Efficient Software Implementation of Pairing-Based Cryptographic Primitives for High-level Security for IoT

March, 2019

Md. Al-Amin KHANDAKER

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

OKAYAMA UNIVERSITY

DOCTORAL THESIS

Efficient Software Implementation of Pairing-Based Cryptographic Primitives for High-level Security for IoT

Author:
Md. Al-Amin KHANDAKER

Supervisor:
Yasuyuki NOGAMI

Co-supervisors:
Nobuo FUNABIKI
Satoshi DENNO

A dissertation submitted to
OKAYAMA UNIVERSITY
in fulfillment of the requirements for the degree of
Doctor of Philosophy in Engineering
in the
Faculty of Engineering
Graduate School of Natural Science and Technology

December 10, 2018

TO WHOM IT MAY CONCERN

We hereby certify that this is a typical copy of the original Doctoral dissertation of

Md. Al-Amin KHANDAKER

Thesis Title:

Efficient Software Implementation of Pairing-Based Cryptographic Primitives for High-level Security for IoT

Seal of Supervisor *Official Seal*

Professor Yasuyuki NOGAMI

Graduate School of Natural Science and Technology

Declaration of Authorship

This dissertation and the work presented here for doctoral studies were conducted under the supervision of Professor Yasuyuki Nogami. I, Md. Al-Amin KHANDAKER, declare that this thesis titled, "Efficient Software Implementation of Pairing-Based Cryptographic Primitives for High-level Security for IoT" and the work presented in it are my own. I confirm that:

- The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, while enrolled in the Faculty of Engineering at Okayama University as a candidate for the degree of Doctor of Philosophy in Engineering.
- This work has not been submitted for a degree or any other qualification at this University or any other institution.
- Some of the previously published works presented in this dissertation listed in "Research Activities".
- The published work of others cited in this thesis is clearly attributed. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help to pursue this work.
- My coauthors contribution is acknowledged in all works.
- The experiments and results presented in this thesis and in the articles where I am the first author were conducted by the myself.

Signed: Md. Al-Amin KHANDAKER Student number: 51427351

Date: December 10, 2018

"If we knew what it was we were doing, it would not be called research, would it? "

Albert Einstein

Abstract

Md. Al-Amin KHANDAKER

*Efficient Software Implementation of Pairing-Based
Cryptographic Primitives for High-level Security for IoT*

Pairing-based cryptography over the elliptic curves is a relative new paradigm in public key cryptography(PKC). In general, pairing calculation involves certain elliptic curve named pairing-friendly curve defined over finite extension of prime field. It is typically defined as bilinear map from rational points of two additive groups to a multiplicative group. Two mathematical tool named as Miller's algorithm and final exponentiation is mostly involved in pairing calculation. However, most protocols also requires two more operation in pairing groups named scalar multiplication and exponentiation in multiplicative group. The above mentioned mathematical tools are the major bottleneck for the efficiency of pairing-based protocols.

Since, the inception at the advent of this century pairing-based cryptography brings monumental amount of research. The results of this vast amount of research brought some novel cryptographic application which was not possible before pairing-based cryptography. However, computation speed of pairing was very slow to consider them as a practical option. Years of research from the mathematicians, cryptographers and computer scientists improves the efficiency of pairing.

The security of pairing-based cryptography is not only rely on the intractability of elliptic curve discrete logarithm problem (ECDLP) of additive elliptic curve group but also discrete logarithm problem (DLP) on multiplicative group. It is known that key size in cryptography based of ECDLP requires fewer bits than cryptography based on DLP. Therefore, it is a crucial to maintain a balance in parameter sizes for both additive and multiplicative groups in pairing-based cryptography. In CRYPTO 2016, Kim and Barbulescu showed a more efficient version of number field sieve algorithm to solve DLP. This new attack makes all previous parameter settings to update.

This thesis presents several improvement technics for pairing-based cryptography over two ordinary pairing-friendly curves named KSS-16 and KSS-18. The motivation behind to work on these curves is, they not widely studied in literature compared to other pairing-friendly curves. After the extNFS algorithm, the security level of widely used pairing-friendly curves were challenged. The technics can also be applied on the ordinary pairing-friendly curves. We also present several improvements in extension field arithmetic operation. We implement the proposed improvements in for experimental purpose. All the sources are bundled in an installable library.

Acknowledgements

The last 3 and a half year was one of the best time of my life that I will cherish forever. I'm immensely blessed throughout this period for which I have many people to thank. I'm grateful to many people who have directly and indirectly helped me finish this work.

This work would not be possible without the unceasing supervision, innumerable counselling and unrelenting persuasion of my Ph.D. advisor Professor Yasuyuki Nogami. I am indebted to *Nogami Sensei* for having me in his lab (*Information Security Lab.*) as a doctoral student and mentoring me on this work. He taught me how to analyze complex problems from different perspectives and express the ideas from pen and papers to a fully publishable article. I enjoyed his insightful comments on the research topics during our discussions. Sometimes his in-depth queries bewildered me and influenced my ideas in this thesis. He guided me in different ways to approach a problem and the need to be persistent to accomplish my goal. His presence and off-work discussion make the lab more than a workplace.

I'm also very grateful for to my doctoral course co-supervisors Professor Nobuo Funabiki (*Distributed Systems Design Lab.*) and Professor Satoshi Denno (*Multimedia Radio Systems Lab.*) for having their time to read my thesis draft. Their insightful comments and helpful advice helped to shape the thesis into this state. I must recall my experience of taking the "Theory of Distributed Algorithm" course taught by Professor Nobuo Funabiki. His strong passion for algorithmic problem solving during the lectures was not only inspiring but also contagious.

I reminisce my encounters with Professor Satoshi Denno during my days at *Secure Wireless System lab.* He provided me with the deep-seated idea of the research works and Japan life. His questions and suggestions for the time of half yearly progress meetings were very intuitive.

I am very grateful to Associate Professor Nobumoto Yamane (*Information Transmission Lab.*) for provided important comments at progress meetings.

I would like to express my gratitude to Senior Assistant Professor Takuya Kusaka (*Information Security Lab.*) for the in-depth discussion of scientific topics. His strong work ethic and passion for research helped us to publish some of the remarkable collaborative works. He was always there to help while any difficulty arose from attending a conference to publishing a paper.

I express my gratitude to Senior Assistant Professor Hiroto Kagotani of (*Information System Design Lab.*) for employing me as a research assistant for a quarter. Since the Information System Design Lab and Information Security Lab. share space, we had encountered more often and share of research discussions. His comments during the progress report were enlightening.

I am also grateful to Assistant Professor Kengo Iokibe (*Optical and Electromagnetic Waves Lab.*) for the collaborative work we had on side-channel analysis of raspberry pi.

I would like to express my deep gratitude of Professor Sylvain Duquesne of Univ Rennes, France for having me at IRMAR as a short-term researcher and allowing me to present my work in front of some brightest audiences. Professor Duquesne's in-depth reviews on my works were not only helpful towards to final acceptability but also intriguing.

My sincere gratitude to post-doctoral fellow Dr. Loubna Ghammam at Normandie University, France for her persistent guidance. Our collaboration with Professor Duquesne and Dr. Loubna helps me to work on the diverse area of mathematical aspects of cryptography.

I am also thankful to Professor Howon Kim of Pusan National University, South Korea and his Ph.D. student Taehwan Park for a great research collaboration on IoT security.

My gratitude to one of the great IoT security expert Professor Hwajeong Seo of Hansung University, South Korea for being a co-author in my first major conference paper.

Thanks to MEXT, Japan for the scholarship which fulfilled my dream to pursue the doctoral study in Japan. I sincerely acknowledge all the funds that afforded me to join several international conferences and conduct research activities.

I am also grateful to all administrative officer of the Faculty of Engineering who directly or indirectly made an impact in my doctoral course studies. My special thanks to Ms. Yumiko Kurooka for her kind support in administrative works.

Special thanks also to my seniors, juniors, and friends in the laboratory for creating a great work atmosphere and their generous support. Thanks to pairing team members of my lab who are one of brightest minds I've worked with.

I can not thank enough to my wife for her sacrifices and generous supports to my bread and butter. I would like to take the opportunity to appreciate my parents Ms. Nasima Akter and Mr. Md. Ali-Azzam Khandaker for their understanding, and encouragements.

So far so general we all are standing on the shoulders of the giants for our works. My profound gratitude to all great cryptographer, cryptographic engineers and researchers whose works keep inspiring students like me. I'm indebted to all my research collaborator, co-authors and reviewers for making my doctoral voyage engaging.

Contents

List of Figures

List of Tables

List of Notations and Symbols

Notation	Description
p	$p > 3$ is an odd prime integer in this thesis.
$x \bmod p$	Modulo operation. the least nonnegative residue of x modulo p.
$\mathbb{F}_p$	Prime field. The field of integers mod p.
$\mathbb{F}_p^*$	The multiplicative group of the field $\mathbb{F}_p$. In other words, $\mathbb{F}_p^* = \{x \mid x \in \mathbb{F}_p \text{ and } x \neq 0\}$.
$\lfloor \cdot \rfloor$	The floor of $\cdot$ is the greatest integer less than or equal to $\cdot$. For example, $\lfloor 1 \rfloor = 1$ and $\lfloor 6.3 \rfloor = 6$.

Dedicated to the people I owe most. To my parents who brought me to this world and to my wife who sacrificed the most during my Ph.D. journey.

Research Activities

Peer-Reviewed Journal Papers (First author)

1. **Md. Al-Amin Khandaker** and Yasuyuki Nogami. “An Improvement of Scalar Multiplication by Skew Frobenius Map with Multi-Scalar Multiplication for KSS Curve”. In: *IEICE Transactions* 100-A.9 (2017), pp. 1838-1845. DOI: 10.1587/transfun.E100.A.1838.
2. **Md. Al-Amin Khandaker**, Taehwan Park, Yasuyuki Nogami, and Howon Kim. “A Comparative Study of Twist Property in KSS Curves of Embedding Degree 16 and 18 from the Implementation Perspective”. In: *J. Inform. and Commun. Convergence Engineering* 15.2 (2017), pp. 97-103. DOI: 10.6109/jicce.2017.15.2.97.

Peer-Reviewed International Conference Papers (First author)

LNCS Proceedings:

3. **Md. Al-Amin Khandaker**, Yuki Nanjo, Loubna Ghammam, Sylvain Duquesne, Yasuyuki Nogami, and Yuta Kodera. “Efficient Optimal Ate Pairing at 128-Bit Security Level”. In: *INDOCRYPT 2017*. Ed. by Arpita Patra and Nigel P. Smart. Vol. 10698. LNCS. Springer, Heidelberg, Dec. 2017, pp. 186–205. DOI: 10.1007/978-3-319-71667-1_10.
4. **Md. Al-Amin Khandaker**, Hirotaka Ono, Yasuyuki Nogami, Masaaki Shirase, and Sylvain Duquesne. “An Improvement of Optimal Ate Pairing on KSS Curve with Pseudo 12-Sparse Multiplication". In: *ICISC 2016*. Ed. by Seokhie Hong and Jong Hwan Park. Vol. 10157. LNCS. Springer, Heidelberg, 2017, pp. 208–219. DOI: 10.1007/978-3-319-53177-9_11.
5. **Md. Al-Amin Khandaker**, Yasuyuki Nogami, Hwajeong Seo, and Sylvain Duquesne. “Efficient Scalar Multiplication for Ate Based Pairing over KSS Curve of Embedding Degree 18”. In: *WISA 2016*. Ed. by Dooho Choi and Sylvain Guilley. Vol. 10144. LNCS. Springer, Heidelberg, Aug. 2016, pp. 221–232. DOI: 10.1007/978-3-319-56549-1_19.

IEEE Xplore indexed:

6. **Md. Al-Amin Khandaker**, Yuki Nanjo, Takuya Kusaka, and Yasuyuki Nogami. "A Comparative Implementation of GLV Technique on KSS-16 Curve." In: *Sixth International Symposium on Computing and Networking, CANDAR 2018*, Gifu, Japan, November 27-30, 2016. 2018, pp. ?–?. DOI: ?. (Acceptance Ratio 28/77 ≈ 36%)

7. **Md. Al-Amin Khandaker** and Yasuyuki Nogami. "Isomorphic Mapping for Ate-Based Pairing over KSS Curve of Embedding Degree 18". In: *Fourth International Symposium on Computing and Networking, CANDAR 2016*, Hiroshima, Japan, November 22-25, 2016. 2016, pp. 629–634. DOI: `10. 1109/CANDAR.2016.`
`0113`.

8. **Md. Al-Amin Khandaker** and Yasuyuki Nogami. "A consideration of towering scheme for efficient arithmetic operation over extension field of degree 18". In: *19th International Conference on Computer and Information Technology (ICCIT) 2016*. Dec. 2016, pp. 276–281. DOI: `10.1109/ICCITECHN.2016. 7860209.:`.

9. **Md. Al-Amin Khandaker** and Yasuyuki Nogami. "An improvement of scalar multiplication on elliptic curve defined over extension field Fq2". In: *IEEE International Conference on Consumer Electronics-Taiwan (ICCE- TW). 2016*, Nantou, Taiwan, May 27-29, 2016. 2016, pp. 1–2. DOI: `10.1109/ICCE-TW.2016.`
`7520894`.

IEICE/IEIE sponsored:

10. **Md. Al-Amin Khandaker** and Yasuyuki Nogami. "Frobenius Map and Skew Frobenius Map for Ate-based Pairing over KSS Curve of Embedding Degree 16 ". In: *International Technical Conference on Circuits / Systems, Computers and Communications (ITC-CSCC), 2017*, Busan, Korea, Jul. 2-5, 2017. IEIE.

Peer-Reviewed Journal Papers (Co-author)

11. Yuki Nanjo, **Md. Al-Amin Khandaker**, Takuya Kusaka, and Yasuyuki Nogami. "Efficient Pairing-Based Cryptography on Raspberry Pi". In: *Journal of Communications (JCM)* 13.2 (2018), pp. 88–93. DOI: `10.12720/jcm.13.2.88-93`.

12. Yuta Hashimoto, **Md. Al-Amin Khandaker**, Yuta Kodera, Taehwan Park, Takuya Kusaka, Howon Kim, and Yasuyuki Nogami. "An Implementation of ECC with Twisted Montgomery Curve over 32nd Degree Tower Field on Arduino Uno". In: *International Journal of Networking and Computing (IJNC)* 8.2 (2018), pp. 341–350. DOI: `10.15803/ijnc.8.2_341`.

13. Yuta Kodera, Takeru Miyazaki, **Md. Al-Amin Khandaker**, Ali Md. Arshad, Takuya Kusaka, Yasuyuki Nogami, and Satoshi Uehara. "Distribution of Digit Patterns in Multi-Value Sequence over the Odd Characteristic Field". In: *IEICE Transactions* 101-A.9 (2018), pp. 1525–1536. DOI: `10.1587/transfun.E101.A.1525`.

14. Shunsuke Ueda, Ken Ikuta, Takuya Kusaka, **Md. Al-Amin Khandaker**, Ali Md. Arshad, and Yasuyuki Nogami. "An Extended Generalized Minimum Distance Decoding for Binary Linear Codes on a 4-Level Quantization over an AWGN Channel". In: *IEICE Transactions* 101-A.8 (2018), pp. 1235–1244. DOI: `10.1587/transfun.E101.A.1235`.

15. Shoma Kajitani, Yasuyuki Nogami, Shunsuke Miyoshi, Thomas Austin, **Md. Al-Amin Khandaker**, Nasima Begum, and Sylvain Duquesne. "Web-based Volunteer Computing for Solving the Elliptic Curve Discrete Logarithm Problem". In: *International Journal of Networking and Computing (IJNC)* 6.2 (2016), pp. 181–194. DOI: `10.15803/ijnc.6.2_181`.

Peer-Reviewed International Conference Papers (Co-author)

LNCS Proceedings:

16. Yuki Nanjo, **Md. Al-Amin Khandaker**, Masaaki Shirase, Takuya Kusaka, and Yasuyuki Nogami. "Efficient Ate-Based Pairing over the Attractive Classes of BN Curves". In: *WISA 2018*. To appear LNCS. Springer, Heidelberg, Aug. 2018. pp. ?–?. DOI: ?. (Acceptance Ratio 22/44 = 50%)

17. Takuya Kusaka, Sho Joichi, Ken Ikuta, **Md. Al-Amin Khandaker**, Yasuyuki Nogami, Satoshi Uehara, Nariyoshi Yamai, and Sylvain Duquesne. "Solving 114-Bit ECDLP for a Barreto-Naehrig Curve". In: *ICISC 2017*. Ed. by Howon Kim and Dong-Chan Kim. Vol. 10779. LNCS. Springer, Heidelberg, Oct. 2017, pp. 231–244. DOI: `10.1007/978-3-319-78556-1_13`.

18. Taehwan Park, Hwajeong Seo, Garam Lee, **Md. Al-Amin Khandaker**, Yasuyuki Nogami, and Howon Kim. "Parallel Implementations of SIMON and SPECK, Revisited". In: *WISA 2017*. Ed. by Brent ByungHoon Kang and Taesoo Kim. Vol. 10763. LNCS. Springer, Heidelberg, Aug. 2017, pp. 283–294. DOI: `10.1007/978-3-319-93563-8_24`.

IEEE Xplore indexed:

19. Yuki Nanjo, **Md. Al-Amin Khandaker**, Takuya Kusaka, and Yasuyuki Nogami. "Consideration of Efficient Pairing Applying Two Construction Methods of Extension Fields." In: *Sixth International Symposium*

on Computing and Networking, CANDAR 2018, Gifu, Japan, November 27-30, 2016. 2018, pp. ?–?. DOI: ?.

20. Yuta Hashimoto, **Md. Al-Amin Khandaker**, Yuta Kodera, Taehwan Park, Takuya Kusaka, Howon Kim, and Yasuyuki Nogami. "An ECC Implementation with a Twisted Montgomery Curve over Fq32 on an 8-Bit Microcontroller". In: *Fifth International Symposium on Computing and Networking, CANDAR 2017*, Aomori, Japan, November 19-22, 2017. 2017, pp. 445–450. DOI: `10.1109/CANDAR.2017.90`.

21. Yuta Kodera, Takuya Kusaka, Takeru Miyazaki, **Md. Al-Amin Khandaker**, Yasuyuki Nogami, and Satoshi Uehara. "An Efficient Implementation of Trace Calculation over Finite Field for a Pseudorandom Sequence". In: *Fifth International Symposium on Computing and Networking, CANDAR 2017*, Aomori, Japan, November 19-22, 2017. 2017, pp. 451–455. DOI: `10.1109/CANDAR.2017.86`.

22. Taehwan Park, Hwajeong Seo, **Md. Al-Amin Khandaker**, Yasuvuki Nogami, and Howon Kim. "Efficient Parallel Simeck Encryption with GPGPU and OpenCL". In: *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). 2018*, Taichung, Taiwan, May 19-21, 2018. 2018, pp. 1–2. DOI: `10.1109/ICCE-China.2018.8448768`.

23. Yuta Kodera, Takeru Miyazaki, **Md. Al-Amin Khandaker**, Ali Md Arshad, Yasuyuki Nogami, and Satoshi Uehara. "Distribution of bit patterns on multi-value sequence over odd characteristics field". In: *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). 2017*, Taipei, Taiwan, June 12-14, 2017. 2017, pp. 137-138. DOI: `10.1109/ICCE-China.2017.7991033`.

24. Akihiro Sanada, Yasuyuki Nogami, Kengo Iokibe, **Md. Al-Amin Khandaker**. "Security analysis of Raspberry Pi against Side-channel attack with RSA cryptography."In: *IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). 2017*, Taipei, Taiwan, June 12-14, 2017. 2017, pp. 287 - 288. DOI: `10.1109/ICCE-China.2017.7991108`.

IEICE/IEIE sponsored:

25. Ken Ikuta, Sho Joichi, Kazuya Kobayashi, **Md. Al-Amin Khandaker**, Takuya Kusaka, and Yasuyuki Nogami. "A Study on the Parameter Size of the Montgomery Trick for ECDLP". In: *International Symposium on Information Theory and its Applications (ISITA), 2018*. IEICE. (To appear in IEEE Xplore).

26. Ken Ikuta, Sho Joichi, Kazuya Kobayashi, **Md. Al-Amin Khandaker**, Takuya Kusaka, and Yasuyuki Nogami. "A Study on the Parameter of the Distinguished Point Method in Pollard's Rho Method for ECDLP". In: *International Symposium on Information Theory and its Applications (ISITA), 2018*. IEICE. (To appear in IEEE Xplore).

27. Ken Ikuta, Takuya Kusaka, **Md. Al-Amin Khandaker**, Yasuyuki Nogami, and Thomas H. Austin. "Estimation of computational complexity of Pollard's rho method based attack for solving ECDLP over Barreto-Naehrig curves". In: *32nd International Technical Conference on Circuits / Systems, Computers and Communications (ITC-CSCC), 2017*. IEIE.

Domestic conferences (First author)

28. **Md. Al-Amin Khandaker**, Hirotaka Ono, Yuki Nanjo, Takuya Kusaka and Yasuyuki Nogami. "Efficient Optimal-Ate Pairing on BLS-12 Curve Using Pseudo 8-Sparse Multiplication". In: *Computer Security Symposium (CSS), 2017*, CD-ROM (3E1-4).

29. **Md. Al-Amin Khandaker** and Yasuyuki Nogami. "Efficient Scalar Multiplication by Skew Frobenius Map with Multi-Scalar Multiplication for KSS Curve". In: *Symposium on Cryptography and Information Security (SCIS), 2017*, CD-ROM (B1-3).

Domestic conferences (Co-author)

30. Yuki Nanjo, **Md. Al-Amin Khandaker**, Takuya Kusaka, Yasuyuki Nogami. "A Study on a Construction Method of Degree 18 Extension Field for Efficient Pairing over KSS Curves". In: *Computer Security Symposium (CSS), 2018*, CD-ROM (??).

31. Hirotaka Ono, **Md. Al-Amin Khandaker**, Yuki Nanjo, Toshifumi Matsumoto, Takuya Kusaka and Yasuyuki Nogami. "An Implementation and Evaluation of ID-based Authentication on Raspberry Pi with Pairing Library". In: *Symposium on Cryptography and Information Security (SCIS), 2018*, CD-ROM (4D2-1).

32. Yuki Nanjo, **Md. Al-Amin Khandaker**, Yuta Kodera and Yasuyuki Nogami. "Implementation method of the pairing over BN curve using two type of extension fields". In: *Symposium on Cryptography and Information Security (SCIS), 2018*, CD-ROM (4D2-3).

33. Yuki Nanjo, **Md. Al-Amin Khandaker**, Takuya Kusaka, and Yasuyuki Nogami. "The relation between the efficient sextic twist and constant of the modular polynomial for BN curve". In: *Computer Security Symposium (CSS), 2017*, CD-ROM (3E1-3).

34. Norito Jitsui, Yuki Nanjo, **Md. Al-Amin Khandaker**, Takuya Kusaka and Yasuyuki Nogami. "Efficient Elliptic Scalar Multiplication for Pairing-Based Cryptography over BLS48". In: *Symposium on Cryptography and Information Security (SCIS), 2018*, CD-ROM (3B4-1).

Chapter 1

Introduction

This chapter introduces the related literature review, motivation and goals of the undertaken research. The chapter begins with a brief preface of cryptology and its importance in the era Internet of Things (IoT) and Big Data. In **Section**. 1.1.2 we present how Public-Key Cryptography (PKC) is shaping the security of our every day life. We introduce the importance of Pairing-Based Cryptography(PBC) in **Section**. 1.1.3 for the next generation of security protocols. **Section**. 1.2 presents the motivation behind the works undertaken to assemble of this thesis. **Section**. 1.4 outlines the overall organization of this thesis.

1.1 Cryptology

Cryptography is the science of communicating with the authentic receiver through an insecure channel in secret. Cryptanalysis is the techniques of breaking the secret communications. Cryptology is the combination of these two domains.

The history of Cryptography dates back to the time of the Greek and Roman empire. Julius Caesar used a simple shift and substitute system. Up until the early '70s of the last century, cryptology was evolved mostly for military purposes. The cryptography got its first democratic form in 1975 when Diffie and Hellman invented the concept of public-key cryptography [DH76]. The concept was first realized by as practical cryptosystem by the works of Rivest, Shamir and Adleman (RSA) in 1977 [RSA78]. At the same time in 1977, National Bureau of Standards published a cryptosystem intended for the governmental agencies or banks with the named Data Encryption Standard (DES). From then a new era of cryptography known as *Modern cryptography* was initiated. The well-organized procedures called *protocols* is the basis of Modern cryptography. One of the most elegant features of modern crypto-protocols is their inner algorithms are not secret yet withstand cryptanalysis from experts/attackers. More importantly, these protocols are easy to use for people with no understanding of the underlying principles. For example, paying by credit cards or withdrawing money using debit cards with a personal identification number (PIN) is doable without concerning what's going on under the hood.

The little basic functionality of modern cryptosystem is to enable a sender (Alice [1]) to convert a message (plaintext) into a cipher (ciphertext) before sending to a legitimate receiver (Bob) over the public communication media. The receiver can convert the cipher back into the original message using secret information named as a key. An adversary (Eve) eavesdrops in the middle of the conversation to retrieve information from the cipher. The cipher is safe from to the adversary until the key is not compromised.

The security of modern cryptosystems depend not on the secrecy of the encryption algorithms but on the difficulty of one way problems. Such problems are easy to calculate in one direction but practically impossible to calculate in reverse direction in a reasonable amount of time using reasonable resources. For example, let us consider a cipher text C and a plaintext $\mathcal{P}$ and a 128-bit key $\mathcal{K}$. The encryption scheme $\mathcal{E}$ takes input $\mathcal{P}$ and $\mathcal{K}$ and output $C = \mathcal{E}(\mathcal{P}, \mathcal{K})$. To obtain the key $\mathcal{K}$ from the $(\mathcal{P}, C)$ pair, we need to try $2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456 \approx 3.4 \times 10^{38}$ (39 decimal digits) combination of 128-bit keys. The most powerful supercomputer till this date can compute 122.3 peta (10^{15}) floating-point operations per second (PFLOPS). Let us consider an optimistic assumption that 1000 (FLOPS) is required to check one key combination. Under this assumption, the supercomputer can compute $122.3. \times 10^{15}/1000 = 122.3 \times 10^{12}$ key combinations per second. Then it will take about $3.4 \times 10^{38}/((122.3 \times 10^{12})(365 \times 24 \times 60 \times 60)) \approx 8.8 \times 10^{16}$ earth years. According to the standard model of physical cosmology [Ade+16] the age of our universe is 13.8^9 or 13.8 billion years. It means finding a key using brute force search will require 6.3 million years more than the age of the universe. We can imagine how big the number 2^{128} is from this comparison.

Cryptography became more important as individuals and business increasingly depend on the Internet as a channel for communication. Therefore, the following four properties are the basis of a cryptosystem.

- Data confidentiality: This property ensures that confidential information such as bank transactions or medical data etc. are secret from unauthorized entities.
- Data integrity: When data is stored, this property ensures that it not only kept secret (Data confidentiality) but also not rigged. Confidentiality and integrity is enforced by encryption.
- Authentication: In a connection-oriented communication authentication proves both parties identity before communication begins. Digital signature is used for this purpose to sign a message electronically. It shields the legitimate party against masquerader from impersonating as a trusted party. This property gives the receiver a confidence to believe that the message sent over the insecure channel is indeed sent by the actual signee.

[1] Alice and Bob are fictional characters first used by Rivest, Shamir and Adleman in [RSA78] as placeholder name in cryptology.

- Non-repudiation: Non-repudiation (with proof of origin and with proof of receipt) ensures that sender and receiver can not deny having taken part in a communication. This is important for many cases especially e-commerce while communicating over the Internet.

The modern crypto-protocols fall in following two major categories.

1.1.1 Symmetric/Private-Key Cryptography

Private-Key Cryptography, also known Symmetric Cryptography is the technique where both the sender and the receiver use same *key* or easily derivable from one another to encrypt and decrypt a message. This type of cryptography is very old history.

Modern cryptosystems offer efficient symmetric cryptography algorithms, e.g Advanced Encryption Standard (AES) [DR02]. Such cryptography has two main obstacles i.e. *Key management* and *key establishment*. Since the keys are same, they need to kept private (*Key management*) in both ends and should be shared securely beforehand (*Key establishment*) without physically meeting.

The Public-key Cryptography offers the solution for *Key establishment* applying Diffie-Hellman key exchange. This work primarily focuses on a certain type of Public-key Cryptography. The subsequent chapters will describe in details.

1.1.2 Public-key Cryptography

The inception of public-key cryptography solved the problem of key distribution of Symmetric-key cryptography. It is also know as Asymmetric Cryptography. The basic idea of public-key cryptography is to use two different keys for each communicating party . One key is public-key which can be used by anyone to encrypt message. The receiver needs correlated private key to decrypt the message. From a given public key and cipher text it is asymptotically difficult to obtain the private key.

As afore mentioned, In 1976, Whitfield Diffie and Martin Hellman published their monumental work as a key exchange protocol[DH76]. **Figure**. 1.1 shows the simple overview of the Diffie-Hellman Key Exchange (DHKE). The problems of key distribution and storage associated with symmetric cryptography were the motivation behind the concept of Asymmetric Cryptography, also referred to as Public- Key Cryptography.

In brief, the protocol has two public parameters, prime number p and a generator g known to all the parties involved in the communication. The main idea is of this protocol is based on the difficulty to solve the one way function i.e. discrete logarithm. Let's say, it is easy to calculate Alice public key k_A using Alice private key k_{Ad} as $k_A = g^{k_{Ad}} (\bmod p)$. However, it will be difficult to obtain k_{Ad} from k_A, g and p. In other words, it is easy to calculate the public key from the private key but the reverse process is practically impossible.

Step	Alice	Eve	Bob
1	Public parameter: p, g		
2	$k_{Ad} = random()$ $k_A = g^{k_{Ad}} \pmod p$		$k_{Bd} = random()$ $k_B = g^{k_{Bd}} \pmod p$
3	$k_A \longrightarrow \longleftarrow k_B$		
4	$S = k_B^{k_{Ad}} \bmod p = g^{k_{Ad}k_{Bd}} \pmod p$		$S = k_A^{k_{Bd}} \bmod p = g^{k_{Bd}k_{Ad}} \pmod p$
5	$\longleftarrow S_{Enc}(Data) \longrightarrow$		

FIGURE 1.1: Exchanging shared secret key using DH-key exchange.

Using this key-exchange we change establish a shared secret which can be used for further encrypted communication.

