ms]
1	7	$-2^{51} - 2^{28} + 2^{11} - 2^{0}$ [CLN11]	4	8	2.82	5.38	8.20
2	7	$+2^{51} - 2^{32} - 2^{20} + 2^{3} - 2^{0}$	5	8	2.84	5.77	8.62
3	7	$-2^{51} - 2^{34} + 2^{24} + 2^{14} - 2^{0}$	5	8	2.84	5.77	8.60
4	7	$-2^{51} + 2^{30} - 2^{24} - 2^{13} - 2^{0}$	5	8	2.85	5.81	8.66
5	7	$-2^{51} - 2^{48} - 2^{21} - 2^{13} - 2^{0}$	5	8	2.84	5.77	8.61
6	16	$+2^{51} + 2^{41} + 2^{34} + 2^{11}$	4	8	2.79	5.49	8.28
7	16	$-2^{51} - 2^{48} + 2^{45} + 2^{39}$ [CLN11]	4	8	2.81	5.54	8.35
8	16	$+2^{51} + 2^{41} - 2^{36} - 2^{5}$	4	8	2.78	5.48	8.27
9	16	$+2^{52} - 2^{49} + 2^{20} + 2^{10}$	4	9	3.31	6.49	9.80
10	16	$+2^{52} - 2^{48} - 2^{46} + 2^{15}$	4	9	3.32	6.52	9.84
11	31	$+2^{51} - 2^{15} - 2^{8} - 2^{0}$ [CLN11]	4	8	2.81	5.36	8.17
12	31	$-2^{52} - 2^{28} + 2^{18} - 2^{0}$ [CLN11]	4	9	3.36	6.37	9.73
13	31	$-2^{51} + 2^{30} - 2^{19} + 2^{11} - 2^{0}$	5	8	2.83	5.77	8.60
14	31	$+2^{51} + 2^{27} - 2^{12} + 2^{3} - 2^{0}$	5	8	2.84	5.77	8.61
15	31	$-2^{51} + 2^{38} - 2^{10} + 2^{4} - 2^{0}$	5	8	2.85	5.81	8.66
16	64	$-2^{51} + 2^{34} - 2^{4}$	3	8	2.79	5.12	7.91
17	64	$-2^{52} - 2^{39} + 2^{16}$ [BD19]	3	9	3.30	6.03	9.34
18	64	$-2^{51} + 2^{35} - 2^{34} - 2^{4}$	4	8	2.82	5.55	8.37
19	64	$+2^{51} + 2^{27} + 2^{17} + 2^{4}$	4	8	2.81	5.55	8.36
20	64	$+2^{51} - 2^{39} + 2^{33} - 2^{10}$	4	8	2.82	5.57	8.39

Table 4.16: Average execution times for computing Miller's algorithm (ML) and final exponentiation (FE) for the pairings on BLS curves with $k = 27$ at the 192-bit security level.

No.	z (mod 6)	Seed z	HW	Word size	ML [ms]	FE [ms]	Total [ms]
1	4	$-2^{22} - 2^{12} + 2^{8} - 2^{6}$	4	7	2.41	13.1	15.5
2	4	$+2^{23} - 2^{18} + 2^{14} - 2^{10}$	4	8	2.80	15.3	18.1
3	4	$-2^{23} - 2^{17} + 2^{8} - 2^{1}$	4	8	2.84	15.3	18.2
4	4	$+2^{22} + 2^{18} + 2^{13} + 2^{4} + 2^{1}$	5	7	2.31	12.7	15.1
5	4	$-2^{22} - 2^{21} - 2^{19} - 2^{6} - 2^{1}$	5	8	2.70	15.6	18.3
6	4	$-2^{23} - 2^{17} - 2^{11} - 2^{10} - 2^{8}$	5	8	2.80	16.1	18.9
7	4	$-2^{23} - 2^{18} - 2^{8} - 2^{7} - 2^{3}$	5	8	2.78	16.0	18.8
8	4	$+2^{22} + 2^{21} + 2^{19} + 2^{14} + 2^{9} + 2^{7}$	6	8	2.72	15.3	18.0
9	4	$+2^{22} + 2^{20} + 2^{14} + 2^{9} + 2^{4} + 2^{2}$	6	7	2.36	13.3	15.7
10	4	$+2^{22} + 2^{14} + 2^{11} + 2^{8} + 2^{4} + 2^{2}$	6	7	2.35	13.2	15.6
11	4	$+2^{22} + 2^{17} + 2^{9} + 2^{7} + 2^{5} + 2^{4}$	6	7	2.35	13.3	15.6
12	4	$-2^{22} - 2^{21} - 2^{15} - 2^{13} - 2^{11} - 2^{9}$	6	8	2.82	16.6	19.4
13	4	$-2^{22} - 2^{11} - 2^{10} - 2^{9} - 2^{8} - 2^{6}$	6	7	2.44	14.3	16.7
14	4	$-2^{22} - 2^{11} - 2^{10} - 2^{9} - 2^{6} - 2^{4}$	6	7	2.43	14.2	16.6
15	4	$-2^{22} - 2^{21} - 2^{17} - 2^{12} - 2^{10} - 2^{8}$	6	8	2.81	16.6	19.4

Chapter 5

Performance Analyses of SIDH with Several Constructions of Quadratic Extension Fields

The supersingular isogeny Diffie-Hellman (SIDH) is one of the isogeny-based key exchange protocols, which is considered it can not be broken even though the post-quantum computers are realized. Since it is an important tool for constructing the supersingular isogeny key encapsulation (SIKE) which is submitted in the NIST post-quantum standardization, there are many works of the optimizations, efficient implementations, and security analyses of the SIDH in recent years. This chapter also provides one of such works for the SIDH, which is introduced as the third work in Sect. 1.3. In the following, the background and motivation are provided.

5.1 Background and motivation

In [JDF11], Jao and De Feo, who were introduced the SIDH, presented that the large-degree isogenies can be efficiently computed by decomposing into low-degree isogenies involving point multiplications on supersingular elliptic curves defined over quadratic extension field. Besides this, there are many works [CLN16; CH17; FH+17; Ren18] for optimizing the SIDH. Particularly, in [CLN16], Costello et al. proposed efficient formulas for computing the low-degree isogenies, i.e., 2, 3, and 4-isogenies, with a projective point associated with fast arithmetic on curves of special form, which are called the Montgomery curves, and updated that in [CH17]. To achieve more optimization, the author focuses on the construction of $\mathbb{F}_{p^2}$ for the following reasons: (i) Since SIDH requires arithmetic operations in $\mathbb{F}_{p^2}$, the performance of the arithmetic operations in $\mathbb{F}_{p^2}$ might affect the performance of SIDH. (ii) Moreover, since the range of the supersingular elliptic curves depends on $\mathbb{F}_{p^2}$ that restricts conditions of field characteristics, there is a possibility that the range can expand by changing the construction of $\mathbb{F}_{p^2}$.

In this context, the author focuses on the following construction methods of $\mathbb{F}_{p^2}$ with efficient performing arithmetics:

- Optimal extension fields (OEFs) [BP01] in which polynomial multiplication is implemented by using Karatsuba multiplication [KO62].
- All-one polynomial extension fields (AOPFs) [NSM03] with an efficient multiplication algorithm named a cyclic vector multiplication algorithm (CVMA).
- Extension fields with normal basis representation (EFNs) of which multiplication is efficiently implemented by the NTT method [KAH00].

Since the OEFs are the most well-known constructions with efficient performing arithmetics, the typical SIDH employed $\mathbb{F}_{p^2}$ which is defined by the OEF with the irreducible polynomial $x^2 + 1$ in $\mathbb{F}_p[x]$ Note that this construction results in the best performing arithmetics among the OEFs. On the other hand, $\mathbb{F}_{p^2}$ can also be constructed by using the AOPFs and EFNs. Then, the numbers of additions in $\mathbb{F}_p$ for the multiplications in $\mathbb{F}_{p^2}$ are smaller than that of OEFs. This means that there is a possibility of the performance improvement of the SIDH by exploiting $\mathbb{F}_{p^2}$ by the AOPFs and EFNs instead of the OEF.

In this chapter, the author confirms that $\mathbb{F}_{p^2}$ based on the AOPFs and EFNs are applicable for the SIDH. Since there are several candidates of $\mathbb{F}_{p^2}$, an isomorphic map between $\mathbb{F}_{p^2}$ to convert the constructions efficiently and conveniently is also presented. The author describes the performance analyses of the SIDH with several candidates of $\mathbb{F}_{p^2}$. As a result of the experiment, the performances of SIDH with $\mathbb{F}_{p^2}$ based on the AOPFs and EFNs are comparable to that of the typical $\mathbb{F}_{p^2}$ based on the OEF. Moreover, one of the candidates of $\mathbb{F}_{p^2}$ based on the EFNs results in a new efficient implementation of the SIDH by using curves that have not been used at this time.

Notations. The calculation costs of the multiplication, squaring, addition, shift operations in $\mathbb{F}_{p^2}$ are denoted as m_2, s_2, a_2, and h_2, respectively.

Organization. In the rest of this chapter, Sect. 5.2 reviews the SIDH key exchange protocols. Sect. 5.3 overviews efficient formulas used for the SIDH. Then, Sect. 5.4 describes the constructions of $\mathbb{F}_{p^2}$, its applicability for the SIDH, and isomorphisms between the possible candidates of $\mathbb{F}_{p^2}$ for SIDH. The performance analyses of SIDH are given in Sect 5.6. Finally, the contributions are summarized in Sect. 5.7.

5.2 Review of SIDH key exchange protocol

This section reviews the SIDH key exchange protocol together with the basic constructions of elliptic curves used for the SIDH. Note that the notations and mathematical fundamentals used in this section are described in Chapter 2.

5.2.1 Supersingular elliptic curves of smooth order

The SIDH requires isogeny classes of supersingular elliptic curves with smooth orders. To achieve that, it is exploited the elliptic curves defined over a finite field with a special characteristic p, such that

$$p = l_A^{e_A} l_B^{e_B} f \pm 1, \tag{5.1}$$

where l_A and l_B are two small distinct primes, e_A and e_B are two positive integers, and f is a cofactor. A prime of the above form is called *SIDH-friendly prime.*

Then, one can find a supersinglar elliptic curve E defined over $\mathbb{F}_{p^2}$ such that $\#E(\mathbb{F}_{p^2}) = (p \mp 1)^2 = l_A^{e_A} l_B^{e_B} f$. For $l \in \{l_A, l_B\}$ and $e \in \{e_A, e_B\}$, there is l^e-torsion subgroup in $E(\mathbb{F}_{p^2})$, i.e., $E[l^e] \subseteq E(\mathbb{F}_{p^2})$. Since l is coprime to p, $E[l^e] \cong \mathbb{Z}/l^e\mathbb{Z} \times \mathbb{Z}/l^e\mathbb{Z}$, which means that $E[l^e]$ consists of $(l^e + 1)$ subgroups of order l^e. Let P and Q be points on $E[l^e]$ such that $\langle P, Q \rangle \cong E[l^e]$. For an integer m, if and only if l does not divide m, there is a point $R = P + [m]Q$ of order l^e. Such points can generate a unique subgroup $\langle R \rangle$ of order l^e, such that $\langle R \rangle \subset E[l^e]$. The SIDH uses a l^e-isogeny of the base curve E where the kernel is $\langle R \rangle$. As described in Sect. 2.5.3, l^e-isogeny can be decomposed into e-times low degree l-isogenies and can be efficiently computed.

5.2.2 SIDH key exchange

This section recalls the SIDH key exchange protocol introduced in [JDF11]. In what follows, the steps of key exchange between the two-person, Alice and Bob, and its security are summarized. Note that the author refers to the construction of SIDH given in [CLN16] for fast implementation.

- Setup: The public parameters are the supersingular curve $E_0/\mathbb{F}_{p^2}$ of which group order is $(l_A^{e_A} l_B^{e_B} f)^2$ as in Sect. 5.2.1, two independent points P_A and Q_A that generate $E_0[l_A^{e_A}]$, and two independent points P_B and Q_B that generate $E_0[l_B^{e_B}]$. Alice and Bob agree to use these public parameters.

- Key generation: Alice chooses her secret integer as $s_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$ such that a point $R_A = P_A + [s_A]Q_A$ has order $l_A^{e_A}$. Her secret key is computed as the degree $l_A^{e_A}$-isogeny $\varphi_A : E_0 \to E_A$ where the kernel is R_A, i.e., $E_A \cong E_0/\langle R_A \rangle$ and her public key is the isogenous curve E_A together with the image points $\varphi_A(P_B)$ and $\varphi_A(Q_B)$.

  Similarly, Bob chooses his secret integer $s_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$ such that a point $R_B = P_B + [s_B]Q_B$ has order $l_B^{e_B}$. His secret key is computed as the degree $l_B^{e_B}$-isogeny $\varphi_B : E_0 \to E_B$ where the kernel is R_B, i.e., $E_B \cong E_0/\langle R_B \rangle$ and his public key is E_B together with the image points $\varphi_B(P_A)$ and $\varphi_B(Q_A)$.

  Finally, they send their public key to each other.

- Shared secret: To compute the shared secret, Alice uses her secret integers and received Bob's public key and computes a point $R'_A = \varphi_B(P_A) + [s_A]\varphi_B(Q_A) = \varphi_B(P_A + [s_A](Q_A)) = \varphi_B(R_A)$. Then, Alice computes the degree $l_A^{e_A}$-isogeny $\varphi'_A : E_B \to E_{BA}$ where the kernel is R'_A, i.e., $E_{BA} \cong E_B/\langle R'_A\rangle$.

  Similarly, Bob computes a point $R'_B = \varphi_A(P_B) + [s_B]\varphi_A(Q_B) = \varphi_A(P_B + [s_B]Q_B) = \varphi_A(R_B)$ from his secret integer s_B and Alice's public key. Bob also computes the degree $l_B^{e_B}$-isogeny $\varphi'_B : E_A \to E_{AB}$ where the kernel is R'_B, i.e., $E_{AB} \cong E_A/\langle R'_B\rangle$

  Then, E_{BA} and E_{AB} are isomorphic since $E_{BA} \cong E/\langle R_A, R_B\rangle \cong E_{AB}$, they can share the same j-invariant $j(E_{BA}) = j(E_{AB})$.

In the following, the operations for computing the l^e-isogeny with images of some points in the key generation phase and l^e-isogeny in the shared secret phase are denoted as `keygen_iso` and `keyshare_iso`, respectively. The operation for computing the generator point of kernel subgroups of order l^e, i.e., $R = P + [m]Q$ with $P, Q \in E[l^e]$ and $m \in \mathbb{Z}/l^e\mathbb{Z}$ is denoted as `kernel_gen`. As seen in the steps of SIDH, these operations occupy almost all the computational complexity of the SIDH.

5.2.3 Security of the SIDH

In [DFJP14], De Feo, Jao, and Plut gave computational problems related to the SIDH and discuss their complexity. The security of the SIDH is based on the assumptions that the following problems are difficult for solving.

Definition 5.1. (Supersingular computational Diffie-Hellman problem (SSCDHP)) Given E_A, E_B and the points $\varphi_A(P_B)$, $\varphi_A(Q_A)$, $\varphi_B(P_A)$, $\varphi_B(Q_A)$, find the j-invariant of $E_0/\langle P_A + [s_A]Q_A, P_B + [s_B]Q_B\rangle$.

Definition 5.2. (Supersingular decision Diffie-Hellman problem (SSDDHP)) Given a tuple sampled with probability 1/2 from one of the following two distributions:

- $(E_A, E_B, \varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A), E_{AB})$, where

  $E_{AB} \cong E_0/\langle P_A + [s_A]Q_A, P_B + [s_B]Q_B\rangle$;

- $(E_A, E_B, \varphi_A(P_B), \varphi_A(Q_B), \varphi_B(P_A), \varphi_B(Q_A), E_C)$, where

  $E_C \cong E_0/\langle P_A + [s_A]Q_A, P_B + [s_B]Q_B\rangle$.

Definition 5.3. (Computational supersingular isogeny problem (CSSIP)) Given E_A and the values $\varphi_A(P_B)$ and $\varphi_A(Q_B)$, find a kernel $R = P + [m]Q$ of the isogeny $\varphi : E \to \tilde{E}$.

According to [DFJP14], given a CSSIP (resp., SSCDHP) solver, it is trivial to solve SSCDHP (resp., SSDDHP).