Rivest, Shamir and Adleman (RSA) realized this protocol in 1977 and published their magnum opus which is widely known as RSA cryptosystem [RSA78]. The security of the RSA depends on the difficulty of factorization of a larger integer into its two prime factors and the trapdoor permutation for encryption. Let us denote two large primes p and q (in practice about 1000-bit). It is easy to calculate their product to get $n = pq$. The reverse process that is for a given integer n it will be very difficult to retain p and q. Using the state-of-the art integer factoring algorithm *general number field sieve* (GNFS), it will take appoximately 2^{90} basic operation to factor a 2048-bit integer. After more than 40 year of the RSA breakthrough, it is still standing as an epitome of public key cryptography. Beside encryption, RSA also enables *digital signature* where the sender uses his private key to sign a message and the receiver verifies the signature by the senders public key. Verification of a digitally signed message gives the receiver the confidence that a senders private key is tied to his public key. It is done to prevent forgery and holds *Non-repudiation* property.

In the mid 80's the independent work of Miller [Mil86] and Koblitz [Kob87] began the journey of elliptic curve cryptosystems (ECC). The security of elliptic curve cryptography protocols depends on the difficulty to solve elliptic curve discrete logarithm problem. The mathematical details of this problem appears in Chapter 2. ECC provides shorter key length for the same level of security than RSA which makes ECC popular among the researchers. Compared to RSA, ECC has other advantages. While RSA provides encryption and digital signature; ECC has a family of algorithms for encryption, signature, key agreement and some advanced high-level cryptographic protocols such as ID-based encryption [BLS01] where user's unique ID e.g. email address can be used as a public key. The high level cryptographic functionalities are provided by paring over elliptic curves [**TODO**] which brings a new paradigm in cryptography called pairing-based cryptography.

1.1.3 Pairing-Based Cryptography

Since the inception by Sakai et al. [Sak00], pairing-based cryptography has gained much attention to cryptographic researchers as well as to mathematicians. It gives flexibility to protocol researcher to innovate applications with provable security and at the same time to mathematicians and cryptography engineers to find efficient algorithms to make pairing implementation more efficient and practical.

Generally, a pairing is a bilinear map e typically defined as $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive cyclic sub-groups of order r on a certain elliptic curve E over a finite extension field $\mathbb{F}_{p^k}$ and $\mathbb{G}_3$ is a multiplicative cyclic group of order r in $\mathbb{F}^*_{p^k}$. Let $E(\mathbb{F}_p)$ be the set of rational points over the prime field $\mathbb{F}_p$ which forms an additive Abelian group together with the point at infinity $\mathcal{O}$. The total number of rational points is denoted as $\#E(\mathbb{F}_p)$. Here, the order r is a large prime number such that $r|\#E(\mathbb{F}_p)$ and $\gcd(r,p) = 1$. The embedding degree k is the smallest positive integer such that $r|(p^k - 1)$. Two basic properties of pairing are

- bilinearity is such that $\forall P_i \in \mathbb{G}_1$ and $\forall Q_i \in \mathbb{G}_2$, where $i = 1, 2$, then $e(Q_1 + Q_2, P_1) = e(Q_1, P_1).e(Q_2, P_1)$ and $e(Q_1, P_1 + P_2) = e(Q_1, P_1).e(Q_1, P_2)$,
- and e is non-degenerate means $\forall P \in \mathbb{G}_1$ there is a $Q \in \mathbb{G}_2$ such that $e(Q, P) \neq 1$ and $\forall Q \in \mathbb{G}_2$ there is a $P \in \mathbb{G}_1$ such that $e(P, Q) \neq 1$.

Such properties allows researchers to come up with various cryptographic applications including ID-based encryption [BF01], group signature authentication [BBS04], and functional encryption [OT10]. However, the security of pairing-based cryptosystems depends on

- the difficulty of solving elliptic curve discrete logarithm problem (ECDLP) in the groups of order r over $\mathbb{F}_p$,
- the infeasibility of solving the discrete logarithm problem (DLP) in the multiplicative group $\mathbb{G}_3 \in \mathbb{F}^*_{p^k}$,
- and the difficulty of pairing inversion.

To maintain the same security level in both groups, the size of the order r and extension field p^k is chosen accordingly. If the desired security level is δ then $\log_2 r \geq 2\delta$ is desirable due to Pollard's rho algorithm [**TODO**]. For efficient pairing, the ratio $\rho = \log_2 p^k / \log_2 r \approx 1$, is expected (usually $1 \leq \rho \leq 2$). In practice, elliptic curves with small embedding degrees k and large r are selected and commonly are knows as "pairing-friendly" elliptic curves.

Galbraith et al. [GPS08] have classified pairings as three major categories based on the underlying group's structure as

- Type 1, where $\mathbb{G}_1 = \mathbb{G}_2$, also known as symmetric pairing.
- Type 2, where $\mathbb{G}_1 \neq \mathbb{G}_2$, known as asymmetric pairing. There exists an efficiently computable isomorphism $\psi : \mathbb{G}_2 \rightarrow \mathbb{G}_1$ but none in reverse direction.

- Type 3, which is also asymmetric pairing, i.e., $\mathbb{G}_1 \neq \mathbb{G}_2$. But no efficiently computable isomorphism is known in either direction between $\mathbb{G}_1$ and $\mathbb{G}_2$.

It all started from

1.2 Motivation

This section tries to outline the over all motivation behind the undertaken works. In this course, some mathematical notations will appear without detailed definitions. The subsequent chapters will define them with further elaboration.

Human civilization is moving to a direction where data generated from the devices used in our daily life will define how smart our society will be. In technical jargon we define that IoT (Internet of Things) era controlled by Data Science. Some data can be mundane with no purpose and some data can be extraordinary important. Let us imagine a case where the adversary takes controls heart beat monitor sensor of our smart watch or control sensors of self-driving car. The outcome of the damage is unimaginable. There is not alternative to protect these data from unwanted access. The challenge is, most of the IoT devices are computationally resource constrained. In some devices it is somewhat impractical to generate key pairs for widely practiced security protocols. There are several innovative solutions e.g. Broadcast encryption, or Identity based encryption that can solve such problems. The above mentioned applications stands on a compelling topic of mathematics name pairing over elliptic curve.

Pairing is a bilinear map from two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a group $\mathbb{G}_3$, where they have respectively same prime order r. In detail, $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively becomes a subgroup in an elliptic curve group $E(\mathbb{F}_q)$ and $E(\mathbb{F}_{q^k})$, and $\mathbb{G}_3$ becomes a subgroup in $\mathbb{F}_{q^k}$, where q is a power of p and an extension degree k is especially called the *embedding degree*.

In pairing-based cryptography, not only pairing calculation but also scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$, and exponentiations in $\mathbb{G}_3$ are carried out. Among these operations, since pairing is the highest cost operation, a lot of improvements for pairing such as η_T pairing over super singular curves and Ate [Hes08], *twisted* Ate[Hes08], *optimized* Ate [Mat+07], optimized *twisted* Ate[Mat+07], R-ate[LLP09], *Optimal*[Ver10], Xate [Nog+08] pairings over ordinary curves, have been proposed in the recent years. Among these pairings, the fastest pairing is η_T pairing. However, η_T pairing has a disadvantage that supersingular curves are restricted to *embedding degree* $k \leq 6$. Since the *embedding degree* is important parameter that determines the security level of pairing-based cryptographies, efficient pairings on ordinary curves whose *embedding degree* are flexibly selectable are required. This thesis targets Ate and *twisted* Ate pairings because they are efficiently calculated on ordinary curves.

On the other hand, in addition to pairing, pairing-based cryptographies need to carry out a lot of scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ in proportion to the number of users. Therefore, efficient scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ can reduce the total cost of pairing-based cryptography.

In this thesis, we propose efficient scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$, and pairings based on Ate and *twisted* Ate pairings.

Let P be a rational point in an elliptic curve group, a scalar multiplication $[s]P$ by scalar $s \in \mathbb{Z}$ means $(s-1)$-times elliptic curve additions of P. General approach to accelerate a scalar multiplication is a binary method. Note that a scalar s is at most the order r. Using a binary method, we can calculate $[s]P$ by $\lfloor \log_2 s \rfloor$-times elliptic curve doublings and $\mathrm{Hw}(s)$-times elliptic curve additions, where $\mathrm{Hw}(s)$ means the number of 1s' in the binary representation of s and it is generally called a *hamming weight* of s.

To accelerate a scalar multiplication, it is important that the scalar multiplication of an intrinsic scalar λ is calculated by efficiently computable endomorphisms. If the scalar λ is smaller than the order of an elliptic curve group, we can decompose a target scalar by λ-adic expansion. By using multi-scalar multiplication techniques, we can reduce the number of elliptic curve doublings to the bit-size of the scalar λ corresponding to the endomorphism. For example, when the elliptic curve is defined over an extension field, Frobenius endomorphism ϕ will efficiently work. Note that a Frobenius endomorphism is free from arithmetic operations. In the case of $\mathbb{G}_2$ of Ate and *twisted* Ate pairings, let t be a Frobenius trace of elliptic curve, a scalar multiplication by $\lambda = (t-1)$ corresponds to $\phi(P)$. Since $t \approx \sqrt{r}$ from Hasse's theorem, the number of elliptic doublings are reduced by about half. This relation $\phi(P) = [t-1]P$ has been considered optimal because the trace is the smallest among parameters construct an elliptic curve.

On the other hand, recent elliptic curves for pairing need one more parameter in addition to parameters such as p, t, and r. Elliptic curves over which pairing can be defined are called pairing-friendly curves. In general, it is difficult to generate such pairing-friendly curves because they need to satisfy some strict conditions. However, several methods to easily generate pairing-friendly curves are proposed in recent years [FST10]. For example, *families* of pairing friendly curves whose parameters such as characteristic p, order r, and trace t are given by polynomials in terms of integer χ are easily constructed. Especially, since *complete families* can generate a lot of elliptic curves, we can select the optimal curve suitable for pairing calculations and scalar multiplications. Among *families* of pairing friendly curves, there is a curve whose trace t is given by polynomial that has larger degree than or equal to 2. In this case, χ becomes the smallest among parameters to construct curves. Therefore, it is possible that we can obtain the relation that joins Frobenius maps and an intrinsic scalar that is smaller than trace t. Thus, it is important to optimize the relation available for a scalar multiplication for *families* of pairing friendly curves. This thesis targets *families* of pairing-friendly curves, and we mainly deal with Barreto-Naehrig (BN) curves of *embedding degree* equal to 12 that is one of the most important *complete families* of pairing-friendly curves.

In the case of BN curves, we can obtain the key relation that joins Frobenius maps and a certain smaller scalar than t. In detail, since the scalar becomes about χ and t is given by $(6\chi^2 + 1)$, its bit-size becomes a half of t.

In the case of $\mathbb{G}_2$, Frobenius endomorphism efficiently works for a scalar multiplication. However, Frobenius endomorphism does not work for a rational point $\mathbb{G}_1$ because a rational point applied Frobenius map becomes itself. Therefore, an efficiently computable endomorphism in $\mathbb{G}_1$ is required to apply the technique with Frobenius map as a scalar multiplication in $\mathbb{G}_2$. This thesis proposes a Frobenius-like endomorphism on $\mathbb{G}_1$. Using twist techniques, we can prepare the another group that is isomorphic to $\mathbb{G}_1$ over an extension field. Therefore, let the group be $\mathbb{G}'_1$, we can use Frobenius endomorphism on $\mathbb{G}'_1$. Focusing on this property, we derive a new endomorphism from the endomorphism on $\mathbb{G}'_1$. Then, we optimize a key relation available for a scalar multiplication in $\mathbb{G}_1$ that joins a new endomorphism and a certain scalar.

Furthermore, we apply the key relations available for scalar multiplications in $\mathbb{G}_1$ and $\mathbb{G}_2$ to accelerating pairing calculations.

The calculation costs of pairing are the highest among operations required for pairing-based cryptographies. Since pairing calculation is inherently sequential, it is difficult to apply the efficient parallelization technique using some recent processors have several computation cores. In general, pairing calculations consists of two calculation parts, one is Miller's algorithm and the other is *final exponentiation*. Since Miller's algorithm is slower than *final exponentiation*, several improvements for Miller's algorithm such as Ate and *twisted* Ate have been proposed. A structure of the algorithm is approximately same as that of a binary method for a scalar multiplication. Therefore, Miller's algorithm iterates a certain process $\lfloor \log_2 s \rfloor$-times as a binary method, where s is a parameter gives an loop iterations of Miller's algorithm. Since the process in Miller's algorithm needs several operations such as elliptic curve additions in elliptic curves and multiplications in extension fields, pairing becomes a fairly complex operation. Though a scalar multiplication uses a random number s, the number of calculations of Miller's algorithm is given by a specific number. For example, the number of calculation loops of Miller's algorithm for Ate pairing is given by $\lfloor \log_2(t-1) \rfloor$. That is, the calculation costs of Miller's algorithm is determined by the number of loop iterations. Therefore, we can reduce the calculation costs of Miller's algorithm by reducing its number of iterations. In addition, Hess shows that the lower bound of the number of iterations of Miller's algorithm for each pairing-friendly curve. In the case of BN curves, it becomes about $\lfloor \log_2 \chi \rfloor$, however Ate and *twisted* Ate pairing does not achieve.

In the case of a scalar multiplication, we can reduce the number of elliptic curve doublings by decomposing a scalar with the key relation. Using divisor theorem, a Miller's algorithm can be also decompose into several Miller's algorithm calculations whose the number of iterations are small. However, since there is no efficiently computable Miller's algorithm calculation for a certain number of iterations, the decomposition does not work. Rather, when we decompose Miller's algorithm, extra operations such as exponentiations in extension fields are required. This thesis shows that when we decompose a Miller's algorithm into several Miller's algorithms, some decomposed Miller's algorithms that has the bilinearity after applying *final exponentiation* can be skipped by using the property of bilinearity. In the case of Ate pairing, Miller's algorithm has the bilinearity when a parameter that gives the number of iteration is equal to $(t-1)$. Therefore, the key relation for a scalar multiplication in $\mathbb{G}_2$ is closely related to Ate pairing. In this thesis, Miller's algorithm is decomposed using the key relation, and then a new pairing whose number of iterations for Miller's algorithm is smaller than that of Ate pairing is proposed. As a result, the proposed pairing achieves the lower bound of the number of iterations for Miller's algorithm. Meanwhile, focusing on the decomposition technique for Miller's algorithm, Vercauteren, Lee et al. and the authors have respectively proposed Optimal, R-ate, and Xate pairings, independently. Optimal and R-ate pairings also achieve the lower bound of the number of iterations for Miller's algorithm. This thesis compares these pairings with our proposed pairing (Xate pairing).

On the other hand, the number of iterations for *twisted* Ate pairing is closely related to the key relation for a scalar multiplication in $\mathbb{G}_1$. That is, a new pairing based on *twisted* Ate pairing is derived, since its Miller's algorithm can be efficiently decomposed by the key relation. This pairing have been proposed by Lee et al. as *twisted R*-ate pairing, but the pairing does not achieve the lower bound of number of iterations for Miller's algorithm. In detail, the number of iterations is twice larger than that of the proposed pairing based on Ate pairing in the case of BN curves. This thesis first derives an another key relation that decomposes Miller's algorithm of *twisted* Ate pairing to two Miller's algorithm calculations whose maximum number of iterations is equal to that of the proposed pairing based on Ate pairing using the key relation available for a scalar multiplication in $\mathbb{G}_1$. Then, using a precomputed scalar multiplication, we propose a method to parallelize the two Miller's algorithm calculations with multi-pairing or *thread-computing*.

Since the proposed methods can substantially improve operations such as scalar multiplications and pairings required for pairing-based cryptographies, we can help to solve the problem on processing times. Therefore, our research contributes to promoting sophisticated cryptographies such as ID-based cryptographies and group signature authentications.

1.3 Our Contribution

1.4 Thesis Outline

Chapter 2

Fundamental Mathematics and Notation

It is necessary to recall some fundamental mathematical concept to understand the subsequent chapters and introduce the notations used in the thesis. This chapter introduces the essential mathematical backgrounds that are directly relevant to the contents of this thesis to help readers to a clear understanding of the subsequent chapters. The theoretical discussion will often appear with minimal definition and citation of the details works since details discussion is beyond the scope of this thesis. For more details of the topics discussed in this chapter we refer to [LN96; MP13]. As an additional purpose, this chapter specifies most of the notations that will appear in the upcoming chapters.

Cryptography deals with numbers mostly integers. To understand modern cryptography it is essential to have a good understanding of the underlying mathematical concepts. The following concepts is the basic for the discussion of the subsequent chapters.

2.1 Modular Arithmetic

Modular arithmetic is the fundamental tool for modern cryptography specially public key cryptosystems.

Definition 1 (Modular Arithmetic) *Let p be a positive integer named as the modulus and a and b are two arbitrary integers. If p divides $b - a$ then we can write*

$$a \equiv b \pmod{p}$$

and express as a and b are congruent modulo p.

Example 2.1 *Let, $p = 7$, $a = 19$ and $b = 5$ then* $19 \equiv 5 \pmod{7}$.

Example 2.2 *Let, $p = 7$, $a = -17$ and $b = 11$. Then* $-17 \pmod{7} = 4$ *and* $11 \pmod{7} = 4$. *We can write*

$$-17 \equiv 11 \pmod{7}$$

and usually express -17 *and* 11 *are congruent modulo* 7.

2.2 Group, Ring, Field

2.2.1 Group

The concept of group is very fundamental for understanding cryptography. It is an algebraic system defined as follows.

Definition 2 (Group) *A group is a non empty set* $\mathbb{G}$ *with a binary operation* $\circ$ *on its elements denoted as* $\langle \mathbb{G}, \circ \rangle$*, sometimes denoted by* $\mathbb{G}$ *only, which satisfies the following axioms.*

Closure *The group is closed under the operation* $\circ$*, i.e.* $\forall a \in \mathbb{G}$*, and* $\forall b \in \mathbb{G}$ *the result of* $(a \circ b) = c \in \mathbb{G}$. [1]

Identity element *There exist an **identity element*** e *also know as neutral element or unit element in* $\mathbb{G}$ *such that* $\forall a \in \mathbb{G}$, $a \circ e = e \circ a = a$.

Inverse element *For* $\forall a \in \mathbb{G}$*, there exists an element* $b \in \mathbb{G}$ *such that* $a \circ b = e = b \circ a$*, where* b *is called inverse element of* a.

Associativity *Elements in group* $\mathbb{G}$ *should follow associativity. i.e.* $(a \circ b) \circ c = a \circ (b \circ c)$ *for all* $a, b, c \in \mathbb{G}$.

Definition 3 (Commutative Group)

A group $\mathbb{G}$ *will be commutative if* $a \circ b = b \circ a$ *for all* $a, b \in \mathbb{G}$.

■

A commutative group is also called *abelian* group.

Example 2.3 *The set of integers* $\mathbb{Z}$ *forms a group under the group operation of addition* $+$ *denoted as* $(\mathbb{Z}, +)$. 0 *is the identity element of the group.*

Example 2.4 *The set of positive integers* $\mathbb{N}$ *under addition does not form a group since elements have not inverse.*

Definition 4 (Order of a Group) *The order of a group* $\mathbb{G}$ *often denoted as* $\#\mathbb{G}$ *is the number of elements in the group* $\mathbb{G}$. ■

Remark 1 *Groups order can be finite and infinite. In example 2.3,* $(\mathbb{Z}, +)$ *has infinite order.*

Definition 5 (Order of group element) *For an element* $a \in \mathbb{G}$*, the smallest positive integer* m *such that* $a^m = e$ *is called the order of* a*, where* e *is the identity element in* $\mathbb{G}$. ■

Example 2.5 *Finite group: As shown in example 2.4, the set* $\mathbb{N}$ *under addition does not form a group since it does not satisfy the group axioms. Let us consider a*

[1] $\forall$ symbol bears is usual notation *"for all"*

set $\mathbb{N}_n$ under the operation $\mod n$ *such that*

$$\mathbb{N}_n = \{0, 1, 2, 3, \cdots, n-1\}$$

where $n \in \mathbb{N}$. It means $\mathbb{N}_n$ is the set of remainders under "$\mod n$*". Recall the modular arithmetic that*

$$a + b \equiv c \mod n \qquad a, b \in \mathbb{N}_n,$$

means c is associated to a remainder on division by n when $a + b = c \notin \mathbb{N}_n$. It makes c belongs to $\mathbb{N}_n$ making $(\mathbb{N}_n, +)$ forming a group. In also includes element 0 which acts as an identity element.

Definition 6 (Group generator) *For a given group $\mathbb{G}$ if there is an element $g \in \mathbb{G}$ such that for any $a \in \mathbb{G}$ there exist an unique integer i with $a = g^i$ then g will be called a generator of $\mathbb{G}$* ■

Definition 7 (Cyclic Group) *A group $\mathbb{G}$ will be* cyclic *if there exist at least one generator $g \in \mathbb{G}$. Cyclic group usually expressed as $\mathbb{G} = \langle g \rangle$* ■

Remark 2 *The number of generator in a group $\mathbb{G}$ of order n is defined by Euler's totient function $\phi(n)$*[2]*. If n is a prime p then the group $\mathbb{G}$ will be called prime order group and it will have $\phi(p) = p - 1$ generators.*

Definition 8 (Cyclic Group) *A group $\mathbb{G}$ will be* cyclic *if there exist at least one generator $g \in \mathbb{G}$. Cyclic group usually expressed as $\mathbb{G} = \langle g \rangle$* ■

In this case we use the notation $\langle \mathbb{G}, \circ \rangle$, there exists some ambiguity which operation we consider. Therefore, the following two types of group nations are very common in literature.

Definition 9 (Additive group) *A cyclic group is called additive if we tend to write its group operation in the same way we do additions, that is*

$$f = g + x$$

can also appear as $[x]g$ meaning applying $x - 1$ times addition operator + on g. It is also common to write as $x \cdot g$. For example, 1 is one of generators in group $(\mathbb{Z}_5, +)$ under addition modular 5, then $1 \cdot 4$ can be written as

$$4 = 1 + 1 + 1 + 1.$$

■

Definition 10 (Multiplicative group) *A cyclic group is called multiplicative if we tend to write its group operation in the same way we do multiplication, that is*

$$f = g \cdot x \text{ or } f = g^x$$

■

[2] When n is a positive integer, Euler's totient function $\phi(n)$ = number of positive integers less than or equal to n that are co-prime to n

Remark 3 *In both notation the x is an integer called the discrete logarithm of h to the base g.*

Remark 4 *Unless otherwise stated, through out this thesis we will use the xg notation for ordinary addition e.g. $a + a = 2a$ and $a + a + a = 3a$ and for multiplicative notation, these will denoted by a^2, a^3.*

From the definition cyclic group, it can be see visualized that any elements in cyclic a group are generated with iterative operations of generator g. **Figure**. 2.1 shows this schematically.

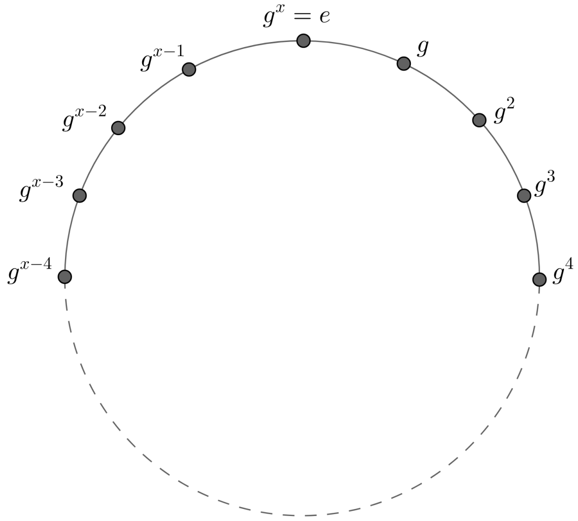


FIGURE 2.1: Cyclic group

A a well known practice of presenting a finite group's operation is *Cayley table* as shown in example 2.6. Cayley table shows all possible group operation that can be performed in a finite group.

Example 2.6 *The Cayley table for the group $\mathbb{Z}_4$ is:*

$\oplus_4$	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

In the above example of group $(\mathbb{Z}_4, +)$, there are $\phi(4) = 2$ generators, 3 and 1.

Definition 11 (Subgroup) *Let $\mathbb{H}$ be a non-empty subset fo group $\mathbb{G}$, $\mathbb{H}$ will be called subgroup of $\mathbb{G}$ if $\mathbb{H}$ itself follows group axioms and $\mathbb{H}$ has the same identity element of group $\mathbb{G}$.* ■

Theorem 1 (Lagrange's Theorem:) *Let $\mathbb{G}$ be a finite abelian group and $\mathbb{H}$ is a subgroup of $\mathbb{G}$. The order of $\mathbb{G}$, $\#\mathbb{G}$ is divisible by the order of subgroup $\mathbb{H}$, $\#\mathbb{H}$ i.e. $\#\mathbb{H}|\#\mathbb{G}$.* ■

Theorem 2 (Fermat's Little Theorem:) *Let p is a prime and $a \in \mathbb{Z}$, then*

$$a^p = a \pmod{p}$$

■

Fermat's *little theorem* is a special case of Lagrange's theorem.

2.2.2 Homomorphism in groups

Morphisms in groups is often used the research of cryptography and inseparable to for pairing-based cryptography research.

Definition 12 (Homomorphism) *Let $(\mathbb{G}, \circ)$ and $(\mathbb{G}^{'}, \star)$ be two groups with identity elements e and e' respectively. A homomorphism is a map f which preserves the group structure while the elements are mapped from $(\mathbb{G}, \circ)$ to $(\mathbb{G}^{'}, \star)$.* ■

A homomorphic map obeys the following conditions:

- $\forall a, b \in \mathbb{G}$, $f(a \circ b) = f(a) \star f(b)$.
- For every $a \in \mathbb{G}$, the inverse map is $f(a^{-1}) = f(a)^{-1}$.
- Identity element mapping also preserves the structure i.e. $f(e) = e'$.

Types of Homomorphism

Isomorphism If element from $\mathbb{G}$ and $\mathbb{G}^{'}$ have bijective relation then $\mathbb{G}$ and $\mathbb{G}^{'}$ are isomorphic to each other.

Endomorphism If elements from group $(\mathbb{G}, \circ)$ is mapped to itself then it is called endomorphism. A frequently used endomorphism in cryptographic algorithms is Frobenius endomorphism.

Authomorphism If element of a group has both endomorphism and isomorphism then it is called automorphism.

Definition 13 (Kernel) *Let $(\mathbb{G}, \circ)$ and $(\mathbb{G}^{'}, \star)$ be two groups with identity elements e and e' respectively and f is homomorphism from $(\mathbb{G}, \circ)$ to $(\mathbb{G}^{'}, \star)$. The kernel of f is denoted as $Ker\{f\}$, defined by*

$$Ker(f) = \{a \in \mathbb{G} : f(a) = e'\}$$

. ■

2.2.3 Ring

The concept of *Ring* will not come as frequently as group and field in the subsequent chapters. However, it is relevant to define ring to understand the related concept.

Definition 14 (Ring) *A **ring** $\mathbb{R}$ is an algebraic structure with two operations, i.e. addition $+$ and multiplication $\cdot$ usually denote as $\mathbb{R}, +, \cdot$.*

- *$\mathbb{R}$ is abelian group under addition operation.*
- *Under multiplication, $\mathbb{R}$ is closed and associative with identity element is 1.*
- *Multiplication is distributive over addition: $\forall a, b, c \in \mathbb{R} : a \cdot (b + c) = a \cdot b + a \cdot c$.*

■

If multiplication operation is commutative, $\mathbb{R}$ forms a commutative ring.

Definition 15 (Multiplicative Inverse Modulo n) *Let $\mathbb{Z}_n$ be a set under modulo n and $a \in \mathbb{Z}_n$. The multiplicative inverse modulo n of a can be written as follows:*

$$a \cdot x \equiv 1 \bmod n.$$

The value x is the multiplicative inverse modulo n of a, often written as a^{-1}. ■

Such value of x only exists if $\gcd(x, n) = 1$. If $n = p$ is a prime then every nonzero element in the set $\mathbb{Z}_p$ will have multiplicative inverse. Such $(\mathbb{Z}_p, +, \cdot)$ will be a ring and having the above property it will form a field.

2.2.4 Field

Definition 16 (Field) *A field $(\mathbb{F}, +, \cdot)$ is a set that obeys two binary operations denoted by $+$ and $\cdot$, such that:*

- *$\mathbb{F}$ is a commutative group with respect to $+$ having identity element 0.*
- *Let $\mathbb{F}^*$is a subset of $\mathbb{F}$ having only not-zero element of $\mathbb{F}$ i.e. $\mathbb{F}^* = \mathbb{F} \setminus \{0\}$. Then $\mathbb{F}^*$will be called a commutative group respect to multiplication$\cdot$ where every element should have multiplicative inverse in $\mathbb{F}^*$.*
- *For all $a, b, c \in \mathbb{F}$ the distributive law will be followed, e.g. $a \cdot (b + c) = a \cdot b + a \cdot c$ and $(b + c) \cdot a = b \cdot a + c \cdot a$.*

■

Definition 17 (Subfield) *Let $\mathbb{F}_1$ is a subset of field $\mathbb{F}$. $\mathbb{F}_1$ will be called a subfeld if $\mathbb{F}_1$ itself obeys the laws of field with respect to the field operation inherited from $\mathbb{F}$.* ■

Remark 5 *In Definition 17, $\mathbb{F}$ is called an* extension field *of $\mathbb{F}_1$. If $\mathbb{F}_1 \neq \mathbb{F}$, then $\mathbb{F}_1$ is a* proper subfield *of $\mathbb{F}$.*

Definition 18 (Order of Finite Field) *The order is the number of elements in $\mathbb{F}$. If the order of $\mathbb{F}$ is finite, $\mathbb{F}$ is called finite field.* ■

Definition 19 (Characteristic of Finite Field) *Let $\mathbb{F}$ be a field and smallest positive number n such that $n \cdot a = 0$ for every $a \in \mathbb{F}$. Such n is called characteristic. If there is no such n in $\mathbb{F}$ then $\mathbb{F}$ has characteristics 0.* ■

Most of the works presented in this dissertation deals with finite fields only. A common property of finite fields often used in cryptographic is fllows:

Theorem 3 *For every finite field* $\mathbb{F}$*, the multiplicative group* $(\mathbb{F}^*, \cdot)$ *is cyclic.* $\blacksquare$

Definition 20 (Prime Field) *Let p be a prime. The ring of integers modulo p is a finite field of characteristics p having field order p denoted as* $\mathbb{F}_p$ *is called a prime field.* $\blacksquare$

Remark 6 *A prime field contains no proper subfield.*

Theorem 4 *Every finite field has a prime field as a subfield.* $\blacksquare$

In this work we classified finite fields into two types, i.e. prime field $\mathbb{F}_p$ and its extension field. Defined 2.2.5 explains more of extension field. The prime field $\mathbb{F}_p$ has the order and characteristic as p. Using the modular arithmetic in the same way as Definition 2.2.5, we can define fundamental operations of prime field $\mathbb{F}_p = \{0, 1, 2 \cdots, p-1\}$. The Cayley table will de

Example 2.7 *The Cayley table for the two operations + and · for elements in* $\mathbb{F}_5$ *are as follows:*

+	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

·	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

As described above, we can define arithmetic operations in $\mathbb{F}_p$ by modular operations (mod p) for integers. However, it does not work in an extension field $\mathbb{F}_{p^m}$. In the next section, arithmetic operations in extension field $\mathbb{F}_{p^m}$ is described in detail.

2.2.5 Extension Field

A subset $\mathbb{F}_0$ of a field $\mathbb{F}$ that is itself a field under the operations of $\mathbb{F}$ will be called a *subfield* of $\mathbb{F}$. In this case, $\mathbb{F}$ is called an *extension field* of $\mathbb{F}_0$. An extension field of a prime field $\mathbb{F}_p$ can be represented as m-dimensional vector space that has m elements in $\mathbb{F}_p$. Let the vector space be the m-th extension field, it is denoted by $\mathbb{F}_{p^m}$. The order of extension fields $\mathbb{F}_{p^m}$ is given as p^m. In what follows, let q be the power of p, the extension field of a prime field $\mathbb{F}_p$ is denoted by $\mathbb{F}_q$.

There are several methods to represent an element in extension fields, such as polynomial basis and normal basis. In this thesis, we use normal basis. Let ω be a root of m-th irreducible polynomial over $\mathbb{F}_q$, we consider the following m elements.

$$\omega,\ \omega^q,\ \omega^{q^2},\ \cdots,\ \omega^{q^{m-1}}$$

All elements in this set are conjugate to each other. When the set of the conjugates become linearly independent, this is called *normal basis*. Using normal basis, an element $\alpha \in \mathbb{F}_q$ is expressed as a polynomial by

$$\alpha = a_1\omega + a_2\omega^q + a_3\omega^{q^2} + \cdots + a_m\omega^{q^{m-1}}, \tag{2.1}$$

where $a_1,\ a_2,\ a_3, \cdots,\ a_m \in \mathbb{F}_q$.

Arithmetic operations in $\mathbb{F}_{q^m}$ are carried out with ordinary addition and multiplication for polynomial and modular reduction by irreducible polynomial.

2.3 Frobenius Map

For any element $\alpha \in \mathbb{F}_{q^m}$, let us consider the following map $\pi_q : \alpha \rightarrow \alpha^q$.

$$\begin{aligned} \pi_q(\alpha) &= \left(a_1\omega + a_2\omega^q + a_3\omega^{q^2} + \cdots + a_m\omega^{q^{m-1}}\right)^q \\ &= a_1\omega^q + a_2\omega^{q^2} + a_3\omega^{q^3} + \cdots + a_m\omega^{q^m} \\ &= a_m\omega + a_1\omega^q + a_2\omega^{q^2} + \cdots + a_{m-1}\omega^{q^{m-1}} \end{aligned} \tag{2.2}$$

Note that the order of $\mathbb{F}^*_{q^m}$ is given by $q^m - 1$, that is, $\omega^{q^m} = \omega$ is satisfied. Furthermore, a^q is equal to a for each coefficients a.

Therefore, the map $\pi_q(\alpha)$ is efficiently calculated by cyclic shift operations among its basis coefficients, which is free from arithmetic operations. From the computational efficiency, the map π_q is especially called Frobenius map.

In ElGamal Encryption, many exponentiations are executed in encryption and decryption processes. When the exponent is equal to p, its calculation cost can be reduced by using Frobenius map. Therefore, Frobenius map is widely used in the cryptographic application.

2.3.1 Quadratic Residue/Quadratic Non-residue, and Cubic Residue/Cubic Non-residue

For any non-zero element $d \in \mathbb{F}_q$, d is called a Quadratic Residue (QR) when x such that $x^2 = d$ exists in $\mathbb{F}_q$. On the other hand, when such an x does not exist in $\mathbb{F}_q$, d is called a Quadratic Non-Residue (QNR). We can identify whether or not d is a QR by following test.

$$d^{(q-1)/2} = \begin{cases} 1 & : \text{QR} \\ -1 & : \text{QNR} \end{cases} \tag{2.3}$$

All elements in finite fields $\mathbb{F}_q$ of odd characteristics become QR in extension fields $\mathbb{F}_{q^{2j}}$. On the other hand, quadratic non-residues also become QNR in $\mathbb{F}_{q^i}$, where i is not divisible by 2.

2.4 Elliptic Curve

In this section, we review elliptic curves and pairings.

2.4.1 Additive Group over Elliptic Curves

In general, let $p > 3$, an elliptic curve $E/\mathbb{F}_p$ over a finite field $\mathbb{F}_p$ is defined as

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b,\ 42a^3 + 27b^2 \neq 0,\ a, b \in \mathbb{F}_p. \tag{2.4}$$

The field that x and y belong to is called the definition field. The solutions (x, y) of Eq.(2.4) is called rational points. $E(\mathbb{F}_q)$ that is the set of rational points on the curve, including the *point at infinity* $\mathcal{O}$, forms an additive abelian group. The *point at infinity* works as an unity element in $E(\mathbb{F}_q)$. When the definition field is $\mathbb{F}_{q^m}$, we denote the additive group by $E(\mathbb{F}_{q^m})$.

For rational points $P_1(x_1, y_1)$, $P_2(x_2, y_2) \in E(\mathbb{F}_q)$, the elliptic curve addition $P_3(x_3, y_3) = P_1 + P_2$ is defined as follows.

$$\begin{aligned}
\lambda &= \left\{ \begin{array}{ll} \dfrac{y_2 - y_1}{x_2 - x_1} & P_1 \neq P_2,\ x_1 \neq x_2 \\[2ex] \dfrac{3x_1^2 + a}{2y_1} & P_1 = P_2 \end{array} \right. \\
x_3 &= \lambda^3 - x_1 - x_2 \\
y_3 &= (x_1 - x_3)\lambda - y_1
\end{aligned}$$

In the case of $P_1 = P_2$, the addition is especially called elliptic curve doubling.

Let a rational point $P(x, y)$, an inverse point $-P$ is given by $-P(x, -y)$. Elliptic curve cryptographies is constructed on elliptic curve groups $E(\mathbb{F}_q)$.

Let $\#E(\mathbb{F}_p)$ be the order of $E(\mathbb{F}_p)$, it is given as

$$\#E(\mathbb{F}_p) = p + 1 - t, \tag{2.5}$$

where t is the Frobenius trace of $E(\mathbb{F}_p)$.

From Hasse's theorem, t satisfies

$$|t| \leq 2\sqrt{p}. \tag{2.6}$$

Let $[s]P$ denote the $(s-1)$-times addition of a rational point P as,

$$[s]P = \sum_{i=0}^{s-1} P. \tag{2.7}$$

This operation is called a scalar multiplication. As a general approach for accelerating a scalar multiplication, the binary method is the most widely used. **Algorithm**. **??** shows the binary method. The binary method iterates elliptic curve doublings and elliptic curve additions using binary representation of scalar. A scalar multiplication needs $\lfloor \log_2 s \rfloor$ elliptic curve doublings and $\lfloor \log_2 s \rfloor/2$ elliptic curve additions on average.

Algorithm 2.1 : Binary method

```
Input :  $P$, $n$-bit integer $s = \sum_{i=0}^{\ell-1} s_i 2^i$, $s_i \in \{0, 1\}$
Output : $R = [s]P$
$R \leftarrow \mathcal{O}$
For $i = \ell - 1$ to 0 by $-1$ do:
    $R \leftarrow R + R$
    If $s_i = 1$ then $R \leftarrow R + P$
Return $R$
```

2.4.2 Elliptic curve [Was03]

Let E be the elliptic curve defined over the prime field $\mathbb{F}_p$ as follows:

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b, \tag{2.8}$$

where $4a^3 + 27b^2 \neq 0$ and $a, b \in \mathbb{F}_p$. Points satisfying Eq.(5.1) are known as rational points on the curve. The set of rational points including the *point at infinity* $\mathcal{O}$ on the curve forms an additive Abelian group denoted by $E(\mathbb{F}_p)$ whose order is denoted as $\#E(\mathbb{F}_p)$, can be obtained as,

$$\#E(\mathbb{F}_p) = p + 1 - t, \tag{2.9}$$

where t is called the Frobenius trace of $E(\mathbb{F}_p)$. When the definition field is the k-th degree extension field $\mathbb{F}_{p^k}$, rational points on the curve E also forms an additive Abelian group denoted as $E(\mathbb{F}_{p^k})$. The order of $E(\mathbb{F}_{p^k})$ is denoted as $\#E(\mathbb{F}_{p^k})$ and given by the Weil's theorem [Coh+05] as follows:

$$\#E(\mathbb{F}_{p^k}) = p^k + 1 - t_k, \tag{2.10}$$

where $t_k = \alpha^k + \beta^k$. α and β are complex conjugate numbers that confirm the relation $f(\alpha) = f(\beta) = 0$, where $f(\pi)$ is a polynomial such that $f(\pi) = \pi^2 - t\pi + p$ and π is the Frobenius map. In practice, t_k is determined recursively with $p = \alpha\beta$ and $t_1 = \alpha + \beta$. Moreover, the $\#E(\mathbb{F}_{p^k})$ is such that $\#E(\mathbb{F}_p) \mid \#E(\mathbb{F}_{p^k})$, which confirms that $E(\mathbb{F}_p)$ is a subgroup of $E(\mathbb{F}_{p^k})$.

2.4.2.1 Point addition

Let's consider two rational points $L = (x_l, y_l)$, $M = (x_m, y_m)$, and their addition $N = L + M$, where $N = (x_n, y_n)$ and $L, M, N \in E(\mathbb{F}_p)$. Then, the x and y

coordinates of N are obtained as follows:

$$(x_n, y_n) = ((\lambda^2 - x_l - x_m), (x_l - x_n)\lambda - y_l), \tag{2.11a}$$

where λ is given as follows:

$$\lambda = \begin{cases} (y_m - y_l)(x_m - x_l)^{-1} & (L \neq M), \\ \\ (3x_l^2 + a)(2y_l)^{-1} & (L = M). \end{cases} \tag{2.11b}$$

Here λ is the tangent at the point on the curve and $\mathcal{O}$ it the additive unity in $E(\mathbb{F}_p)$. When $L \neq M$ then $L + M$ is called elliptic curve addition (ECA). If $L = M$ then $L + M = 2L$, which is known as elliptic curve doubling (ECD).

2.4.2.2 Scalar multiplication

Let r be the *order* of the target rational point group and s be the scalar such that $0 \leq s < r$. Scalar multiplication of rational point M, typically denoted as $[s]M$ can be calculated by $(s-1)$-times additions of M as,

$$[s]M = \underbrace{M + M + \cdots + M}_{s-1 \text{ times additions}}. \tag{2.12}$$

If $s = r$, where r is the order of the curve then $[r]M = \mathcal{O}$. When $[s]M = N$, if s is unknown, then the solving s from M and N is known as elliptic curve discrete logarithm problem (ECDLP). The security of elliptic curve cryptography lies on the difficulty of solving ECDLP.

2.4.2.2.1 Binary method The binary method is an extensively applied method for calculating the elliptic curve scalar multiplication. The pseudo code of left-to-right binary scalar multiplication algorithm is shown in Algorithm 6. This algorithm scans the bits of scalar s from the most significant bit to the least significant bit. When $s[i] = 1$, it performs ECA and ECD otherwise only ECD is calculated. This method is easy to implement but the important drawback of this method is not resistant to *side channel attack* [Koc96].

Algorithm 1: Left-to-right binary algorithm for elliptic curve scalar multiplication

```
Input: $P, s$
Output: $[s]P$
$T \leftarrow 0$
for $i = \lfloor \log_2 s \rfloor$ to $0$ do
    $T \leftarrow T + T$
    if $s[i] = 1$ then
        $T \leftarrow T + P$
return $T$
```

2.4.2.2.2 Montgomery ladder method Montgomery ladder algorithm is said to be resistant to *side channel attack*. Such resistance comes by paying tolls as calculation overhead which slows down this method than binary method. Algorithm 7 shows the Montgomery ladder algorithm for scalar multiplication. Montgomery ladder has some similarity with binary method except in each iteration it performs ECA and ECD.

Algorithm 2: Montgomery ladder algorithm for elliptic curve scalar multiplication

```
Input: A point $P$, an integer $s$
Output: $[s]P$
$T_0 \leftarrow 0, T_1 \leftarrow P$
for $i = \lfloor \log_2 s \rfloor$ to $0$ do
    if $s[i] = 1$ then
        $T_0 \leftarrow T_0 + T_1$
        $T_1 \leftarrow T_1 + T_1$
    else if $s[i] = 0$ then
        $T_1 \leftarrow T_0 + T_1$
        $T_0 \leftarrow T_0 + T_0$
return $T_0$
```

2.4.2.2.3 Sliding-window method Sliding-window [Coh+05] algorithm is also resistant to *side channel attack* and at the same time it is faster than Montgomery ladder. In this method the scalar s is processed in blocks of length w, known as window size. Algorithm 3 shows the sliding-window algorithm for scalar multiplication.

Algorithm 3: Sliding window algorithm for elliptic curve scalar multiplication

```
Input: A point $P$, an integer $s = \sum_{j=0}^{l-1} s_j 2^j$, $s_j \in \{0,1\}$, window size $w \geq 1$
Output: $Q = [s]P$
Pre-computation.
$P_1 \leftarrow P, P_2 \leftarrow [2]P$
for $i = 1$ to $2^{w-1} - 1$ do
    $P_{2i+1} \leftarrow P_{2i-1} + P_2$
$j \leftarrow l - 1, Q \leftarrow \mathcal{O}$.
Main loop.
while $j \geq 0$ do
    if $s_j = 0$ then
        $Q \leftarrow [2]Q, j \leftarrow j - 1$
    else
        Let $t$ be the least ineger such that
        $j - t + 1 \leq w$ and $s_t = 1$
        $h_j \leftarrow (s_j s_{j-1} \cdots s_t)_2$
        $Q \leftarrow [2^{j-t+1}]Q + P_{h_j}$
        $j \leftarrow t - 1$
return $Q$
```

2.4.3 Frobenius Map on Elliptic Curve Groups

In this section, we introduce Frobenius map for a rational point in $E(\mathbb{F}_q)$. For any rational point $P = (x, y)$, Frobenius map ϕ is given by $\phi : P(x, y) \longrightarrow (x^q, y^q)$. Then, the following relation holds for any rational points in $E(\mathbb{F}_q)$ with regard to Frobenius map.

$$\left(\phi^2 - [t]\phi + [q]\right) P = \mathcal{O}.$$

Thus, we have

$$[q]P = \left([t]\phi - \phi^2\right) P. \tag{2.13}$$

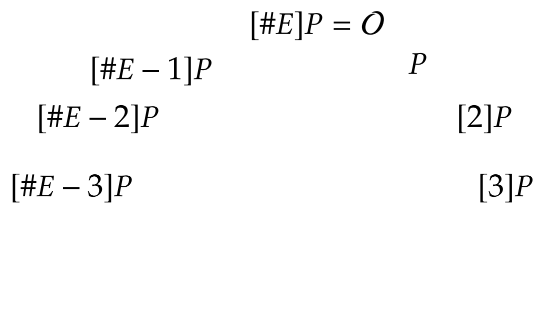


FIGURE 2.2: An image of elliptic curve group

From Hasse's theorem, note the bit-size of Frobenius trace t is about a half of the characteristic p. Using Eq.(2.13), we can efficiently calculate scalar multiplication [Kob92].

Chapter 3

Improved Optimal-Ate Pairing for KSS-18 Curve

3.1 Introduction

3.1.1 Background and Motivation

From the very beginning of the cryptosystems that utilizes elliptic curve pairing; proposed independently by Sakai et al. [SK03] and Joux [Jou04], has unlocked numerous novel ideas to researchers. Many researchers tried to find out security protocol that exploits pairings to remove the need of certification by a trusted authority. In this consequence, several ingenious pairing based encryption scheme such as ID-based encryption scheme by Boneh and Franklin [BF01] and group signature authentication by Nakanishi et al. [NF05] have come into the focus. In such outcome, Ate-based pairings such as Ate [Coh+05], Optimal-ate [Ver10], twisted Ate [Mat+07], R-ate [LLP09], and u-Ate [Nog+08] pairings and their applications in cryptosystems have caught much attention since they have achieved quite efficient pairing calculation. But it has always been a challenge for researchers to make pairing calculation more efficient for being used practically as pairing calculation is regarded as quite time consuming operation.

3.1.2 General Notation

As aforementioned, pairing is a bilinear map from two rational point groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a multiplicative group $\mathbb{G}_3$ [SCA86]. Bilinear pairing operation consist of two predominant parts, named as Miller's algorithm and final exponentiation. In the case of Ate-based pairing using KSS-18 pairing-friendly elliptic curve of embedding degree $k = 18$, the bilinear map is denoted by $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$, The groups $\mathbb{G}_1 \subset E(\mathbb{F}_p)$, $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ and $\mathbb{G}_3 \subset \mathbb{F}^*_{p^{18}}$ and p denotes the characteristic of $\mathbb{F}_p$. The elliptic curve E is defined over the extension field $\mathbb{F}_{p^{18}}$. The rational point in $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ has a special vector representation where out of 18 $\mathbb{F}_p$ coefficients, continuously 3 of them are non-zero and the others are zero. By utilizing such representation along with the sextic twist and isomorphic mapping in subfield of $\mathbb{F}_{p^{18}}$, this chapter has computed the elliptic curve doubling and elliptic curve addition in the

Miller's algorithm as $\mathbb{F}_{p^3}$ arithmetic without any explicit mapping from $\mathbb{F}_{p^{18}}$ to $\mathbb{F}_{p^3}$.

3.1.3 Contribution Outline

This chapter proposes *pseudo 12-sparse multiplication* in affine coordinates for line evaluation in the Miller's algorithm by considering the fact that multiplying or dividing the result of Miller's loop calculation by an arbitrary non-zero $\mathbb{F}_p$ element does not change the result as the following final exponentiation cancels the effect of multiplication or division. Following the division by a non-zero $\mathbb{F}_p$ element, one of the 7 non-zero $\mathbb{F}_p$ coefficients (which is a combination of 1 $\mathbb{F}_p$ and 2 $\mathbb{F}_{p^3}$ coefficients) becomes 1 that yields calculation efficiency. The calculation overhead caused from the division is canceled by isomorphic mapping with a quadratic and cubic residue in $\mathbb{F}_p$. This chapter does not end by giving only the theoretic proposal of improvement of Optimal-Ate pairing by pseudo 12-sparse multiplication. In order to evaluate the theoretic proposal, this chapter shows some experimental results with recommended parameter settings.

3.1.4 Related Works

Finding pairing friendly curves [FST06] and construction of efficient extension field arithmetic are the ground work for any pairing operation. Many research has been conducted for finding pairing friendly curves [BLS03; DEM05] and efficient extension field arithmetic [BP01]. Some previous work on optimizing the pairing algorithm on pairing friendly curve such Optimal-Ate pairing by Matsuda et al. [Mat+07] on Barreto-Naehrig (BN) curve [BN06] is already carried out. The previous work of Mori et al. [Mor+14] has showed the *pseudo 8-sparse multiplication* to efficiently calculate Miller's algorithm defined over BN curve. Apart from it, Aranha et al. [Ara+13] has improved Optimal-Ate pairing over KSS-18 curve for 192 bit security level by utilizing the relation $t(u) - 1 \equiv u + 3p(u) \bmod r(u)$ where $t(u)$ is the Frobenius trace of KSS-18 curve, u is an integer also known as *mother parameter*, $p(u)$ is the prime number and $r(u)$ is the order of the curve. This chapter has exclusively focused on efficiently calculating the Miller's loop of Optimal-Ate pairing defined over KSS-18 curve [KSS07] for 192-bit security level by applying *pseudo 12-sparse multiplication* technique along with other optimization approaches. The parameter settings recommended in [Ara+13] for 192 bit security on KSS-18 curve is used in the simulation implementation. But in the recent work, Kim et al. [KB16] has suggested to update the key sizes associated with pairing-based cryptography due to the development new algorithm to solve discrete logarithm problem over finite field. The parameter settings of [Ara+13] does not end up at the 192 bit security level according to [KB16]. However the parameter settings of [Ara+13] is primarily adapted in this chapter in order to show the resemblance of the proposal with the experimental result.

3.2 Fundamentals

This section briefly reviews the fundamentals of towering extension field with irreducible binomials [BP01], sextic twist, pairings and sparse multiplication [Mor+14] with respect to KSS-18 curve [KSS07].

3.2.1 KSS Curve

Kachisa-Schaefer-Scott (KSS) curve [KSS07] is a non supersingular pairing friendly elliptic curve of embedding degrees $k = \{16, 18, 32, 36, 40\}$. This chapter considers KSS curve of embedding degree $k = 18$, in short KSS-18 curve. The equation of KSS-18 curve defined over $\mathbb{F}_{p^{18}}$ is given as follows:

$$E : y^2 = x^3 + b, \quad b \in \mathbb{F}_p \tag{3.1}$$

together with the following parameter settings,

$$\begin{aligned} p(u) &= (u^8+5u^7+7u^6+37u^5+188u^4+259u^3+343u^2+1763u+2401)/21, && \text{(3.2-a)}\\ r(u) &= (u^6+37u^3+343)/343, && \text{(3.2-b)}\\ t(u) &= (u^4+16u+7)/7, && \text{(3.2-c)} \end{aligned}$$

where $b \neq 0, x, y \in \mathbb{F}_{p^{18}}$ and characteristic p (prime number), Frobenius trace t and order r are obtained systematically by using the integer variable u, such that $u \equiv 14 \pmod{42}$.

3.2.2 Towering Extension Field

In extension field arithmetic, higher level computations can be improved by towering. In towering, higher degree extension field is constructed as a polynomial of lower degree extension fields. Since KSS-18 curve is defined over $\mathbb{F}_{p^{18}}$, this chapter has represented extension field $\mathbb{F}_{p^{18}}$ as a tower of sub-fields to improve arithmetic operations. In some previous works, such as Bailey et al. [BP01] explained tower of extension by using irreducible binomials. In what follows, let $(p - 1)$ be divisible by 3 and c is a certain quadratic and cubic non residue in $\mathbb{F}_p$. Then for KSS-18-curve [KSS07], where $k = 18$, $\mathbb{F}_{p^{18}}$ is constructed as tower field with irreducible binomial as follows:

$$\begin{cases} \mathbb{F}_{p^3} &= \mathbb{F}_p[i]/(i^3 - c), \\ \mathbb{F}_{p^6} &= \mathbb{F}_{p^3}[v]/(v^2 - i), \\ \mathbb{F}_{p^{18}} &= \mathbb{F}_{p^6}[\theta]/(\theta^3 - v). \end{cases} \tag{3.3}$$

Here isomorphic sextic twist of KSS-18 curve is available in the base extension field $\mathbb{F}_{p^3}$ where the original curve is defined over $\mathbb{F}_{p^{18}}$

3.2.3 Sextic Twist of KSS-18 Curve

Let z be a certain quadratic and cubic non residue in $\mathbb{F}_{p^3}$. The sextic twisted curve E' of KSS-18 curve E (Eq.(3.1)) and their isomorphic mapping ψ_6 are given as follows:

$$\begin{aligned} E' &: y^2 = x^3 + bz, \quad b \in \mathbb{F}_p \\ \psi_6 &: E'(\mathbb{F}_{p^3})[r] \longmapsto E(\mathbb{F}_{p^{18}})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ & \quad (x, y) \longmapsto (z^{-1/3}x, z^{-1/2}y) \end{aligned} \tag{3.4}$$

where Ker(·) denotes the kernel of the mapping. Frobenius mapping π_p for rational point is given as

$$\pi_p : (x, y) \longmapsto (x^p, y^p). \tag{3.5}$$

The order of the sextic twisted isomorphic curve $\#E'(\mathbb{F}_{p^3})$ is also divisible by the order of KSS-18 curve E defined over $\mathbb{F}_p$ denoted as r. Extension field arithmetic by utilizing the sextic twisted subfield curve $E'(\mathbb{F}_{p^3})$ based on the isomorphic twist can improve pairing calculation. In this chapter, $E'(\mathbb{F}_{p^3})[r]$ shown in Eq. (12.10) is denoted as $\mathbb{G}'_2$.

3.2.4 Isomorphic Mapping between $E(\mathbb{F}_p)$ and $\hat{E}(\mathbb{F}_p)$

Let us consider $\hat{E}(\mathbb{F}_p)$ is isomorphic to $E(\mathbb{F}_p)$ and $\hat{z}$ as a quadratic and cubic residue in $\mathbb{F}_p$. Mapping between $E(\mathbb{F}_p)$ and $\hat{E}(\mathbb{F}_p)$ is given as follows:

$$\begin{aligned} \hat{E} &: y^2 = x^3 + b\hat{z}, \\ & \quad \hat{E}(\mathbb{F}_p)[r] \longmapsto E(\mathbb{F}_p)[r], \\ & \quad (x, y) \longmapsto (\hat{z}^{-1/3}x, \hat{z}^{-1/2}y), \end{aligned}$$

where

$$\hat{z}, \hat{z}^{-1/2}, \hat{z}^{-1/3} \in \mathbb{F}_p$$

.

3.2.5 Pairing over KSS-18 Curve

As described earlier bilinear pairing requires two rational point groups to be mapped to a multiplicative group. In what follows, Optimal-Ate pairing over KSS-18 curve of embedding degree $k = 18$ is described as follows.

3.2.5.1 Ate Pairing

Let us consider the following two additive groups as $\mathbb{G}_1$ and $\mathbb{G}_2$ and multiplicative group as $\mathbb{G}_3$. The Ate pairing α is defined as follows:

$$\begin{aligned} \mathbb{G}_1 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]). \end{aligned}$$

$$\alpha : \mathbb{G}_2 \times \mathbb{G}_1 \longrightarrow \mathbb{F}'_{p^k} / (\mathbb{F}^*_{p^k})^r. \tag{3.6}$$

where $\mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ in the case of KSS-18 curve.

Let $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, Ate pairing $\alpha(Q, P)$ is given as follows.

$$\alpha(Q, P) = f_{t-1,Q}(P)^{\frac{p^k-1}{r}}, \tag{3.7}$$

where $f_{t-1,Q}(P)$ symbolize the output of Miller's algorithm. The bilinearity of Ate pairing is satisfied after calculating the final exponentiation. It is noted that improvement of final exponentiation is not the focus of this chapter. Several works [STO06; Sco+09] have been already done for efficient final exponentiation.

3.2.5.2 Optimal-Ate Pairing

The previous work of Aranha et al. [Ara+13] has mentioned about the relation $t(u) - 1 \equiv u + 3p(u) \bmod r(u)$ for Optimal-Ate pairing. Exploiting the relation, Optimal-Ate pairing on the KSS-18 curve is defined by the following representation.

$$(Q, P) = (f_{u,Q} \cdot f_{3,Q}^{p} \cdot l_{[u]Q,[3p]Q})^{\frac{p^{18}-1}{r}}, \tag{3.8}$$

where u is the mother parameter. The calculation procedure of Optimal-Ate pairing is shown in **Algorithm**. 4. In what follows, the calculation steps from 1 to 5 shown in **Algorithm**. 4 is identified as Miller's loop. Step 3 and 5 are line evaluation along with elliptic curve doubling and addition. These two steps are key steps to accelerate the loop calculation. As an acceleration technique *pseudo 12-sparse multiplication* is proposed in this chapter.

3.2.6 Sparse multiplication

In the previous work, Mori et al. [Mor+14] has substantiated the pseudo 8-sparse multiplication for BN curve. Adapting affine coordinates for representing rational points, we can apply Mori's work in the case of KSS-18 curve. The doubling phase and addition phase in Miller's loop can be carried out efficiently by the following calculations. Let $P = (x_P, y_P)$, $T = (x, y)$ and $Q = (x_2, y_2) \in E'(\mathbb{F}_{p^3})$ be given in affine coordinates, and let $T + Q = (x_3, y_3)$ be the sum of T and Q.

3.2.6.1 Step 3: Elliptic curve doubling phase ($T = Q$)

$$\begin{gathered} A = \frac{1}{2y}, B = 3x^2, C = AB, D = 2x, x_3 = C^2 - D, \\ E = Cx - y, y_3 = E - Cx_3, F = C\overline{x}_P, \\ l_{T,T}(P) = y_P + Ev + F\theta = y_P + Ev - Cx_P\theta, \end{gathered} \tag{3.9}$$

where $\overline{x}_P = -x_P$ will be pre-computed. Here $l_{T,T}(P)$ denotes the tangent line at the point T.

3.2.6.2 Step 5: Elliptic curve addition phase ($T \neq Q$)

$$\begin{aligned}
A = \tfrac{1}{x_2 - x}, B = y_2 - y, C = AB, D = x + x_2, x_3 = C^2 - D,\\
E = Cx - y, y_3 = E - Cx_3, F = C\overline{x}_P,\\
l_{T,Q}(P) = y_P + Ev + F\theta = y_P + Ev - Cx_P\theta,
\end{aligned} \tag{3.10}$$

where $\overline{x}_P = -x_P$ will be pre-computed. Here $l_{T,Q}(P)$ denotes the tangent line between the point T and Q.

Analyzing Eq.(9.14) and Eq.(9.16), we get that E and Cx_P are calculated in $\mathbb{F}_{p^3}$. After that, the basis element 1, v and θ identifies the position of y_P, E and Cx_P in $\mathbb{F}_{p^{18}}$ vector representation. Therefore vector representation of $l_{\psi_6(T),\psi_6(T)}(P) \in \mathbb{F}_{p^{18}}$ consists of 18 coefficients. Among them at least 11 coefficients are equal to zero. In the other words, only 7 coefficients $y_P \in \mathbb{F}_p$, $Cx_P \in \mathbb{F}_{p^3}$ and $E \in \mathbb{F}_{p^3}$ are perhaps to be non-zero. $l_{\psi_6(T),\psi_6(Q)}(P) \in \mathbb{F}_{p^{18}}$ also has the same vector structure. Thus, the calculation of multiplying $l_{\psi_6(T),\psi_6(T)}(P) \in \mathbb{F}_{p^{18}}$ or $l_{\psi_6(T),\psi_6(Q)}(P) \in \mathbb{F}_{p^{18}}$ is called sparse multiplication. In the above mentioned instance especially called 11-sparse multiplication. This sparse multiplication accelerates Miller's loop calculation as shown in **Algorithm**.4. This chapter comes up with pseudo 12-sparse multiplication.