5.3 Review of Efficient operations for SIDH

As described in [CLN16], supersingular elliptic curves in the Montgomery form are suggested using for fast SIDH. In this section, the formulas of efficient Montgomery operations for the SIDH are summarized.

5.3.1 Montgomery curves

An elliptic curve E over $\mathbb{F}_q$ of the Weierstrass equation $y^2 = x^3+ax+b$ can be transformed into a special equation if 4 divides $\#E(\mathbb{F}_q)$, which is found by Montgomery in [Mon87]. Indeed, let α and β be elements in $\overline{\mathbb{F}}_q$ such that $\alpha^3 + a\alpha + b = 0$ and $\beta^2 = 3\alpha^2 + a$. Then, the substitution of $x \mapsto (x - \alpha)/\beta$ results in

$$E : By^2 = x^3 + Ax^2 + x, \tag{5.2}$$

where A and B are coefficients in $\mathbb{F}_q$ satisfying $B \neq 0$ and $A^2 \neq 4$. The elliptic curve of the above equation is called the *Montgomery curve*. The Montgomery curve has the j-invariant given by $j(E) = 256(A^2 - 3)^3/(A^2 - 4)$.

All the rational points on the Montgomery curve can be represented in homogenized coordinates in $\mathbb{P}^2$ over $\mathbb{F}_q$ such that $(X : Y : Z)$ with $x = X/Z, y = Y/Z$ with $Z \neq 0$, which a point at infinity becomes $\mathcal{O} = (0 : 1 : 0)$. There are more efficient formulas in projective coordinates without Y-coordinate by using a 2-to-1 mapping as shown below.

$$\begin{aligned} x :& E \to E/\langle - \rangle, \\ P \mapsto & \begin{cases} (X : Z) & \text{if } Z \neq 0, \\ (1 : 0) & \text{if } Z = 0, \end{cases} \end{aligned} \tag{5.3}$$

where $-$ is a negation automorphism given as $- : (x, y) \mapsto (x, -y)$. Then, $E/\langle - \rangle$ is a projective 1-space $\mathbb{P}^1$ over $\mathbb{F}_q$.

Since $-$ is commutative with $[s]$, a point multiplication $x(P) \mapsto x([s]P)$ can also be available in $\mathbb{P}^1$. An isogeny $\varphi : E \to \tilde{E}$ between Montgomery curves E and $\tilde{E}$ can also be computed in $\mathbb{P}^1$, i.e. $x(P) \mapsto x(\varphi(P))$, since the x-coordinate of $\varphi(P)$ is determined without the y-coordinate of a point P. The above operations on E typically depend on only the coefficient A, which is typically taken as $(A \pm 2)/4$ for efficient formulas. More details of the facts of the Montgomery curves are described in [CS18].

5.3.2 Projective Montgomery operations for SIDH

The author refers to [CLN16; CH17] and presents the formulas of projective Montgomery operations for the SIDH. In this subsection, the SIDH with $p = l_A^{e_A} l_B^{e_B} f \pm 1$ where $l_A = 2$,

$l_B = 3$, and $2 \mid e_A$ is considered.

Projective point operations

The author uses the projective coordinates not only the points of the curve but also the curve coefficients since they are not fixed but moved in isogeny graphs. Thus, the constant term $(A-2)/4$ in the projective coordinates is denoted as $(A_{24} : C_{24})$. Assuming $x(P) = (X_P : Z_P)$, $x(Q) = (X_Q : Z_Q)$, and $x(Q-P) = (X_{Q-P} : Z_{Q-P})$, a point doubling operation `xDBL` : $(x(P), (A-2)/4) \mapsto x([2]P)$, a tripling operation `xTPL` : $(x(P), (A-2)/4) \mapsto x([3]P)$, and a point addition `xADD` : $(x(P), x(Q), x(Q-P)) \mapsto x(Q+P)$ are given as follows:

- Doubling operation (`xDBL`)

$$\begin{aligned}[2](X_P : Z_P) = (&C_{24}(X_P + Z_P)^2(X_P - Z_P)^2 : \\ &4X_PZ_P(C_{24}(X_P + Z_P)^2 + 4A_{24}X_PZ_P).\end{aligned} \tag{5.4}$$

- Tripling operation (`xTPL`)

$$\begin{aligned}[3](X_P : Z_P) = (&X_P(16A_{24}X_PZ_P^3 - C_{24}(X_P - 3Z_P)(X_P + Z_P)^3)^2 : \\ &Z_P(16A_{24}X_P^3Z_P + C_{24}(3X_P - Z_P)(X_P + Z_P)^3)^2).\end{aligned} \tag{5.5}$$

- Addition operation (`xADD`)

$$\begin{aligned}(X_Q : Z_Q) + (X_P : Z_P) = (&Z_{Q-P}((X_Q - Z_Q)(X_P + Z_P) + (X_Q + Z_Q)(X_P - Z_P))^2 : \\ &X_{Q-P}((X_Q - Z_Q)(X_P + Z_P) - (X_Q + Z_Q)(X_P - Z_P))^2).\end{aligned} \tag{5.6}$$

According to [CH17], `xTPL` can be computed by taking a coefficient as $(A_{24}, K_{24} = A_{24} + C_{24})$. The operations `xDBL` and `xTPL` are used for the computations of the points of order 2 and 3 required for 2- or 4-isogeny and 3-isogeny computations, respectively. Although `xADD` is typically does not exploited for SIDH, an operation to compute `xDBL` and `xADD` simultaneously, i.e., `xDBLADD` : $(x(P), x(Q), x(Q-P), (A+2)/4) \mapsto (x([2]P), x(Q-P))$ is used for the SIDH operation `kernel_gen` as described in [DFJP14; FH+17].

Projective isogenies computation

As for the computation of the 2^{e_A}-isogeny with $2 \mid e_A$, 4-isogenies are typically adopted for the SIDH. Let $(X'_P : Z'_P)$ and $(A'_{24} : C'_{24})$ be an image of $(X_P : Z_P)$ and coefficient of an elliptic curve given by ϕ, respectively. Assuming $(X_3 : Z_3)$ and $(X_4 : Z_4)$ denote rational points of order 3 and 4, the isogenies of degrees 3 and 4 are computed as follows:

Table 5.1: The calculation costs of the projective operations for SIDH.

Operation/
from
	Input(s)
	type(s)
		Output(s)
		type(s)
			Operations			

			m_2	s_2	a_2	h_2
`xDBL` [Mon87]	$x(P), A_{24}, C_{24}$ $\mathbb{P}^1 \times \mathbb{F}_{p^2} \times \mathbb{F}_{p^2}$	$x([2]P)$ $\mathbb{P}^1$	4	2	4	0
`xTPL` App. A in [CH17]	$x(P), A_{24}, K_{24}$ $\mathbb{P}^1 \times \mathbb{F}_{p^2} \times \mathbb{F}_{p^2}$	$x([3]P)$ $\mathbb{P}^1$	7	5	10	0
`xDBLADD` [CLN16]	$x(P), x(Q), x(Q-P), \frac{A+2}{4}$ $\mathbb{P}^1 \times \mathbb{P}^1 \times \mathbb{P}^1 \times \mathbb{F}_{p^2}$	$x([2]P), x(Q-P)$ $\mathbb{P}^1$	7	4	8	0
`3_iso_curve` App. A in [CH17]	$x(P_3)$ $\mathbb{P}^1$	$\mathbf{c}_2, A'_{24}, C'_{24}$ $(\mathbb{F}_{p^2})^2 \times \mathbb{F}_{p^2} \times \mathbb{F}_{p^2}$	2	3	14	0
`3_iso_curve*` This work	$x(P_3)$ $\mathbb{P}^1$	$\mathbf{c}_2, A'_{24}, K'_{24}$ $(\mathbb{F}_{p^2})^2 \times \mathbb{F}_{p^2} \times \mathbb{F}_{p^2}$	2	3	13	0
`3_iso_point` App. A in [CH17]	$(\mathbf{c}_2, x(P))$ $(\mathbb{F}_{p^2})^2 \times \mathbb{P}^1$	$x(\phi(P))$ $\mathbb{P}^1$	4	2	4	0
`4_iso_curve` App. A in [CH17]	$x(P_4)$ $\mathbb{P}^1$	$\mathbf{c}_3, A'_{24}, C'_{24}$ $(\mathbb{F}_{p^2})^3 \times \mathbb{F}_{p^2} \times \mathbb{F}_{p^2}$	-	4	3	1
`4_iso_point` App. A in [CH17]	$(\mathbf{c}_3, x(P))$ $(\mathbb{F}_{p^2})^3 \times \mathbb{P}^1$	$x(\phi(P))$ $\mathbb{P}^1$	6	2	6	0

- 3-isogeny operations (`3_iso_curve`, `3_iso_point`)

$$(A'_{24} : C'_{24}) = ((X_3 + Z_3)(Z_3 - 3X_3)^3 : 16X_3Z_3^3), \tag{5.7}$$

$$(X'_P : Z'_P) = (X_P(X_3X_P - Z_3Z_P)^2 : Z_P(Z_3X_P - X_3Z_P)^2). \tag{5.8}$$

- 4-isogeny operations (`4_iso_curve`, `4_iso_point`)

$$(A'_{24} : C'_{24}) = (X_4^4 - Z_4^4 : Z_4^4), \tag{5.9}$$

$$\begin{aligned}(X'_P : Z'_P) = (&X_P(2X_4Z_4Z_P - X_P(X_4^2 + Z_4^2))(X_4X_P - Z_4Z_P)^2 : \\ &Z_P(2X_4Z_4X_P - Z_P(X_4^2 + Z_4^2))(Z_4X_P - X_4Z_P)).\end{aligned} \tag{5.10}$$

The author modifies `3_iso_curve` by using $K_{24} = A_{24} + C_{24}$ and defines an operation `3_iso_curve*` which compute $(A'_{24}, K'_{24} = A'_{24} + C'_{24})$ as follows:

$$(A'_{24} : K'_{24}) = ((X_3 + Z_3)(Z_3 - 3X_3)^3 : (Z_3 - X_3)(Z_3 + 3X_3)^3), \tag{5.11}$$

which results in a reduction of single addition in $\mathbb{F}_{p^2}$.

The calculation costs and I/O specifications of `xDBL`, `xTPL`, `xDBLADD`, `3_iso_curve`, `3_iso_point`, `3_iso_curve*`, `4_iso_curve`, and `4_iso_point` are summarized in Table 5.1, where $\mathbf{c}_2$ and $\mathbf{c}_3$ are common variables for the curve determination and point evaluation.

5.4 Constructions of quadratic extension fields for fast SIDH

In this section, several attractive constructions of $\mathbb{F}_{p^2}$ with efficient performing arithmetics are described. This section also presents the applicability of these constructions of $\mathbb{F}_{p^2}$ for the SIDH with $p = 2^{e_A}3^{e_B}f \pm 1$.

5.4.1 Construction methods

A quadratic extension field applied for the SIDH has to be particularly efficient since the efficiency of the SIDH strongly depends on the efficiency of arithmetics in $\mathbb{F}_{p^2}$. Thus, the author constructs implementation-friendly $\mathbb{F}_{p^2}$ by exploiting the existing construction methods of extension fields with efficient performing arithmetics, which are introduced below.

(i) Optimal extension field

In [BP01], Bailey and Paar proposed the OEFs which are defined by using irreducible binomials. An OEF of degree m of $\mathbb{F}_p$ is defined as $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(x^m - c_0) \cong \mathbb{F}_p(\alpha)$, where $f(x) = x^m - c_0$ is an irreducible binomial in $\mathbb{F}_p[x]$ and α is an element in $\mathbb{F}_{p^m}$ such that $f(\alpha) = 0$. Any element $a \in \mathbb{F}_{p^m}$ is represented as $a = a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}$ where a_i for $0 \leq i \leq m-1$ are elements in $\mathbb{F}_p$. A set $\{1, \alpha, \dots, \alpha^{m-1}\}$ is a basis of a which is classified into a *polynomial basis*. In $\mathbb{F}_{p^m}$, there are several efficient multiplication algorithms such that Karatsuba multiplication [KO62] and Toom-Cook multiplication [Too63; CA69]. Although the field characteristics of the original OEF are pseudo-Mersenne primes, it is possible to extend for the general characteristics including the SIDH-friendly primes. Thus, the author considers the following definition of $\mathbb{F}_{p^2}$.

$$\mathbb{F}_{p^2} \cong \mathbb{F}_p[x]/(x^2 - c_0) \cong \mathbb{F}_p(\alpha), \tag{5.12}$$

where $x^2 - c_0$ is an irreducible polynomial in $\mathbb{F}_p[x]$ and α is an element in $\mathbb{F}_{p^2}$ such that $\alpha^2 = c_0$. The small value of c_0 leads to efficient performing arithmetics. Indeed, the choice of $c_0 = -1$ results in the best performing arithmetics among the OEFs and is used for the typical SIDH.

(ii) All one polynomial field

In [NSM03], Nogami et al. proposed other attractive extension fields, i.e., AOPFs. An AOPF of degree m of $\mathbb{F}_p$ is defined as $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(x^m + x^{m-1} + \cdots + 1) \cong \mathbb{F}_p(\beta)$, where $f(x) = x^m + x^{m-1} + \cdots + 1$ is an irreducible all-one polynomial in $\mathbb{F}_p[x]$ and

β is an element in $\mathbb{F}_{p^m}$ such that $f(\beta) = 0$. Any element $a \in \mathbb{F}_{p^m}$ is represented as $a = a_1\beta + a_1\beta^2 + \cdots + a_m\beta^m$ where a_i for $1 \leq i \leq m$ are elements in $\mathbb{F}_p$. A set $\{\beta, \beta^2, \ldots, \beta^m\}$ is a basis of a classified into an *optimal normal basis* [Mul+88]. In $\mathbb{F}_{p^m}$, an efficient multiplication algorithm which is named the CVMA in [NSM03] is available. Since the AOPF can also be extended for the fields with SIDH-friendly primes, the author considers the following definition of $\mathbb{F}_{p^2}$.

$$\mathbb{F}_{p^2} \cong \mathbb{F}_p[x]/(x^2 + x + 1) \cong \mathbb{F}_p(\beta), \tag{5.13}$$

where $x^2 + x + 1$ is an irreducible polynomial in $\mathbb{F}_p[x]$ and β is an element in $\mathbb{F}_{p^2}$ such that such that $\beta^2 = -\beta - 1$.

(iii) Extension field with a normal basis

There exist extension fields such that any elements are represented by using a basis classified into a *normal basis*. Such fields are called the EFNs in this thesis. An EFN of degree m of $\mathbb{F}_p$ is defined as $\mathbb{F}_{p^m} \cong \mathbb{F}_p[x]/(x^m + c_{m-1}x^{m-1} + \cdots + c_0) \cong \mathbb{F}_p(\gamma)$, where $f(x) = x^m + c_{m-1}x^{m-1} + \cdots + c_0$ is an irreducible polynomial with non-zero trace in $\mathbb{F}_p[x]$ and γ is an element in $\mathbb{F}_{p^m}$ such that $f(\gamma) = 0$. Any element $a \in \mathbb{F}_{p^m}$ is represented as $a = a_0\gamma + a_1\gamma^p + \cdots + a_{m-1}\gamma^{p^{m-1}}$ where a_i for $0 \leq i \leq m-1$ are elements in $\mathbb{F}_p$. Note that a set $\{\gamma, \gamma^p, \ldots, \gamma^{p^{m-1}}\}$ is a basis of a which is called the normal basis. The EFNs are efficiently implemented by using the NTT method [KAH00]. From the above, the author also considers the following definition of $\mathbb{F}_{p^2}$.

$$\mathbb{F}_{p^2} \cong \mathbb{F}_p[x]/(x^2 + c_1x + c_0) \cong \mathbb{F}_p(\gamma), \tag{5.14}$$

where $x^2 + c_1x + c_0$ with non-zero c_1 is an irreducible polynomial in $\mathbb{F}_p[x]$ and γ is an element in $\mathbb{F}_{p^2}$ such that $\gamma^2 = -c_1\gamma - c_0$.

5.4.2 Attractive candidates of $\mathbb{F}_{p^2}$

According to the constructions of $\mathbb{F}_{p^2}$ described in the previous subsection, the author considers the following candidates of $\mathbb{F}_{p^2}$.