Algorithm 4: Optimal-Ate pairing on KSS-18 curve

```
Input: $u, P \in \mathbb{G}_1, Q \in \mathbb{G}_2'$
Output: $(Q, P)$
$f \leftarrow 1, T \leftarrow Q$
for $i = \lfloor \log_2(u) \rfloor$ downto 1 do
    $f \leftarrow f^2 \cdot l_{T,T}(P), T \leftarrow [2]T$
    if $u[i] = 1$ then
        $f \leftarrow f \cdot l_{T,Q}(P), T \leftarrow T + Q$
$f_1 \leftarrow f_{3,Q}^p, f \leftarrow f \cdot f_1$
$Q_1 \leftarrow [u]Q, Q_2 \leftarrow [3p]Q$
$f \leftarrow f \cdot l_{Q_1,Q_2}(P)$
$f \leftarrow f^{\frac{p^{18}-1}{r}}$
return $f$
```

3.3 Improved Optimal-Ate Pairing for KSS-18 Curve

In this section we describe the main proposal. Before going to the details, at first we give an overview of the improvement procedure of Optimal-Ate pairing in KSS-18 curve. The following two ideas are proposed in order to efficiently apply 12-sparse multiplication on Optimal-Ate pairing on KSS-18 curve.

1. In Eq.(9.14) and Eq.(9.16) among the 7 non-zero coefficients, one of the non-zero coefficients is $y_P \in \mathbb{F}_p$. And y_P remains uniform through Miller's loop calculation. Thereby dividing both sides of those Eq.(9.14) and Eq.(9.16) by y_P, the coefficient becomes 1 which results in a more efficient sparse multiplication by $l_{\psi_6(T),\psi_6(T)}(P)$ or $l_{\psi_6(T),\psi_6(Q)}(P)$. This chapter calls it *pseudo 12-sparse multiplication*.
2. Division by y_P in Eq.(9.14) and Eq.(9.16) causes a calculation overhead for the other non-zero coefficients in the Miller's loop. To cancel this additional cost in Miller's loop, the map introduced in Eq.(9.18) is applied.

It is to be noted that this chapter doesn't focus on making final exponentiation efficient in Miller's algorithm since many efficient algorithms are available. From Eq.(9.14) and Eq.(9.16) the above mentioned ideas are introduced in details.

3.3.1 Pseudo 12-sparse Multiplication

As said before y_P shown in Eq.(9.14) is a non-zero elements in $\mathbb{F}_p$. Thereby, dividing both sides of Eq.(9.14) by y_P we obtain as follows:

$$y_P^{-1} l_{T,T}(P) = 1 + E y_P^{-1} v - C(x_P y_P^{-1})\theta. \tag{3.11}$$

Replacing $l_{T,T}(P)$ by the above $y_P^{-1} l_{T,T}(P)$, the calculation result of the pairing does not change, since $final\ exponentiation$ cancels $y_P^{-1} \in \mathbb{F}_p$. One of the non-zero coefficients becomes 1 after the division by y_P, which results in more efficient vector multiplications in Miller's loop. This chapter calls it $pseudo\ 12 - sparse\ multiplication$. **Algorithm**. 5 introduces the detailed calculation procedure of pseudo 12-sparse multiplication.

Algorithm 5: Pseudo 12-sparse multiplication

```
Input: $a, b \in \mathbb{F}_{p^{18}}$
$a = (a_0 + a_1\theta + a_2\theta^2) + (a_3 + a_4\theta + a_5\theta^2)v,\ b = 1 + b_1\theta + b_3 v$
where $a_i, b_j, c_i \in \mathbb{F}_{p^3} (i = 0, \cdots, 5, j = 1, 3)$
Output: $c = ab = (c_0 + c_1\theta + c_2\theta^2) + (c_3 + c_4\theta + c_5\theta^2)v \in \mathbb{F}_{p^{18}}$
$c_1 \leftarrow a_0 \times b_1, c_5 \leftarrow a_2 \times b_3, t_0 \leftarrow a_0 + a_2, S_0 \leftarrow b_1 + b_3$
$c_3 \leftarrow t_0 \times S_0 - (c_1 + c_5)$
$c_2 \leftarrow a_1 \times b_1, c_6 \leftarrow a_3 \times b_3, t_0 \leftarrow a_1 + a_3$
$c_4 \leftarrow t_0 \times S_0 - (c_2 + c_6)$
$c_5 \leftarrow c_5 + a_4 \times b_1, c_6 \leftarrow c_6 + a_5 \times b_1$
$c_7 \leftarrow a_4 \times b_3, c_8 \leftarrow a_5 \times b_3$
$c_0 \leftarrow c_6 \times i$
$c_1 \leftarrow c_1 + c_7 \times i$
$c_2 \leftarrow c_2 + c_8 \times i$
$c \leftarrow c + a$
return $c = (c_0 + c_1\theta + c_2\theta^2) + (c_3 + c_4\theta + c_5\theta^2)v$
```

3.3.2 Line Calculation in Miller's Loop

The comparison of Eq.(9.14) and Eq.(3.11) shows that the calculation cost of Eq.(3.11) is little bit higher than Eq.(9.14) for Ey_P^{-1}. The cancellation process of $x_P y_P^{-1}$ terms by utilizing isomorphic mapping is introduced next. The $x_P y_P^{-1}$ and y_P^{-1} terms are pre-computed to reduce execution time complexity. The map introduced in Eq.(9.18) can find a certain isomorphic rational point $\hat{P}(x_{\hat{P}}, y_{\hat{P}}) \in \hat{E}(\mathbb{F}_p)$ such that

$$x_{\hat{P}} y_{\hat{P}}^{-1} = 1. \tag{3.12}$$

Here the twist parameter z of Eq.(12.10) is considered to be $\hat{z} = (x_P y_P^{-1})^6$ of Eq.(9.18), where $\hat{z}$ is a quadratic and cubic residue in $\mathbb{F}_p$ and $\hat{E}$ denotes the KSS-18 curve defined by Eq.(9.18). From the isomorphic mapping Eq.(12.10), such z is obtained by solving the following equation considering the input $P(x_P, y_P)$.

$$z^{1/3} x_P = z^{1/2} y_P, \tag{3.13}$$

Afterwards the $\hat{P}(x_{\hat{P}}, y_{\hat{P}}) \in \hat{E}(\mathbb{F}_p)$ is given as

$$\hat{P}(x_{\hat{P}}, y_{\hat{P}}) = (x_P^3 y_P^{-2}, x_P^3 y_P^{-2}). \tag{3.14}$$

As the x and y coordinates of $\hat{P}$ are the same, $x_{\hat{P}} y_{\hat{P}}^{-1} = 1$. Therefore, corresponding to the map introduced in Eq.(9.18), first mapping not only P to $\hat{P}$ shown above but also Q to $\hat{Q}$ shown below.

$$\hat{Q}(x_{\hat{Q}}, y_{\hat{Q}}) = (x_P^2 y_P^{-2} x_Q, x_P^3 y_P^{-3} y_Q). \tag{3.15}$$

When we define a new variable $L = (x_P^{-3} y_P^2) = y_{\hat{P}}^{-1}$, the line evaluations, Eq.(9.14) and Eq.(9.16) become the following calculations. In what follows, let $\hat{P} = (x_{\hat{P}}, y_{\hat{P}}) \in E(\mathbb{F}_p)$, $T = (x, y)$ and $Q = (x_2, y_2) \in E'(\mathbb{F}_{p^3})$ be given in affine coordinates and let $T + Q = (x_3, y_3)$ be the sum of T and Q.

3.3.2.1 Step 3: Doubling Phase ($T = Q$)

$$\begin{gathered} A = \tfrac{1}{2y}, B = 3x^2, C = AB, D = 2x, x_3 = C^2 - D, \\ E = Cx - y, y_3 = E - Cx_3, \\ \hat{l}_{T,T}(P) = y_P^{-1} l_{T,T}(P) = 1 + ELv - C\theta, \end{gathered} \tag{3.16}$$

where $L = y_{\hat{P}}^{-1}$ will be pre-computed.

3.3.2.2 Step 5: Addition Phase ($T \neq Q$)

$$\begin{gathered} A = \tfrac{1}{x_2 - x}, B = y_2 - y, C = AB, D = x + x_2, x_3 = C^2 - D, \\ E = Cx - y, y_3 = E - Cx_3, \\ \hat{l}_{T,Q}(P) = y_P^{-1} l_{T,Q}(P) = 1 + ELv - C\theta, \end{gathered} \tag{3.17}$$

where $L = y_{\hat{P}}^{-1}$ will be pre-computed.

As we compare the above equation with to Eq.(9.14) and Eq.(9.16), the third term of the right-hand side becomes simple since $x_{\hat{P}} y_{\hat{P}}^{-1} = 1$.
In the above procedure, calculating $\hat{P}$, $\hat{Q}$ and L by utilizing x_P^{-1} and y_P^{-1} will create some computational overhead. In spite of that, calculation becomes efficient as it is performed in isomorphic group together with pseudo 12-sparse multiplication in the Miller's loop. Improvement of Miller's loop calculation is presented by experimental results in the next section.

3.4 Cost Evaluation and Experimental Result

This section shows some experimental results with evaluating the calculation costs in order to the signify efficiency of the proposal. It is to be noted here that in the following discussions "Previous method" means Optimal-Ate pairing with no use the sparse multiplication, "11-sparse multiplication" means Optimal-Ate pairing with 11-sparse multiplication and "Proposed method" means Optimal-Ate pairing with Pseudo 12-sparse multiplication.

3.4.1 Parameter Settings and Computational Environment

In the experimental simulation, this chapter has considered the 192 bit security level for KSS-18 curve. Table 3.1 shows the parameters settings suggested in [Ara+13] for 192 bit security over KSS-18 curve. However this parameter settings does not necessarily comply with the recent suggestion of key size by Kim et al. [KB16] for 192 bit security level. The sole purpose to use this parameter settings in this chapter is to compare the literature with the experimental result.

TABLE 3.1: Parameters for Optimal-Ate pairing over KSS-18 curve.

Security level	u	$p(u)$ [bit]	c Eq.(3.3)	b Eq.(3.1)
192-bit	$-2^{64} - 2^{51} + 2^{46} + 2^{12}$	508	2	2

To evaluate the operational cost and to compare the execution time of the proposal based on the recommended parameter settings, the following computational environment is considered. Table 3.2 shows the computational environment.

3.4.2 Cost Evaluation

Let us consider m, s, a and i to denote the times of multiplication, squaring, addition and inversion $\in \mathbb{F}_p$. Similarly, $\tilde{m}, \tilde{s}, \tilde{a}$ and $\tilde{i}$ denote the number of multiplication, squaring, addition and inversion $\in \mathbb{F}_{p^3}$ and $\hat{m}, \hat{s}, \hat{a}$ and $\hat{i}$ to denote the count of multiplication, squaring, addition and inversion $\in \mathbb{F}_{p^{18}}$

TABLE 3.2: Computing environment of Optimal-Ate pairing over KSS-18 curve.

CPU	Core i5 6600
Memory	8.00GB
OS	Ubuntu 16.04 LTS
Library	GMP 6.1.0 [Gt15]
Compiler	gcc 5.4.0
Programming language	C

respectively. Table 3.3 and Table 3.4 show the calculation costs with respect to operation count.

TABLE 3.3: Operation count of line evaluation.

$E(\mathbb{F}_{p^{18}})$ Operations	Previous method	11-sparse multiplication	Proposed method
Precomputation	-	$\tilde{a}$	$6\tilde{m} + 2\tilde{i}$
Doubling + $l_{T,T}(P)$	$9\hat{a} + 6\hat{m} + 1\hat{i}$	$7\tilde{a} + 6\tilde{m} + 1\tilde{i}$	$7\tilde{a} + 6\tilde{m} + 1\tilde{i}$
Addition + $l_{T,Q}(P)$	$8\hat{a} + 5\hat{m} + 1\hat{i}$	$6\tilde{a} + 5\tilde{m} + 1\tilde{i}$	$6\tilde{a} + 5\tilde{m} + 1\tilde{i}$

TABLE 3.4: Operation count of multiplication.

$\mathbb{F}_{p^{18}}$ Operations	Previous method	11-sparse multiplication	Proposed method
Vector Multiplication	$30\tilde{a} + 18\tilde{m} + 8a$	$1\hat{a} + 11\tilde{a} + 10\tilde{m} + 3a + \mathbf{18m}$	$1\hat{a} + 11\tilde{a} + 10\tilde{m} + 3a$

By analyzing the Table 3.4 we can find that 11-sparse multiplication requires 18 more multiplication in $\mathbb{F}_p$ than pseudo 12-sparse multiplication.

3.4.3 Experimental Result

Table 3.5 shows the calculation times of Optimal-Ate pairing respectively. In this execution time count, the time required for final exponentiation is excluded. The results (time count) are the averages of 10000 iterations on PC respectively. According to the experimental results, pseudo 12-sparse contributes to a few percent acceleration of 11-sparse.

3.5 Contribution Summary

Acceleration of a pairing calculation of an Ate-based pairing such as Optimal-Ate pairing depends not only on the optimization of Miller algorithm's loop

TABLE 3.5: Calculation time of Optimal-Ate pairing at the 192-bit security level.

Operation	Previous method	11-sparse multiplication	Proposed method
Doubling+ $l_{T,T}(P)$ [μs]	681	44	44
Addition+ $l_{T,Q}(P)$ [μs]	669	39	37
Multiplication [μs]	119	74	65
Miller's Algorithm [ms]	524	142	140

parameter but also on efficient elliptic curve arithmetic operation and efficient final exponentiation. This chapter has proposed a *pseudo 12-sparse multiplication* to accelerate Miller's loop calculation in KSS-18 curve by utilizing the property of rational point groups. In addition, this chapter has shown an enhancement of the elliptic curve addition and doubling calculation in Miller's algorithm by applying implicit mapping of its sextic twisted isomorphic group. Moreover this chapter has implemented the proposal with recommended security parameter settings for KSS-18 curve at 192 bit security level. The simulation result shows that the proposed *pseudo 12-sparse multiplication* gives more efficient Miller's loop calculation of an Optimal-Ate pairing operation along with recommended parameters than pairing calculation without sparse multiplication.

3.6 Conclusion

This chapter has proposed pseudo 12-sparse multiplication for accelerating Optimal-Ate pairing on KSS-18 curve. According to the calculation costs and experimental results shown in this chapter, the proposed method can calculate Optimal-Ate pairing more efficiently.

Chapter 4

Improved $\mathbb{G}_2$ Scalar Multiplication over KSS-18 Curve

4.1 Introduction

4.1.1 Background and Motivation

Recall that, pairing based cryptography has attracted many researchers since Sakai et al. [SK03] and Joux et al. [Jou04] independently proposed a cryptosystem based on elliptic curve pairing. This has encouraged to invent several innovative pairing based cryptographic applications such as broadcast encryption [BGW05] and group signature authentication [BBS04], that has increased the popularity of pairing based cryptographic research.

However, using pairing based cryptosytem in industrial state is still restricted by its expensive operational cost with respect to time and computational resources in practical case. In order to make it practical, several pairing techniques such as Ate [Coh+05], Optimal-ate [Ver10], twisted Ate [Mat+07], χ-Ate [Nog+08] and *subfield twisted* Ate [DSD07] pairings have gained much attention since they have achieved quite efficient pairing calculation in certain pairing friendly curve. Researchers still continues on finding efficient way to implement pairing to make it practical enough for industrial standardization.

In such consequences, this chapter focuses on a peripheral technique of Ate-based pairings that is scalar multiplication defined over Kachisa-Schaefer-Scott (KSS) curve [KSS07] of embedding degree 18. Scalar multiplication over higher degree rational point groups is often regarded as the bottleneck for faster pairing based cryptography.

As aforementioned, pairing is a bilinear map of two rational point groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a multiplicative group $\mathbb{G}_3$ [SCA86]. The typical notation of pairing is $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$. In Ate-based pairing, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as:

$$\begin{aligned}
\mathbb{G}_1 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [1]), \\
\mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\
\mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r,
\end{aligned}$$

$$\alpha : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3,$$

where α denotes Ate pairing. Pairings are often defined over certain extension field $\mathbb{F}_{p^k}$, where p is the prime number, also know as characteristics and k is the minimum extension degree for pairing also called *embedding* degree. The set of rational points $E(\mathbb{F}_{p^k})$ are defined over a certain pairing friendly curve of embedded extension field of degree k. This chapter has considered Kachisa-Schaefer-Scott (KSS) [KSS07] pairing friendly curves of emebdding degree $k = 18$ described in [FST06].

4.1.2 Related Works

There are several works [Nog+09][Sak+08] on efficiently computing scalar multiplication defined over Barreto-Naehrig[BN06] curve along with efficient extension field arithmetic [BP98]. This chapter focuses on scalar multiplication on KSS-18 curve.

4.1.3 Contribution Outline

Scalar multiplication is often considered to be one of the most time consuming operation in cryptographic scene. Efficient scalar multiplication is one of the important factors for making the pairing practical over KSS-18 curve.

This chapter focuses on efficiently performing scalar multiplication on rational points defined over rational point group $\mathbb{G}_2$ by scalar s, since scalar multiplication is required repeatedly in cryptographic calculation. However in asymmetric pairing such as Ate-based pairing, scalar multiplication of $\mathbb{G}_2$ rational points is important as no mapping function is explicitly given between $\mathbb{G}_1$ to $\mathbb{G}_2$. By the way, as shown in the definition, $\mathbb{G}_1$ is a set of rational points defined over prime field and there are several researches [Sak+08] for efficient scalar multiplication in $\mathbb{G}_1$.

The common approach to accelerate scalar multiplication are log-step algorithm such as binary and non-adjacent form (NAF) methods, but more efficient approach is to use Frobenius mapping in the case of $\mathbb{G}_2$ that is defined over $\mathbb{F}_{p^k}$. Moreover when sextic twist of the pairing friendly curve exists, then we apply skew Frobenius map on the isomorphic sextic-twisted subfield rational points. Such technique will reduce the computational cost in a great extent.

In this chapter we have exploited the sextic twisted property of KSS-18 curve and utilized skew Frobenius map to reduce the computational time of scalar multiplication on $\mathbb{G}_2$ rational point. Utilizing the relation $z \equiv -3p + p^4 \bmod r$,[1] derived by Aranha et al,[Ara+13] and the properties of $\mathbb{G}_2$ rational point, the scalar can be expressed as z-adic representation. Together with skew Frobenius mapping and z-adic representation the scalar multiplication can be further accelerated. We have utilized this relation to construct z-adic

[1] z is the mother parameter of KSS-18 curve and z is about six times smaller than the size of order r.

representation of scalar s which is introduced in **Section**. 4.3.4. In addition with Frobenius mapping and z-adic representation of s, we applied the multi-scalar multiplication technique to compute elliptic curve addition in parallel in the proposed scalar multiplication. We have compared our proposed method with three other well studied methods named binary method, sliding-window method and non-adjacent form method. The comparison shows that our proposed method is about 60 times faster than the plain implementations of above mentioned methods in execution time. The comparison also reveals that the proposed method requires more than 5 times less elliptic curve doubling than any of the compared methods.

4.2 Preliminaries

In this section we recall some already introduced preliminaries for comprehensible understanding of the proposal. We will briefly review elliptic curve scalar multiplication. Throughout this chapter, p and k denote characteristic and embedding extension degree, respectively. $\mathbb{F}_{p^k}$ denotes k-th extension field over prime field $\mathbb{F}_p$ and $\mathbb{F}_{p^k}^*$ denotes the multiplicative group in $\mathbb{F}_{p^k}$.

4.2.1 Elliptic Curve

An elliptic curve [Was03] defined over $\mathbb{F}_p$ is generally represented by *affine coordinates* [SCA86] as follows;

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b, \tag{4.1}$$

where $4a^3 + 27b^2 \neq 0$ and $a, b \in \mathbb{F}_p$. A pair of coordinates x and y that satisfy Eq.(4.1) are known as *rational points* on the curve.

4.2.1.1 Elliptic Curve Point Operation

Let $E(\mathbb{F}_p)$ be the set of all rational points on the curve E including the point at infinity $\mathcal{O}$. $\#E(\mathbb{F}_p)$ denotes the order of $E(\mathbb{F}_p)$. Let us consider two rational points using affine coordinates as $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, and their addition $R = P_1 + P_2$, where $R = (x_3, y_3)$ and $P_1, P_2, R \in E(\mathbb{F}_p)$. Then the x and y coordinates of R are calculated as follows:

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, && (4.2a) \\ y_3 &= (x_1 - x_3)\lambda - y_1, && (4.2b) \end{aligned}$$

where λ is given as follows:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}; & P_1 \neq P_2, \\ \frac{3x_1^2 + a}{2y_1}; & P_1 = P_2, \end{cases} \tag{4.2c}$$

λ is the tangent at the point on the curve and $\mathcal{O}$ is the additive unity in $E(\mathbb{F}_p)$. If $P_1 \neq P_2$ then $P_1 + P_2$ is called elliptic curve addition (ECA). If $P_1 = P_2$ then $P_1 + P_2 = 2P_1$, which is known as elliptic curve doubling (ECD).

4.2.1.2 Elliptic Curve Scalar Multiplication

Let scalar s is $0 \leq s < r$, where r is the order of the target rational point group. Scalar multiplication of rational points P_1, denoted as $[s]P_1$ is calculated by $(s-1)$-times additions of P_1 as,

$$[s]P_1 = \sum_{i=0}^{s-1} P_1, \quad 0 \leq s < r, \tag{4.3}$$

When $s = r$, then $[r]P_1 = \mathcal{O}$ where r is the order of the curve. Let $[s]P_1 = P_2$, and value of s is not obtained, then the solving s from P_1 and P_2 is known as elliptic curve discrete logarithm problem (ECDLP). The difficulty level of solving ECDLP defines the security strength of elliptic curve cryptography.

4.2.2 KSS Curve of Embedding Degree $k = 18$

In [KSS07], Kachisa, Schaefer, and Scott proposed a family of non supersingular Brezing-Weng pairing friendly elliptic curves using elements in the cyclotomic field. In what follows this chapter considers the KSS curve of embedding degree $k = 18$ since it holds *sextic twist*. The equation of KSS curve defined over $\mathbb{F}_{p^{18}}$ is given as follows:

$$E : Y^2 = X^3 + b, \quad (b \in \mathbb{F}_p), \tag{4.4}$$

where $b \neq 0$ and $X, Y \in \mathbb{F}_{p^{18}}$. Its characteristic p, Frobenius trace t and order r are given systematically by using an integer variable z as follows:

$$\begin{aligned} p(z) &= (z^8 + 5z^7 + 7z^6 + 37z^5 + 188z^4 + 259z^3 \\ &\quad +343z^2 + 1763z + 2401)/21, && (4.5a) \\ r(z) &= (z^6 + 37z^3 + 343)/343, && (4.5b) \\ t(z) &= (z^4 + 16z + 7)/7, && (4.5c) \end{aligned}$$

where z is such that $z \equiv 14 \pmod{42}$ and the ρ value is $\rho = (\log_2 p / \log_2 r) \approx 1.33$.

In some previous work of Aranha et al. [Ara+13] and Scott et al. [Sco11] has mentioned that the size of the characteristics p to be 508 to 511-bit with order r of 384-bit for 192-bit security level. Therefore this chapter used parameter settings according to the suggestion of [Ara+13] for 192 bit security on KSS-18 curve in the simulation implementation. In the recent work, Kim et al. [KB16] has suggested to update the key sizes in pairing-based cryptography due to the development of new discrete logarithm problem over finite field. The parameter settings used in this chapter doesn't completely end up at the 192 bit security level according to [KB16]. However the parameter settings used in this chapter in order to show the resemblance of the proposal with the experimental result.

4.2.3 $\mathbb{F}_{p^{18}}$ Extension Field Arithmetic

Pairing based cryptography requires to perform arithmetic operation in extension fields of degree $k \geq 6$[SCA86]. We recall **Section**. 3.2.2 of **Chapter**. 3 for $\mathbb{F}_{p^{18}}$ construction.

Let $(p-1)$ is divisible by 3 and c is a quadratic and cubic non residue in $\mathbb{F}_p$. In KSS curve [KSS07], where $k = 18$, $\mathbb{F}_{p^{18}}$ is constructed with irreducible binomials by the following towering scheme.

$$\begin{cases} \mathbb{F}_{p^3} = \mathbb{F}_p[i]/(i^3 - c), \text{where } c = 2 \text{ is the best choice,} \\ \mathbb{F}_{p^6} = \mathbb{F}_{p^3}[v]/(v^2 - i), \\ \mathbb{F}_{p^{18}} = \mathbb{F}_{p^6}[\theta]/(\theta^3 - v). \end{cases}$$

where the base extension field is $\mathbb{F}_{p^3}$ for the *sextic twist* of KSS-18 curve.

4.2.4 Frobenius Mapping of Rational Points in $E(\mathbb{F}_{p^{18}})$

Let (x, y) be certain rational point in $E(\mathbb{F}_{p^{18}})$. Frobenius map $\pi_p : (x, y) \mapsto (x^p, y^p)$ is the p-th power of the rational point defined over $\mathbb{F}_{p^{18}}$. Sakemi et al. [Sak+08] showed an efficient scalar multiplication by applying skew Frobenius mapping in the context of Ate-based pairing in BN curve of embedding degree $k = 12$. In this chapter we have utilized skew Frobenius mapping technique for efficient scalar multiplication for the KSS-18 curve.

4.2.5 Sextic Twist of KSS-18 Curve

Let the embedding degree $k = 6e$, where e is positive integer, *sextic* twist is given as follows:

$$E : \quad y^2 = x^3 + b, \quad b \in \mathbb{F}_p, \tag{4.6}$$

$$E'_6 : \quad y^2 = x^3 + bu^{-1}, \tag{4.7}$$

where u is a quadratic and cubic non residue in $E(\mathbb{F}_{p^e})$ and $3|(p^e - 1)$. Isomorphism between $E'_6(\mathbb{F}_{p^e})$ and $E(\mathbb{F}_{p^{6e}})$, is given as follows:

$$\psi_6 : \begin{cases} E'_6(\mathbb{F}_{p^e}) \longrightarrow E(\mathbb{F}_{p^{6e}}), \\ (x, y) \quad \mapsto (xu^{1/2}, yu^{1/2}). \end{cases} \tag{4.8}$$

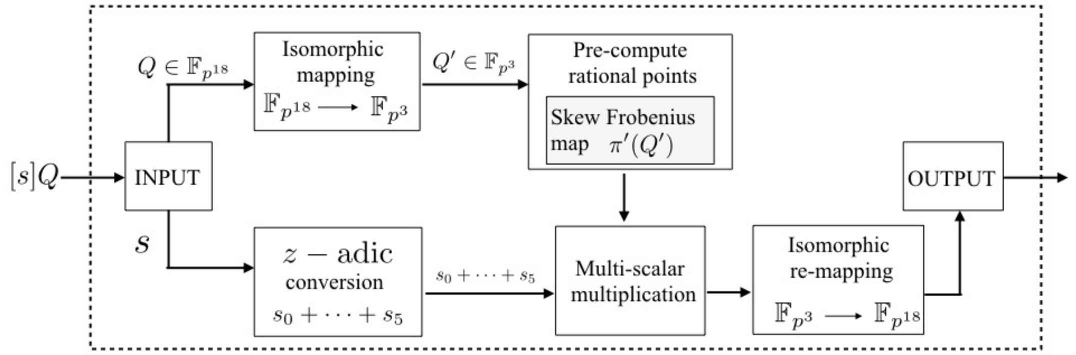
In context of Ate-based pairing for KSS curve of embedding degree 18, sextic twist is considered to be the most efficient.

4.3 Improved Scalar Multiplication for $\mathbb{G}_2$ rational point

This section will introduce the proposal for efficient scalar multiplication of $\mathbb{G}_2$ rational points defined over KSS curve of embedding degree $k = 18$ in context of Ate-based pairing. An overview the proposed method is given next before diving into the detailed procedure.

4.3.1 Overview of the Proposal

Figure 4.1 shows an overview of overall process of proposed scalar multiplication. Rational point groups $\mathbb{G}_1$, $\mathbb{G}_2$ and multiplicative group $\mathbb{G}_3$ groups will be defined at the beginning. Then a rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ will be calculated. Q has a special vector representation with 18 $\mathbb{F}_p$ elements for each coordinates. A random scalar s will be considered for scalar multiplication of $[s]Q$ which is denoted as input in Figure 4.1. After that we will consider an isomorphic map of rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ to its sextic twisted rational point $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$. At the same time we will obtain the z-adic representation of the scalar s. Next the some rational points defined over $E'(\mathbb{F}_{p^3})$ will be pre-computed by applying the skew Frobenius mapping. After that a multi-scalar multiplication technique will be applied to calculate the scalar multiplication in parallel. The result of this scalar multiplication will be defined over $\mathbb{F}_{p^3}$. Finally the result of the multi-scalar multiplication will be re-mapped to rational point in $E(\mathbb{F}_{p^{18}})$ to get the final result.


$Q \in \mathbb{F}_{p^{18}}$
Isomorphic mapping
$\mathbb{F}_{p^{18}} \longrightarrow \mathbb{F}_{p^3}$
$Q' \in \mathbb{F}_{p^3}$
Pre-compute rational points
Skew Frobenius map $\pi'(Q')$
$[s]Q$
INPUT
OUTPUT
s
$z-$ adic conversion
$s_0 + \cdots + s_5$
$s_0 + \cdots + s_5$
Multi-scalar multiplication
Isomorphic re-mapping
$\mathbb{F}_{p^3} \longrightarrow \mathbb{F}_{p^{18}}$

* s is a random scalar of size about 376-bit in the experiment

FIGURE 4.1: Overview of the proposed scalar multiplication for KSS-18 Curve.

4.3.2 $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ groups

In the context of pairing-based cryptography, especially on KSS-18 curve, three groups $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_3$ are considered. From [Mor+14], we define $\mathbb{G}_1$,

$\mathbb{G}_2$ and $\mathbb{G}_3$ as follows:

$$\begin{aligned}
\mathbb{G}_1 &= E(\mathbb{F}_{p^{18}})[r] \cap \mathrm{Ker}(\pi_p - [1]), \\
\mathbb{G}_2 &= E(\mathbb{F}_{p^{18}})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\
\mathbb{G}_3 &= \mathbb{F}^*_{p^{18}}/(\mathbb{F}^*_{p^{18}})^r,
\end{aligned}$$

$$\alpha : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3, \tag{4.9}$$

where α denotes Ate pairing. In the case of KSS-18 curve, $\mathbb{G}_1, \mathbb{G}_2$ are rational point groups and $\mathbb{G}_3$ is the multiplicative group in $\mathbb{F}_{p^{18}}$. They have the same order r.

In context of KSS-18 curve, let us consider a rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ where Q satisfies the following relations,

$$\begin{aligned}
\left[p+1-t\right]Q &= \mathcal{O}, \\
\left[t-1\right]Q &= \left[p\right]Q.
\end{aligned} \tag{4.10}$$

$$\begin{aligned}
[\pi_p - p]Q &= \mathcal{O}, \\
\pi_p(Q) &= [p]Q.
\end{aligned} \tag{4.11}$$

where $[t-1]Q = \pi_p(Q)$, by substituting $[p]Q$ in Eq.(4.10).

4.3.3 Isomorphic Mapping between Q and Q'

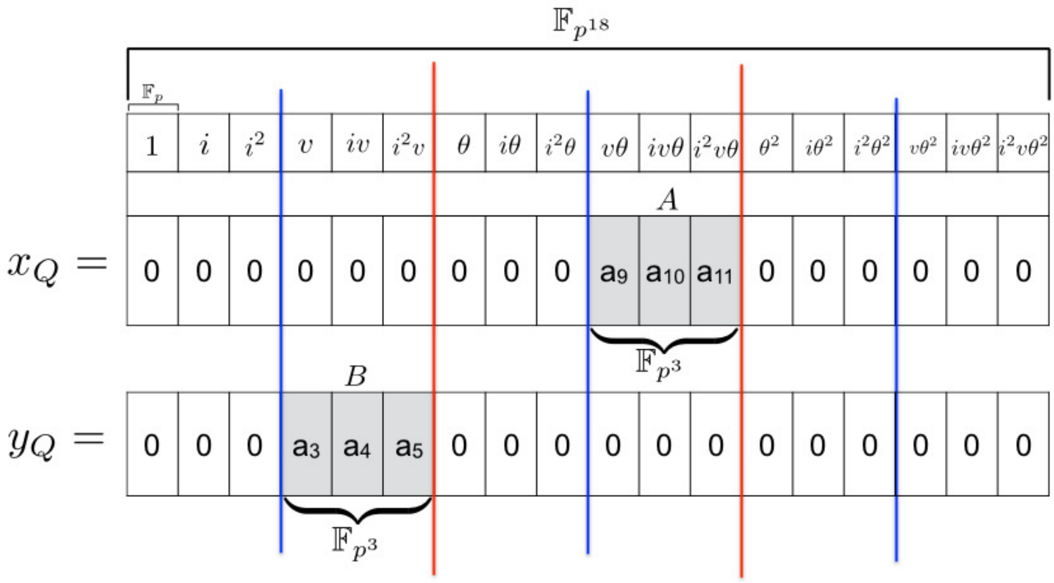
Let us consider E is the KSS-18 curve in base field $\mathbb{F}_{p^3}$ and E' is sextic twist of E given as follows:

$$E : y^2 = x^3 + b, \tag{4.12}$$
$$E' : y^2 = x^3 + bi, \tag{4.13}$$

where $b \in \mathbb{F}_p$; $x, y, i \in \mathbb{F}_{p^3}$ and basis element i is the quadratic and cubic non residue in $\mathbb{F}_{p^3}$.

Rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ has a special vector representation with 18 $\mathbb{F}_p$ elements for each x_Q and y_Q coordinates. Figure 4.2 shows the structure of the coefficients of $Q \in \mathbb{F}_{p^{18}}$ and its sextic twisted isomorphic rational point $Q' \in \mathbb{F}_{p^3}$ in KSS-18 curve. Among 18 elements, there are 3 continuous nonzero $\mathbb{F}_p$ elements which belongs to a $\mathbb{F}_{p^3}$ element. The other coefficients are zero. In this chapter, considering parameter settings given in Table 4.2 of section 4; Q is given as $Q = (Av\theta, Bv)$, showed in Figure 4.2, where $A, B \in \mathbb{F}_{p^3}$ and v and θ are the basis elements of $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{18}}$ respectively.

Let us consider the sextic twisted isomorphic subfield rational point of Q as $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$ and x' and y' as the coordinates of Q'.


$\mathbb{F}_{p^{18}}$
$\mathbb{F}_p$
1 i i^2 v iv i^2v θ $i\theta$ $i^2\theta$ $v\theta$ $iv\theta$ $i^2v\theta$ θ^2 $i\theta^2$ $i^2\theta^2$ $v\theta^2$ $iv\theta^2$ $i^2v\theta^2$
A
$x_Q =$
0 0 0 0 0 0 0 0 0 a9 a10 a11 0 0 0 0 0 0
$\mathbb{F}_{p^3}$
B
$y_Q =$
0 0 0 a3 a4 a5 0 0 0 0 0 0 0 0 0 0 0 0
$\mathbb{F}_{p^3}$

$$a_j \in \mathbb{F}_p, \quad \text{where} \quad a_j = (0, 1, \cdots, 17)$$
$$Q = (x_Q, y_Q) = (Av\theta, Bv) \in \mathbb{F}_{p^{18}}$$
$$Q' = (x'_Q, y'_Q) = (Ai, Bi) \in \mathbb{F}_{p^3}$$

FIGURE 4.2: $Q \in \mathbb{F}_{p^{18}}$ and its sextic twisted isomorphic rational point $Q' \in \mathbb{F}_{p^3}$ structure in KSS-18 curve.

4.3.3.1 Mapping $Q = (Av\theta, Bv)$ to the rational point $Q' = (x', y')$

Let's multiply θ^{-6} with both side of Eq.(4.13), where $i = \theta^6$ and $v = \theta^3$.

$$E' : \left(\frac{y}{\theta^3}\right)^2 = \left(\frac{x}{\theta^2}\right)^3 + b. \tag{4.14}$$

Now θ^{-2} and θ^{-3} of Eq.(4.14) can be represented as follows:

$$\theta^{-2} = i^{-1}\theta^4, \tag{4.15a}$$
$$\theta^{-3} = i^{-1}\theta^3. \tag{4.15b}$$

Let us represent $Q = (Av\theta, Bv)$ as follows:

$$Q = (A\theta^4, B\theta^3), \quad \text{where } v = \theta^3. \tag{4.16}$$

From Eq.(4.15a) and Eq.(4.15b) $\theta^4 = i\theta^{-2}$ and $\theta^3 = i\theta^{-3}$ is substituted in Eq.(4.16) as follows:

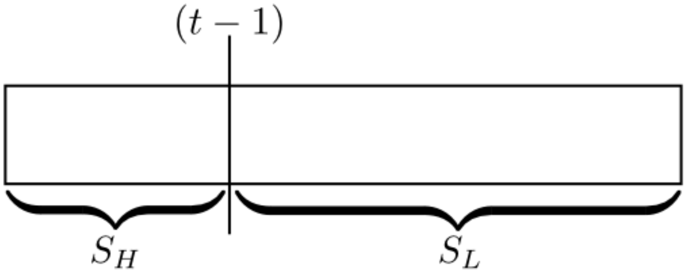
$$Q = (Ai\theta^{-2}, Bi\theta^{-3}), \tag{4.17}$$

where $Ai = x'$ and $Bi = y'$ are the coordinates of $Q' = (x', y') \in \mathbb{F}_{p^3}$. From the structure of $\mathbb{F}_{p^{18}}$, given in 4.2.3, this mapping has required no expensive arithmetic operation. Multiplication by the basis element i in $\mathbb{F}_{p^3}$ can be done by 1 bit wise left shifting since $c = 2$ is considered for towering in 4.2.3.

4.3.4 z-adic Representation of Scalar s

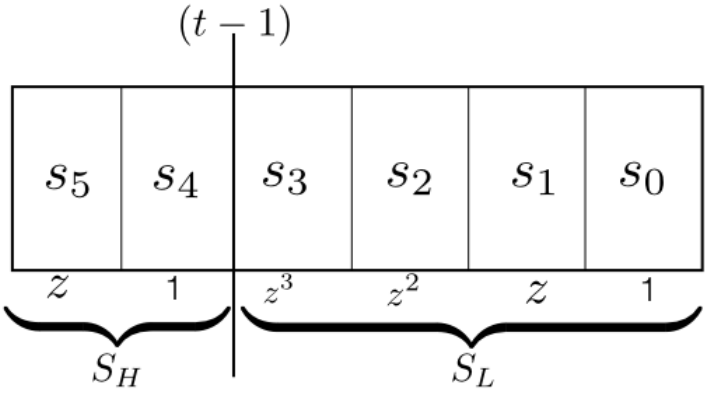
In context of KSS-18 curve, properties of Q will be obtained to define the Eq.(4.11) relation. Next, a random scalar s will be considered for scalar multiplication of $[s]Q$. Then $(t-1)$-adic representation of s will be considered as Figure 4.3. Here s will be divided into two smaller coefficients S_H, S_L where S_L denotes lower bits of s, will be nearly equal to the size of $(t-1)$. On the other hand the higher order bits S_H will be the half of the size of $(t-1)$. Next, z-adic representation of S_H and S_L will be considered. Figure 4.4, shows the z-adic representation from where we find that scalar s is divided into 6 coefficients of z, where the size of z is about 1/4 of that of $(t-1)$ according to Eq.(4.5c).

Figure 4.3 shows $(t-1)$-adic representation of scalar s.



$$s = S_H(t-1) + S_L$$

FIGURE 4.3: $(t-1)$ -adic representation of scalar s.



$$s = S_H(t-1) + S_L = (s_5z + s_4)(t-1) + (s_3z^3 + s_2z^2 + s_1z + s_0)$$

FIGURE 4.4: z-adic and $(t-1)$-adic representation of scalar s.

Figure 4.4 shows the z-adic representation of scalar s. In the previous work on Optimal-Ate pairing, Aranha et al. [Ara+13] derived a relation from the parameter setting of KSS-18 curve as follows:

$$z + 3p - p^4 \equiv 0 \bmod r, \tag{4.18}$$

where z is the *mother parameter* of KSS-18 curve which is about six times smaller than order r.

Since Q is mapped to its ismorphic sextic twisted rational point Q', therefore we can consider scalar multiplication $[s]Q'$ where $0 \le s < r$. $[s]Q'$ will be calculated in $\mathbb{F}_{p^3}$ and eventually the result will be mapped to $\mathbb{F}_{p^{18}}$ to get the final result. From Eq.(4.5b) we know r is the order of KSS-18 curve where $[r]Q = \mathcal{O}$. Here, the bit size of s is nearly equal to r. In KSS-18 curve t is 4/6 times of r. Therefore, let us first consider $(t-1)$-adic representation of s as follows:

$$s = S_H(t-1) + S_L, \tag{4.19}$$

where s will be separated into two coefficients S_H and S_L. S_L will be nearly equal to the size of $(t-1)$ and S_H will be about half of $(t-1)$. In what follows, z-adic representation of S_H and S_L is given as:

$$\begin{aligned} S_H &= s_5 + s_4, \\ S_L &= s_3z^3 + s_2z^2 + s_1z + s_0. \end{aligned}$$

Finally s can be represented as 6 coefficients as follows:

$$\begin{aligned} s &= \sum_{i=0}^{3} s_iz^i + (s_4 + s_5z)(t-1), \\ s &= (s_0 + s_1z) + (s_2 + s_3z)z^2 + (s_4 + s_5z)(t-1). \end{aligned} \tag{4.20}$$

4.3.5 Reducing Elliptic Curve Doubling in $[s]Q'$

Let us consider a scalar multiplication of $Q' \in \mathbb{G}_2'$ in Eq.(4.20) as follows:

$$[s]Q' = (s_0 + s_1z)Q' + (s_2 + s_3z)z^2Q' + (s_4 + s_5z)(t-1)Q'. \tag{4.21}$$

In what follows, z^2Q', $(t-1)Q'$ of Eq.(4.21) is denoted as Q_1' and Q_2' respectively. From Eq.(4.18) and Eq.(4.11) we can derive the Q_1' as follows:

$$\begin{aligned} Q_1' &= z^2Q', \\ &= (9p^2 - 6p^5 + p^8)Q', \\ &= 9\pi'^2(Q') - 6\pi'^5(Q') + \pi'^8(Q'). \end{aligned} \tag{4.22}$$

where $\pi'(Q')$ is called the **skew Frobenius mapping** of rational point $Q' \in E'(\mathbb{F}_{p^3})$. Eq.(4.22) is simplified as follows by utilizing the properties of cyclotomic polynomial.

$$\begin{aligned} Q_1' &= 8\pi'^2(Q') - 5\pi'^5(Q'), \\ &= \pi'^2(8Q') - \pi'^5(5Q'). \end{aligned} \tag{4.23}$$

And from the Eq.(4.10) and Eq.(4.11), Q_2' is derived as,

$$Q_2' = \pi'(Q'). \tag{4.24}$$

Substituting Eq.(4.23) and Eq.(4.24) in Eq.(4.21), the following relation is obtained.

$$s[Q'] = (s_0 + s_1 z)Q' + (s_2 + s_3 z)Q'_1 + (s_4 + s_5 z)Q'_2. \tag{4.25}$$

Using $z \equiv -3p + p^4 \pmod r$ from Eq.(4.18), $z(Q')$ can be pre-computed as follows:

$$z(Q') = \pi'(-3Q') + \pi'^4(Q'). \tag{4.26}$$

Table 4.1 shows all the pre-computed values of rational points defined over $\mathbb{F}_{p^3}$ for the proposed method. Pre-computed rational points are denoted inside angular bracket such as $< Q' + Q'_2 >$ in this chapter.

TABLE 4.1: 13 pre-computed values of rational points.

Pre-computed rational points	Skew Frobenius mapped rational points
	$z(Q')$
Q'_1	$z(Q'_1)$
Q'_2	$z(Q'_2)$
$Q'_1 + Q'_2$	$z(Q'_1) + z(Q'_2)$
$Q' + Q'_2$	$z(Q') + z(Q'_2)$
$Q' + Q'_1$	$z(Q') + z(Q'_1)$
$Q' + Q'_1 + Q'_2$	$z(Q') + z(Q'_1) + z(Q'_2)$

4.3.6 Skew Frobenius Map of $\mathbb{G}_2$ Points in KSS-18 Curve

Similar to Frobenius mapping, skew Frobenius map is the p-th power over the sextic twisted isomorphic rational points such as $Q' = (x', y')$ as follows:

$$\pi' : (x', y') \mapsto (x'^p, y'^p) \tag{4.27}$$

The detailed procedure to obtain the skew Frobenius map of $Q' = (x', y') \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$ is given bellow:

$$\begin{aligned}
\pi'(x') &= (x')^p(i)^{1-p}(v)^{p-1}(\theta)^{p-1} \\
&= (x')^p(i)^{1-p}(\theta^4)^{p-1} \\
&= (x')^p(i^{-1})^p i(\theta^{p-1})^4 \\
&= (x')^p(i^{-1})^p i(i^{\frac{p-1}{6}})^4 \quad \text{where } \theta^6 = i \\
&= (x')^p(i^{-1})^p i(i^{\frac{p-1}{6}-1}i)^4 \\
&= (x')^p(i^{-1})^p i(i^{3\frac{\frac{p-7}{6}}{3}})^4 i^4 \\
&= (x')^p(i^{-1})^p i(2^{\frac{p-7}{18}})^4 2i \quad \text{where } i^3 = 2 \\
&= (x')^p(i^{-1})^p i(2^{\frac{2p-14}{9}+1})i \\
&= (x')^p(i^{-1})^p i(2^{\frac{2p-5}{9}})i, \qquad (4.28a)
\end{aligned}$$

$$\begin{aligned}
\pi'(y') &= (y')^p(i)^{1-p}(v)^{p-1} \\
&= (y')^p(i^{-1})^p i(v^{6\frac{p-1}{6}}) \\
&= (y')^p(i^{-1})^p i(i^{3\frac{p-1}{6}}) \\
&= (y')^p(i^{-1})^p i2^{\frac{p-1}{6}}. \qquad (4.28b)
\end{aligned}$$

Here $(i^{-1})^p i$, $(2^{\frac{2p-5}{9}})i$ and $2^{\frac{p-1}{6}}$ can be pre-computed.

4.3.7 Multi-Scalar Multiplication

Applying the the multi-scalar multiplication technique in Eq.(4.25) we can efficiently calculate the scalar multiplication in $\mathbb{F}_{p^3}$. Figure 4.5 shows an example of this multiplication. Suppose in an arbitrary index, from left to right, bit pattern of s_1, s_3, s_5 is 101 and at the same index s_0, s_2, s_4 is 111. Therefore we apply the pre-computed points $< z(Q') + z(Q'_2) >$ and $< Q' + Q'_1 + Q'_2 >$ as ECA in parallel. Then we perform ECD and move to the right next bit index to repeat the process until maximum length z-adic coefficient becomes zero.

As shown in Figure 4.5, during scalar multiplication, we are considering 3 pair of coefficients of z-adic representation as shown in Eq.(4.20). If we consider 6-coefficients for parallelization, it will require $2^6 \times 2$ pre-computed points. The chance of appearing each pre-computed point in the calculation will be only once that will cause redundancy.

4.3.7.1 Re-mapping Rational Points from $E'(\mathbb{F}_{p^3})$ to $E(\mathbb{F}_{p^{18}})$

After the multi-scalar multiplication, we need to remap the result to $\mathbb{F}_{p^{18}}$. For example let us consider re-mapping of $Q' = (x', y') \in E'(\mathbb{F}_{p^3})$ to $Q = (Av\theta, Bv) \in$

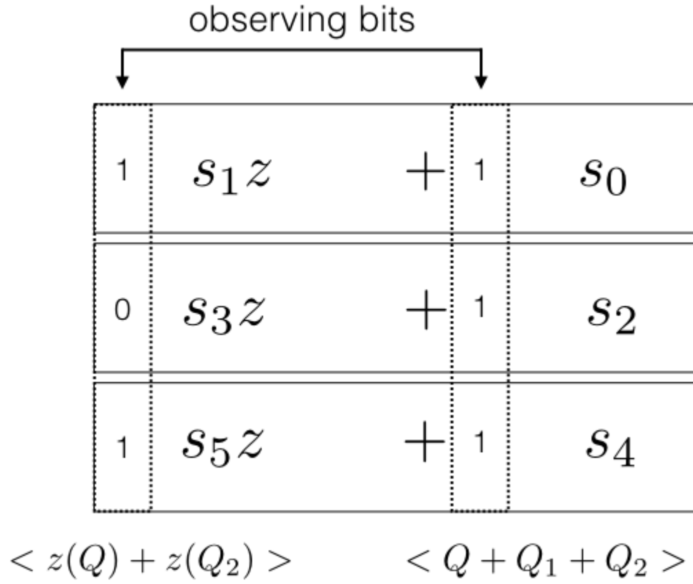


FIGURE 4.5: Multi-scalar multiplication of s with Frobenius mapping.

$E(\mathbb{F}_{p^{18}})$. From Eq.(4.15a), Eq.(4.15b) and Eq.(4.14) it can be obtained as follows:

$$\begin{aligned} xi^{-1}\theta^4 &= Av\theta, \\ yi^{-1}\theta^3 &= Bv, \end{aligned}$$

which resembles that $Q = (Av\theta, Bv)$. Therefore it means that multiplying i^{-1} with the Q' coordinates and placing the resulted coefficients in the corresponding position of the coefficients in Q, will map Q' to Q. This mapping costs one $\mathbb{F}_{p^3}$ inversion of i which can be pre-computed and one $\mathbb{F}_p$ multiplication.

4.4 Simulation Result Evaluation

This section shows experimental result with the calculation cost. In the experiment we have compared the proposed method with three well studied method of scalar multiplication named binary method, sliding-window method and non-adjacent form (NAF) method. The mother parameter z is selected according to the suggestion of Scott et al. [Sco11] to obtain $p = 508 \approx 511$-bit and $r = 376 \approx 384$-bit to simulate in 192-bit security level. Table 4.2 shows the parameter settings considered for the simulation.

Table 4.3 shows the environment, used to experiment and evaluate the proposed method.

In the experiment 100 random scalar numbers of size less than order r (378-bit) is generated. 13 ECA counted for pre-computed rational points is taken

TABLE 4.2: Parameter settings used in the experiment

Defined KSS-18 curve	$y^2 = x^3 + 11$
Mother parameter z	65-bit
Characteristics $p(z)$	511-bit
Order $r(z)$	376-bit
Frobenius trace $t(z)$	255-bit
Persuadable security level	192-bit

TABLE 4.3: Computational Environment

	PC	iPhone6s
CPU *	2.7 GHz Intel Core i5	Apple A9 Dual-core 1.84 GHz
Memory	16 GB	2 GB
OS	Mac OS X 10.11.6	iOS 10.0
Compiler	gcc 4.2.1	gcc 4.2.1
Programming Language	C	Objective-C, C
Library	GMP 6.1.0	GMP 6.1.0

*Only single core is used from two cores.

into account while the average is calculated for the proposed method. Window size of 4-bit is considered for sliding-window method. Therefore 14 pre-computed ECA is required. In addition, average execution time of the proposed method and the three other methods is also compared along with the operation count.

In what follows, "***With isomorphic mapping***" refers that skew Frobenius mapping technique is applied for Binary, Sliding-window and NAF methods. Therefore the scalar multiplication is calculated in $\mathbb{F}_{p^3}$ extension field. And for Proposed method it is skew Frobenius mapping with multi-scalar multiplication. On the other hand "***Without isomorphic mapping***" denotes that Frobenius map is not applied for any of the methods. In this case, all the scalar multiplication is calculated in $\mathbb{F}_{p^{18}}$ extension field.

In Table 4.4 the operations of the *Proposed* method are counted in $\mathbb{F}_{p^3}$. On the other hand for Binary, Sliding-window and NAF method, the operations are counted in $\mathbb{F}_{p^{18}}$. The table clearly shows that in the *Proposed* method requires about 6 times less ECD than any other methods. The number of ECA is also reduced in the *Proposed* method by about 30% than binary method and almost same number of ECA of NAF.

Analyzing Table 4.5, we can find that when isomorphic mapping and skew Frobenius mapping is not adapted for Binary, Sliding-window and NAF, then the scalar multiplication of proposed method is more than 60 times faster than other methods. However when isomorphic mapping is applied for the

TABLE 4.4: Comparison of average number of ECA and ECD for $\mathbb{G}_2$ SCM in KSS-18.

	Count of average number of ECA, ECD	
Methods	ECA	ECD
Binary	186	375
Sliding-window	102	376
NAF	127	377
Proposed	123	64

TABLE 4.5: Comparison of execution time in [ms] for scalar multiplication in KSS-18 curve.

	Execution time in [ms]			
	With isomorphic mapping		Without isomorphic mapping	
Methods	PC	iPhone6s	PC	iPhone6s
Binary	5.4×10^1	8.4×10^1	1.2×10^3	1.8×10^3
Sliding-window	4.8×10^1	7.5×10^1	1.0×10^3	1.6×10^3
NAF	5.3×10^1	7.7×10^1	1.6×10^3	1.7×10^3
Proposed	1.6×10^1	2.4×10^1	-	-
Multi-scalar (only)	-	-	3.4×10^2	5.5×10^2

other methods then our proposed technique is more than 3 times faster. Another important comparison shows that when only multi-scalar multiplication is applied then our proposed methods is about 20 times faster. In every scenario our proposed method is faster than the other commonly used approaches.

The main focus of this experiment is to evaluate the acceleration ratio of scalar multiplication by applying the proposed approach on $\mathbb{G}_2$ rational point group of KSS curve of embedding degree 18. The experiment does not focus on efficiently implementing scalar multiplication for certain environment.

4.5 Contribution Summary

In this chapter we have proposed an efficient method to calculate elliptic curve scalar multiplication using skew Frobenius mapping over KSS-18 curve in context of pairing based cryptography. Utilizing the skew Frobenius map along with multi-scalar multiplication procedure, an efficient scalar multiplication method for KSS-18 curve is proposed in the chapter. In addition to the theoretic proposal, this chapter has also presented a comparative simulation of the proposed approach with plain binary method, sliding window method and non-adjacent form (NAF) for scalar multiplication. We have also applied $(t-1)$-adic and z-adic representation on the scalar and have applied multi-scalar multiplication technique to calculate scalar multiplication in parallel. We have evaluated and analyzed the improvement by implementing a simulation for large size of scalar in 192-bit security level. According to the simulation result multi-scalar multiplication after applying skew Frobenius mapping in $\mathbb{G}_2'$ can accelerate the scalar multiplication in $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ by more than 60 times than scalar multiplication of $\mathbb{G}_2$ rational point directly in $\mathbb{F}_{p^{18}}$.

4.6 Future Work

In the previous work of Sakemi et al. [Sak+08] have proposed skew Frobenius map for $\mathbb{G}_1$ rational point defined over BN curve. As a future work we would like to apply such approach on $\mathbb{G}_1$ rational point defined over KSS-18 curve. Together with the proposed method, the skew Frobenius mapping of $\mathbb{G}_1$ will remarkably accelerate scalar multiplication over KSS-18 curve in the context of pairing based cryptography.

Chapter 5

CANDAR 2016

Pairing based cryptography is considered as the next generation of security for which it attracts many researcher to work on faster and efficient pairing to make it practical. Among the several challenges of efficient pairing; efficient scalar multiplication of rational point defined over extension field of degree $k \geq 12$ is important. However, there exists isomorphic rational point group defined over relatively lower degree extension field. Exploiting such property, this thesis has showed a mapping technique between isomorphic rational point groups in the context of Ate-based pairing with Kachisa-Schaefer-Scott (KSS) pairing friendly curve of embedding degree $k = 18$. In the case of KSS curve, there exists subfield sextic twisted curve that includes sextic twisted isomorphic rational point group defined over $\mathbb{F}_{p^3}$. This thesis has showed the mapping procedure from certain $\mathbb{F}_{p^{18}}$ rational point group to its subfield isomorphic rational point group in $\mathbb{F}_{p^3}$ and vice versa. This thesis has also showed that scalar multiplication is about 20 times faster after applying the proposed mapping which in-turns resembles that the impact of this mapping will greatly enhance the pairing operation in KSS curve.

5.1 Introduction

At the advent of this century, Sakai et al. [SK03] and Joux et al. [Jou04] independently proposed a cryptosystem based on elliptic curve pairing. Since then, pairing based cryptography has attracted many researchers and it has been considered as the basis of next generation security. Many researchers have proposed several innovative pairing based cryptographic applications such as ID-based encryption [SK03], broadcast encryption [BGW05] and group signature authentication [BBS04] that upsurge the popularity of pairing based cryptography. In such outcome, Ate-based pairings such as Ate [Coh+05], R-ate [LLP09], Optimal-ate [Ver10], twisted Ate [Mat+07] and χ-Ate [Nog+08] pairings have gained much attention since they have achieved quite efficient pairing calculation. There is no alternative of efficient and fast pairing calculation for deploying pairing-based cryptographic applications in practical case. This thesis focuses on a peripheral technique of Ate-based pairings with Kachisa-Schaefer-Scott (KSS) curve [KSS07].

In general, pairing is a bilinear map from two rational point group $\mathbb{G}_1$ and $\mathbb{G}_2$ to a multiplicative group $\mathbb{G}_3$ [SCA86], typically denoted by $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$. In the context of Ate-based pairing, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\begin{aligned}\mathbb{G}_1 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [1]),\\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]),\\ \mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r,\end{aligned}$$

$$\alpha : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3,$$

where α denotes Ate pairing. Pairings are often found in certain extension field $\mathbb{F}_{p^k}$, where p is the prime number, also know as characteristics and the minimum extension degree k is called *embedding* degree. The rational points $E(\mathbb{F}_{p^k})$ are defined over a certain pairing friendly curve E of embedded extension field of degree k. This thesis has considered Kachisa-Schaefer-Scott (KSS) [KSS07] pairing friendly curves of emebdding degree $k = 18$ described in [FST06].

In Ate-based pairing with KSS curve, where $k = 18$, pairing computations are done in higher degree extension field $\mathbb{F}_{p^{18}}$. However, KSS curves defined over $\mathbb{F}_{p^{18}}$ have the sextic twisted isomorphism over $\mathbb{F}_{p^3}$. Therefore we can execute computations in the subfield $\mathbb{F}_{p^3}$. Exploiting such a property, different arithmetic operation of Ate-based pairing can be efficiently performed in $\mathbb{G}_2$. In this thesis we have mainly focused on mapping $\mathbb{G}_2$ rational point from extension field $\mathbb{F}_{p^{18}}$ to its sextic twisted subfield $\mathbb{F}_{p^3}$ and its reverse procedure.

The advantage of such mapping is examined by performing scalar multiplication on $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ rational point, since scalar multiplication is required repeatedly in cryptographic calculation. We have considered subfield sextic twisted curve of KSS curve, denoted as E'. It includes sextic twisted isomorphic rational point group denoted as $\mathbb{G}'_2 \subset E(\mathbb{F}_{p^3})$. In KSS curve, $\mathbb{G}_2$ is defined over $\mathbb{F}_{p^{18}}$ whereas its subfield isomorphic group $\mathbb{G}'_2$ is defined over $\mathbb{F}_{p^3}$. Then the proposed mapping technique is applied to map rational points of $\mathbb{G}_2$ to its isomorphic $\mathbb{G}'_2$. After that the scalar multiplication in $\mathbb{G}'_2$ performed and the resulted points are re-mapped to $\mathbb{G}_2$ in $\mathbb{F}_{p^{18}}$. The experiment result shows that efficiency of binary scalar multiplication is increased by more than 20 times in subfield sextic twisted curve than scalar multiplication in $\mathbb{F}_{p^{18}}$ without applying proposed mapping. The mapping and remapping requires one bit wise shifting in $\mathbb{F}_p$, one $\mathbb{F}_{p^3}$ inversion which can be pre-computed and one $\mathbb{F}_p$ multiplication; hence the mapping procedure has no expensive arithmetic operation.

The rest of the thesis is organized as follows. The fundamentals of elliptic curve arithmetic, scalar multiplication along with KSS curve over $\mathbb{F}_{p^{18}}$ extension field and *sextic twist* of KSS curve are described in section II. In section III, this thesis describes the isomorphic mapping between the rational point Q and Q' in details. The experimental result is presented in section IV which shows that our scalar multiplication on $\mathbb{G}_2$ point can be accelerated by 20 times by applying the proposed mapping technique in KSS curve. Finally section V draws the conclusion with some outline how this work can be enhanced more as a future work.

5.2 Preliminaries

In this section this thesis briefly overviews the fundamentals of elliptic curve operations. Elliptic curve scalar multiplication is reviewed briefly. Pairing friendly curve of embedded degree $k = 18$, i.e., KSS curve and its properties are introduced in combination with its construction procedure by towering.

5.2.1 Elliptic curve

Let $\mathbb{F}_p$ be a prime field and $\mathbb{F}_q$ be its extension field. An elliptic curve [Was03] defined over $\mathbb{F}_p$ is generally represented by *affine coordinates* [SCA86] as follows;

$$E/\mathbb{F}_p : y^2 = x^3 + ax + b, \tag{5.1}$$

where $4a^3 + 27b^2 \neq 0$ and $a, b \in \mathbb{F}_p$. A pair of coordinates x and y that satisfy Eq.(5.1) are known as *rational points* on the curve.

$E(\mathbb{F}_{q^k})$ denotes an elliptic curve group where the definition field is $\mathbb{F}_{q^k}$ and $\#E(\mathbb{F}_{q^k})$ denotes its order. When the definition field is prime field $\mathbb{F}_p$ then $\#E(\mathbb{F}_p)$ can be represented as,

$$\#E(\mathbb{F}_p) = p + 1 - t, \tag{5.2}$$

where t is called the Frobenius trace of $E(\mathbb{F}_p)$.