`OEF_x2+1` : $\mathbb{F}_p[x]/(x^2 + 1)$, `OEF_x2+2` : $\mathbb{F}_p[x]/(x^2 + 2)$,
`OEF_x2-2` : $\mathbb{F}_p[x]/(x^2 - 2)$, `OEF_x2+3` : $\mathbb{F}_p[x]/(x^2 + 3)$,
`OEF_x2-3` : $\mathbb{F}_p[x]/(x^2 - 3)$, `OEF_x2+4` : $\mathbb{F}_p[x]/(x^2 + 4)$,
`OEF_x2+5` : $\mathbb{F}_p[x]/(x^2 + 5)$, `OEF_x2-5` : $\mathbb{F}_p[x]/(x^2 - 5)$,
`AOPF_x2+x+1` : $\mathbb{F}_p[x]/(x^2 + x + 1)$,
`EFN_x2-x+1` : $\mathbb{F}_p[x]/(x^2 - x + 1)$, `EFN_x2-x-1` : $\mathbb{F}_p[x]/(x^2 - x - 1)$,

Table 5.2: The calculation costs of arithmetic operations in $\mathbb{F}_{p^2}$.

Constructions	Multiplication				Squaring			
	m_1	s_1	a_1	h_1	m_1	s_1	a_1	h_1
OEF_x2+1	3	0	5	0	2	0	3	0
OEF_x2+2	3	0	6	0	2	0	5	0
OEF_x2-2	3	0	5	0	2	0	5	0
OEF_x2+3	3	0	5	1	2	0	3	2
OEF_x2-3	3	0	6	0	2	0	5	0
OEF_x2+4	3	0	5	1	2	0	5	1
OEF_x2+5	3	0	6	1	2	0	4	2
OEF_x2-5	3	0	5	1	2	0	4	2
AOPF_x2+x+1	3	0	4	0	2	0	4	0
EFN_x2-x+1	3	0	4	0	2	0	4	0
EFN_x2-x-1	3	0	4	0	0	3	3	0

where $\mathbb{F}_{p^2}$ defined by a certain polynomial is denoted as [field_name]_[polynomial], e.g., $\mathbb{F}_{p^2}$ based on OEFs given by a polynomial $x^2 + 1$ is denoted as OEF_x2+1. Note that OEF_x2+1 is employed for the typical SIDH. The details of the operation algorithms for OEF_x2+1, OEF_x2-5, AOPF_x2+x+1, EFN_x2-x+1, and EFN_x2-x-1 are especially presented in App. D.

The calculation costs of multiplication and squaring in $\mathbb{F}_{p^2}$ based on the OEF, AOPF, and EFN are summarized in Table 5.2. According to Table 5.2, it is found that OEF_x2+1 is the best performing arithmetic among $\mathbb{F}_{p^2}$ based on the OEFs. In contrast, 1 addition in $\mathbb{F}_p$ for the multiplications in AOPF_x2+x+1, EFN_x2-x+1, and EFN_x2-x-1 is reduced from that of OEF_x2+1. However, 1 addition in $\mathbb{F}_p$ for squarings in AOPF_x2+x+1 and EFN_x2-x+1 is increased from that of OEF_x2+1, which is a degradation. As for the squaring in EFN_x2-x-1, 2 multiplications in $\mathbb{F}_p$ are replaced with 3 squarings in $\mathbb{F}_p$ from that of OEF_x2+1. According to Table 5.1, multiplications in $\mathbb{F}_{p^2}$ are more often required for the SIDH operations than squarings in $\mathbb{F}_{p^2}$. If it is possible to apply AOPF_x2+x+1, EFN_x2-x+1, and EFN_x2-x-1 for the SIDH, the performance of the SIDH might be competitive to or rather better than that of OEF_x2+1.

5.4.3 Applicability of the candidates of $\mathbb{F}_{p^2}$ for SIDH

The author confirms the applicability of the candidates of $\mathbb{F}_{p^2}$ described in Sect. 5.4.2 for the SIDH. Note that not all SIDH-friendly primes results in $\mathbb{F}_{p^2}$ based on any constructions since there exist restrictions of field characteristics from the irreducibility of a polynomial. In the following, the author especially describes the applicability of $\mathbb{F}_{p^2}$ based on the target constructions for the SIDH with $p = 2^{e_A}3^{e_B}f \pm 1$.

Theorem 5.4. The field characteristic p has to satisfy the following conditions for constructing $\mathbb{F}_{p^2}$.

`OEF_x2+1` : $p \equiv 3 \pmod{4}$, `OEF_x2+2` : $p \equiv 5, 7 \pmod{8}$,

`OEF_x2-2` : $p \equiv 3, 5 \pmod{8}$, `OEF_x2+3` : $p \equiv 2 \pmod{3}$,

`OEF_x2-3` : $p \equiv 5, 7 \pmod{12}$, `OEF_x2+4` : $p \equiv 3 \pmod{4}$,

`OEF_x2+5` : $p \equiv 11, 13, 17, 19 \pmod{20}$, `OEF_x2-5` : $p \equiv 2, 3 \pmod{5}$,

`AOPF_x2+x+1` : $p \equiv 2 \pmod{3}$,

`EFN_x2-x+1` : $p \equiv 2 \pmod{3}$, `EFN_x2-x-1` : $p \equiv 2, 3 \pmod{5}$.

Proof of Theorem 5.4. To construct $\mathbb{F}_{p^2}$, a polynomial $f(x) = x^2 + c_1x + c_0$ in $\mathbb{F}[x]$ has to be irreducible over $\mathbb{F}_p$. The irreducibility of $f(x)$ depends on the quadratic residue properties of the discriminant $D = c_1^2 - 4c_0$ since a root of $f(x)$ is given as $(-c_1 \pm \sqrt{D})/2$. If D is a quadratic non-residue in $\mathbb{F}_p^*$, the polynomial is irreducible in $\mathbb{F}[x]$. For `OEF_x2+1`, `OEF_x2+2`, `OEF_x2-2`, `OEF_x2+3`, `OEF_x2-3`, `OEF_x2+4`, `OEF_x2+5`, `OEF_x2-5`, `AOPF_x2+x+1`, `EFN_x2-x+1`, and `EFN_x2-x-1`, the discriminants are given as $D = -4$, -8, 8, -12, 12, -16, -20, 20, -3, -3, and 5, respectively. Applying the properties of the Legendre symbol described in [Kob94], the restriction of the characteristic for the certain discriminant can be uniquely obtained as follows: $\left(\frac{-4}{p}\right) = \left(\frac{-16}{p}\right) = -1 \Leftrightarrow p \equiv 3 \pmod{4}$, $\left(\frac{-8}{p}\right) = -1 \Leftrightarrow p \equiv 5, 7 \pmod{12}$, $\left(\frac{8}{p}\right) = -1 \Leftrightarrow p \equiv 3, 5 \pmod{8}$, $\left(\frac{-12}{p}\right) = \left(\frac{-3}{p}\right) = -1 \Leftrightarrow p \equiv 2 \pmod{3}$, $\left(\frac{12}{p}\right) = -1 \Leftrightarrow p \equiv 5, 7 \pmod{12}$, $\left(\frac{-20}{p}\right) = -1 \Leftrightarrow p \equiv 11, 13, 17, 19 \pmod{20}$, and $\left(\frac{20}{p}\right) = \left(\frac{5}{p}\right) = -1 \Leftrightarrow p \equiv 2, 3 \pmod{5}$. Thus, the restrictions to apply $\mathbb{F}_{p^2}$ are obtained as shown in the theorem. □

The SIDH-friendly prime given by $p = 2^{e_A}3^{e_B}f \pm 1$ are clearly satisfy the condition $p \equiv \pm 1 \pmod{2^{e_A}}$ and $p \equiv \pm 1 \pmod{3^{e_B}}$, respectively. When comparing to the restrictions to exploit $\mathbb{F}_{p^2}$ given in Lemma 5.4, the applicability of $\mathbb{F}_{p^2}$ for the SIDH with $p = 2^{e_A}3^{e_B}f \pm 1$ is obtained as shown in Table 5.3 where ✓ and X denote applicable and inapplicable, respectively.

From Table 5.3, the new candidates of $\mathbb{F}_{p^2}$ such that `AOPF_x2+x+1` and `EFN_x2-x+1` can be available for the SIDH with $p = 2^{e_A}3^{e_B}f - 1$. Moreover, if the primes satisfy $p \equiv 2, 3 \pmod{5}$, `EFN_x2-x-1` can also be applied not only for $p = 2^{e_A}3^{e_B}f - 1$ but also for $p = 2^{e_A}3^{e_B}f + 1$ which have not so many choices of $\mathbb{F}_{p^2}$ based on the OEFs. Thus, there is a possibility that the SIDH with $p = 2^{e_A}3^{e_B}f + 1$ also results in an efficient implementation, however, the previous SIDH implementation does not focus on that.

Note that the sign of the constant term of the SIDH-friendly prime might not affect the performance of the modular reduction described in [CLN16], which is based on Montgomery reduction [Mon85]. Assuming $p = 2^{e_A}3^{e_B}f \pm 1$ and R is slightly larger than the size of p given as $R = 2^m$ with an integer m, one can compute the Montgomery residue

Table 5.3: Applicability of the constructions of $\mathbb{F}_{p^2}$ for the typical SIDH.

Constructions	Applicability	
	$p = 2^{e_A}3^{e_B}f - 1$	$p = 2^{e_A}3^{e_B}f + 1$
OEF_x2+1	✓	X
OEF_x2+2	✓	X
OEF_x2-2	X	X
OEF_x2+3	X	X
OEF_x2-3	✓	X
OEF_x2+4	✓	X
OEF_x2+5	✓**	✓*
OEF_x2-5	✓*	✓*
AOPF_x2+x+1	✓	X
EFN_x2-x+1	✓	X
EFN_x2-x-1	✓*	✓*

*If only a SIDH-friendly prime satisfies $p \equiv 2, 3 \pmod 5$
**If only a SIDH-friendly prime satisfies $p \equiv 1, 4 \pmod 5$

$c = aR^{-1} (\text{mod } p)$ for an input $a < pR$ as $c = (a + (aM'(\text{mod } R))p)/R = (a \pm aM'(\text{mod } R))/R + ((p \mp 1)(aM'(\text{ mod } R))) = (a \pm aM'(\text{ mod } R))/R + (2^{e_A}3^{e_B}f(aM'(\text{ mod } R)))$ where $M' = -p^{-1}(\text{mod} R)$.

5.5 Isomorphisms between the candidates of $\mathbb{F}_{p^2}$

Since there are several candidates of $\mathbb{F}_{p^2}$ which are applicable for the SIDH, the author provides an isomorphic map between $\mathbb{F}_{p^2}$ to convert the constructions efficiently and conveniently. Indeed, the author presents a construction method of an isomorphic map from $\mathbb{F}_{p^2}$ of the typical construction OEF_x2+1 for the SIDH with the typical prime $p = 2^{e_A}3^{e_B}f - 1$ to $\mathbb{F}_{p^2}$ of any constructions. Before describing the proposed map, the following lemma is required.

Lemma 5.5. If a field characteristic is $p = 2^{e_A}3^{e_B}f - 1$, there exists a primitive cube root of identity in $\mathbb{F}_p^*$.

Proof of Lemma 5.5. Since the primitive cube root of identity is written as $\sqrt[3]{1} = (-1 \pm \sqrt{-3})/2$, it is defined over $\mathbb{F}_{p^2}$ if $\sqrt{-3} \in \mathbb{F}_{p^2}$. According to [Lem13], if $3 \nmid (p - 1)$ is satisfied, 3 and -1 are quadratic residue and non-residue in $\mathbb{F}_p^*$ which leads to -3 is quadratic non-residue in $\mathbb{F}_p^*$, which mans that $\sqrt{-3} \in \mathbb{F}_{p^2}^*$. Since $p = 2^{e_A}3^{e_B}f - 1$ is satisfied the condition, $\sqrt[3]{1} \in \mathbb{F}_{p^2}$. □

In the following, let us define $\mathbb{F}_{p^2}$ as $\mathbb{F}_p[x]/(x^2+1) \cong \mathbb{F}_p(\alpha)$ and $\mathbb{F}_{p^2}[x]/(x^2+c_1x+c_0) \cong \mathbb{F}_p(\omega)$ where α and ω are elements in $\mathbb{F}_{p^2}$ such that $\alpha^2 = -1$ and $\omega^2 = -c_1\omega - x_0$, respectively. From Lemma 5.5, there exists a primitive cube root of identity in $\mathbb{F}_p(\alpha)$ and $\mathbb{F}_p(\omega)$ with a SIDH-friendly prime given by $p = 2^{e_A}3^{e_B}f - 1$. In the following, let $\delta = \delta_0 + \delta_1\alpha \in \mathbb{F}_p(\alpha)$ and $\zeta = \zeta_0 + \zeta_1\omega \in \mathbb{F}_p(\omega)$ be primitive cube roots of the identity in $\mathbb{F}_p(\alpha)$ and $\mathbb{F}_p(\omega)$ where $\delta_0, \delta_1, \zeta_0, \zeta_1 \in \mathbb{F}_p$, respectively. Indeed, δ and ζ can be written by using $\delta_0 = -1/2$, $\delta_1 = \pm\sqrt{3}/2$, $\zeta_0 = (-1 \pm c_1\sqrt{-3/D})/2$, and $\zeta_1 = \pm\sqrt{-3/D}$ where $D = c_1^2 - 4c_0 \in \mathbb{F}_p^*$, respectively. Note that we have $\sqrt{3}, \sqrt{-3/D} \in \mathbb{F}_p^*$ from the quadratic residue property of 3 and quadratic non-residue property of -3 and D in $\mathbb{F}_p^*$.

Theorem 5.6. If a field characteristic is $p = 2^{e_A}3^{e_B}f - 1$, an isomorphic map from $\mathbb{F}_p(\alpha)$ to any $\mathbb{F}_p(\omega)$ is defined as follows:

$$\begin{aligned} M :& \mathbb{F}_p(\alpha) \to \mathbb{F}_p(\omega), \\ & a_0 + a_1\alpha \mapsto (a_0 + ma_1) + na_1\omega, \end{aligned} \tag{5.16}$$

where $m = (\zeta_0 - \delta_0)/\delta_1, n = \zeta_1/\delta_1 \in \mathbb{F}_p$.

Proof of Theorem 5.6. Let a and b be elements in $\mathbb{F}_p(\alpha)$ represented by $a = a_0 + a_1\alpha$ with $a_0, a_1 \in \mathbb{F}_p$ and $b = b_0 + b_1\alpha$ with $b_0, b_1 \in \mathbb{F}_p$, respectively.

(i) Additive homomorphism. It is clearly satisfied that $M(a+b) = ((a_0+b_0) + m(a_1 + b_1)) + n(a_1 + b_1)\omega = M(a) + M(b)$.

(ii) Multiplicative homomorphism. It is obtained that $M(a \cdot b) = (a_0b_0 + m(a_0b_1 + a_1b_0) - a_1b_1) + n(a_0b_0 + a_1b_0)\omega$ and $M(a) \cdot M(b) = (a_0b_0 + m(a_0b_1 + a_1b_0) + d_0a_1b_1) + n(a_0b_1 + a_0b_1 - d_1a_1b_1)\omega$ where $d_0 = m^2 - c_0n^2$ and $d_1 = n(c_1n - 2m)$. Since $m = \pm c_1\sqrt{-1/D}$ and $n = \pm 2\sqrt{-1/D}$ with $D = c_1^2 - 4c_0 \in \mathbb{F}_p^*$, we have $d_0 = -1$ and $d_1 = 0$ which leads to $M(a \cdot b) = M(a) \cdot M(b)$.

(iii) Monomorphism. Since $n \neq 0$, it is satisfied that $M(a) \neq M(b)$ if $a \neq b \in \mathbb{F}_p(\alpha)$.

From the above (i)–(iii), M is an isomorphism. □

From the above, the isomorphic map $M : \mathbb{F}_p(\alpha) \to \mathbb{F}_p(\omega)$ is easily constructed once the primitive cube root of identity $\delta \in \mathbb{F}_p(\alpha)$ and $\zeta \in \mathbb{F}_p(\omega)$ are obtained. The elements δ and ζ are obtained without square root computation by computing a cubic non-residue element to the power of $(p^2 - 1)/3$ in $\mathbb{F}_p(\alpha)$ and $\mathbb{F}_p(\omega)$, respectively. The calculation cost to compute an image of $a \in \mathbb{F}_p(\alpha)$ is enough low since it requires only 2 multiplications and 1 addition in $\mathbb{F}_p$. Note that $M(x) \in \mathbb{F}_p(\omega)$ with the polynomial basis representation can also be deformed to the optimal normal basis and normal basis representations as

$$\begin{aligned} M(a) &= (a_0 + ma_1) + na_1\omega \\ &= ((-c_1a_0 + (c_0n - c_1m)a_1)/c_0)\omega - ((a_0 + ma_1)/c_0)\omega^2 \end{aligned}$$

$$= ((-a_0 + (c_1 n - m)a_1)/c_1)\omega - ((a_0 + ma_1)/c_1)\omega^p, \tag{5.17}$$

where c_0 and c_1 are non-zero coefficients.