Let $E(\mathbb{F}_p)$ be the set of all rational points on the curve defined over $\mathbb{F}_p$ and it includes the point at infinity denoted by $\mathcal{O}$. The order of $E(\mathbb{F}_p)$ is denoted by $\#E(\mathbb{F}_p)$ where $E(\mathbb{F}_p)$ forms an additive group for the elliptic addition. The set of rational points over $\mathbb{F}_q$, including $\mathcal{O}$ satisfying Eq.(5.1) is denoted by $E(\mathbb{F}_q)$. The order of $E(\mathbb{F}_q)$ is denoted by $\#E(\mathbb{F}_q)$.

Let us consider two rational points using affine coordinates as $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, and their addition $R = P_1 + P_2$, where $R = (x_3, y_3)$ and $P_1, P_2, R \in E(\mathbb{F}_q)$. Then the x and y coordinates of R are calculated as follows:

$$x_3 = \lambda^2 - x_1 - x_2, \tag{5.3a}$$

$$y_3 = (x_1 - x_3)\lambda - y_1, \tag{5.3b}$$

where λ is given as follows:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}; & P_1 \neq P_2, \\ \frac{3x_1^2 + a}{2y_1}; & P_1 = P_2, \end{cases} \tag{5.3c}$$

λ is the tangent at the point on the curve and O is the additive unity in $E(\mathbb{F}_q)$. When $P_1 \neq P_2$ then $P_1 + P_2$ is called elliptic curve addition (ECA). If $P_1 = P_2$ then $P_1 + P_2 = 2P_1$, which is known as elliptic curve doubling (ECD).

Let $[s]P_1$ be the scalar multiplication for the rational point P_1 with scalar s as,

$$[s]P_1 = \sum_{i=0}^{s-1} P_1, \quad 0 \leq s < r, \tag{5.4}$$

where r is the order of the target rational point group. If $s = r$, where r is the order of the curve then $[r]P_1 = O$. When $[s]P_1 = P_2$, if s is unknown, then the solving s from P_1 and P_2 is known as elliptic curve discrete logarithm problem (ECDLP). The security of elliptic curve cryptography depends on the difficulty of solving ECDLP.

The binary method is a widely recognized method for calculating the elliptic curve scalar multiplication. Algorithm 6 shows the binary scalar multiplication algorithm. This algorithm scans the bits of scalar s from most significant bit to least significant bit. When $s[i] = 1$, it will perform ECA and ECD otherwise only ECD will be calculated. But this method is not resistant to side channel attack [Koc96].

Algorithm 6: Left-to-right binary algorithm for elliptic curve scalar multiplication

```
Input: $P, s$
Output: $[s]P$
$T \leftarrow 0$
for $i = \lfloor \log_2 s \rfloor$ to 0 do
    $T \leftarrow T + T$
    if $s[i] = 1$ then
        $T \leftarrow T + P$
return $T$
```

On the other hand Montgomery ladder algorithm is said to be resistant of side channel attack. Algorithm 7 shows the Montgomery ladder algorithm for scalar multiplication. Montgomery ladder has some similarity with binary method except in each iteration it performs ECA and ECD.

Algorithm 7: Montgomery ladder algorithm for elliptic curve scalar multiplication

```
Input: $P, s$
Output: $[s]P$
$T_0 \leftarrow 0, T_1 \leftarrow P$
for $i = \lfloor \log_2 s \rfloor$ to 0 do
    if $s[i] = 1$ then
        $T_0 \leftarrow T_0 + T_1$
        $T_1 \leftarrow T_1 + T_1$
    else if $s[i] = 0$ then
        $T_1 \leftarrow T_0 + T_1$
        $T_0 \leftarrow T_0 + T_0$
return $T_0$
```

This thesis has considered left-to-right binary scalar multiplication for evaluating the efficiency of the proposed mapping operation. But from the view point of security binary method is vulnarable to side channel attack. Therefore this thesis has also experimented with Montgomery ladder [SCA86] for scalar multiplication evaluation.

5.2.2 KSS curve

Kachisa-Schaefer-Scott (KSS) curve [KSS07] is a non super-singular pairing friendly elliptic curve of embedding degree 18, defined over $\mathbb{F}_{p^{18}}$ as follows:

$$E/\mathbb{F}_{p^{18}} : Y^2 = X^3 + b, \quad b \in \mathbb{F}_p, \tag{5.5}$$

where $b \neq 0$ and $X, Y \in \mathbb{F}_{p^{18}}$. Its characteristic p, Frobenius trace t and order r are given systematically by using an integer variable u as follows:

$$\begin{aligned} p(u) &= (u^8 + 5u^7 + 7u^6 + 37u^5 + 188u^4 + 259u^3 \\ &\quad +343u^2 + 1763u + 2401)/21, && (5.6a) \\ r(u) &= (u^6 + 37u^3 + 343)/343, && (5.6b) \\ t(u) &= (u^4 + 16u + 7)/7, && (5.6c) \end{aligned}$$

where u is such that $u \equiv 14 \pmod{42}$ and the ρ value is $\rho = (\log_2 p / \log_2 r) \approx 1.33$.

5.2.3 $\mathbb{F}_{p^{18}}$ extension field arithmetic

In pairing, arithmetic operations are performed in higher degree extension fields, such as $\mathbb{F}_{p^k}$ for moderate value of k [SCA86]. Concequently, such higher extension field needs to be constructed as tower of extension fields [BS09] to perform arithmetic operation cost effectively.

This thesis has represented extension field $\mathbb{F}_{p^{18}}$ as a tower of subfield to improve arithmetic operations. It has also used irreducible binomials introduced by Bailey et al. [BP01]. In what follows, this thesis considers $3|(p-1)$ and c is a quadratic and cubic non residue in $\mathbb{F}_p$. In context of KSS-curve [KSS07], where $k = 18$, $\mathbb{F}_{p^{18}}$ is constructed as tower field with irreducible binomial as follows:

$$\begin{cases} \mathbb{F}_{p^3} = \mathbb{F}_p[i]/(i^3 - c), \\ \mathbb{F}_{p^6} = \mathbb{F}_{p^3}[v]/(v^2 - i), \\ \mathbb{F}_{p^{18}} = \mathbb{F}_{p^6}[\theta]/(\theta^3 - v), \end{cases} \tag{5.7}$$

where $c = 2$ is considered to be the best choice for efficient arithmetic. From the above towering construction we can find that $i = v^2 = \theta^6$, where i is the basis element of the base extension field $\mathbb{F}_{p^3}$. In the previous work of Aranha et al. [Ara+13], explained the base extension field $\mathbb{F}_{p^3}$ for the *sextic twist* of KSS curve.

5.2.4 $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ groups.

In the context of pairing-based cryptography, especially on KSS curve, three groups $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_3$ are considered. From [Mor+14], we define $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ as follows:

$$\begin{aligned} \mathbb{G}_1 &= E(\mathbb{F}_{p^{18}})[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 &= E(\mathbb{F}_{p^{18}})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ \mathbb{G}_3 &= \mathbb{F}^*_{p^{18}}/(\mathbb{F}^*_{p^{18}})^r, \end{aligned}$$

$$\alpha : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3, \tag{5.8}$$

where α denotes Ate pairing. In the case of KSS curve, $\mathbb{G}_1, \mathbb{G}_2$ are rational point groups and $\mathbb{G}_3$ is the multiplicative group in $\mathbb{F}_{p^{18}}$. They have the same order r.

5.2.5 Sextic twist of KSS curve

When the embedding degree $k = 6e$, where e is positive integer, *sextic* twist is given as follows:

$$E : \quad y^2 = x^3 + b, \quad b \in \mathbb{F}_p, \tag{5.9}$$

$$E'_6 : \quad y^2 = x^3 + bz^{-1}, \tag{5.10}$$

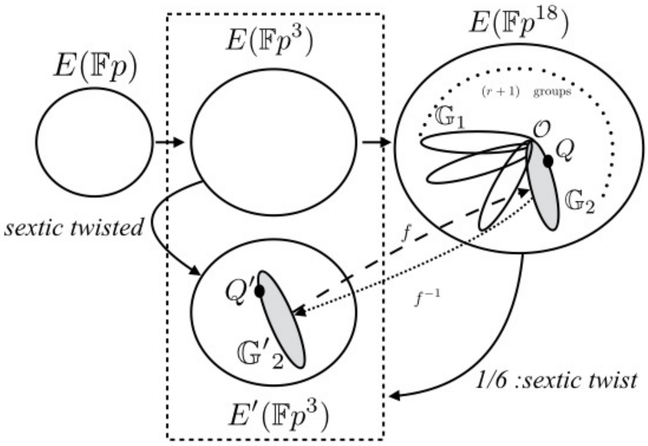


FIGURE 5.1: *sextic twist* in KSS curve.

where z is a quadratic and cubic non residue in $E(\mathbb{F}_{p^e})$ and $3|(p^e - 1)$. Isomorphism between $E'_6(\mathbb{F}_{p^e})$ and $E(\mathbb{F}_{p^{6e}})$, is given as follows:

$$\psi_6 : \begin{cases} E'_6(\mathbb{F}_{p^e}) \longrightarrow E(\mathbb{F}_{p^{6e}}), \\ (x, y) \quad\ \ \mapsto (xz^{1/2}, yz^{1/2}). \end{cases} \tag{5.11}$$

In context of Ate-based pairing for KSS curve of embedding degree 18, sextic twist is considered to be the most efficient. This papers considers mapping of sextic twisted subfield isomorphic group of $\mathbb{F}_{p^{18}}$.

5.3 Isomorphic mapping between Q and Q'

This section introduces our proposal of mapping procedure of $\mathbb{G}_2$ rational point group to its sextic twisted isomorphic group $\mathbb{G}'_2$ for Ate-based pairing with KSS curve.

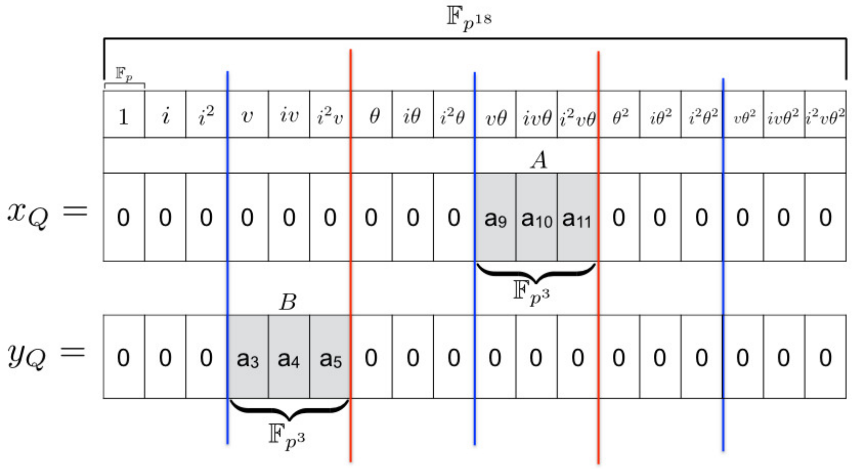
Figure 5.1 shows an overview of sextic twisted curve $E'(\mathbb{F}_{p^3})$ of $E(\mathbb{F}_{p^{18}})$. Let us consider E is the KSS curve in base field $\mathbb{F}_{p^3}$ and E' is sextic twist of E' given as follows:

$$E : y^2 = x^3 + b, \tag{5.12}$$

$$E' : y^2 = x^3 + bi, \tag{5.13}$$

where $b \in \mathbb{F}_p$; $x, y, i \in \mathbb{F}_{p^3}$ and basis element i is the quadratic and cubic non residue in $\mathbb{F}_{p^3}$.

In context of KSS curve, let us consider a rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$. Q has a special vector representation with 18 $\mathbb{F}_p$ elements for each x_Q and y_Q coordinates. Figure 5.2 shows the structure of the coefficients of $Q \in \mathbb{F}_{p^{18}}$ and its sextic twisted isomorphic rational point $Q' \in \mathbb{F}_{p^3}$ in KSS curve. Among 18 elements, there are 3 continuous nonzero $\mathbb{F}_p$ elements. The others are zero. However the set of these nonzero elements belongs to $\mathbb{F}_{p^3}$.



	1	i	i^2	v	iv	i^2v	θ	$i\theta$	$i^2\theta$	$v\theta$	$iv\theta$	$i^2v\theta$	θ^2	$i\theta^2$	$i^2\theta^2$	$v\theta^2$	$iv\theta^2$	$i^2v\theta^2$
$x_Q =$	0	0	0	0	0	0	0	0	0	a_9	a_{10}	a_{11}	0	0	0	0	0	0
$y_Q =$	0	0	0	a_3	a_4	a_5	0	0	0	0	0	0	0	0	0	0	0	0

$$a_j \in \mathbb{F}_p, \quad \text{where} \quad a_j = (0, 1, \cdots, 17)$$
$$Q = (x_Q, y_Q) = (Av\theta, Bv) \in \mathbb{F}_{p^{18}}$$
$$Q' = (x'_Q, y'_Q) = (Ai, Bi) \in \mathbb{F}_{p^3}$$

FIGURE 5.2: $Q \in \mathbb{F}_{p^{18}}$ and its sextic twisted isomorphic rational point $Q' \in \mathbb{F}_{p^3}$ structure in KSS curve.

This thesis considers the mother parameter of KSS curve u = 65-bit and characteristics p = 511-bit. In such consideration, Q is given as $Q = (Av\theta, Bv)$, showed in Figure 5.2, where $A, B \in \mathbb{F}_{p^3}$ and v and θ are the basis elements of $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{18}}$ respectively.

Let us consider the sextic twisted isomorphic subfield rational point of Q as $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$. Considering x' and y' as the coordinates of Q', we can map the rational point $Q = (Av\theta, Bv)$ to the rational point $Q' = (x', y')$ as follows.

Multiplying both side of Eq.(5.13) with θ^{-6}, where $i = \theta^6$ and $v = \theta^3$.

$$E' : \left(\frac{y}{\theta^3}\right)^2 = \left(\frac{x}{\theta^2}\right)^3 + b. \tag{5.14}$$

θ^{-2} of Eq.(5.14) can be represented as follows:

$$\begin{aligned} \theta^{-2} &= i^{-1} i\theta^{-2}, \\ &= i^{-1}\theta^4, \end{aligned} \tag{5.15a}$$

and multiplying i with both sides.

$$\theta^4 = i\theta^{-2}. \tag{5.15b}$$

Similarly θ^{-3} can be represented as follows:

$$\begin{aligned} \theta^{-3} &= i^{-1} i\theta^{-3} \\ &= i^{-1}\theta^3. \end{aligned} \tag{5.15c}$$

Multiplying i with both sides of Eq.(5.15c) we get θ^3 as,

$$\theta^3 = i\theta^{-3}, \tag{5.15d}$$

5.3.0.1 Q to Q' mapping

Let us represent $Q = (Av\theta, Bv)$ as follows:

$$Q = (A\theta^4, B\theta^3), \quad \text{where } v = \theta^3. \tag{5.16}$$

From Eq.(5.15b) and Eq.(5.15d), we substitute $\theta^4 = i\theta^{-2}$ and $\theta^3 = i\theta^{-3}$ in Eq.(5.16) as follows:

$$Q = (Ai\theta^{-2}, Bi\theta^{-3}), \tag{5.17}$$

where $Ai = x'$ and $Bi = y'$ are the coordinates of $Q' = (x', y') \in \mathbb{F}_{p^3}$. Which implies that we can map $Q \in \mathbb{F}_{p^{18}}$ to $Q' \in \mathbb{F}_{p^3}$ by first selecting the 3 nonzero $\mathbb{F}_p$ coefficients of each coordinates of Q. Then these nonzero $\mathbb{F}_p$ elements form an $\mathbb{F}_{p^3}$ element. After that multiplying the basis element i with that $\mathbb{F}_{p^3}$ element, we get the final $Q' \in \mathbb{F}_{p^3}$. From the structure of $\mathbb{F}_{p^{18}}$, given in Eq.(5.7), this mapping has required no expensive arithmetic operation. Multiplication by the basis element i in $\mathbb{F}_{p^3}$ can be done by 1 bit wise left shifting since $c = 2$ is considered for towering in Eq.(5.7).

5.3.0.2 Q' to Q mapping

The reverse mapping $Q' = (x', y') \in \mathbb{F}_{p^3}$ to $Q = (Av\theta, Bv) \in \mathbb{F}_{p^{18}}$ can be obtained as from Eq.(5.15a), Eq.(5.15c) and Eq.(5.14) as follows:

$$\begin{aligned} xi^{-1}\theta^4 &= Av\theta, \\ yi^{-1}\theta^3 &= Bv, \end{aligned}$$

which resembles that $Q = (Av\theta, Bv)$. Therefore it means that multiplying i^{-1} with the Q' coordinates and placing the resulted coefficients in the corresponding position of the coefficients in Q, will map Q' to Q. This mapping costs one $\mathbb{F}_{p^3}$ inversion of i which can be pre-computed and one $\mathbb{F}_p$ multiplication.

5.4 Result Analysis

In order to determine the advantage of the proposal, first we have applied the proposed mapping technique to map rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ to its isomorphic point $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$. After that we performed the scalar multiplication of Q'. Then the resulted points are re-mapped to $\mathbb{G}_2$ in $\mathbb{F}_{p^{18}}$. On the other hand we performed scalar multiplication of Q without mapping. In the experiment, 100 scalar numbers of size (about 377-bit) less than order r is generated randomly and then scalar multiplication is calculated for both case. Average value of execution time is considered for comparison. The comparative result is shown in Table 5.2.

TABLE 5.1: Computational Environment

•	PC	iPhone6s
CPU [*]	2.7 GHz Intel Core i5	Apple A9 Dual-core 1.84 GHz
Memory	16 GB	2 GB
OS	Mac OS X 10.11.4	iOS 9.3.1
Compiler	gcc 4.2.1	gcc 4.2.1
Programming Language	C	Objective-C, C
Library	GNU MP [Gt15]	GNU MP

[*] Only single core is used from two cores.

TABLE 5.2: Comparative result of average execution time in [ms] for scalar multiplication

	Average execution time [ms] comparison	
	PC	iPhone 6s
	Execution time	Execution time
Binary method with mapping	5.4×10^1	6.4×10^1
Binary method without mapping	1.1×10^3	1.2×10^3
Montgomery ladder with mapping	6.8×10^1	8.4×10^1
Montgomery ladder without mapping	1.5×10^3	1.6×10^3

In the experiment, mother parameter u is also selected accordingly to find out $\mathbb{G}_2$ rational point Q. In addition p = 511-bit is considered, since Scott et al. [Sco11] has proposed the size of the characteristics p to be 508 to 511-bit with order r of 384-bit for 192-bit security level.

In the experiment, KSS curve over $\mathbb{F}_{p^{18}}$ is given as $y^2 = x^3 + 11$, considering the following parameters

$$\begin{aligned} u &= \text{65-bit}, \\ p &= \text{511-bit}, \\ r &= \text{378-bit}, \\ t &= \text{255-bit}. \end{aligned}$$

Table 11.2 shows the experiment environment, used to evaluate usefulness of the proposed mapping.

Analyzing Table 5.2, we can find that scalar multiplication using the proposed mapping technique is more than 20 times faster than scalar multiplication without the proposed mapping. It this experiment we used binary method and Montgomery ladder for scalar multiplication in both case. In the previous work of Nogami et al. [Nog+09], has showed the procedure to apply Frobenious mapping on twisted elliptic curve for Ate-based pairing. This multiplication can be done more efficiently if skew Frobenius mapping is applied on sextic twisted isomorphic rational point after applying the proposed mapping.

In the experiment we have used two execution environments; such as PC and iPhone with different CPU frequencies. In both environments only one processor core is utilized. The result also shows that the ratio of execution time of PC and iPhone without mapping of both methods is about 0.9. On the other hand the ratio of execution time with mapping of both methods is about 0.8. But the ratio of CPU frequencies of iPhone and PC is about $1.84/2.7 \approx 0.68$. Since PC and iPhone has different processor architectures therefore it's frequency ratio has no relation with the execution time ratio.

The main focus of this experiment is to evaluate the acceleration ratio of scalar multiplication by applying the proposed mapping on $\mathbb{G}_2$ rational point group of KSS curve of embedding degree 18. The experiment does not focus on efficiently implementing scalar multiplication for certain environment. There are other pairing friendly curves such as BLS-12, BLS-24 [FST06] where sextic twist is available. We will try to apply the proposed mapping on those curves as our future work.

5.5 Conclusion and future work

In this thesis we have proposed mapping procedure of $\mathbb{G}_2$ rational point group to its sextic twisted subfield isomorphic rational point group $\mathbb{G}'_2$ and its reverse mapping on KSS curve in context of Ate based pairing. We have also presented the advantages of such mapping by applying binary scalar multiplication and Montgomery ladder on sextic twisted isomorphic rational points in $\mathbb{G}'_2$. Then result of scalar multiplication in $\mathbb{G}'_2$ can accelerate the scalar multiplication in $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ by more than 20 times than scalar multiplication of $\mathbb{G}_2$ rational point directly in $\mathbb{F}_{p^{18}}$. In the previous work of Sakemi et al. [Sak+08] has proposed skew Frobenious map for $\mathbb{G}_1$ rational point defined over BN curve. As a future work we would like to apply such approach on $\mathbb{G}_1$ rational point defined over KSS curve. Together with the proposed mapping and the skew Frobenius mapping of $\mathbb{G}_1$ will remarkably accelerate scalar multiplication over KSS curve in the context of pairing based cryptography.

Acknowledgment

This work was partially supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of Ministry of Internal Affairs and Communications, Japan.

// Mergend in IJNC

Chapter 6

Mapping over Quartic and Sextic twisted KSS Curves

6.1 Introduction

6.1.1 Background and Motivation

In Ate-based pairing with KSS curve, pairing computations are done in higher degree extension field $\mathbb{F}_{p^k}$. However, KSS curves defined over $\mathbb{F}_{p^{18}}$ have the sextic twisted isomorphic rational point group defined over $\mathbb{F}_{p^3}$ and KSS curves defined over $\mathbb{F}_{p^{16}}$ have the quartic twisted isomorphism over $\mathbb{F}_{p^4}$. Therefore we can execute computations in the subfield $\mathbb{F}_{p^{k/d}}$ where d is the twist degree. Exploiting such a property, different arithmetic operations of Ate-based pairing can be efficiently performed in $\mathbb{G}_2$. However, performing elliptic curve operations in small extension field brings security issue since they are vulnerable to small subgroup attack [LL97]. Recently Barreto et al. [Bar+15] have studied the resistance of KSS-18 curves to small subgroup attacks. Such resistible KSS-16 curve is also studied by Loubna et al. [GF16a] at 192-bit security level. Therefore isomorphic mapping of KSS-18 and KSS-16 curves and implementing arithmetic operation can be done securely in subfield twisted curves for 192-bit security level. This chapter has mainly focused on isomorphic mapping of $\mathbb{G}_2$ rational points from extension field $\mathbb{F}_{p^k}$ to its twisted (sextic and quartic) subfield $\mathbb{F}_{p^{k/d}}$ and its reverse procedure for both KSS-18 and KSS-16 curves.

The advantage of such isomorphic mapping is examined by performing scalar multiplication on $\mathbb{G}_2 \subset E(\mathbb{F}_{p^k})$ rational point, since scalar multiplication is required repeatedly in cryptographic calculation. Three well-known scalar multiplication algorithms are considered for the comprehensive experimental implementation named as binary method, Montgomery ladder and sliding-window method. This chapter has considered subfield twisted curve of both KSS-16 and KSS-18 curve, denoted as E'. KSS-18 curve E' includes sextic twisted isomorphic rational point group denoted as $\mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$, whereas for KSS-16 curve E' contains the quartic twisted isomorphic rational point group denoted as $\mathbb{G}'_2 \subset E'(\mathbb{F}_{p^4})$. Then the proposed mapping technique is applied to map rational points of $\mathbb{G}_2$ to its isomorphic $\mathbb{G}'_2$. After that the scalar multiplication is performed in $\mathbb{G}_2'$ and then resulted points are re-mapped to $\mathbb{G}_2$.

The experiment result shows that efficiency of scalar multiplication is increased by more than 20 to 10 times in subfield twisted curve E' than scalar multiplication in $E(\mathbb{F}_{p^{18}})$ and $E(\mathbb{F}_{p^{16}})$ respectively without applying the proposed mapping. The mapping and remapping for sextic twisted curves requires one bit wise shifting in $\mathbb{F}_p$, one $\mathbb{F}_{p^3}$ inversion which can be pre-computed and one $\mathbb{F}_p$ multiplication; hence the sextic twisted mapping procedure has no expensive arithmetic operation. On the other hand, quartic twisted mapping requires no arithmetic operation rather it needs some attention since elliptic curve doubling in the twisted curve has a tricky part. The experiment also reveals that sextic twist is preferable since it gives better performance than quartic twist. Performance of such isomorphic mapping can be fully realized when it is applied in some pairing-based protocols. It is obvious that efficiency of Ate-based pairing protocols depends not only on improved scalar multiplication but also on efficient Miller's algorithm and final exponentiation implementation.

6.1.2 Related Works

Pairings are often found in certain extension field $\mathbb{F}_{p^k}$, where p is the prime number, also know as characteristics of the field and the minimum extension degree k is called *embedding* degree. The rational points $E(\mathbb{F}_{p^k})$ are defined over a certain pairing-friendly curve E of embedded extension field of degree k. In [Ara+13], Aranha et al. have presented pairing calculation for 192-bit security level where KSS curve of embedding degree 18 is regarded as one of the good candidates for 192-bit security level. Recently Zhang et al. [ZL12] have shown that the KSS curve of embedding degree 16 are more suitable for 192-bit security level. Therefore this chapter has considered KSS pairing-friendly curves of embedding degree $k = 16$ and 18.

6.1.3 Contribution Outline

Implementing asynchronous pairing operation on a certain pairing-friendly non-supersingular curve requires two rational points typically denoted as P and Q. Generally, P is spotted on the curve $E(\mathbb{F}_p)$, defined over the prime field $\mathbb{F}_p$ and Q is placed in a group of rational points on the curve $E(\mathbb{F}_{p^k})$, defined over $\mathbb{F}_{p^k}$, where k is the *embedding degree* of the pairing-friendly curve. In the case of Kachisa-Schaefer-Scott (KSS) pairing-friendly curve family, $k \geq 16$. Therefore performing pairing calculation on such curves requires calculating elliptic curve operations in higher degree extension field, which is regarded as one of the major bottlenecks to the efficient pairing operation. However, there exists a *twisted* curve of $E(\mathbb{F}_{p^k})$, denoted as $E'(\mathbb{F}_{p^{k/d}})$, where d is the twist degree, on which calculation is faster than the k-th degree extension field. Rational points group defined over such twisted curve has an isomorphic group in $E(\mathbb{F}_{p^k})$. This chapter explicitly shows the mapping procedure between the isomorphic groups in the context of Ate-based pairing over KSS family of pairing-friendly curves. This chapter considers *quartic twist* and *sextic twist* for KSS curve of embedding degree $k = 16$ and $k = 18$ receptively. To evaluate the performance enhancement of isomorphic mapping, this chapter shows the experimental result by comparing the scalar multiplication. The result shows that scalar multiplication in $E(\mathbb{F}_{p^{k/d}})$ is 10 to 20 times faster than scalar multiplication in $E(\mathbb{F}_{p^k})$. It also shows that sextic twist is faster than the quartic twist for KSS curve when parameter settings for 192-bit security level are considered.

6.2 Fundamentals

Most of the fundamentals related to this chapter is already discussed in the previous chapters. In this section we briefly recall KSS family of pairing-friendly curves and twisted property of KSS curve is discussed concisely in this section.

6.2.1 Kachisa-Schaefer-Scott (KSS) Curve Family

In [KSS07], Kachisa, Schaefer, and Scott proposed a family of non super-singular Brezing-Weng pairing-friendly elliptic curves of embedding degree $k = \{16, 18, 32, 36, 40\}$, using elements in the cyclotomic field. Similar to other pairing-friendly curves, *characteristic* p, *Frobenius trace* t and *order* r of these curves are given systematically by using an integer variable also known as mother parameter. In what follows, this chapter considers two curves of this family named as *KSS-16* of embedding degree $k = 16$ and *KSS-18* of $k = 18$.

KSS-18 curve, defined over $\mathbb{F}_{p^{18}}$, is given by the following equation

$$E/\mathbb{F}_{p^{18}} : Y^2 = X^3 + b, \quad b \in \mathbb{F}_p \text{ and } b \neq 0\,, \tag{6.1}$$

where $X, Y \in \mathbb{F}_{p^{18}}$. KSS-18 curve is parameterized by an integer variable u as follows:

$$\begin{aligned} p(u) &= (u^8 + 5u^7 + 7u^6 + 37u^5 + 188u^4 + 259u^3 + 343u^2 + 1763u \\ &\quad +2401)/21, & (6.2a)\\ r(u) &= (u^6 + 37u^3 + 343)/343, & (6.2b)\\ t(u) &= (u^4 + 16u + 7)/7. & (6.2c) \end{aligned}$$

The necessary condition for u is $u \equiv 14 \pmod{42}$ and the ρ value is $\rho = (\log_2 p/\log_2 r) \approx 1.33$.

On the other hand, KSS-16 curve is defined over $\mathbb{F}_{p^{16}}$, represented by the following equation

$$E/\mathbb{F}_{p^{16}} : Y^2 = X^3 + aX, \quad (a \in \mathbb{F}_p) \text{ and } a \neq 0, \tag{6.3}$$

where $X, Y \in \mathbb{F}_{p^{16}}$. Its characteristic p, Frobenius trace t and order r are given the integer variable u as follows:

$$\begin{aligned} p(u) &= (u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 + 625u^2 + 2398u \\ &\quad +3125)/980, && \text{(6.4a)} \\ r(u) &= u^8 + 48u^4 + 625, && \text{(6.4b)} \\ t(u) &= (2u^5 + 41u + 35)/35, && \text{(6.4c)} \end{aligned}$$

where u is such that $u \equiv 25$ or $45 \pmod{70}$ and the ρ value is $\rho = (\log_2 p/\log_2 r) \approx 1.25$.

6.2.2 Extension Field Construction for KSS Curves

Pairing-based cryptography requires performing the arithmetic operation in extension fields of degree $k \geq 6$ [SCA86]. We recall the extension field construction of KSS-18 here and introduce the towering for KSS-18 curve. Since this chapter uses two curves of different extension degree, therefore, the construction process of $\mathbb{F}_{p^{18}}$ and $\mathbb{F}_{p^{16}}$ are represented in the following as a tower of subfields.

6.2.2.1 Towering of $\mathbb{F}_{p^{18}}$ Extension Field

Let $3|(p-1)$, where p is the characteristics of KSS-18 and c is a quadratic and cubic non residue in $\mathbb{F}_p$. In the context of KSS-18, where $k = 18$, $\mathbb{F}_{p^{18}}$ is constructed as tower field with irreducible binomial as follows:

$$\begin{cases} \mathbb{F}_{p^3} = \mathbb{F}_p[i]/(i^3 - c), \\ \mathbb{F}_{p^6} = \mathbb{F}_{p^3}[v]/(v^2 - i), \\ \mathbb{F}_{p^{18}} = \mathbb{F}_{p^6}[\theta]/(\theta^3 - v). \end{cases} \tag{6.5}$$

Here $c = 2$ is considered to be the best choice for efficient extension field arithmetic. From the above towering construction we can find that $i = v^2 = \theta^6$, where i is the basis element of the base extension field $\mathbb{F}_{p^3}$.

6.2.2.2 Towering of $\mathbb{F}_{p^{16}}$ Extension Field

Let the characteristics p of KSS-16 is such that $4|(p-1)$ and z is a quadratic non residue in $\mathbb{F}_p$. By using irreducible binomials, $\mathbb{F}_{p^{16}}$ is constructed for KSS-16 curve as follows:

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 - z), \\ \mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/(\beta^2 - \alpha), \\ \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\gamma]/(\gamma^2 - \beta), \\ \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\omega]/(\omega^2 - \gamma), \end{cases} \tag{6.6}$$

Here $z = 11$ is chosen along with the value of mother parameter u as given in **Table** 6.3.

6.2.3 $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ Groups

In the context of pairing-based cryptography, especially on KSS curve, two additive rational point groups $\mathbb{G}_1, \mathbb{G}_2$ and a multiplicative group $\mathbb{G}_3$ of order r are considered. From [Mor+14], $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\begin{aligned}\mathbb{G}_1 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [1]),\\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]),\\ \mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r,\end{aligned}$$

$$\xi : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3, \tag{6.7}$$

where ξ denotes Ate pairing. In the case of KSS curves, the above $\mathbb{G}_1$ is just $E(\mathbb{F}_p)$. In what follows, rest of this chapter considers $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G}_2$ where $\mathbb{G}_2$ is a subset of $E(\mathbb{F}_{p^{16}})$ and $E(\mathbb{F}_{p^{18}})$ for KSS-16 and KSS-18 curves respectively.

6.2.4 Twist of KSS Curves

Let us consider performing an asynchronous type of pairing operation on KSS curves. Let it be the Ate pairing $\xi(P,Q)$, one of asynchronous variants. P is defined over the prime field $\mathbb{F}_p$ and Q is typically placed on the k-th degree extension field $\mathbb{F}_{p^k}$ on the defined KSS curve. There exists a *twisted curve* with a group of rational points of order r which are isomorphic to the group where rational point $Q \in E(\mathbb{F}_{p^k})$ belongs to. This subfield isomorphic rational point group includes a twisted isomorphic point of Q, typically denoted as $Q' \in E'(\mathbb{F}_{p^{k/d}})$, where k is the embedding degree and d is the twist degree.

Since points on the twisted curve are defined over a smaller field than $\mathbb{F}_{p^k}$, therefore ECA and ECD becomes faster. However, when required in the pairing calculation such as for line evaluation they can be quickly mapped to a point on $E(\mathbb{F}_{p^k})$. Defining such mapping and re-mapping techniques is the main focus of this chapter. Since the pairing-friendly KSS-16 [KSS07] curve has CM discriminant of $D = 1$ and $4|k$, therefore quartic twist is available. For sextic twist, the curve should have $D = 3$ and $6|k$, which exists in KSS-18.

6.2.4.1 Sextic Twist of KSS-18 Curve

When the embedding degree $k = 6e$, where e is positive integer, *sextic* twist is given as follows:

$$E : \quad y^2 = x^3 + b, \quad b \in \mathbb{F}_p, \tag{6.8}$$

$$E'_6 : \quad y^2 = x^3 + bv^{-1}, \tag{6.9}$$

where v is a quadratic and cubic non residue in $E(\mathbb{F}_{p^e})$ and $3|(p^e - 1)$. For KSS-18 curve $e = 3$. Isomorphism between $E'_6(\mathbb{F}_{p^e})$ and $E(\mathbb{F}_{p^{6e}})$, is given as

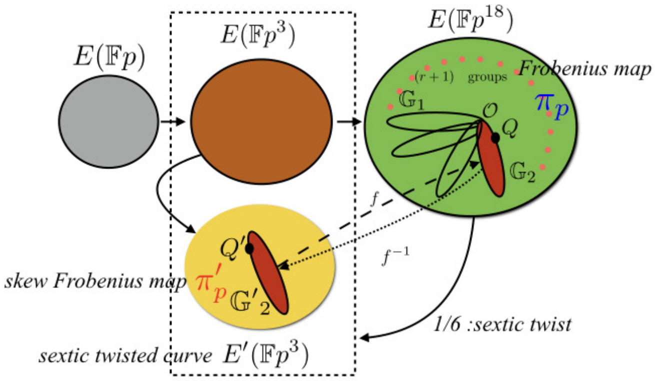


FIGURE 6.1: *sextic twist* in KSS-18 curve.

follows:

$$\psi_6 : \begin{cases} E'_6(\mathbb{F}_{p^e}) \longrightarrow E(\mathbb{F}_{p^{6e}}), \\ (x, y) \qquad \mapsto (xv^{1/3}, yv^{1/2}). \end{cases} \tag{6.10}$$

6.2.4.2 Quartic Twist of KSS-16 Curve

The quartic twist of KSS-16 curve is given as follows:

$$E : \quad y^2 \quad = \quad x^3 + ax, \quad a \in \mathbb{F}_p, \tag{6.11}$$

$$E'_4 : \quad y^2 \quad = \quad x^3 + a\sigma^{-1}x, \tag{6.12}$$

where σ is a quadratic non residue in $E(\mathbb{F}_{p^4})$ and $4|(p-1)$. The Isomorphism between $E'_4(\mathbb{F}_{p^4})$ and $E(\mathbb{F}_{p^{16}})$, is given as follows:

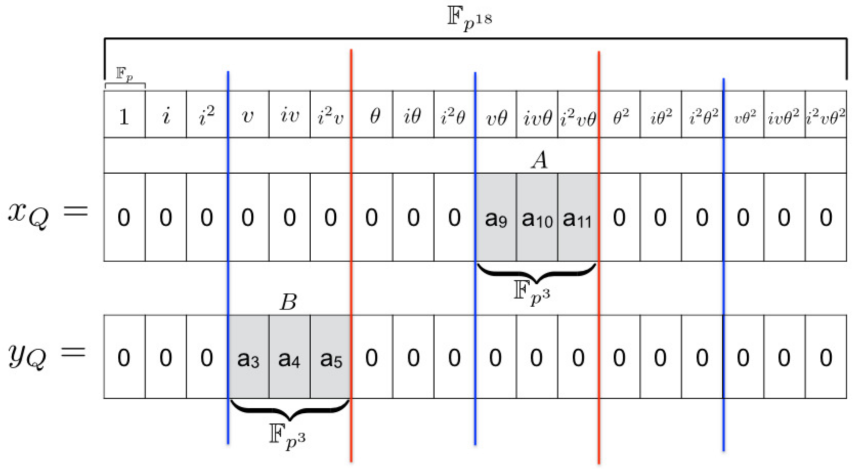
$$\psi_4 : \begin{cases} E'_4(\mathbb{F}_{p^4}) \longrightarrow E(\mathbb{F}_{p^{16}}), \\ (x, y) \qquad \mapsto (x\sigma^{1/2}, y\sigma^{3/4}). \end{cases} \tag{6.13}$$

6.3 Isomorphic Mapping between Q and Q'

This section introduces the proposed mapping procedure of $\mathbb{G}_2$ rational point group to its twisted (quartic and sextic) isomorphic group $\mathbb{G}'_2$ for Ate-based pairing for the considered KSS curves.

6.3.1 Sextic twisted Isomorphic Mapping between $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ and $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$

Figure 6.1 shows an overview of sextic twisted curve $E'(\mathbb{F}_{p^3})$ of $E(\mathbb{F}_{p^{18}})$.



$$a_j \in \mathbb{F}_p, \quad \text{where} \quad a_j = (0, 1, \cdots, 17)$$
$$Q = (x_Q, y_Q) = (Av\theta, Bv) \in \mathbb{F}_{p^{18}}$$
$$Q' = (x'_Q, y'_Q) = (Ai, Bi) \in \mathbb{F}_{p^3}$$

FIGURE 6.2: $Q \in \mathbb{F}_{p^{18}}$ and its sextic twisted isomorphic rational point $Q' \in \mathbb{F}_{p^3}$ structure in KSS-18 curve.

Let us consider E be the KSS-18 curve in base field $\mathbb{F}_{p^3}$ and E' is sextic twist of E' given as follows:

$$E : y^2 = x^3 + b, \tag{6.14}$$
$$E' : y^2 = x^3 + bi, \tag{6.15}$$

where $b \in \mathbb{F}_p$; $x, y, i \in \mathbb{F}_{p^3}$ and basis element i is the quadratic and cubic non residue in $\mathbb{F}_{p^3}$.

In the context of KSS-18 curve, let us consider a rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$. Q has a special vector representation with 18 $\mathbb{F}_p$ elements for each x_Q and y_Q coordinate. Figure 6.2 shows the structure of the coefficients of $Q \in \mathbb{F}_{p^{18}}$ and its sextic twisted isomorphic rational point $Q' \in \mathbb{F}_{p^3}$ in KSS-18 curve. Among 18 elements, there are 3 continuous nonzero $\mathbb{F}_p$ elements. The others are zero. However, the set of these nonzero elements belongs to a $\mathbb{F}_{p^3}$ field.

This chapter considers parameter given in **Table** 6.2 for KSS-18 curve where mother parameter u = 65-bit and characteristics p = 511-bit. In such consideration, Q is given as $Q = (Av\theta, Bv)$, showed in Figure 6.2, where $A, B \in \mathbb{F}_{p^3}$ and v and θ are the basis elements of $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{18}}$ respectively.

Let us consider the sextic twisted isomorphic subfield rational point of Q as $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^3})$. Considering x' and y' as the coordinates of Q', we can map the rational point $Q = (Av\theta, Bv)$ to the rational point $Q' = (x', y')$ as follows.

Multiplying both side of Eq.(6.15) with θ^{-6}, where $i = \theta^6$ and $v = \theta^3$.

$$E' : \left(\frac{y}{\theta^3}\right)^2 = \left(\frac{x}{\theta^2}\right)^3 + b. \tag{6.16}$$

θ^{-2} of Eq.(6.16) can be represented as follows:

$$\begin{aligned} \theta^{-2} &= i^{-1}i\theta^{-2}, \\ &= i^{-1}\theta^4, \end{aligned} \tag{6.17a}$$

and multiplying i with both sides.

$$\theta^4 = i\theta^{-2}. \tag{6.17b}$$

Similarly θ^{-3} can be represented as follows:

$$\begin{aligned} \theta^{-3} &= i^{-1}i\theta^{-3}, \\ &= i^{-1}\theta^3. \end{aligned} \tag{6.17c}$$

Multiplying i with both sides of Eq.(6.17c) we get θ^3 as,

$$\theta^3 = i\theta^{-3}, \tag{6.17d}$$

6.3.1.1 Q to Q' Mapping in KSS-18

Let us represent $Q = (Av\theta, Bv)$ as follows:

$$Q = (A\theta^4, B\theta^3), \quad \text{where } v = \theta^3. \tag{6.18}$$

From Eq.(6.17b) and Eq.(6.17d), we substitute $\theta^4 = i\theta^{-2}$ and $\theta^3 = i\theta^{-3}$ in Eq.(6.18) as follows:

$$Q = (Ai\theta^{-2}, Bi\theta^{-3}), \tag{6.19}$$

where $Ai = x'$ and $Bi = y'$ are the coordinates of $Q' = (x', y') \in \mathbb{F}_{p^3}$. Which implies that we can map $Q \in \mathbb{F}_{p^{18}}$ to $Q' \in \mathbb{F}_{p^3}$ by first selecting the 3 nonzero $\mathbb{F}_p$ coefficients of each coordinate of Q. Then these nonzero $\mathbb{F}_p$ elements form a $\mathbb{F}_{p^3}$ element. After that multiplying the basis element i with that $\mathbb{F}_{p^3}$ element, we get the final $Q' \in \mathbb{F}_{p^3}$. From the structure of $\mathbb{F}_{p^{18}}$, given in Eq.(6.5), this mapping has required no expensive arithmetic operation. Multiplication by the basis element i in $\mathbb{F}_{p^3}$ can be done by 1 bitwise left shifting since $c = 2$ is considered for towering in Eq.(6.5).

6.3.1.2 Q' to Q Mapping in KSS-18

The reverse mapping $Q' = (x', y') \in \mathbb{F}_{p^3}$ to $Q = (Av\theta, Bv) \in \mathbb{F}_{p^{18}}$ can be obtained as from Eq.(6.17a), Eq.(6.17c) and Eq.(6.16) as follows:

$$\begin{aligned} xi^{-1}\theta^4 &= Av\theta, \\ yi^{-1}\theta^3 &= Bv, \end{aligned}$$

which resembles that $Q = (Av\theta, Bv)$. Therefore it means that multiplying i^{-1} with the Q' coordinates and placing the resulted coefficients in the corresponding position of the coefficients in Q, will map Q' to Q. This mapping costs one $\mathbb{F}_{p^3}$ inversion of i which can be pre-computed and one $\mathbb{F}_p$ multiplication.

6.3.2 Quartic Twisted Isomorphic Mapping

For quartic twisted mapping first we need to obtain certain ration point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ of subgroup order r. One necessary condition for obtaining such Q is $r^2 \mid \#E(\mathbb{F}_{p^{16}})$, where $\#E(\mathbb{F}_{p^{16}})$ is the number of rational points in $E(\mathbb{F}_{p^{16}})$. But it is carefully observed that $\#E(\mathbb{F}_{p^{16}})$ is not divisible by r^2 when r is given by Eq.(6.4b). Therefore polynomial of r, given in [KSS07] is divided as follows:

$$r(u) = (u^8 + 48u^4 + 625)/61250, \tag{6.21}$$

to make it dive $\#E(\mathbb{F}_{p^{16}})$ completely.

Let us consider the rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ and its quartic twisted rational point $Q' \in \mathbb{G}_2 \subset E'(\mathbb{F}_{p^4})$. Rational point Q has a special vector representation given in **Table** 6.1.

TABLE 6.1: Vector representation of $Q = (x_Q, y_Q) \in \mathbb{F}_{p^{16}}$

	1	α	β	$\alpha\beta$	γ	$\alpha\gamma$	$\beta\gamma$	$\alpha\beta\gamma$	ω	$\alpha\omega$	$\beta\omega$	$\alpha\beta\omega$	$\gamma\omega$	$\alpha\gamma\omega$	$\beta\gamma\omega$	$\alpha\beta\gamma\omega$
x_Q	0	0	0	0	n_4	n_5	n_6	n_7	0	0	0	0	0	0	0	0
y_Q	0	0	0	0	0	0	0	0	0	0	0	0	n_{12}	n_{13}	n_{14}	n_{15}

From **Table** 6.1 co-ordinates of $Q = (x_Q, y_Q) \in \mathbb{F}_{p^{18}}$ is obtained as $Q = (x_Q, y_Q) = (\gamma x_{Q'}, \omega\gamma y_{Q'})$ where $x_{Q'}, y_{Q'}$ are the co-ordinates of the rational point Q' in the twisted curve. Now let's find the twisted curve of Eq.(6.3) in $\mathbb{F}_{p^4}$ as follows:

$$\begin{aligned}
(\omega\gamma y_{Q'})^2 &= (\gamma x_{Q'})^3 + a(\gamma x_{Q'}), \\
\gamma\beta y_{Q'}^2 &= \gamma\beta x_{Q'}^3 + a\gamma x_{Q'}, \\
y_{Q'}^2 &= x_{Q'}^3 + a\beta^{-1}x_{Q'}, \quad \text{multiplying } (\gamma\beta)^{-1} \text{ both sides.}
\end{aligned} \tag{6.22}$$

The twisted curve of E' is obtained as $y^2 = x^3 + a\beta^{-1}x$ where β is the basis element in $\mathbb{F}_{p^4}$. There is a tricky part that needs attention when calculating the ECD in $E'(\mathbb{F}_{p^4})$ presented in the following equation.

$$\lambda = (3x_{Q'}^2 + \mathbf{a})(2y_{Q'})^{-1}, \tag{6.23}$$

where $\mathbf{a} \in \mathbb{F}_{p^4}$, since $\mathbf{a} = a\beta^{-1}$ and $\beta \in \mathbb{F}_{p^4}$. The calculation of $\mathbf{a} = a\beta^{-1}$ is given as follows:

$$\begin{aligned}
a\beta^{-1} &= (a + 0\alpha + 0\beta + 0\alpha\beta)\beta^{-1}, \\
&= z^{-1}a\alpha\beta \quad \text{where } \alpha^2 = z
\end{aligned} \tag{6.24}$$

Now let us denote the quartic mapping as follows:

$$Q = (x_Q, y_Q) = (\gamma x_{Q'}, \omega\gamma y_{Q'}) \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}}) \longmapsto Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4}).$$

For mapping from Q to Q' no extra calculation is required. By picking the non-zero coefficients of Q and placing it to the corresponding basis position is enough to get Q'. Similarly, re-mapping from Q' to Q can also be done without any calculation rather multiplying with basis elements.

6.4 Result Analysis

The main focus of this proposed mapping is to find out the isomorphic mapping of two well-known pairing-friendly curves, KSS-16 and KSS-18. In order to determine the advantage of the proposal, this chapter has implemented 3 well-known elliptic curve scalar multiplication method named as the binary method, Montgomery ladder method, and sliding-window method.

For the experiment first we have applied the proposed mapping technique to map rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^k})$ to its isomorphic point $Q' \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^{k/d}})$ in both KSS curves. After that we performed the scalar multiplication of Q'. Then the resulted points are re-mapped to $\mathbb{G}_2$ in $\mathbb{F}_{p^k}$. Lets define this strategy as ***with mapping***. On the other hand, we have performed scalar multiplication of Q without mapping which is denoted as ***w/o mapping***.

In the experiment, after many careful searches, the mother parameter u is selected to find out $\mathbb{G}_2$ rational point Q for KSS-18 curve. On the other hand, for KSS-16 curve, parameters are given by Loubna et al. [GF16a]. In pairing-based cryptosystems, both KSS-16 and KSS-18 are regarded as good candidates for implementing 192-bit security. Therefore, while choosing parameters for the experiment, this chapter has adapted 192-bit security level. But the main focus of this chapter is not to find out efficient parameters for certain security levels. The main purpose of the selected the parameters is to compare the twisted isomorphic mappings on the nominated curves at standard security levels.

Table 6.2 and **Table** 6.3 show the parameters used in the experiment. **Table** 6.4 shows the experiment environment, used to evaluate the usefulness of the proposed mapping. In the experiment, 100 scalar numbers of size less than order r is generated randomly and then scalar multiplication is calculated for both cases. Average value of execution time in [ms] is considered for comparison. **Table** 6.5 shows the settings considered during the experiment. The comparative result is shown in **Table** 6.6.

Parameter of KSS curves are given in decimal value used for evaluating the mapping efficiency in the experiment.

Analyzing **Table** 6.6, we can find that scalar multiplication on the sextic twisted KSS-18 curve using the proposed mapping technique is more than 20 times faster than scalar multiplication without the proposed mapping. On

TABLE 6.2: KSS-18 Parameters

$y^2 =$	$x^3 + 11$	bit size
$u =$	23058430092138432950	65
$p =$	380556013753003852484338059727997572538865139076812 970560732143111526346817611942575176069026109216559 8021019048849831001675531254097766654664544068613131	511
$r =$	4382120271066581232104344084955320374849908135951851 5268755202336574860904936668100704293777799119708528 7495125001	378
$t =$	4038507576637353290391809403638366577735736214369368 5385569578231170388739601	255

TABLE 6.3: KSS-16 Parameters

$y^2 =$	$x^3 + 17x$	bit size
$u =$	1266366845779935	51
$p =$	108235379323342249430403752839634417782861787922010 5831937449880701267192580688017668298801139820714475 1031509661694254867934067997516170939905853281	492
$r =$	10798667332013548302444682759479306650777434983428752 081956116352950853566245965258810783523700606376869560 4209229873	386
$t =$	186105672625714085505985902011330755941369113096635058 9745550013872708970	247

the other hand, in the quartic twisted KSS-16 curve, scalar multiplication becomes at most 10 times faster after applying proposed mapping techniques than no mapping. Another important difference is sextic twisted mapped points take less time for scalar multiplication in both experiment environments. Therefore we can certainly say sextic twist over KSS-18 is more efficient than the quartic twisted KSS-16 curve for implementing pairing operations.

In the experiment we have used two execution environments; such as PC and iPhone with different CPU frequencies. In both environments only one processor core is utilized. The ratio of CPU frequencies of iPhone and PC is about $1.84/2.7 \approx 0.68$. The result shows that the ratio of execution time of PC and iPhone without mapping for KSS-18 curve is around 0.62 to 0.66. Which is close to CPU frequency ratio. On the other hand, the ratio of execution time with mapping of KSS-18 curve is also around 0.6. For KSS-16 curve, the ratio with no mapping case is more than 0.8 and for mapping case it is around 0.7 to 0.9. Since PC and iPhone has different processor architectures therefore it's frequency ratio has modest relation with the execution time ratio. The ratio may also be effected by the other processes, running in certain environment

TABLE 6.4: Computational Environment

	PC	iPhone6s
CPU *	2.7 GHz Intel Core i5	Apple A9 Dual-core 1.84 GHz
Memory	16 GB	2 GB
OS	Mac OS X 10.12.3	iOS 10.2.1
Compiler	gcc 4.2.1	gcc 4.2.1
Programming Language	C	Objective-C, C
Library	GNU MP 6.1.1[Gt15]	GNU MP 6.1.1

*Only single core is used from two cores.

TABLE 6.5: Additional settings used in the experiment

	KSS-18	KSS-16
Number of sample s	100	100
Average bit size of s	377-bit	385-bit
Average hamming weight of s	187	193
Window size for sliding window method	4	4
No. of Pre-computed ECA in sliding window	14	14
Perceived level of security	192-bit	192-bit

during the experiment time.

The main focus of this experiment is to evaluate the acceleration ratio of scalar multiplication by applying the proposed mapping on $\mathbb{G}_2$ rational point group of the nominated KSS curves. The experiment does not focus on efficiently implementing scalar multiplication for certain environment. There are other pairing-friendly curves such as BLS-12, BLS-24 [FST10] where sextic twist is available. As our future work, we will try to apply the proposed mapping on those curves.

6.5 Conclusion

In this chapter, we have demonstrated isomorphic mapping procedure of $\mathbb{G}_2$ rational point group to its sextic and quartic twisted subfield isomorphic rational point group $\mathbb{G}_2'$ and its reverse mapping for KSS-18 and KSS-16 curves in the context of Ate-based pairing.

We have also evaluated the advantage of such mapping by applying binary scalar multiplication, Montgomery ladder, and sliding- window method on twisted isomorphic rational points in $\mathbb{G}_2'$. Then result of scalar multiplication in $\mathbb{G}_2'$ can accelerate the scalar multiplication in $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{18}})$ by 20 to 10 times than scalar multiplication of $\mathbb{G}_2$ rational point directly in $\mathbb{F}_{p^{18}}$ and $\mathbb{F}_{p^{16}}$.

TABLE 6.6: Comparative result of average execution time in [ms] for scalar multiplication

	Average execution time [ms] comparison			
	KSS-18		KSS-16	
	PC	iPhone 6s	PC	iPhone 6s
Binary with mapping	5.7×10^1	8.2×10^1	1.3×10^2	1.4×10^2
Binary w/o mapping	1.2×10^3	1.8×10^3	1.2×10^3	1.3×10^3
Montgomery ladder with mapping	7.1×10^1	1.1×10^2	1.7×10^2	1.8×10^2
Montgomery ladder w/o mapping	1.5×10^3	2.4×10^3	1.6×10^3	1.8×10^3
Sliding-window with mapping	4.9×10^1	7.5×10^1	1.0×10^2	1.3×10^2
Sliding-window w/o mapping	1.0×10^3	1.6×10^3	1.0×10^3	1.2×10^3

Chapter 7

ICCIT 2016

A Consideration of Towering Scheme for Efficient Arithmetic Operation over Extension Field of Degree 18

Barreto-Naehrig (BN) curve is a well studied pairing friendly curve of embedding degree 12, that uses arithmetic in $\mathbb{F}_{p^{12}}$. Therefore the arithmetic of $\mathbb{F}_{p^{12}}$ extension field is well studied. In this thesis, we have proposed an efficient approach of arithmetic operation over the extension field of degree 18 by towering. $\mathbb{F}_{p^{18}}$ extension field arithmetic is considered to be the basis of implementing the next generation pairing based security protocols. We have proposed to use $\mathbb{F}_p$ element to construct irreducible binomial for building tower of extension field up to $\mathbb{F}_{p^6}$, where conventional approach uses the root of previous irreducible polynomial to create next irreducible polynomials. Therefore using $\mathbb{F}_p$ elements in irreducible binomial construction, reduces the number of multiplications in $\mathbb{F}_p$ to calculate inversion and multiplication over $\mathbb{F}_{p^{18}}$, which effects acceleration in total arithmetic operation over $\mathbb{F}_{p^{18}}$.

7.1 Introduction

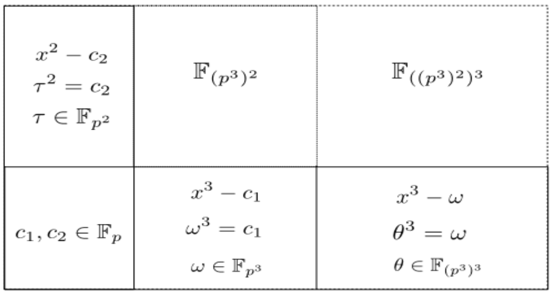
The emerging information security of computer system stands on the strong base of cryptography. Compared to RSA cryptography, elliptic curve cryptography [Kob87] gained much attention for its faster key generation, shorter key size with same security level and less memory and computing power consumption. Intractability of Elliptic Curve Discrete Logarithm Problem (ECDLP) encourages many innovative cryptographic protocols. At the very beginning of the twenty first century, a cyptosystems based on elliptic curve pairing was proposed independently by Sakai et al. [SK03] and Joux [Jou04]. Since then this pairing based cryptosystem has unlocked several novel ideas to researchers such as Identity based encryption scheme explained by Boneh et al. [BF01]. In addition, group signature authentication [BBS04],[NF05] and broadcast encryption [BGW05] has increased the popularity of pairing based cryptography. Pairings such as Weil[**Weil_p**], Tate and Optimal-ate [Ver10], Eta [HSV06] and χ-Ate [Nog+08] pairings has gained much attention in recent years. Pairing is a bilinear map from two rational point groups denoted by $\mathbb{G}_1$ and $\mathbb{G}_2$ to a multiplicative group denoted by $\mathbb{G}_3$ [SCA86]. It is generally denoted by $\mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_3$. In addition, these groups are defined over a certain extension field $\mathbb{F}_{p^k}$, where p is the prime number, also called characteristics and k is the extension degree, especially called *embedding* degree. Therefore it is important to efficiently construct extension field arithmetic in order to make pairing based cryptography efficient.

In prairing based cryptography, rational points are defined over a certain pairing friendly elliptic curve. Let $E(\mathbb{F}_{p^k})$ be a set of rational points such as (x, y), $x, y \in \mathbb{F}_{p^k}$ lies in the elliptic curve E, defined over extension field $\mathbb{F}_{p^k}$ of embedding degree k. Security level of pairing based cryptography depends on the sizes of both r and p^k, where r denotes the largest prime number that divides the order of $E(\mathbb{F}_p)$. It is said that the next generation pairing-based cryptography needs $\log_2 r \approx 256$ and $\log_2 p^k \approx 3000$ to 5000. Supposing the most efficient case of $\rho = (\log_2 p)/(\log_2 r) = 1$, k needs to be 12 to 20. In this thesis we are considering $k = 18$ and 18 degree pairing friendly curve described in [FST06].

While using pairing based protocols, it is required to perform arithmetic in higher fields, such as $\mathbb{F}_{p^k}$ for moderate value of k [SCA86]. It is important to represent the field in such a way that, the arithmetic can be performed efficiently. One of the most efficient way is to use the tower of extension field [BS09]. Which explains that, higher level computations can be calculated as a function of lower level computations. Because of that, efficient implementation of lower level arithmetic results in the good performance of arithmetic in higher degree fields. Recently the implementation of pairing based cryptosystems for different low power and mobile devices are increasing. Moreover, the hardware capabilities of the embedded devices are improving which can make pairing implementations efficient and faster. Therefore efficiency of extension field arithmetic is important to improve the performance of pairing. In this thesis we have presented an efficient way to construct $\mathbb{F}_{p^{18}}$ extension field and performing arithmetic operation on that field. In current approach of constructing extension field by towering, root of previous irreducible polynomial is used to construct the irreducible polynomial for next extension field. In our proposal, element in prime field $\mathbb{F}_p$ is used to construct the irreducible polynomial for the first two extension field and for in the last extension field root of base extension field is used for constructing irreducible polynomial.

7.2 Preliminaries

In this section we will go though the background how tower of extension field is constructed in practice and some basic idea of basis to construct extension field.



$x^2 - c_2$ $\tau^2 = c_2$ $\tau \in \mathbb{F}_{p^2}$	$\mathbb{F}_{(p^3)^2}$	$\mathbb{F}_{((p^3)^2)^3}$
$c_1, c_2 \in \mathbb{F}_p$	$x^3 - c_1$ $\omega^3 = c_1$ $\omega \in \mathbb{F}_{p^3}$	$x^3 - \omega$ $\theta^3 = \omega$ $\theta \in \mathbb{F}_{(p^3)^3}$

FIGURE 7.1: Construction overview of $\mathbb{F}_{((p^3)^2)^3}$

7.2.1 Basis of extension field and towering

In order to construct the arithmetic operations in $\mathbb{F}_{p^k}$, we generally need an irreducible polynomial $f(x)$ of degree k over $\mathbb{F}_p$. Let ω be a zero of $f(x)$, that is $\omega \in \mathbb{F}_{p^k}$, then the following set forms a basis of $\mathbb{F}_{p^k}$ over $\mathbb{F}_p$

$$\left\{1, \omega, \omega^2, \cdots, \omega^{k-1}\right\}, \tag{7.1}$$

which is known as polynomial basis. An arbitrary element A in $\mathbb{F}_{p^k}$ is written as

$$A = a_0 + a_1\omega + a_2\omega^2 + \cdots + a_{k-1}\omega^{k-1}. \tag{7.2}$$

The vector representation of A is $v_A = (a_0, a_1, a_2, \cdots a_{k-1})$. Multiplication and inversion in $\mathbb{F}_{p^k}$ are carried out by using the relation $f(\omega) = 0$, and therefore $f(x)$ is called the *modular reduction polynomial* of $\mathbb{F}_{p^k}$. Frobenious mapping should be efficient while calculating conjugates of ω.