The proposed isomorphic map supports the generation of the public parameters of SIDH with $\mathbb{F}_{p^2}$ based on several constructions from the existing parameters defied over `OEF_x2+1`, e.g., `SIKEp434` given in [Cam+19]. The details of the application of the isomorphism are described in App. E.

5.6 Performance analyses of SIDH

The author picks up the four implementation-friendly $\mathbb{F}_{p^2}$, i.e., `OEF_x2+1`, `OEF_x2-5`, `AOPF_x2+x+1`, and `EFN_x2-x-1` and compares the performance of the operations `keygen_iso`, `keyshare_iso` and `ker_gen` which occupy almost all computational complexity of SIDH. The author also confirms the performance of the SIDH with $p = 2^{e_A}3^{e_B}f - 1$ and $p = 2^{e_A}3^{e_B}f + 1$.

5.6.1 Assumptions

In the following, the author presents the details of the experimental assumptions such as the parameter setting, environment, optimization, and evaluation methods.

Parameters setup

The author chooses the SIDH-friendly primes satisfying $p \equiv 2, 3 \pmod 5$ to use various constructions of $\mathbb{F}_{p^2}$. The SIDH-friendly primes which can ensure quantum security at the 128-bit levels are given as follows:

$$p_{434-} = 2^{216}3^{137} - 1, \tag{5.18}$$

$$p_{441+} = 2^{216}3^{137}139 + 1, \tag{5.19}$$

where the sizes of the primes are given by 434-bit and 441-bit, respectively. Note that p_{434-} is presented in [Cam+19] where the parameter set is called `SIKEp434` and p_{441+} is found by this work. It is considered that the proposed parameter p_{441+} can also ensure the same security level since e_A and e_B which are parameterized the size of the kernel of isogenies are the same size as p_{434-} ones.

The author uses supersingular elliptic curves of Montgomery form defined over $\mathbb{F}_{p^2}$ of which orders are $(p+1)^2$ and $(p-1)^2$ for the prime p_{434-} and p_{441+}, respectively. For p_{434-}, the supersingular elliptic curve is given as $E/\mathbb{F}_p : y^2 = x^3 + 6x^2 + x$. For p_{441+}, the curve can be found by using a quadratic twist as described in App. F.

Experimental environment.

To evaluate the performance of the SIDH with several candidates of $\mathbb{F}_{p^2}$, the protocol is implemented by C language. In the implementation, the big integer arithmetics are implemented by using `mpz_t` data type of GMP library [tea15]. The software is compiled with GCC 8.3.0 with the option `-O2 -march=native`, and is executed on 3.50GHz Intel(R) Core(TM) i7-7567U CPU running macOS High Sierra version 10.13.6.

The four categories of arithmetic functions of GMP which are `mpz_mul`, `mpz_add/ mpz_sub`, `mpz_mul_2exp/ mpz_tdiv_q_2exp`, `mpz_invert`, `mpz_mod`, and `mpz_set` are employed in the software. The categories are referred as `mul`, `add`, `shift`, `mod`, and `set` respectively. If `mpz_mul` has the same operands, it is denoted as the sixth category `sqr`. To minimize the number of function calls of `mod` which has one of the highest computational complexity among the categories, the author allows the operands with the twice size of characteristic for `add`. The size of the operand(s) is denoted as a subscript of the category's name, e.g., `mpz_mul` with s-bit operands is denoted as $\texttt{mul}_s$.

The weight of these operation categories with the specific size of the operand(s) used for the implementation is given in Table 5.4. The weight is derived from one hundred million trials of execution time excluding the overhead on this environment. Unlike `mul`, `sqr`, and `mod`, the differences of the weights of `add`, `shift`, and `set` between p_{434-} and p_{441+} are invisible since these operations are low computational complexity.

Optimization

All arithmetics are performed on Montgomery curves and applied the optimization proposed in [DFJP14; CLN16] as described in Sect. 5.3.2. The author refers to Sect. 4.2.2 in [DFJP14] and finds the optimal paths of computing 4^{108}- and 3^{137}-isogenies from the ratio of a single point multiplication and isogeny evaluation. The ratio is derived from the computational complexities of these operations which are calculated by the sum of the number of operation categories multiplied by the weight given in Table 5.4. From the optimal paths, it is found that the numbers of operations `xDBL` and `4_iso_point` for computing 4^{108}-isogeny are specifically given by 666 and 405 for all candidates of $\mathbb{F}_{p^2}$ in this implementation, respectively. Similarly, the numbers of operations `xTPL` and `3_iso_point` required for computing 4^{108}-isogeny are also specifically given as 407 and 597, respectively. Note that this implementation does not adopt the Montgomery reduction described in Sect. 5.4.3 since the performance of that of p_{434-} and p_{441-} are might be competitive.

Table 5.4: Weight of the operation categories employed in the implementation of SIDH.

$\log_2\lfloor s \rfloor$	$\texttt{mul}_s$	$\texttt{sqr}_s$	$\texttt{add}_s$	$\texttt{add}_{2s}$	$\texttt{shift}_s$	$\texttt{mod}_{2s}$	$\texttt{set}_s$
434	5.12	3.46	1.00	1.14	1.04	16.4	0.63
441	5.13	3.47	1.00	1.14	1.04	16.8	0.63

Evaluation

The author measures the number of function calls required for the SIDH operations, i.e., keygen_iso, keyshare_iso, and kernel_gen, which occupy almost all the computational complexity of SIDH. Since the number of function calls of kernel_gen typically depends on the secret key, the average of the result of 1,000 random secret keys are calculated. The computational complexity of the SIDH operations is computed by the sum of the numbers of the function calls multiplied by the weight of the operation categories. Besides, average execution times of 100,000 trials of the operations are measured. Note that the measurement is performed by repeating the operations for 1,000 random secret keys 100 times.

5.6.2 Results and analyses

Tables 5.5 and 5.6 show the numbers of the function call of the operations (a) Alice's keygen_iso, (b) Bob's keygen_iso, (c) Alice's keyshare_iso, (d) Bob's keyshare_iso. (e) Alice's kernel_gen, and (f) Bob's kernel_gen for the primes p_{434-} and p_{441+}, respectively. The tables also involve computational complexity and average execution time. Figs. 5.1 and 5.2 also provide the results of the computational complexity and execution time for p_{434-} and p_{441+}. The details of the results and their analyses are described below.

From Table 5.5 and Fig. 5.1, the performance of the SIDH operations with p_{434-} applied AOPF_x2+x+1 and EFN_x2-x-1 are competitive to that of OEF_x2+1 which is exploited for the previous implementations. The results are caused by the complexities of the multiplication and squaring in $\mathbb{F}_{p^2}$ as described in Sect. 5.4.2. Moreover, EFN_x2-x-1 can achieve more 1% improvement than that of OEF_x2+1 since the computational complexity of 3 squarings in $\mathbb{F}_p$ is lower than that of 2 multiplications in $\mathbb{F}_p$ which results in more efficient performing squaring in EFN_x2-x-1 than that of OEF_x2+1. Therefore, the performance improvement for the entire SIDH can be expected by using AOPF_x2+x+1 or EFN_x2-x-1 as a replacement for OEF_x2+1. Since the calculation costs of arithmetic operations in EFN_x2+x-1 are exactly the same as AOPF_x2+x+1, EFN_x2+x-1 is yet another candidate for the replacement. However, the results of the execution time with OEF_x2+1 are slightly better than that of AOPF_x2+x+1 despite the reduction of the complexity. The author confirms the software by GNU profiler and finds that the number of function calls of the operations applied OEF_x2+1 and AOPF_x2+x+1 is exactly correct, however, the exe-

Table 5.5: The number of function calls, computational complexity and execution time of the SIDH operations (a) Alice's keygen_iso, (b) Bob's keygen_iso, (c) Alice's keyshare_iso, (d) Bob's keyshare_iso. (e) Alice's kernel_gen, and (f) Bob's kernel_gen with p_{434-}.

Construction-	Operation	Function calls							Complexity	Time [ms]
		$\texttt{mul}_{434}$	$\texttt{sqr}_{434}$	$\texttt{add}_{434}$	$\texttt{add}_{868}$	$\texttt{shift}_{434}$	$\texttt{mod}_{868}$	$\texttt{set}_{434}$		
OEF_x2+1	(a)	27,557	0	35,759	23,820	216	20,520	1,284	541,567.20	5.04
	(b)	30,389	0	43,473	25,453	0	23,234	1,632	610,146.86	5.66
	(c)	20,429	0	27,186	16,841	216	15,336	1,284	403,525.18	3.70
	(d)	23,813	0	35,146	19,806	0	18,302	1,632	480,828.36	4.47
	(e)	6,231	2	8,291	5,251	0	4,729	16	123,752.46	1.15
	(f)	6,271	0	8,331	5,292	0	4,757	16	124,496.28	1.15
OEF_x2-5	(a)	27,557	0	35,709	27,092	13,698	20,520	1,284	559,268.56	5.24
	(b)	30,389	0	43,403	29,985	16,079	23,234	1,632	631,965.50	5.88
	(c)	20,429	0	27,145	19,456	10,458	15,336	1,284	417,116.96	3.86
	(d)	23,813	0	35,091	23,501	12,791	18,302	1,632	498,288.30	4.64
	(e)	6,231	2	8,278	6,124	3,224	4,729	16	128,087.64	1.19
	(f)	6,271	0	8,318	6,170	3,243	4,757	16	128,856.92	1.20
AOPF_x2+x+1	(a)	27,557	0	42,711	13,052	216	20,520	1,284	536,243.68	5.08
	(b)	30,389	0	53,085	13,148	0	23,234	1,632	605,731.16	5.73
	(c)	20,429	0	32,462	9,045	216	15,336	1,284	399,913.74	3.78
	(d)	23,813	0	42,925	10,156	0	18,302	1,632	477,606.36	4.52
	(e)	6,231	2	10,142	2,755	0	4,729	16	122,758.02	1.16
	(f)	6,271	0	10,199	2,776	0	4,757	16	123,496.04	1.17
EFN_x2-x-1	(a)	21,113	9,666	33,771	18,770	216	20,520	1,284	534,273.28	4.97
	(b)	21,465	13,386	40,582	21,189	0	23,234	1,632	603,019.58	5.59
	(c)	15,281	7,722	25,329	13,604	216	15,336	1,284	398,338.36	3.64
	(d)	16,533	10,920	32,723	16,718	0	18,302	1,632	475,394.64	4.41
	(e)	4,512	2,581	7,749	4,289	0	4,729	16	122,235.84	1.14
	(f)	4,541	2,595	7,787	4,322	0	4,757	16	122,967.58	1.15

cution time of single $\texttt{add}_{434}$ of AOPF_x2+x+1 is strangely slower than that of OEF_x2+1. At this time, the author considers that it might come from the effects of cache and parallel processing.

The results Table 5.5 and Fig. 5.1 also show that the performance of the SIDH operations applied OEF_x2-5 compares unfavorably to OEF_x2+1. Thus, such constructions of $\mathbb{F}_{p^2}$ should be kept away from practical implementations. However, as described in Sect. 5.4.3, there do not exist good choices of $\mathbb{F}_{p^2}$ based on the OEFs for the SIDH with $p = 2^{e_A}3^{e_B}f + 1$. In contrast, the author finds the new candidate of $\mathbb{F}_{p^2}$, i.e., EFN_x2-x-1, for such the SIDH. According to Table 5.6 and Fig. 5.2, EFN_x2-x-1 contributes to improve the performance of the SIDH operations around 4% compared with the previous best choice of $\mathbb{F}_{p^2}$ based on OEFs, i.e., OEF_x2-5. Moreover, the performance of the SIDH

Table 5.6: The number of function calls, computational complexity and execution time of the SIDH operations (a) Alice's keygen_iso, (b) Bob's keygen_iso, (c) Alice's keyshare_iso, (d) Bob's keyshare_iso. (e) Alice's kernel_gen, and (f) Bob's kernel_gen with p_{441+}.

Construction	Operation	Function calls							Complexity	Time [ms]
		$\texttt{mul}_{441}$	$\texttt{sqr}_{441}$	$\texttt{add}_{441}$	$\texttt{add}_{882}$	$\texttt{shift}_{441}$	$\texttt{mod}_{882}$	$\texttt{set}_{441}$		
OEF_x2-5	(a)	27,557	0	35,244	27,557	13,698	20,520	1,284	567,817.23	5.23
	(b)	30,389	0	43,000	30,388	16,079	23,234	1,632	641,619.41	5.88
	(c)	20,429	0	26,814	19,787	10,458	15,336	1,284	423,501.99	3.84
	(d)	23,813	0	34,780	23,812	12,791	18,302	1,632	505,890.77	4.64
	(e)	6,232	1	8,181	6,221	3,224	4,729	16	130,056.81	1.20
	(f)	6,271	0	8,223	6,265	3,243	4,757	16	130,835.73	1.21
EFN_x2-x-1	(a)	21,113	9,666	32,028	20,513	216	20,520	1,284	543,033.09	5.01
	(b)	21,465	13,386	38,544	23,227	0	23,234	1,632	612,947.01	5.66
	(c)	15,281	7,722	24,030	14,903	216	15,336	1,284	404,884.65	3.68
	(d)	16,533	10,920	31,145	18,296	0	18,302	1,632	483,210.89	4.46
	(e)	4,512	2,581	7,321	4,716	0	4,729	16	124,257.15	1.16
	(f)	4,541	2,595	7,357	4,752	0	4,757	16	125,001.94	1.16

with p_{441+} applied EFN_x2-x-1 is competitive to that of p_{434-} applied OEF_x2+1. Thus, the author concludes that the efficient implementation of the SIDH with $p = 2^{e_A}3^{e_B}f + 1$ can exist.

5.7 Summary of contributions

In this chapter, the author considers the SIDH using elliptic curves defined over $\mathbb{F}_{p^2}$ that are specified by several constructions such as OEFs, AOPFs, and EFNs. It is found that not only the OEFs but also AOPFs and EFNs can be applied for the SIDH with $p = 2^{e_A}3^{e_B}f - 1$. Moreover, the EFN is also available for the SIDH with $p = 2^{e_A}3^{e_B}f + 1$, which leads to expanding the range of the elliptic curves used for the SIDH. With the possible candidates of $\mathbb{F}_{p^2}$, the author implements the SIDH and analyzes the performance of the SIDH. The results of the experiment show that the performance of the SIDH with $p = 2^{e_A}3^{e_B}f - 1$ is competitive between the possible candidates of $\mathbb{F}_{p^2}$. Besides, the SIDH with $p = 2^{e_A}3^{e_B}f + 1$ applied the EFNs are almost competitive to the SIDH with $p = 2^{e_A}3^{e_B}f - 1$ applied the typical OEFs. Thus, there are many candidates of $\mathbb{F}_{p^2}$ for fast SIDH, which involves the constructions of $\mathbb{F}_{p^2}$ that have not been considered in the previous works. Note that changing the constructions of $\mathbb{F}_{p^2}$ requires not so much effort by using isomorphisms between $\mathbb{F}_{p^2}$.

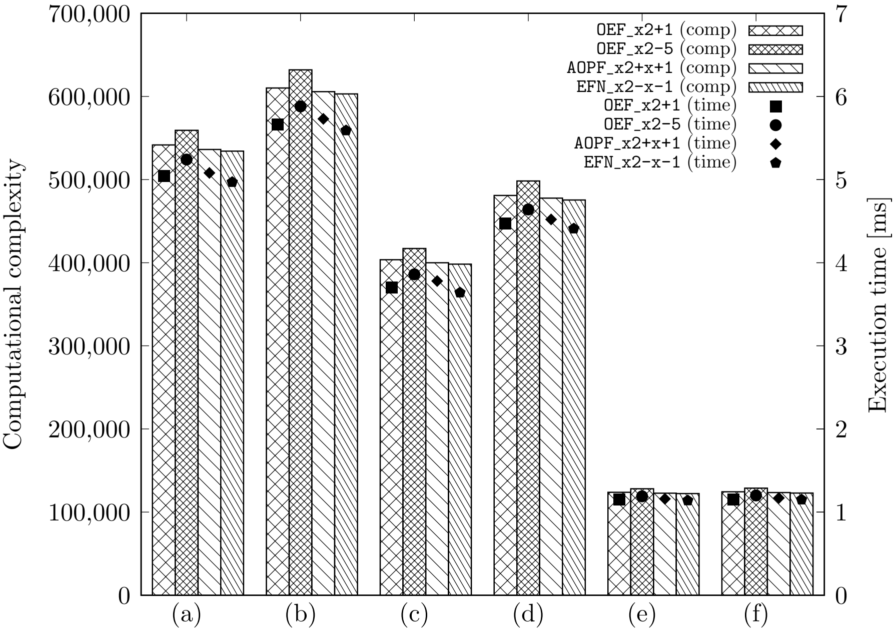


Figure 5.1: Computational complexity and execution time of the SIDH operations (a) Alice's `keygen_iso`, (b) Bob's `keygen_iso`, (c) Alice's `keyshare_iso`, (d) Bob's `keyshare_iso`. (e) Alice's `kernel_gen`, and (f) Bob's `kernel_gen` with p_{434-}.

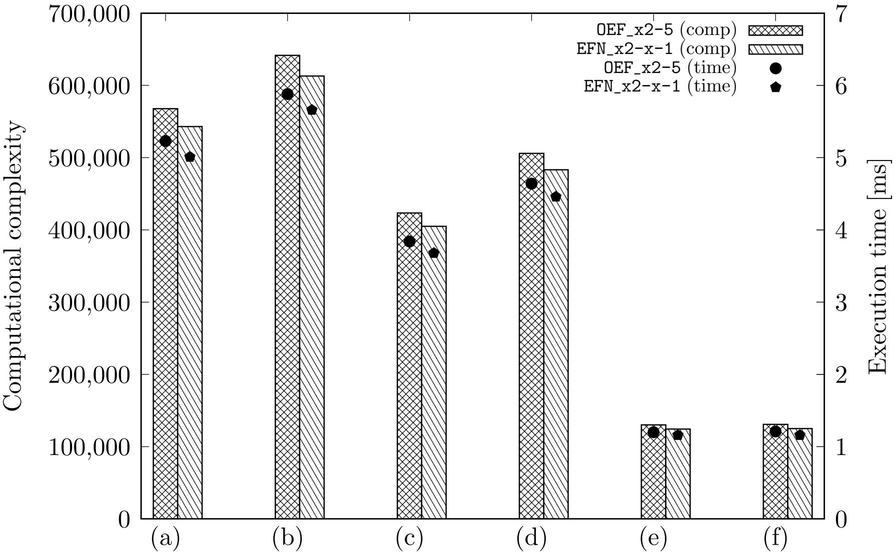


Figure 5.2: Computational complexity and execution time of the SIDH operations (a) Alice's `keygen_iso`, (b) Bob's `keygen_iso`, (c) Alice's `keyshare_iso`, (d) Bob's `keyshare_iso`. (e) Alice's `kernel_gen`, and (f) Bob's `kernel_gen` with p_{441+}.

Chapter 6

Conclusion and Future Works

The mathematical fundamentals of cryptography using elliptic curves are introduced from the description of algebraic systems in Chapter 2. The author presented efficient algorithms for computing the final exponentiation for several candidates of curves suggested for the pairings at the 128-bit security level in Chapter 3. The author also provided the formulas for generating fixed final exponentiation algorithms that are applicable for a certain family of curves with any generalized embedding degrees. In Chapter 4, the author described methods for obtaining attractive subfamilies of pairing-friendly curves with many embedding degrees, which are one of the extended works of [CLN11] by Costello et al. The author presented concrete parameters suggested for the pairings at the 128- and 192-bit security levels. In Chapter 5, the author discussed the SIDH by using several constructions of quadratic extension fields. The performance of SIDH was analyzed by an implementation.

The results of Chapters 3 and 4 contribute to optimizing the pairings and determining the attractive curves and algorithms for computing pairings efficiently. Besides, the result of Chapter 5 contributes to finding new constructions of SIDH which have the competitive performance of the previous ones. The author considers that the results involve important achievements for the practical applications of pairing and SIDH.

Finally, the author briefly shows the future works and outlook. As seen in Chapters 3 and 4, the author is interested in completely operating the settings and algorithms for computing the pairings by the curves and their parameters. To achieve that for much more curves, the author considers that a deeper understanding of the structures of pairing computations specified by the families of curves is required. The author is also interested in an alternative method [Sta07] for computing the Tate pairing via the elliptic nets, however, it is slower than the typical Miller's algorithm. There is a possibility that an efficient pairing computation is provided by improving the elliptic nets. The author also hopes to find such new alternative methods for computing the isogenies and curves for the SIDH.

Bibliography

[AD97] Miklós Ajtai and Cynthia Dwork. "A public-key cryptosystem with worst-case/average-case equivalence". In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing.* 1997, pp. 284–293.

[Adj+18] Gora Adj et al. "On the cost of computing isogenies between supersingular elliptic curves". In: *International Conference on Selected Areas in Cryptography.* Springer. 2018, pp. 322–343.

[AM93] A Oliver L Atkin and François Morain. "Elliptic curves and primality proving". In: *Mathematics of computation* 61.203 (1993), pp. 29–68.

[Aok+00] Kazumaro Aoki et al. "Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis". In: *International workshop on selected areas in cryptography.* Springer. 2000, pp. 39–56.

[Ara+11] Diego F Aranha et al. "Faster explicit formulas for computing pairings over ordinary curves". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* Springer. 2011, pp. 48–68.

[Ara13] Diego F Aranha. *RELIC is an efficient library for cryptography.* `https://github.com/relic-toolkit/relic`. 2013.

[Aza+17] Reza Azarderakhsh et al. "Supersingular isogeny key encapsulation". In: *Submission to the NIST Post-Quantum Standardization project* 152 (2017), pp. 154–155.

[Bar+02] Paulo SLM Barreto et al. "Efficient algorithms for pairing-based cryptosystems". In: *Annual international cryptology conference.* Springer. 2002, pp. 354–369.

[BD19] Razvan Barbulescu and Sylvain Duquesne. "Updating key size estimations for pairings". In: *Journal of Cryptology* 32.4 (2019), pp. 1298–1336.

[BEMG19] Razvan Barbulescu, Nadia El Mrabet, and Loubna Ghammam. *A taxonomy of pairings, their security, their complexity.* Cryptology ePrint Archive, Report 2019/485. `https://eprint.iacr.org/2019/485`. 2019.

[Ber08] Daniel J Bernstein. “The Salsa20 family of stream ciphers”. In: *New stream cipher designs*. Springer, 2008, pp. 84–97.

[Ber+08] Daniel J Bernstein et al. “ChaCha, a variant of Salsa20”. In: *Workshop record of SASC*. Vol. 8. 2008, pp. 3–5.

[Beu+10] Jean-Luc Beuchat et al. “High-speed software implementation of the optimal ate pairing over Barreto–Naehrig curves”. In: *International Conference on Pairing-Based Cryptography*. Springer. 2010, pp. 21–39.

[BF01] Dan Boneh and Matt Franklin. “Identity-based encryption from the Weil pairing”. In: *Annual international cryptology conference*. Springer. 2001, pp. 213–229.

[BGK15] Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. “The tower number field sieve”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2015, pp. 31–55.

[BKV19] Ward Beullens, Thorsten Kleinjung, and Frederik Vercauteren. “CSI-FiSh: Efficient isogeny based signatures through class group computations”. In: *International Conference on the Theory and Application of Cryptology and Information Security*. Springer. 2019, pp. 227–247.

[BLS02] Paulo SLM Barreto, Ben Lynn, and Michael Scott. “Constructing elliptic curves with prescribed embedding degrees”. In: *International Conference on Security in Communication Networks*. Springer. 2002, pp. 257–267.

[BN05] Paulo SLM Barreto and Michael Naehrig. “Pairing-friendly elliptic curves of prime order”. In: *International Workshop on Selected Areas in Cryptography*. Springer. 2005, pp. 319–331.

[Bon+04] Dan Boneh et al. “Public key encryption with keyword search”. In: *International conference on the theory and applications of cryptographic techniques*. Springer. 2004, pp. 506–522.

[BP01] Daniel V Bailey and Christof Paar. “Efficient arithmetic in finite field extensions with application in elliptic curve cryptography”. In: *Journal of cryptology* 14.3 (2001), pp. 153–176.

[BS10] Naomi Benger and Michael Scott. “Constructing tower extensions of finite fields for implementation of pairing-based cryptography”. In: *International Workshop on the Arithmetic of Finite Fields*. Springer. 2010, pp. 180–195.

[BW05] Friederike Brezing and Annegret Weng. “Elliptic curves suitable for pairing based cryptography”. In: *Designs, Codes and Cryptography* 37.1 (2005), pp. 133–141.

[CA69] Stephen A Cook and Stål O Aanderaa. "On the minimum computation time of functions". In: *Transactions of the American Mathematical Society* 142 (1969), pp. 291–314.

[Cam+19] Matthew Campagna et al. *Supersingular isogeny key encapsulation.* `https://sike.org/files/SIDH-spec.pdf`. 2019.

[Cas+18] Wouter Castryck et al. "CSIDH: an efficient post-quantum commutative group action". In: *International Conference on the Theory and Application of Cryptology and Information Security.* Springer. 2018, pp. 395–427.

[CH17] Craig Costello and Huseyin Hisil. "A simple and compact algorithm for SIDH with arbitrary degree isogenies". In: *International Conference on the Theory and Application of Cryptology and Information Security.* Springer. 2017, pp. 303–329.

[CLN10] Craig Costello, Tanja Lange, and Michael Naehrig. "Faster pairing computations on curves with high-degree twists". In: *International Workshop on Public Key Cryptography.* Springer. 2010, pp. 224–242.

[CLN11] Craig Costello, Kristin Lauter, and Michael Naehrig. "Attractive subfamilies of BLS curves for implementing high-security pairings". In: *International conference on cryptology in India.* Springer. 2011, pp. 320–342.

[CLN16] Craig Costello, Patrick Longa, and Michael Naehrig. "Efficient algorithms for supersingular isogeny Diffie-Hellman". In: *Annual International Cryptology Conference.* Springer. 2016, pp. 572–601.

[CM11] Sanjit Chatterjee and Alfred Menezes. "On cryptographic protocols employing asymmetric pairings—the role of Ψ revisited". In: *Discrete Applied Mathematics* 159.13 (2011), pp. 1311–1322.

[Cos12] Craig Costello. *Particularly friendly members of family trees.* Cryptology ePrint Archive, Report 2012/072. `https://eprint.iacr.org/2012/072`. 2012.

[Cos20] Craig Costello. "B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion". In: *International Conference on the Theory and Application of Cryptology and Information Security.* Springer. 2020, pp. 440–463.

[Cou06] Jean Marc Couveignes. *Hard homogeneous spaces.* Cryptology ePrint Archive, Report 2006/291. `https://eprint.iacr.org/2006/291`. 2006.

[CP01] Clifford Cocks and Richard G.E. Pinch. *Identity-based cryptosystems based on the Weil pairing.* Unpublished manuscript. 2001.

[CS18] Craig Costello and Benjamin Smith. "Montgomery curves and their arithmetic". In: *Journal of Cryptographic Engineering* 8.3 (2018), pp. 227–240.

[DCC05] Pu Duan, Shi Cui, and Choong Wah Chan. *Special polynomial families for generating more suitable elliptic curves for pairing-based cryptosystems.* `https://eprint.iacr.org/2005/342`. 2005.

[DF+20] Luca De Feo et al. “SQISign: Compact post-quantum signatures from quaternions and isogenies”. In: *International Conference on the Theory and Application of Cryptology and Information Security.* Springer. 2020, pp. 64–93.

[DFJP14] Luca De Feo, David Jao, and Jérôme Plût. “Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies”. In: *Journal of Mathematical Cryptology* 8.3 (2014), pp. 209–247.

[DG16] Christina Delfs and Steven D Galbraith. “Computing isogenies between supersingular elliptic curves over $\mathbb{F}_p$”. In: *Designs, Codes and Cryptography* 78.2 (2016), pp. 425–440.

[DH76] Whitfield Diffie and Martin Hellman. “New directions in cryptography”. In: *IEEE transactions on Information Theory* 22.6 (1976), pp. 644–654.

[Elg85] Taher Elgamal. “A public key cryptosystem and a signature scheme based on discrete logarithms”. In: *IEEE transactions on information theory* 31.4 (1985), pp. 469–472.

[EMJ17] Nadia El Mrabet and Marc Joye. *Guide to pairing-based cryptography.* CRC Press, 2017.

[FCKRH11] Laura Fuentes-Castaneda, Edward Knapp, and Francisco Rodríguez-Henríquez. “Faster hashing to $\mathbb{G}_2$”. In: *International workshop on selected areas in cryptography.* Springer. 2011, pp. 412–430.

[FH+17] Armando Faz-Hernández et al. “A faster software implementation of the supersingular isogeny Diffie-Hellman key exchange protocol”. In: *IEEE Transactions on Computers* 67.11 (2017), pp. 1622–1636.

[FK19] Georgios Fotiadis and Elisavet Konstantinou. “TNFS resistant families of pairing-friendly elliptic curves”. In: *Theoretical Computer Science* 800 (2019), pp. 73–89.

[FM19] Georgios Fotiadis and Chloe Martindale. *Optimal TNFS-secure pairings on elliptic curves with composite embedding degree.* Cryptology ePrint Archive, Report 2019/555. `https://eprint.iacr.org/2019/555`. 2019.

[FMP20] Emmanuel Fouotsa, Nadia El Mrabet, and Aminatou Pecha. “Optimal ate pairing on elliptic curves with embedding degree $9, 15$ and 27”. In: *journal of Groups, Complexity, Cryptology* 12, issue 1 (2020). URL: `https://gcc.episciences.org/6285`.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. “A taxonomy of pairing-friendly elliptic curves”. In: *Journal of cryptology* 23.2 (2010), pp. 224–280.

[Gal99] Steven D Galbraith. “Constructing isogenies between elliptic curves over finite fields”. In: *LMS Journal of Computation and Mathematics* 2 (1999), pp. 118–138.

[GMT20] Aurore Guillevic, Simon Masson, and Emmanuel Thomé. “Cocks–Pinch curves of embedding degrees five to eight and optimal ate pairing computation”. In: *Designs, Codes and Cryptography* 88.6 (2020), pp. 1047–1081.

[GPS08] Steven D Galbraith, Kenneth G Paterson, and Nigel P Smart. “Pairings for cryptographers”. In: *Discrete Applied Mathematics* 156.16 (2008), pp. 3113–3121.

[GS10] Robert Granger and Michael Scott. “Faster squaring in the cyclotomic subgroup of sixth degree extensions”. In: *International Workshop on Public Key Cryptography*. Springer. 2010, pp. 209–223.

[GS19] Aurore Guillevic and Shashank Singh. *On the alpha value of polynomials in the tower number field sieve algorithm*. Cryptology ePrint Archive, Report 2019/885. `https://eprint.iacr.org/2019/885`. 2019.

[Gui20] Aurore Guillevic. “A short-list of pairing-friendly curves resistant to special TNFS at the 128-bit security level”. In: *PKC 2020-IACR International Conference on Practice and Theory of Public-Key Cryptography*. Vol. 12111. Springer. 2020, pp. 535–564.

[Har+75] Robin Hartshorne et al. *Algebraic geometry, Arcata 1974*. Vol. 29. American Mathematical Soc., 1975.

[HHT20] Daiki Hayashida, Kenichiro Hayasaka, and Tadanori Teruya. *Efficient final exponentiation via cyclotomic structure for pairings over families of elliptic curves*. Cryptology ePrint Archive, Report 2020/875. `https://eprint.iacr.org/2020/875`. 2020.

[HSV06] Florian Hess, Nigel P Smart, and Frederik Vercauteren. “The eta pairing revisited”. In: *IEEE transactions on information theory* 52.10 (2006), pp. 4595–4602.

[Jan18] Ján Jančár. *Ecgen: Tool for generating Elliptic curve domain parameters*. `https://github.com/J08nY/ecgen`. 2018.

[JDF11] David Jao and Luca De Feo. “Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies”. In: *International Workshop on Post-Quantum Cryptography*. Springer. 2011, pp. 19–34.

[Jou00] Antoine Joux. "A one round protocol for tripartite Diffie–Hellman". In: *International algorithmic number theory symposium*. Springer. 2000, pp. 385–393.

[JS19] Samuel Jaques and John M Schanck. "Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE". In: *Annual International Cryptology Conference*. Springer. 2019, pp. 32–61.

[KAH00] Tetsutaro Kobayashi, Kazumaro Aoki, and Fumitaka Hoshino. "OEF using a successive extension". In: *The 2000 Symposium on Cryptography and Information Security*. B02. 2000.

[Kar13] Koray Karabina. "Squaring in cyclotomic subgroups". In: *Mathematics of Computation* 82.281 (2013), pp. 555–579.

[KB16] Taechan Kim and Razvan Barbulescu. "Extended tower number field sieve: A new complexity for the medium prime case". In: *Annual International Cryptology Conference*. Springer. 2016, pp. 543–571.

[KD13] Cameron F Kerry and Charles Romine Director. *FIPS PUB 186-4 federal information processing standards publication digital signature standard (DSS)*. `https://cryptome.org/2013/07/NIST.FIPS.186-4.pdf`. 2013.

[KJ17] Taechan Kim and Jinhyuck Jeong. "Extended tower number field sieve with application to finite fields of arbitrary composite extension degree". In: *IACR International Workshop on Public Key Cryptography*. Springer. 2017, pp. 388–408.

[KM05] Neal Koblitz and Alfred Menezes. "Pairing-based cryptography at high security levels". In: *IMA International Conference on Cryptography and Coding*. Springer. 2005, pp. 13–36.

[KO62] Anatolii Alekseevich Karatsuba and Yu P Ofman. "Multiplication of many-digital numbers by automatic computers". In: *Doklady Akademii Nauk*. Vol. 145. 2. Russian Academy of Sciences. 1962, pp. 293–294.

[Kob87] Neal Koblitz. "Elliptic curve cryptosystems". In: *Mathematics of computation* 48.177 (1987), pp. 203–209.

[Kob94] Neal Koblitz. *A course in number theory and cryptography*. Vol. 114. Springer Science & Business Media, 1994.

[Koj20] Nuida Koji. *Post-quantum cryptography*. Morikita Publishing Co., Ltd., 2020.

[KSS08] Ezekiel J Kachisa, Edward F Schaefer, and Michael Scott. "Constructing Brezing-Weng pairing-friendly elliptic curves using elements in the cyclotomic field". In: *International Conference on Pairing-Based Cryptography*. Springer. 2008, pp. 126–135.

[Lem13] Franz Lemmermeyer. *Reciprocity laws: from Euler to Eisenstein*. Springer Science & Business Media, 2013.

[Len+93] Arjen K Lenstra et al. "The number field sieve". In: *The development of the number field sieve*. Springer, 1993, pp. 11–42.

[Lin+08] Xibin Lin et al. "Computing the ate pairing on elliptic curves with embedding degree $k = 9$". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 91.9 (2008), pp. 2387–2393.

[LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. "Factoring polynomials with rational coefficients". In: *Mathematische annalen* 261.ARTICLE (1982), pp. 515–534.

[Mat19] Kazuto Matsuo. "SIDH over quadratic twists". In: *The 2019 Symposium on Cryptography and Information Security*. 2019.

[McE78] Robert J McEliece. "A public-key cryptosystem based on algebraic". In: *Coding Thv* 4244 (1978), pp. 114–116.

[MI88] Tsutomu Matsumoto and Hideki Imai. "Public quadratic polynomial-tuples for efficient signature-verification and message-encryption". In: *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer. 1988, pp. 419–453.

[Mil04] Victor S Miller. "The Weil pairing, and its efficient calculation". In: *Journal of cryptology* 17.4 (2004), pp. 235–261.

[Mil85] Victor S Miller. "Use of elliptic curves in cryptography". In: *Conference on the theory and application of cryptographic techniques*. Springer. 1985, pp. 417–426.

[MNT01] Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. "New explicit conditions of elliptic curve traces for FR-reduction". In: *IEICE transactions on fundamentals of electronics, communications and computer sciences* 84.5 (2001), pp. 1234–1243.

[Mon85] Peter L Montgomery. "Modular multiplication without trial division". In: *Mathematics of computation* 44.170 (1985), pp. 519–521.

[Mon87] Peter L Montgomery. "Speeding the Pollard and elliptic curve methods of factorization". In: *Mathematics of computation* 48.177 (1987), pp. 243–264.

[MSS16] Alfred Menezes, Palash Sarkar, and Shashank Singh. "Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography". In: *International Conference on Cryptology in Malaysia*. Springer. 2016, pp. 83–108.

[Mul+88] Ronald C Mullin et al. "Optimal normal bases in GF(pn)". In: *Discrete Applied Mathematics* 22.2 (1988), pp. 149–161.

[Nog+08] Yasuyuki Nogami et al. "Integer variable χ–based ate pairing". In: *International Conference on Pairing-Based Cryptography*. Springer. 2008, pp. 178–191.

[NSM03] Yasuyuki Nogami, Akinori Saito, and Yoshitaka Morikawa. "Finite extension field with modulus of all-one polynomial and representation of its elements for fast arithmetic operations". In: *IEICE transactions on fundamentals of electronics, communications and computer sciences* 86.9 (2003), pp. 2376–2387.

[Ogu+12] Naoki Ogura et al. "A note on the pairing computation using normalized Miller functions". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* 95.1 (2012), pp. 196–203.

[Per+11] Geovandro CCF Pereira et al. "A family of implementation-friendly BN elliptic curves". In: *Journal of systems and software* 84.8 (2011), pp. 1319–1326.

[Pol78] John M Pollard. "Monte Carlo methods for index computation (mod p)". In: *Mathematics of computation* 32.143 (1978), pp. 918–924.

[Pol93] John M Pollard. "Factoring with cubic integers". In: *The development of the number field sieve*. Springer, 1993, pp. 4–10.

[Ren18] Joost Renes. "Computing isogenies between Montgomery curves using the action of (0, 0)". In: *International Conference on Post-Quantum Cryptography*. Springer. 2018, pp. 229–247.

[RNL19] Barbulescu Razvan, El Mrabet Nadia, and Ghammam Loubna. *A taxonomy of pairings, their security, their complexity*. Cryptology ePrint Archive, Report 2020/875. `https://eprint.iacr.org/2019/485.pdf`. 2019.

[RS06] Alexander Rostovtsev and Anton Stolbunov. *Public-key cryptosystem based on isogenies*. Cryptology ePrint Archive, Report 2006/145. `https://eprint.iacr.org/2006/145`. 2006.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems". In: *Communications of the ACM* 21.2 (1978), pp. 120–126.

[Sch93] Oliver Schirokauer. “Discrete logarithms and local units”. In: *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences* 345.1676 (1993), pp. 409–423.

[Sco+09] Michael Scott et al. “On the final exponentiation for calculating pairings on ordinary elliptic curves”. In: *International Conference on Pairing-Based Cryptography*. Springer. 2009, pp. 78–88.

[Sha06] Hovav Shacham. *New paradigms in signature schemes*. Stanford University, 2006.

[Shi+07] Taizo Shirai et al. “The 128-bit blockcipher CLEFIA”. In: *International workshop on fast software encryption*. Springer. 2007, pp. 181–195.

[Shi10] Masaaki Shirase. *Barreto-Naehrig curve with fixed coefficient- Efficiently constructing pairing-friendly curves-*. Cryptology ePrint Archive, Report 2010/134. `https://eprint.iacr.org/2010/134`. 2010.

[Shi15] Mitsunari Shigeo. *Applied cryptography for the cloud*. SHUWA SYSTEM CO., Ltd., 2015.

[Sho94] Peter W Shor. “Algorithms for quantum computation: discrete logarithms and factoring”. In: *Proceedings 35th annual symposium on foundations of computer science*. Ieee. 1994, pp. 124–134.

[Sil09] Joseph H Silverman. *The arithmetic of elliptic curves*. Vol. 106. Springer, 2009.

[SL02] Martijn Stam and Arjen K Lenstra. “Efficient subgroup exponentiation in quadratic and sixth degree extensions”. In: *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer. 2002, pp. 318–332.

[SOK00] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. “Cryptosystems based on pairing”. In: *The 2000 Symposium on Cryptography and Information Security*. 2000.

[SSM21] Zihao Song, Junichi Sakamoto, and Tsutomu Matsumoto. “An implementation of a pairing on Cocks-Pinch curve with embedding degree 6”. In: *IEICE Tech. Rep.* 121.1 (2021), pp. 19–24.

[Sta07] Katherine E Stange. “The Tate pairing via elliptic nets”. In: *International Conference on Pairing-Based Cryptography*. Springer. 2007, pp. 329–348.

[SW05] Amit Sahai and Brent Waters. “Fuzzy identity-based encryption”. In: *Annual international conference on the theory and applications of cryptographic techniques*. Springer. 2005, pp. 457–473.

[tea15] The GMP development team. *GNU MP: the GNU Multiple Precision Arithmetic Library, 6.1.2.* `https://gmplib.org`. 2015.

[Ter+13] Tadanori Teruya et al. "Constructing symmetric pairings over supersingular elliptic curves with embedding degree three". In: *International Conference on Pairing-Based Cryptography.* Springer. 2013, pp. 97–112.

[Too63] Andrei L Toom. "The complexity of a scheme of functional elements realizing the multiplication of integers". In: *Soviet Mathematics Doklady.* Vol. 3. 4. 1963, pp. 714–716.

[Vél71] Jacques Vélu. "Isogénies entre courbes elliptiques". In: *CR Acad. Sci. Paris, Séries A* 273 (1971), pp. 305–347.

[Ver09] Frederik Vercauteren. "Optimal pairings". In: *IEEE Transactions on Information Theory* 56.1 (2009), pp. 455–461.

[Wat69] William C Waterhouse. "Abelian varieties over finite fields". In: *Annales scientifiques de l'École Normale Supérieure.* Vol. 2. 4. 1969, pp. 521–560.

[YTS15] Takanori Yasuda, Tsuyoshi Takagi, and Kouichi Sakurai. "Constructing pairing-friendly elliptic curves using global number fields". In: *International symposium on computing and networking.* IEEE. 2015, pp. 477–483.

[ZL12] Xusheng Zhang and Dongdai Lin. "Analysis of optimum pairing products at high security levels". In: *International Conference on Cryptology in India.* Springer. 2012, pp. 412–430.

Appendix A

Formulas of Cubings in Cyclotomic Subgroup

The author describes cubings in the cyclotomic subgroup $G_{\Phi_k(p)}$ of $\mathbb{F}_{p^k}^*$. In the following, let $k \mid (p-1)$, $3 \mid k$, and $q = p^{k/3}$. Let $\mathbb{F}_{q^3}$ is defined by $\mathbb{F}_q[x]/(x^3 - \zeta) \cong \mathbb{F}_q(\omega)$ where ω is an element in $\mathbb{F}_{q^3}$ such that $\omega^3 = \zeta$. Then, there is a cyclotomic subgroup of $\mathbb{F}_{q^3}^*$ of order $\Phi_3(q)$, which is denoted by $G_{\Phi_3(q)}$. Since $G_{\Phi_3(q)}$ involves a cyclotomic subgroup $G_{\Phi_k(p)}$, arithmetic operations in $G_{\Phi_3(q)}$ is also available in $G_{\Phi_k(p)}$. In the following, the author describes the cubing in $\mathbb{F}_{q^3}$ and derives efficient cubing in $G_{\Phi_3(q)}$ based on [GS10] which shows efficient squarings.

A.1 Typical cubing

Let a be an element in $\mathbb{F}_{q^3}^*$ represented as $a = a_0 + a_1\omega + a_2\omega^2$ where $a_0, a_1, a_2 \in \mathbb{F}_q$. Then, the formula of typical cubing is given as follows:

$$\begin{aligned}a^3 =&(a_0 + a_1\omega + a_2\omega^2)^3\\ =&a_0^3 + (6a_0a_1a_2 + a_1^3)\zeta + a_2^3\zeta^2\\ &\quad + 3\left(a_0^2a_1 + a_2(a_0a_2 + a_1^2)\zeta\right)\omega\\ &\qquad + 3\left(a_0(a_0a_2 + a_1^2) + a_1a_2^2\zeta\right)\omega^2.\end{aligned} \tag{A.1}$$

Assuming $a^3 = b_0 + b_1\omega + b_2\omega^2 \in \mathbb{F}_{q^3}$, the elements $b_0, b_1, b_2 \in \mathbb{F}_q$ are obtained as follows:

$$\begin{aligned}&t_0 = \left((a_0 + a_2)^2 - (a_0^2 + a_2^2)\right)/2 + a_1^2,\\ &t_1 = a_0t_0, t_2 = a_2t_0, t_3 = a_0^2a_1, t_4 = a_1a_2^2,\\ &t_5 = (a_0 + a_1 + a_2)^3 - (a_0^3 + a_2^3 + t_1 + t_2 + t_3 + t_4),\\ &b_0 = a_0^3 + t_5\zeta + a_2^3\zeta^2, b_1 = 3(t_3 + t_2\zeta), b_2 = 3(t_1\zeta + t_4),\end{aligned} \tag{A.2}$$

which leads to the following sequence of operations.

$$\begin{aligned}
&t_0 = a_0^2, t_1 = a_1^2, t_2 = a_2^2, t_3 = t_0a_0, t_4 = t_2a_2,\\
&t_5 = a_0 + a_2, t_6 = t_5 + a_1, t_7 = t_6^2, t_6 = t_6t_7, t_5 = t_5^2,\\
&t_5 = t_5 - t_0, t_5 = t_5 - t_2, t_5 = t_5/2, t_1 = t_1 + t_5,\\
&t_5 = t_1a_2, t_1 = t_1a_0, t_0 = t_0a_1, t_2 = t_2a_1, t_7 = t_5\zeta,\\
&t_7 = t_0 + t_7, t_8 = 2t_7, b_1 = t_7 + t_8, t_7 = t_2\zeta,\\
&t_7 = t_1 + t_7, t_8 = 2t_7, b_2 = t_7 + t_8, t_0 = t_0 + t_1,\\
&t_0 = t_0 + t_2, t_0 = t_0 + t_5, t_1 = 2t_0, t_0 = t_0 + t_1,\\
&t_0 = t_0 + t_3, t_0 = t_0 + t_4, t_0 = t_6 - t_0, t_0 = t_0\zeta,\\
&t_4 = t_4\zeta^2, t_0 = t_0 + t_4, b_0 = t_0 + t_3. \qquad \text{(A.3)}
\end{aligned}$$

Thus, the typical cubing takes 7 multiplications, 5 squarings, 18 additions, 4 shift operations, and 4 multiplication by ζ in $\mathbb{F}_q$.

A.2 Cyclotomic cubing

Let $\alpha = \alpha_0 + \alpha_1\omega + \alpha_2\omega^2$ with $\alpha_0, \alpha_1, \alpha_2 \in \mathbb{F}_q$ be an element in $G_{\Phi_3(q)}$. Since α has a specific order, there is a relation associated with α_i for $0 \le i \le 2$ as shown in the following lemma.

Lemma A.1. For $\alpha \in G_{\Phi_3(q)}$, the following is satisfied.

$$\alpha^{\Phi_3(q)} = \alpha_0^3 + (-3\alpha_0\alpha_1\alpha_2 + \alpha_1^3)\zeta + \alpha_2^3\alpha^2 = 1. \qquad \text{(A.4)}$$

Proof of Lemma A.1. It is clearly satisfied the equation $\alpha^{\Phi_3(q)} = \alpha \cdot \alpha^q \cdot \alpha^{q^2} = 1$ where α^q is given as follows: $\alpha^q = (\alpha_0 + \alpha_1\omega + \alpha_2\omega)^q = \alpha_0 + \alpha_1\omega^{q-1}\omega + \alpha_2(\omega^{q-1})^2\omega^2 = \alpha_0 + \alpha_1\zeta^{\frac{q-1}{3}}\omega + \alpha_2(\zeta^{\frac{q-1}{3}})^2\omega^2$. Note that $\zeta^{\frac{q-1}{3}}$ is a primitive cube root of unity since ζ is cubic non-residue in $\mathbb{F}_q^*$. As the same manner, it is obtained that $\alpha^{q^2} = \alpha_0 + \alpha_1(\zeta^{\frac{q-1}{3}})^2\omega + \alpha_2\zeta^{\frac{q-1}{3}}\omega^2$. Assuming $\epsilon = \zeta^{\frac{q-1}{3}}$, $\epsilon^3 = 1$ and $\epsilon^2 + \epsilon + 1 = 0$. Then, the equation can be deformed as follows:

$$\begin{aligned}
\alpha^{\Phi_3(q)} &= \alpha \cdot \alpha^q \cdot \alpha^{q^2}\\
&= \alpha_0^3 + \alpha_0^3\alpha_0^2\alpha_1(\epsilon^2 + \epsilon + 1)\omega\\
&\qquad + \alpha_0\left((\alpha_0\alpha_2 + \alpha_1^2)(\epsilon^2 + \epsilon) + (\alpha_0\alpha_2 + \alpha_1^2\epsilon^3)\right)\omega^2\\
&\qquad + \left(\alpha_0\alpha_1\alpha_2(\epsilon^4 + 3\epsilon^2 + 2\epsilon) + \alpha_1^3\epsilon^3\right)\omega^3\\
&\qquad + \alpha_2\left((\alpha_0\alpha_2 + \alpha_1^2)(\epsilon^3 + \epsilon^2) + (\alpha_0\alpha_2 + \alpha_1^2\epsilon^3)\epsilon\right)\omega^4\\
&\qquad + a_1a_2^2\epsilon^2(\epsilon^2 + \epsilon + 1)\omega^5 + a_2^3\epsilon^3\omega^6
\end{aligned}$$

$$= \alpha_0^3 + (-3\alpha_0\alpha_1\alpha_2 + \alpha_1^3)\zeta + \alpha_2^3\zeta^2 = 1. \tag{A.5}$$

Thus, Lemma 1 is obtained. □

When applying Lemma A.1 for the typical cubing given in Eq. (A.1), the formula of cyclotomic cubing is obtained as follows:

$$\begin{aligned}\alpha^3 =&(\alpha_0 + \alpha_1\omega + \alpha_2\omega^2)^3\\ =&1 + 9\alpha_0\alpha_1\alpha_2\zeta\\ &+ 3\left(\alpha_0^2\alpha_1 + \alpha_2(\alpha_0\alpha_2 + \alpha_1^2)\zeta\right)\omega\\ &+ 3\left(\alpha_0(\alpha_0\alpha_2 + \alpha_1^2) + \alpha_1\alpha_2^2\zeta\right)\omega^2.\end{aligned} \tag{A.6}$$

Assuming $\alpha^3 = \beta_0 + \beta_1\omega + \beta_2\omega^2 \in \mathbb{F}_{q^3}$, the elements $\beta_0, \beta_1, \beta_2 \in \mathbb{F}_q$ are obtained as follows:

$$\begin{aligned}&t_0 = \left((\alpha_0 + \alpha_2)^2 - (\alpha_0^2 + \alpha_2^2)\right)/2, t_1 = t_0 + \alpha_1^2,\\ &t_2 = t_0\alpha_1, t_3 = \alpha_0 t_1, t_4 = \alpha_2 t_1, t_5 = \alpha_0^2\alpha_1, t_6 = \alpha_1\alpha_2^2,\\ &\beta_0 = 1 + 9t_2\zeta, \beta_1 = 3(t_5 + t_4\zeta), \beta_2 = 3(t_3\zeta + t_6),\end{aligned} \tag{A.7}$$

which leads to the following sequence of operations.

$$\begin{aligned}&t_0 = \alpha_0^2, t_1 = \alpha_1^2, t_2 = \alpha_2^2, t_3 = \alpha_0 + \alpha_2, t_3 = t_3^2,\\ &t_3 = t_3 - t_0, t_3 = t_3 - t_2, t_3 = t_3/2, t_4 = t_3\alpha_1,\\ &t_5 = 2^3 t_4, t_4 = t_4 + t_5, t_4 = t_4\zeta, \beta_0 = t_4 + 1,\\ &t_1 = t_1 + t_3, t_0 = t_0\alpha_1, t_3 = t_1\alpha_2, t_3 = t_3\zeta,\\ &t_3 = t_0 + t_3, t_4 = 2t_3, \beta_1 = t_3 + t_4, t_2 = t_2\alpha_1,\\ &t_2 = t_2\zeta, t_1 = t_1\alpha_0, t_1 = t_1 + t_2, t_2 = 2t_1, \beta_2 = t_2 + t_2.\end{aligned} \tag{A.8}$$

This takes 5 multiplications, 4 squarings, 9 additions, 4 shift operations, and 3 multiplication by ζ in $\mathbb{F}_q$, and 1 addition in $\mathbb{F}_p$. When comparing the operation counts of the cubings, it is found that the cyclotomic cubing can reduce 2 multiplications, 1 squaring, 9 additions, and 1 multiplication by ζ in $\mathbb{F}_q$ from the calculation cost of the typical one.

Appendix B

Reproduced Calculation Costs of Final Exponentiations

We reproduce the calculation costs for the final exponentiation for curves with $k = 6$ and 12 by using state-of-the-art algorithms. In the following, the author refers to [GMT20] and assumes the calculation costs of the arithmetics in $\mathbb{F}_{p^k}$ as Table B.1.

B.1 Cocks-Pinch curve with $k = 6$

According to [GMT20], the Cocks-Pinch curve has the parameterizations of $p(x)$, $r(x)$, and $t(x)$ as follows:

$$\begin{cases} p(x) &= \frac{1}{12}((9h_y^2 + 6h_y + 4)x^4 - (18h_y^2 + 6h_y + 12)x^3 \\ & \quad +(27h_y^2+18h_y+16)x^2 - (18h_y^2+12h_y)x + 9h_y^2+12h_y+4), \\ r(x) &= \Phi_6(x) = x^2 - x + 1, \\ t(x) &= x + 1 - r(x), \end{cases} \tag{B.1}$$

where h_y is an integer. In the following, let z be an integer making $p(z)$ and $r(z)$ being primes. Then, the exponent of the final exponentiation is given by $(p(z)^6 - 1)/r(z) = (p(z)^3 - 1) \cdot (p(z) + 1) \cdot (p(z)^2 - p(z) + 1)/r(x)$ where $(p(z)^3 - 1) \cdot (p(z) + 1)$ and $d(x) = (p(z)^2 - p(z) + 1)/r(z)$ are easy and hard parts, respectively. Although [GMT20] provided the hard part representation, it does not work. In [SSM21], Song et al. corrected that and proposed a multiple $d'(z) = 3d(z)$ of $d(z)$ that is denoted by

$$d'(z) = 3(z^2 + 3(-z + 1)) + 3c(z) \cdot (p(z) - (z^2 + 2(-z + 1))), \tag{B.2}$$

$$3c(z) = ((9w^2 + 3w + 1) \cdot (z - 1) + (9w^2 + 6w)) \cdot (z - 1) + 9w^2 + 9w + 3, \tag{B.3}$$

Table B.1: The calculation costs of the arithmetics in $\mathbb{F}_{p^k}$ of $k = 1$, 2, 6 and 12.

k	m_k	s_{ck}	i_k	i_{ck}	f_k^i
1	m_1	m_1	$25m_1$	-	-
2	$3m_1$	$2m_1$	$29m_1$	-	-
6	$18m_1$	$6m_1$	$59m_1$	$4m_1$	$4m_1$
12	$54m_1$	$18m_1$	$119m_1$	$10m_1$	$10m_1$

where $w = h_y/2$ and which results in the following computations.

$$\begin{aligned}
&t_0 = f^z, t_1 = t_0^z, t_0 = t_0^{-1}, t_0 = t_0 \cdot f,\\
&t_2 = t_0^2, t_0 = t_0 \cdot t_2, t_0 = t_0 \cdot t_1, t_3 = t_0^2, t_0 = t_3 \cdot t_0,\\
&t_1 = t_1 \cdot t_2, t_1 = t_1^{-1}, t_2 = f^p, t_1 = t_1 \cdot t_2,\\
&t_2 = t_1^2, t_2 = t_2 \cdot t_1, t_3 = t_2^w, t_4 = t_3^2, t_5 = t_4 \cdot t_3, t_5 = t_5^w, t_4 = t_5 \cdot t_4,\\
&t_5 = t_3^{-1}, t_5 = t_5 \cdot t_4, t_5 = t_5 \cdot t_1, t_3 = t_4 \cdot t_3, t_3 = t_3 \cdot t_2,\\
&t_5 = t_5^{z-1}, t_5 = t_5 \cdot t_4, t_5 = t_5^{z-1}, t_5 = t_5 \cdot t_3, \mu = t_5 \cdot t_0.
\end{aligned} \tag{B.4}$$

This requires the calculation cost $2u_6^z + 2u_6^{z-1} + 2u_6^w + 16m_6 + 4s_{c6} + 3i_{c6} + f_6^1$. Adding the cost of the easy part $2m_6 + i_6 + f_6^1 + f_6^3$, we obtain the calculation cost of the final exponentiation as $2u_6^z + 2u_6^{z-1} + 2u_6^w + 18m_6 + 4s_{c6} + i_6 + 3i_{c6} + 2f_6^1 + f_6^3$.

In contrast, we propose to use the simpler decomposition of $d'(z)$ such that

$$d'(z) = 3c'(z) \cdot (p(z) + z - 1) + 3, \tag{B.5}$$

$$3c'(z) = ((9w^2 + 3w + 1)z - (9w^2 + 2))z + (9w^2 + 3w + 1). \tag{B.6}$$

Assuming f is an element after computing the easy part, $\mu = f^{d'(z)}$ is computed by the following sequence of the operations.

$$\begin{aligned}
&t_0 = f^2, t_1 = t_0 \cdot f, t_2 = t_1^w, t_3 = t_2^2, t_4 = t_3 \cdot t_2, t_4 = t_4^w,\\
&t_0 = t_0 \cdot t_4, t_0 = t_0^{-1}, t_4 = t_4 \cdot f, t_3 = t_3 \cdot t_4, t_2 = t_2 \cdot t_4,\\
&t_2 = t_2^z, t_0 = t_2 \cdot t_0, t_0 = t_0^z, t_0 = t_0 \cdot t_3,\\
&t_2 = t_0^p, t_0 = t_0^{z-1}, t_0 = t_0 \cdot t_2, \mu = t_0 \cdot t_1.
\end{aligned} \tag{B.7}$$

The above requires the calculation costs $2u_6^z + u_6^{z-1} + 2u_6^w + 10m_6 + 2s_{c6} + i_{c6} + f_6^1$. Adding the cost of the easy part, we obtain the calculation cost of the final exponentiation as $2u_6^z + u_6^{z-1} + 2u_6^w + 12m_6 + 2s_{c6} + i_6 + i_{c6} + 2f_6^1 + f_6^3$. Since the proposal results in reducing $u_6^{z-1} + 6m_6 + 2s_{c6} + 2i_{c6}$ from the previous one, here we adopt the proposed decomposition for the calculation cost estimation of the final exponentiation.

For the pairing at the 128-bit security level, in [GMT20], Guillevic suggested using z and h_y such that

$$z = 2^{128} - 2^{124} - 2^{69}, \tag{B.8}$$

$$h_y = 2^{80} - 2^{70} - 2^{66} - 2^{14} + 2^5. \tag{B.9}$$

Then, the calculation costs of the exponents by z and w are given by $u_6^z = 4(128 - 1)m_1 + (12 - 3)m_1 + 2m_6 + 6s_1 + i_1 + i_{c6} = 588m_1$, $u_6^{z-1} = u_6^z + m_6 = 606m_1$, and $u_6^w = 4(79-1)m_1 + (24-3)m_1 + 4m_6 + 12s_1 + i_1 + i_{c6} = 446m_1$, respectively (see Corollary 4.1 in [Kar13]). Thus, the calculation cost of the final exponentiation is estimated as $2(588m_1) + (606m_1) + 2(446m_1) + 12(18m_1) + 2(6m_1) + (59m_1) + (4m_1) + 2(4m_1) + (4m_1) = 2977m_1$.

B.2 BLS curve with $k = 12$

We recall that the BLS family of pairing-friendly curves with $k = 12$ has the parameters $p(x) = \frac{1}{3}(x-1)^2 \cdot r(x) + x$, $r(x) = \Phi_{12}(x) = x^4 - x^2 + 1$, and $t(x) = x + 1$. For an integer z making $p(z)$ and $r(z)$ being primes, the exponent of the final exponentiation is given as $(p(z)^{12} - 1)/r(z) = (p(z)^6 - 1) \cdot (p(z)^2 + 1) \cdot (p(z)^4 - p(z)^2 + 1)/r(z)$ where $(p(z)^6 - 1) \cdot (p(z)^2 + 1)$ and $d(z) = (p(z)^4 - p(z)^2 + 1)/r(z)$ is easy and hard parts, respectively. In [HHT20], Hayashida et al. proposed to use a multiple $d'(z) = 3d(z) = (z-1)^2 \cdot (z + p(z)) \cdot (z^2 + p(z)^2 - 1) + 3$. If $2 \mid z$, the calculation cost of the hard part is given by $4u_{12}^z + u_{12}^w + 7m_{12} + s_{c12} + 2i_{c12} + f_{12}^1 + f_{12}^2$ (see Example 3.7). Adding the cost of the easy part given by $2m_{12} + i_{12} + f_{12}^2 + f_{12}^6$, the cost of the final exponentiation is $4u_{12}^z + u_{12}^w + 9m_{12} + s_{c12} + i_{12} + 2i_{c12} + 2f_{12}^1 + f_{12}^2 + f_{12}^6$.

For the pairing at the 128-bit security level, it is suggested using $z = -2^{77} + 2^{50} + 2^{33}$ in [BD19], which leads to $u_{12}^z = 4(77-1)m_2 + (12-3)m_2 + 2m_{12} + 6s_2 + i_2 + i_{c12} = 1098m_1$ and $u_{12}^w = 4(76-1)m_2 + (12-3)m_2 + 2m_{12} + 6s_2 + i_2 + i_{c12} = 1086m_1$ (see Corollary 4.1 in [Kar13]). Then, the calculation cost is estimated as $4(1098m_1) + (1086m_1) + 9(54m_1) + (18m_1) + (119m_1) + 2(10m_1) + 4(10m_1) = 6161m_1$

Appendix C

Miller's Algorithm for Pairings on Curves with Embedding Degree of Multiple of Three

Algorithm C.1 shows Miller's algorithm for computing the ate variant pairings e_{a_T} which adopt the rational function $f_{T,Q}$ where T is a loop parameter and Q is a point trace-zero subgroup $\mathcal{G}_2$. For the curves with embedding degree k of multiple of 3, Zhang et al. proposed an alternative function of $l_{Q_1,Q_2}/v_{Q_1+Q_2}$ for points Q_1 and Q_2 in $\mathcal{G}_2$ without denominator in [ZL12]. This results in avoiding the execution of the inversions in $\mathbb{F}_{p^k}$ in `DBL`, `ADD`, and `SUB` steps. However, unfortunately, there still remain v_Q^{-1} in `INIT`$_-$ and `SUB` steps. Thus, for fast Miller's algorithm, it is advantageous to use not only T with low Hamming weight but also T such that it does not contain $t_i = -1$ for $0 \leq i < n$ as much as possible to avoid `SUB` step. However, since there are not many elliptic curves that fulfill the above requirements, the range of the practical choices of elliptic curves is limited especially for odd embedding degrees. To ease this restriction, the author proposes to compute the ate pairing by using a rational function $f_{-T,Q}$ instead of $f_{T,Q}$ which results in swapping the `ADD` and `SUB` steps by using a technique given by Hess et al. in [HSV06].

Theorem C.1. The value $e_{a_T}(Q,P)$ can also be computed as follows (see Sect. 2 of [HSV06]):

$$e_{a_T}(Q,P) = \left((f_{-T,Q}(P) \cdot v_{-TQ}(P))^{-1}\right)^{\frac{p^k-1}{r}}. \tag{C.1}$$

Proof of Theorem C.1. A map with the above image also ensures an ate pairing since $\mathrm{div}((f_{-T,Q} \cdot v_{-TQ})^{-1}) = -(-T(Q) - (-TQ) + (T+1)(\mathcal{O})) - ((-TQ) + (TQ) - 2(\mathcal{O})) = T(Q) - (TQ) - (T-1)(\mathcal{O}) = \mathrm{div}(f_{T,Q})$. □

From Theorem C.1, it is found that the ate pairing can be computed by using $f_{-T,Q}$

Algorithm C.1: Miller's algorithm for computing the ate variant pairings.

```
Input: $T = t_l 2^l + \cdots + t_1 2^1 + t_0 2^0 \in \mathbb{Z}$ where $t_i \in \{-1,0,1\}$, $P \in \mathcal{G}_1$, and $Q \in \mathcal{G}_2$
Output: $f_{T,Q}(P)$
If $t_l = 1$ then
  $f \leftarrow 1, R \leftarrow Q;$                                                  //INIT$_+$
else if $t_l = -1$ then
  $f \leftarrow v_Q^{-1}(P), R \leftarrow -Q;$                                        //INIT$_-$
endif
For $i$ from $l-1$ downto 0 do
  $f \leftarrow f^2 \cdot \frac{l_{R,R}(P)}{v_{R+R}(P)}, R \leftarrow R+R;$                //DBL
  If $t_i = 1$ then
    $f \leftarrow f \cdot \frac{l_{R,Q}(P)}{v_{R+Q}(P)}, R \leftarrow R+Q;$               //ADD
  else if $t_i = -1$ then
    $f \leftarrow f \cdot \frac{l_{R,-Q}(P)}{v_{R-Q}(P)} \cdot v_Q^{-1}(P), R \leftarrow R-Q;$   //SUB
  endif
endfor
return $f$;
```

with some adjustments instead of $f_{T,Q}$, however, it involves an additional inversion in $\mathbb{F}_{p^k}^*$. Fortunately, for the target ate pairings which are defined on the elliptic curves with $3 \mid k$, this inversion can be removed by extending the Aranha et al.'s work [Ara+11] for an optimal ate pairing on BN curves (see Lemma 1 of [Ara+11]). Indeed, the inversion is replaced with inexpensive exponentiations according to the following corollary.

Corollary C.2. If $3 \mid k$, the value $e_{a_T}(Q,P)$ is deformed as follows:

$$e_{a_T}(Q,P) = \left((f_{-T,Q}(P)\cdot v_{-TQ}(P))^{q^2+q}\right)^{\frac{p^k-1}{r}}, \tag{C.2}$$

where $q = p^{k/3}$.

Proof of Corollary C.2. The corollary is true since the exponent is represented as $-(p^k-1)/r = (1-q)\cdot(q^2+q+1)/r \equiv (q^3-q)\cdot(q^2+q+1)/r = (q^2+q)\cdot(q-1)\cdot(q^2+q+1)/r = (q^2+q)\cdot(p^k-1)/r \bmod (p^k-1)$. □

The exponentiation by q and q^2 are performed by the Frobenius endomorphism in $\mathbb{F}_{p^k}$. Although there remains one multiplication by $v_{-TQ}(P)$ in $\mathbb{F}_{p^k}$, it is less expensive than the typical multiplication in $\mathbb{F}_{p^k}$ since $v_{-TQ}(P)$ gives a sparse element in $\mathbb{F}_{p^k}$.

As a result, the algorithm for computing the value of the proposed rational function is given in Algorithm C.1, where `ADD` and `SUB` steps are swapped from the previous computation given in Algorithm C.2. INIT$_+$ and INIT$_-$ steps are also swapped. Although the proposed algorithm requires an additional `ADJ` step, it is suitable for computing the

Algorithm C.2: An alternation of Miller's algorithm for computing the ate variant pairings.

```
Input: $T = t_l 2^l + \cdots + t_1 2^1 + t_0 2^0 \in \mathbb{Z}$ where $t_i \in \{-1, 0, 1\}$, $P \in \mathcal{G}_1$, and $Q \in \mathcal{G}_2$
Output: $(f_{-T,Q}(P) \cdot v_{-TQ}(P))^{q^2+q}$
If $t_l = 1$ then
  $f \leftarrow v_Q^{-1}(P), R \leftarrow -Q$;                                  //INIT$_-$
else if $t_l = -1$ then
  $f \leftarrow 1, R \leftarrow Q$;                                             //INIT$_+$
endif
For $i$ from $l-1$ downto $0$ do
  $f \leftarrow f^2 \cdot \frac{l_{R,R}(P)}{v_{R+R}(P)}, R \leftarrow R + R$;    //DBL
  If $t_i = 1$ then
    $f \leftarrow f \cdot \frac{l_{R,-Q}(P)}{v_{R-Q}(P)} \cdot v_Q^{-1}(P), R \leftarrow R-Q$;   //SUB
  else if $t_i = -1$ then
    $f \leftarrow f \cdot \frac{l_{R,Q}(P)}{v_{R+Q}(P)}, R \leftarrow R + Q$;    //ADD
  endif
endfor
$f \leftarrow (f \cdot v_R(P))^{q^2+q}$;                                        //ADJ
return $f$;
```

pairings on curves with T such that it does not contain $t_i = 1$ for $0 \leq i < n$, which opposite holds for the typical algorithm.

Appendix D

Algorithms for Computing Arithmetic Operations in $\mathbb{F}_{p^2}$

Let $A = (a_0, a_1)$ and $B = (b_0, b_1)$ be any elements in $\mathbb{F}_{p^2}$, where $a_0, a_1, b_0, b_1 \in \mathbb{F}_p$. Then, multiplication of A and B and squaring of A, i.e., $A \cdot B = (u_0, u_1)$ and $A^2 = (v_0, v_1)$ with $u_0, u_1, v_0, v_1 \in \mathbb{F}_p$, can be computed by using variable elements $t_1, t_2, t_3, t_4 \in \mathbb{F}_p$ as follows:

- `OEF_x2+1`

  Multiplication: $t_1 = a_0b_0, t_2 = a_1b_1, t_3 = a_0 + a_1, t_4 = b_0 + b_1, u_0 = t_1 - t_2, u_1 = t_3t_4, u_1 = u_1 - t_1, u_1 = u_1 - t_2$.

  Squaring: $t_1 = a_0 + a_1, t_2 = a_0 - a_1, v_1 = a_0a_1, v_1 = v_1 + v_1, v_0 = t_1t_2$.

- `OEF_x2-5`

  Multiplication: $t_1 = a_0 + a_1, t_2 = b_0 + b_1, t_1 = t_1t_2, t_2 = a_0b_0, t_3 = a_1b_1, t_2 = t_2 + t_3, u_1 = t_1 - t_2, t_3 = 4t_3, u_0 = t_2 + t_3$.

  Squaring: $t_1 = a_0 + a_1, t_2 = 4a_1, t_2 = t_2 + a_1, t_2 = a_0 + t_2, t_1 = t_1t_2, t_2 = a_0a_1, v_1 = t_2 + t_2, t_3 = 4v_1, t_3 = t_3 - v_1, v_0 = t_1 - t_3$.

- `AOPF_x2+x+1`:

  Multiplication: $t_1 = a_0 - a_1, t_2 = b_0 - b_1, t_1 = t_1t_2, t_2 = a_0b_0, t_3 = a_1b_1, u_0 = t_1 - t_2, u_1 = t_1 - t_3$.

  Squaring: $t_1 = a_0 + a_0, t_1 = a_1 - t_1, t_2 = a_1 + a_1, t_2 = a_0 - t_2, v_0 = t_1a_1, v_1 = t_2a_0$.

- `EFN_x2-x+1`

  Multiplication: $t_1 = a_0 - a_1, t_2 = b_0 - b_1, t_1 = t_1t_2, t_2 = a_0b_0, t_3 = a_1b_1, u_0 = t_2 - t_1, u_1 = t_3 - t_1$.

  Squaring: $t_1 = a_0 + a_0, t_1 = t_1 - a_1, t_2 = a_1 + a_1, t_2 = t_2 - a_0, v_0 = t_1a_1, v_1 = t_2a_0$,

- `EFN_x2-x-1`

  Multiplication: $t_1 = a_0 - a_1, t_2 = b_0 - b_1, t_1 = t_1 t_2, t_2 = a_0 b_0, t_3 = a_1 b_1, u_0 = t_1 + t_2, u_1 = t_1 + t_3$.

  Squaring: $t_1 = a_0 - a_1, t_1 = t_1^2, t_2 = a_0^2, t_3 = a_1^2, v_0 = t_1 + t_2, v_1 = t_1 + t_3, v_1 = t_1 + t_3$.

Appendix E

Conversion of Public Parameters of SIDH

The author applies the proposed isomorphism and obtains the public parameter set for SIDH with $\mathbb{F}_{p^2}$ defined by $\mathbb{F}_p[x]/(x^2+x+1) \cong \mathbb{F}_p(\beta)$ where β is an element in $\mathbb{F}_{p^2}$ such that $\beta^2 = -\beta - 1$. It is adopted the existing public parameter set `SIKEp434` defined over $\mathbb{F}_p[x]/(x^2+1) \cong \mathbb{F}_p(\alpha)$ where α is an element in $\mathbb{F}_{p^2}$ such that $\alpha^2 = -1$, which consists of the following materials (see Chapter 1.6.1 in [Cam+19]):

- $p = 2^{216}3^{137} - 1$, i.e., $l_A = 2$, $l_B = 3$, $e_A = 216$, and $e_B = 137$;
- $E_0 : y^2 = x^3 + Ax^2 + x$ where $A = 6 + 0\alpha \in \mathbb{F}_p(\alpha)$;
- x-coordinates of the initial points P_A, Q_A, $R_A = P_A - Q_A \in E[l_A^{e_A}]$, P_B, Q_B, $R_B = P_B - Q_B \in E[l_B^{e_B}]$, which are elements in $\mathbb{F}_p(\alpha)$. Note that since we work on the Montgomery curves without y-coordinates, $x(R_A)$ and $x(R_B)$ are required.

The author constructs an isomorphic map $M : \mathbb{F}_p(\alpha) \to \mathbb{F}_p(\beta)$ and derives a public parameter set of SIDH defined over $\mathbb{F}_p(\omega)$ by computing $M(A)$, $M(x(P_A))$, $M(x(Q_A))$, $M(x(R_A))$, $M(x(P_B))$, $M(x(Q_B))$, and $M(x(R_B))$.

According to Theorem 5.6, the isomorphism map $\mathbb{F}_p(\alpha) \to \mathbb{F}_p(\beta)$ is obtained by $M : \mathbb{F}_p(\alpha) \to \mathbb{F}_p(\beta), x_0 + x_1\alpha \mapsto (x_0 + mx_1) + nx_1\beta = (-x_0 + (n-m)x_1)\beta - (x_0 + mx_1)\beta^2$. where m and n are given as follows:

$$m = \texttt{00db6794 b8c6558d e8372711 9cd51000 00000000 00000000 00000000},$$

$$n = \texttt{01b6cf29 718cab1b d06e4e23 39aa2000 00000000 00000000 00000000}.$$

Applying M, a curve coefficient $A \in \mathbb{F}_p(\alpha)$ is mapped to $M(A) = -6\beta - 6\beta^2 \in \mathbb{F}_p(\beta)$. For $S \in \{P, Q, R\}$ and $X \in \{A, B\}$, the x-coordinates of initial point $x(S_X)$ in $\mathbb{F}_p(\alpha)$ can be mapped to an element in $\mathbb{F}_p(\beta)$ by computing $M(x(S_X)) = x(S'_X) = x_{S'_X,0}\beta + x_{S'_X,1}\beta^2$

where $x_{S'_X,0}$ and $x_{S'_X,1}$ are elements consists of as follows:

$$\begin{aligned}
x_{P'_A,0} =& \texttt{0001b7ec 3cb83805 31034815 ffcce3b5 40693f5a fb9bbd81 80395c7b}\\
& \texttt{9cfbb4fb 30ad5bdd 3cba824f 73f213fe e7125ecc 8be39afc 2fcf4c60},\\
x_{P'_A,1} =& \texttt{00000293 5e9b5a9f 35f24ff3 5de41dac a2843950 b9f07d05 b49cbb3b}\\
& \texttt{12d96a45 d64a0409 5dceb9dd ea4aaeaa 0c29fc7a df7a8ab4 a3a31d0f},\\
x_{Q'_A,0} =& \texttt{0001bcec 6753b4d5 c8dd8561 a57eeca8 cc29930f a7b9a009 d83cb9b5}\\
& \texttt{a109001f 13c48a6a 2f9ff3c3 c6f7de48 67ad08b5 e671097a 225bc897},\\
x_{Q'_A,1} =& \texttt{00011cc5 b86ac995 173a0084 4c1e862d b9733e81 129c3bd1 59924a7c}\\
& \texttt{3ec1ba05 5ed21eb2 55da228c b8565f38 ceee876b 1dd4a10d c1ce1e8f},\\
x_{R'_A,0} =& \texttt{0000b936 ddd16a1e 503f960c 9c71a2fc 210958e0 306a79c0 573cb62c}\\
& \texttt{c04a31b8 462b666b acf65cb4 ccc79553 2d9ad510 582b7a6f 55726594},\\
x_{R'_A,1} =& \texttt{0001c812 09c63acd 2e8f4126 ae76e1a3 7c4fd316 6921dcf9 d3f29fa4}\\
& \texttt{559a7dac c167f8c0 08dcd073 b6c29408 5cb6fc9a cd8d5b69 1e93503e}.\\
x_{P'_B,0} =& \texttt{0001adba a0b8cb6c 560c24a4 9fa15de9 3b5c300b 6094d83c b7611fcf}\\
& \texttt{faa76a13 c8c97403 ff620503 4c26819c 609a161b a0b9a8c4 f9c84856},\\
x_{P'_B,1} =& \texttt{0001adba a0b8cb6c 560c24a4 9fa15de9 3b5c300b 6094d83c b7611fcf}\\
& \texttt{faa76a13 c8c97403 ff620503 4c26819c 609a161b a0b9a8c4 f9c84856}.\\
x_{Q'_B,0} =& \texttt{0001059a 4fb24deb 8667a051 bfc945a6 e20e2135 ca957fdd a2b130ff}\\
& \texttt{1806b39c 14f9c97e 174e18c6 73f4dbe3 e64699a0 2461ebf9 25c2c7b9},\\
x_{Q'_B,1} =& \texttt{0001059a 4fb24deb 8667a051 bfc945a6 e20e2135 ca957fdd a2b130ff}\\
& \texttt{1806b39c 14f9c97e 174e18c6 73f4dbe3 e64699a0 2461ebf9 25c2c7b9}.\\
x_{R'_B,0} =& \texttt{00004a01 53e81db2 b207c2d4 9cc9c890 c660622d 7785390f 637fa6d6}\\
& \texttt{f44e6787 266dbc35 100f2130 c5c6f60b 3351c140 4ce94455 a3517d60},\\
x_{R'_B,1} =& \texttt{000083ec 47621b2c 28213cd2 95cf9731 dc0d41f9 a79332cd 53df0535}\\
& \texttt{e132f50e ddc026b7 66d32c9a 1ba4f05d 732eeed5 7e031f07 480913c6}.
\end{aligned}$$

Appendix F

Construction of Supersingular Elliptic Curves of Order $(p-1)^2$

Let E be a supersingular elliptic curve of which order is $\#E(\mathbb{F}_{p^2}) = (p-1)^2$. According to [Wat69], a twist of E, which is denoted as $E'/\mathbb{F}_p : y^2 = x^3 + ax + b$, has an order $\#E'(\mathbb{F}_{p^2}) = (p+1)^2$. For $p = p_{441+} = 2^{216}3^{137}139 + 1$, the coefficients of E' are easily found by using ecgen library [Jan18].

$$\begin{aligned}
a = {} & \texttt{00627426 b720ddfa 4e7970c2 25f07717 f583111e 9cba318c 9bba7fcd} \\
& \texttt{d4e49249 24924924 92492492 49249249 24924924 92492492 49249245}, \\
b = {} & \texttt{00cfd8c3 829ab82c de8e98b6 501817dd 3f312424 2e6ca17e 2c50d4eb} \\
& \texttt{6c1b6db6 db6db6db 6db6db6d b6db6db6 db6db6db 6db6db6d b6db6dbc}.
\end{aligned}$$

Since E is a quadratic twist of E' defined over $\mathbb{F}_{p^2}$, the curve E is obtained as $E/\mathbb{F}_{p^2} : y^2 = x^3 + \delta^{2/3}ax + \delta b$ where δ is quadratic non-residue and cubic residue in $\mathbb{F}_{p^2}$. One can convert the elliptic curve of the Weierstrass to the Montgomery form. As a result, the Montgomery curve of $\#E(\mathbb{F}_{p^2}) = (p-1)^2$ is obtained.