Extension field of $\mathbb{F}_{p^k}$ with moderate value of k, such as $k \geq 6$ needs to be represented as a tower of sub extension field to improve pairing calculation. In [Lan08] explained tower of extension by using irreducible binomial. In case of Barreto-Naehrig (BN) curves [BN06], where $k = 12$, towering extension field with irreducible binomial is represented as follows:

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_q[\omega]/(\omega^2 - \beta), \text{where } \beta = c \text{ and } c \in \mathbb{F}_p. \\ \mathbb{F}_{p^6} = \mathbb{F}_{q^2}[\tau]/(\tau^3 - \xi), \text{where } \xi = \omega + 1. \\ \mathbb{F}_{p^{12}} = \mathbb{F}_{q^6}[\theta]/(\theta^2 - \tau), \text{where } \tau = \xi. \end{cases}$$

Here p needs to be prime and $p - 1$ needs to be divisible by 4 and c should be quadratic and cubic non residue over $\mathbb{F}_p$.

In this section we will construct the extension field of degree 18 as a tower of three sub extension field. The extension field $\mathbb{F}_{p^3}$ is the sextic twist of $\mathbb{F}_{p^{18}}$. Therefore its is considered as the base field for constructing $\mathbb{F}_{((p^3)^2)^3}$ extension field in our proposal. Figure 7.1 shows the top level overview of our proposal to construct the tower of extension fields.

7.2.2 Arithmetic operations over extension field $\mathbb{F}_{p^3}$

At first, let us consider arithmetic operations in $\mathbb{F}_{p^3}$, which is the degree 3 extension field over $\mathbb{F}_p$. In order to perform arithmetic operations in $\mathbb{F}_{p^3}$, we generally need an irreducible polynomial $f(x)$ of degree 3 over $\mathbb{F}_p$. Specifically irreducible binomial is efficient to use as reduction modular polynomial. In order to obtain such binomial, Legendre symbol $\left({}^{c_1}/_{p}\right)$ is convenient. Let us consider $3|(p-1)$ and a non-zero element $c_1 \in \mathbb{F}_p$.

$$c_1^{\frac{p-1}{3}} = \begin{cases} 0 & c_1 = 0, \\ 1 & \text{CPR}, \\ otherwise & \text{CPNR}, \end{cases} \tag{7.3}$$

where CPR and CPNR are abbreviations of cubic power residue and cubic power non residue, respectively. If c_1 does not have any cubic root in $\mathbb{F}_p$, $f(x) = x^3 - c_1$ becomes an irreducible binomial over $\mathbb{F}_p$. Let ω be a zero of $f(x)$, which is an element in $\mathbb{F}_{p^3}$. Therefore the set $\left\{1, \omega, \omega^2\right\}$ forms a polynomial basis of $\mathbb{F}_{p^3}$ over $\mathbb{F}_p$. Now let us consider two arbitrary element **a, b** in $\mathbb{F}_{p^3}$, can be represented as follows:

$$\begin{aligned} \mathbf{a} &= a_0 + a_1\omega + a_2\omega^2, \\ \mathbf{b} &= b_0 + b_1\omega + b_2\omega^2, \\ & \quad a_i, b_j \in \mathbb{F}_p. \end{aligned}$$

7.2.2.1 Addition and subtraction in $\mathbb{F}_{p^3}$

Addition, subtraction within the elements and multiplication by a scalar with any element in $\mathbb{F}_{p^3}$ are carried out by coefficient wise operations over $\mathbb{F}_p$ as follows,

$$\mathbf{a} \pm \mathbf{b} = (a_0 \pm b_0, a_1 \pm b_1, a_2 \pm b_2), \tag{7.4}$$

$$k\mathbf{a} = (ka_0, ka_1, ka_2),\ k \in \mathbb{F}_p. \tag{7.5}$$

7.2.2.2 Multiplication in $\mathbb{F}_{p^3}$

Multiplication of two arbitrary vectors is performed as follows:

$$\begin{aligned} \mathbf{ab} &= (a_0 + a_1\omega + a_2\omega^2)(b_0 + b_1\omega + b_2\omega^2) \\ &= a_0b_0 + (a_0b_1 + a_1b_0)\omega + (a_0b_2 + a_1b_1 + a_2b_0)\omega^2 \\ &\quad +(a_1b_2 + a_2b_1)\omega^3 + a_2b_2\omega^4. \end{aligned} \tag{7.6}$$

Here in Eq.(7.6), there are 9 multiplications and 4 additions in $\mathbb{F}_p$. To reduce the number of multiplications in Eq.(7.6), we apply Fast Polynomial Multiplication introduced in [BP01] as follows:

$$\begin{aligned}
A_0 &= a_0b_0\\
A_1 &= a_1b_1\\
A_2 &= a_2b_2\\
A_3 &= (a_0+a_1)(b_0+b_1)\\
A_4 &= (a_0+a_2)(b_0+b_2)\\
A_5 &= (a_1+a_2)(b_1+b_2), \qquad (7.7)
\end{aligned}$$

where $A_i, i = 0, 1, \cdots, 5$ are the auxiliary products. Let us consider $\mathbf{ab} = t(\omega) = \sum_{i=0}^{4} t_i\omega^i$. Now we can represent the coefficients $t(\omega)$ as only additions and subtractions of A_i,

$$\begin{aligned}
t_0 &= A_0\\
t_1 &= A_3 - A_1 - A_0\\
&= (a_0b_0 + a_0b_1 + a_1b_0 + a_1b_1) - a_1b_1 - a_0b_0\\
t_2 &= A_4 - A_2 - A_0 + A_1\\
&= (a_0b_0 + a_2b_0 + a_0b_2 + a_2b_2) - a_2b_2 - a_0b_0 + a_1b_1\\
t_3 &= A_5 - A_1 - A_2\\
&= (a_1b_1 + a_1b_2 + a_2b_1 + a_2b_2) - a_1b_1 - a_2b_2\\
t_4 &= A_2. \qquad (7.8)
\end{aligned}$$

Considering subtractions as additions, from the above equations we find that only 6 multiplications and 13 additions are required in $\mathbb{F}_p$ for multiplying two arbitrary vectors in $\mathbb{F}_{p^3}$. Therefore, compared to Eq.(7.6) the above method will accelerate the vector multiplication, since in most processors multiplication is slower than addition. Substituting $\omega^3 = c_1$ in Eq.(7.6), owing to the fact that $f(\omega) = 0$ of the irreducible binomial $f(x) = x^3 - c_1$; $\mathbf{ab}$ becomes as follows:

$$\begin{aligned}
\mathbf{ab} &= t_0 + t_1\omega + t_2\omega^2 + t_3\omega^3 + t_4\omega^4\\
&= (t_0 + c_1t_3) + (t_1 + c_1t_4)\omega + t_2\omega^2. \qquad (7.9)
\end{aligned}$$

Here it requires 2 more $\mathbb{F}_p$ additions. Multiplication with c_1 will not increase the number of multiplications in $\mathbb{F}_p$ since c_1 is small such as 2 and it can be achieved using bit wise shifting. Finally 6 multiplications and 15 additions are required in $\mathbb{F}_p$ to multiply two elements in $\mathbb{F}_{p^3}$.

7.2.2.3 Squaring in $\mathbb{F}_{p^3}$

Squaring of an $\mathbb{F}_{p^3}$ element A is performed by applying Chung-Hasan method [CH07] as following.

$$\begin{aligned} A^2 &= (a_0 + a_1\omega + a_2\omega^2)^2 \\ &= a_0^2 + 2c_1a_1a_2 + [2a_0a_1 + c_1a_2^2]\omega + [(a_0 + a_1 + a_2)^2 \\ &\quad -(a_0^2 + a_2^2 + 2a_1a_2 + 2a_0a_1)]\omega^2. \end{aligned} \tag{7.10}$$

In what follows, let us consider Eq.(7.10) be written as $\mathbf{AB} = S_1 + S_2\omega + S_3\omega^2$ and the coefficients are expressed as Eq.(7.11). The following terms can be pre-calculated to reduce the number of operations. $T_1 = 2a_1$, $T_2 = a_0^2$, $T_3 = a_2^2$, $T_4 = T_1a_2$, $T_5 = T_1a_0$, $T_6 = (a_0 + a_1 + a_2)^2$.

$$S_1 = T_2 + c_1T_4, \tag{7.11a}$$

$$S_2 = T_5 + c_1T_3, \tag{7.11b}$$

$$S_3 = T_6 - (T_2 + T_3 + T4 + T_5). \tag{7.11c}$$

When $c_1 = 2$, the operation cost of a squaring in $\mathbb{F}_{p^3}$ is 2 multiplications, 3 squaring and 8 additions in $\mathbb{F}_p$ and 2 bit wise left shifting.

7.2.2.4 Vector inversion in $\mathbb{F}_{p^3}$

The inverse element $\mathbf{a}^{-1} \in \mathbb{F}_{p^3}$, can be easily calculated using Frobenius mapping (FM) $\pi(\mathbf{a})$. At first we find the conjugates $\mathbf{a}^p$, $\mathbf{a}^{p^2}$ of $\mathbf{a}$ by applying FM. Then the inverse element $\mathbf{a}^{-1}$ is calculated as follows.

$$\mathbf{a}^{-1} = n(\mathbf{a})^{-1}(\mathbf{a}^p\mathbf{a}^{p^2}), \tag{7.12}$$

where $n(\mathbf{a}) = (\mathbf{a}\mathbf{a}^p\mathbf{a}^{p^2}) \in \mathbb{F}_p^*$ is the product of conjugates. Conjugate $\mathbf{a}^p = (a_0 + a_1\omega + a_2\omega^2)^p$ can be easily calculated as follows:

$$\begin{aligned} (a_0 + a_1\omega + a_2\omega^2)^p &= (a_0 + a_1\omega)^p + (a_2\omega^2)^p \\ &= a_0 + a_1(\omega^3)^{\frac{p-1}{3}}\omega \\ &\quad +a_2((\omega^3)^{\frac{p-1}{3}})^2\omega^2 \\ &= a_0 + a_1(c_1)^{\frac{p-1}{3}}\omega \\ &\quad +a_2((c_1)^{\frac{p-1}{3}})^2\omega^2 \\ &= a_0 + a_1c_1'\omega + a_2c_1''\omega^2 \\ &= a_0 + a_1'\omega + a_2'\omega^2, \end{aligned} \tag{7.13}$$

where $a_1', a_2' \in \mathbb{F}_p$ and $c_1' = (c_1)^{\frac{p-1}{3}}$ is already known from Eq.(7.3) and $c_1'' = (c_1')^2$ can be precalculated. In the above computation, 2 multiplications in $\mathbb{F}_p$ is required. Now the other conjugate $\mathbf{a}^{p^2}$ can be calculated with the same number of operations according to the above procedure as follows:

$$\begin{aligned}\mathbf{a}^{p^2} &= (\mathbf{a}^p)^p \\ &= (a_0 + a_1'\omega + a_2'\omega^2)^p \\ &= a_0 + a_1'c_1'\omega + a_2'c_1''\omega^2 \\ &= a_0 + a_1''\omega + a_2''\omega^2, \end{aligned} \tag{7.14}$$

where $a_1'', a_2'' \in \mathbb{F}_p$. Before calculating $n(\mathbf{a})$ we first calculate the multiplication of $(\mathbf{a}^p\mathbf{a}^{p^2})$ like Eq.(7.6) as follows

$$\mathbf{a}^p\mathbf{a}^{p^2} = (a_0 + a_1'\omega + a_2'\omega^2)(a_0 + a_1''\omega + a_2''\omega^2). \tag{7.15}$$

Now let us consider the following representation.
$\mathbf{T} = \mathbf{a}^p\mathbf{a}^{p^2} = (t_0, t_1, t_2), \quad n(\mathbf{a}) = s = \mathbf{aT},$
Thereby the inversion of $\mathbf{a}$ can be expressed as $\mathbf{a}^{-1} = s^{-1}\mathbf{T}$. The vector representation of the non-zero scalar s is written as $s = (s, 0, 0)$. In addition, $\mathbf{a}^p$ and $\mathbf{a}^{p^2}$ is represented by the following equations by using the relation $c_1'^2 + c_1' + 1 = 0$, where $c_1'^3 = 1$.

$$\mathbf{a}^p = (a_0, c_1'a_1, c_1'^2a_2) = (a_0, c_1'a_1, -a_2 - c_1'a_2), \tag{7.16a}$$

$$\mathbf{a}^{p^2} = (a_0, c_1'^2a_1, c_1'a_2) = (a_0, -a_1 - c_1'a_1, c_1'a_2). \tag{7.16b}$$

Now let us consider the variables $T_0 \sim T_5$ as following expressions.

$$\begin{aligned} T_0 &= a_0^2, \\ T_1 &= a_1^2, \\ T_2 &= a_2^2, \\ T_3 &= (c_1'a_1 + c_1'^2a_2)(c_1'^2a_1 + c_1'a_2) \\ &= a_1^2 - a_1a_2 + a_2^2 \\ T_4 &= (a_0 + c_1'a_1)(a_0 + c_1'^2a_1) \\ &= a_0^2 - a_0a_1 + a_1^2 \\ T_5 &= (a_0 + c_1'^2a_2)(a_0 + c_1'a_2) \\ &= a_0^2 - a_0a_2 + a_2^2. \end{aligned}$$

The elements of $\mathbf{T} = (t_0, t_1, t_2)$ can be obtained as follows:

$$\begin{aligned} t_1 &= T_0 + c_1(T_3 - T_1 - T_2) \\ &= a_0^2 - c_1a_1a_2, && (7.18a) \\ t_2 &= T_4 - T_0 - T_1 + c_1T_2 \\ &= c_1a_2^2 - a_0a_1, && (7.18b) \\ t_3 &= T_5 - T_0 - T_2 + T_1 \\ &= a_1^2 - a_0a_2. && (7.18c) \end{aligned}$$

The calculation cost of t_1, t_2, t_3 is 3 multiplications, 3 squaring, 3 additions and 2 bit shifting. The vector multiplication for getting $s = \mathbf{aT} = (s, 0, 0)$ can be done by calculating $s = a_0b_0 + c_1(a_1b_2 + a_2b_1)$ which costs 3 multiplication, 2 additions and 1 bit shifting.

Finally the inversion of the scalar s and multiplication by the inverse of scalar s with vector $\mathbf{T} = \mathbf{a}^p\mathbf{a}^{p^2}$ can be obtained by distributive law which takes 1 inversion and 3 multiplication in $\mathbb{F}_p$. Therefore the total cost of inversion is 9 multiplications, 3 squaring, 5 additions, 3 bit shifting and 1 inversion in $\mathbb{F}_p$.

7.2.3 Arithmetic operations over extension field $\mathbb{F}_{(p^3)^2}$

$\mathbb{F}_{(p^3)^2}$ is constructed with the irreducible binomial $g(x) = x^2 - c_2$ where $c_2 \in \mathbb{F}_p$. Here it differs from the existing method to towering. Existing method uses $x^2 - \omega$ as the irreducible polynomial in $\mathbb{F}_{p^6}$; that is the root of irreducible binomial of $\mathbb{F}_{p^3}$ is used to construct irreducible binomial in $\mathbb{F}_{p^6}$. In this proposed approach, such binomial can be easily obtained by applying Legendre Symbol $({}^{c_2}/_{p})$ over $\mathbb{F}_p$. Then let its zero be $\tau, \tau \in \mathbb{F}_{(p^3)^2}$, therefore the set $\{1, \tau\}$ forms the polynomial basis in $\mathbb{F}_{(p^3)^2}$. If we choose p such that $p \equiv 3 \pmod 4$, that will accelerate the arithmetic operation significantly; since multiplication by $c_2 = -1$ will be calculated only by substitution. Let us consider $\mathbf{m}, \mathbf{n}$ as two arbitrary elements in $\mathbb{F}_{(p^3)^2}$ as follows:

$$\begin{aligned} \mathbf{m} &= \mathbf{a}_0 + \mathbf{a}_1\tau, \\ \mathbf{n} &= \mathbf{b}_0 + \mathbf{b}_1\tau, \\ & \quad \mathbf{a}_i, \mathbf{b}_j \in \mathbb{F}_{p^3}. \end{aligned}$$

Addition and Subtraction is done coefficient wise similar to those in $\mathbb{F}_{p^3}$. Multiplication of $\mathbf{m}, \mathbf{n}$ is done as follows:

$$\begin{aligned} \mathbf{mn} &= (\mathbf{a}_0 + \mathbf{a}_1\tau)(\mathbf{b}_0 + \mathbf{b}_1\tau) \\ &= \mathbf{a}_0\mathbf{b}_0 + (\mathbf{a}_0\mathbf{b}_1 + \mathbf{a}_1\mathbf{b}_0)\tau + \mathbf{a}_1\mathbf{b}_1\tau^2 \\ &= (\mathbf{a}_0\mathbf{b}_0 + c_2\mathbf{a}_1\mathbf{b}_1) + (\mathbf{a}_0\mathbf{b}_1 + \mathbf{a}_1\mathbf{b}_0)\tau \qquad (7.19) \\ &= (\mathbf{a}_0\mathbf{b}_0 + c_2\mathbf{a}_1\mathbf{b}_1) + (\mathbf{a}_0 + \mathbf{a}_1)(\mathbf{b}_0 + \mathbf{b}_1)\tau \\ & \quad -(\mathbf{a}_0\mathbf{b}_0)\tau - (\mathbf{a}_1\mathbf{b}_1)\tau. \qquad (7.20) \end{aligned}$$

Here Karatsuba method [KO62] is applied. In this calculation, we have substituted $\tau^2 = c_2$, as τ is a zero of the irreducible binomial $g(x) = x^2 - c_2$. Since prime number p is chosen such that $p \equiv 3 \pmod 4$, therefore c_2 is just substituted with -1. That means multiplication with c_2 needs no countable computations in $\mathbb{F}_p$. Moreover multiplication of $\mathbf{a}_1\mathbf{b}_1$ and $\mathbf{a}_0\mathbf{b}_0$ will be reused. Therefore we need 3 multiplications and 5 additions in $\mathbb{F}_{p^3}$ to multiply two vectors over $\mathbb{F}_{(p^3)^2}$, where we consider subtractions as additions.

7.2.3.1 Vector inversion in $\mathbb{F}_{(p^3)^2}$

For calculating the multiplicative inverse vector of a non-zero vector $\mathbf{m} \in \mathbb{F}_{(p^3)^2}$, first we calculate the conjugate of $\mathbf{m}$ that is given by Frobenius mapping $\pi_{p^3}(\mathbf{m}) = \mathbf{m}^{p^3}$. Then the inverse of $\mathbf{m}$, $\mathbf{m}^{-1}$ is calculated as follows:

$$\mathbf{m}^{-1} = n(\mathbf{m})^{-1}(\mathbf{m}^{p^3}), \tag{7.21}$$

where $\mathbf{m}$, $\mathbf{m}^{p^3}$ are the conjugates and $n(\mathbf{m})$ is their product. FM of $\mathbf{m}$, $\pi_{p^3}(\mathbf{m}) = (\mathbf{a}_0 + \mathbf{a}_1\tau)^{p^3}$ can be easily calculated using the defined irreducible binomial $g(x)$ as follows:

$$\begin{aligned}
(\mathbf{a}_0 + \mathbf{a}_1\tau)^{p^3} &= \mathbf{a}_0 + \mathbf{a}_1\tau^{p^3}\\
&= \mathbf{a}_0 + \mathbf{a}_1(\tau^2)^{\frac{p^3-1}{2}}\tau\\
&= \mathbf{a}_0 + \mathbf{a}_1(c_2)^{\frac{p^3-1}{2}}\tau\\
&= \mathbf{a}_0 - \mathbf{a}_1\tau,
\end{aligned} \tag{7.22}$$

where the modular relation $\tau^2 = c_2$ and $c_2 = -1$ is substituted. In other words, the conjugate of $\mathbf{m}$ is given as $\mathbf{a}_0 - \mathbf{a}_1\tau$. No addition and multiplication is required here. Now the calculation procedure for $n(\mathbf{m}) = \mathbf{m}\mathbf{m}^{p^3}$ is as follows:

$$\begin{aligned}
n(\mathbf{m}) &= (\mathbf{a}_0 + \mathbf{a}_1\tau)(\mathbf{a}_0 - \mathbf{a}_1\tau)\\
&= \mathbf{a}_0^2 - \mathbf{a}_1^2\tau^2\\
&= \mathbf{a}_0^2 - c_2\mathbf{a}_1^2\\
&= \mathbf{a}_0^2 + \mathbf{a}_1^2.
\end{aligned} \tag{7.23}$$

Here 2 squaring and 1 addition is required over $\mathbb{F}_{p^3}$. Since $n(\mathbf{m})$ is given without τ, it is found that $n(\mathbf{m}) \in \mathbb{F}_{p^3}$. Therefore, the inversion element $n(\mathbf{m})^{-1}$ is calculated using Eq.(7.12) over $\mathbb{F}_{p^3}$. Finally 2 multiplications, 2 squaring, 1 inversion and 1 addition in $\mathbb{F}_{p^3}$ is required to get an inverse element over $\mathbb{F}_{(p^3)^2}$.

7.2.4 Arithmetic operations over extension field $\mathbb{F}_{((p^3)^2)^3}$

To construct $\mathbb{F}_{((p^3)^2)^3}$ arithmetic operation let us consider irreducible binomial $h(x) = x^3 - \omega$ where $\omega \in \mathbb{F}_{p^3}$ and ω is the root of $f(x)$. Then let θ be a root of $h(x)$, where $\theta \in \mathbb{F}_{((p^3)^2)^3}$, therefore the set $\{1, \theta, \theta^2\}$ forms the polynomial basis in $\mathbb{F}_{((p^3)^2)^3}$. Let us consider $\mathbf{u}, \mathbf{v}$ as two arbitrary elements in $\mathbb{F}_{((p^3)^2)^3}$ as follows:

$$\begin{aligned}
\mathbf{u} &= \mathbf{m}_0 + \mathbf{m}_1\theta + \mathbf{m}_2\theta^2,\\
\mathbf{v} &= \mathbf{n}_0 + \mathbf{n}_1\theta + \mathbf{n}_2\theta^2,\\
&\quad \mathbf{m}_i, \mathbf{n}_j \in \mathbb{F}_{(p^3)^2}.
\end{aligned}$$

In $\mathbb{F}_{((p^3)^2)^3}$, vector addition and subtraction is performed coefficient wise over $\mathbb{F}_{(p^3)^2}$. Multiplication of $\mathbf{u}, \mathbf{v}$ is performed by using $h(x)$ as follows:

$$\mathbf{uv} \;=\; (\mathbf{m}_0 + \mathbf{m}_1\theta + \mathbf{m}_2\theta^2)(\mathbf{n}_0 + \mathbf{n}_1\theta + \mathbf{n}_2\theta^2). \tag{7.24}$$

After applying fast polynomial multiplication according to Eq.(7.7) and Eq.(7.8), here we have 6 multiplications and 15 additions in $\mathbb{F}_{(p^3)^2}$ as follows:

$$\begin{aligned}\mathbf{uv} &= t'_0 + t'_1\theta + t'_2\theta^2 + t'_3\theta^3 + t'_4\theta^4 \\ &= (t_0 + \omega t_3) + (t_1 + \omega t_4)\theta + t'_2\theta^2. \end{aligned} \tag{7.25}$$

Multiplication of basis element with vector will not effect the calculation since it is comparatively small, which will be calculated as bit wise shifting.

7.2.4.1 Vector inversion in $\mathbb{F}_{((p^3)^2)^3}$

Inversion of $\mathbb{F}_{((p^3)^2)^3}$ vector can be easily carried out by applying the similar steps of $\mathbb{F}_{p^3}$ vector inversion. For calculating the multiplicative inverse vector of a non-zero vector $\mathbf{v} \in \mathbb{F}_{((p^3)^2)^3}$, at first we find the conjugates $\mathbf{v}^{p^6}$, $\mathbf{v}^{p^{12}}$ of $\mathbf{v}$ applying FM. Then the inverse element $\mathbf{v}^{-1}$ is calculated as follows:

$$\mathbf{v}^{-1} = n(\mathbf{v})^{-1}(\mathbf{v}^{p^6}\mathbf{v}^{p^{12}}), \tag{7.26}$$

where $\mathbf{v}$, $\mathbf{v}^{p^6}, \mathbf{v}^{p^{12}}$ are the conjugates and $n(\mathbf{v})$ is their product. Here we first calculate $\pi_{p^6}(\mathbf{v}) = (\mathbf{n}_0 + \mathbf{n}_1\theta + \mathbf{n}_2\theta^2)^{p^6}$ using the defined irreducible binomial $h(x)$ as follows:

$$\begin{aligned}(\mathbf{n}_0 + \mathbf{n}_1\theta + \mathbf{n}_2\theta^2)^{p^6} &= (\mathbf{n}_0 + \mathbf{n}_1\theta)^{p^6} + (\mathbf{n}_2\theta^2)^{p^6} \\ &= \mathbf{n}_0 + \mathbf{n}_1(\theta^3)^{\frac{p^6-1}{3}}\theta \\ &\quad +\mathbf{n}_2((\theta^3)^{\frac{p^6-1}{3}})^2\theta^2 \\ &= \mathbf{n}_0 + \mathbf{n}_1(\omega)^{\frac{p^6-1}{3}}\theta \\ &\quad +\mathbf{n}_2((\omega)^{\frac{p^6-1}{3}})^2\theta^2 \\ &= \mathbf{n}_0 + \mathbf{n}_1(\omega^3)^{\frac{p^6-1}{9}}\theta \\ &\quad +\mathbf{n}_2((\omega^3)^{\frac{p^6-1}{9}})^2\theta^2 \\ &= \mathbf{n}_0 + \mathbf{n}_1(c_1)^{\frac{p^6-1}{9}}\theta \\ &\quad +\mathbf{n}_2((c_1)^{\frac{p^6-1}{9}})^2\theta^2 \\ &= \mathbf{n}_0 + \mathbf{n}_1 c'_\omega\theta + \mathbf{n}_2 c''_\omega\theta^2 \\ &= \mathbf{n}_0 + \mathbf{n}'_1\theta + \mathbf{n}'_2\theta^2, \end{aligned} \tag{7.27}$$

where $n'_1, n'_2 \in \mathbb{F}_{(p^3)^2}$ and $c'_\omega = (c_1)^{\frac{p^6-1}{9}}$, $c''_\omega = (c'_\omega)^2$ can be precalculated. Therefore only 6 multiplications in $\mathbb{F}_p$ is required in the above calculation. Now the other conjugate $\mathbf{v}^{p^{12}}$ can be calculated according to the above procedure with the same number of operations as follows:

$$\begin{aligned} \mathbf{v}^{(p^6)^2} &= (\mathbf{v}^{p^{12}}) \\ &= (\mathbf{n}_0 + \mathbf{n}'_1\theta + \mathbf{n}'_2\theta^2)^{p^6} \\ &= \mathbf{n}_0 + \mathbf{n}'_1 c'_\omega \theta + \mathbf{n}'_2 c''_\omega \theta^2 \\ &= \mathbf{n}_0 + \mathbf{n}''_1\theta + \mathbf{n}''_2\theta^2. \end{aligned} \tag{7.28}$$

Now computation of $(\mathbf{v}^{p^6}\mathbf{v}^{p^{12}})$ according to Eq.(7.25) will cost 6 multiplication and 15 additions in $\mathbb{F}_{(p^3)^2}$ as follows:

$$\mathbf{v}^{p^6}\mathbf{v}^{p^{12}} = (\mathbf{n}_0 + \mathbf{n}'_1\theta + \mathbf{n}'_2\theta^2)(\mathbf{n}_0 + \mathbf{n}''_1\theta + \mathbf{n}''_2\theta^2). \tag{7.29}$$

The next calculation procedure is identical of $\mathbb{F}_{p^3}$ vector inversion which also results the same number of operation counts in $\mathbb{F}_{p^6}$. Finally the total cost of 1 vector inversion in $\mathbb{F}_{p^{18}}$ is 9 multiplications, 3 squaring, 5 additions, 3 bit shifting and 1 inversion in $\mathbb{F}_{p^6}$.

7.3 Result evaluation

The main focus of this proposal is to show the construction procedure of $\mathbb{F}_{p^{18}}$ extension field in a new approach of towering that will lead to efficient arithmetic operation. We can also apply subfield isomorphic group arithmetic or Cyclic Vector Multiplication Algorithm (CVMA) to reduce the number of additions and multiplication in each extension field which will make this towering construction more efficient. But that is not focused in this thesis.

Table 7.1 shows the environment, used to experiment and evaluate the proposed method.

TABLE 7.1: Computational Environment

•	PC
CPU *	2.7 GHz Intel Core i5
Memory	16 GB
OS	Mac OS X 10.11.4
Compiler	gcc 4.2.1
Programming Language	C
Library	GNU MP

*Only single core is used from two cores.

In the experiment we have used Kachisa-Schaefer-Scott (KSS) [KSS07] pairing friendly curves with embedding degree $k = 18$ at the 192-bit security level. The prime number p = 511-bit is considered and the curve is defined as $y^2 = x^3 + 11$.

In what follows, let us consider m, s, a and i to denote the times of multiplication, squaring, addition and inversion respectively. The bit wise shifting operation is not taken into account during the final operation count. Table 11.2 shows the calculation cost in the context of operation count and Table 7.3 shows the execution time.

TABLE 7.2: $\mathbb{F}_{((p^3)^2)^3}$ operation count

Operation in	1 inversion in $\mathbb{F}_{p^{18}}$	1 multiplication in $\mathbb{F}_{p^{18}}$
$\mathbb{F}_p$	$199m + 9s + 660a + 1i$	$108m + 402a$

TABLE 7.3: Execution time [ms] for inversion and multiplication in $\mathbb{F}_{((p^3)^2)^3}$

Operation	Execution time[ms]
Inversion	5.4×10^{-1}
Multiplication	3.3×10^{-1}

From Table 11.2 we find that only 199 multiplication, 9 squaring, 660 additions and 1 inversion is required in $\mathbb{F}_p$ to perform 1 inversion in $\mathbb{F}_{p^{18}}$. There exist a competitive toweting scheme prsented by Aranha et al. [Ara+13] that uses subfield isomorphic group to reduce number of arithmetic operation. Such isomorphic subfield isomorphic rational point group technique can also be applied in the proposed towering approach which will be presented as our future work.

7.4 Conclusion and future work

In this thesis we have presented a new towering scheme to construct $\mathbb{F}_{p^{18}}$ extension field arithmetic. This towering approach is one of the most important step for constructing the basis of pairing based cryptography defined over extension field of degree 18. This thesis also presented the mathematical derivation for efficiently constructing the $\mathbb{F}_{((p^3)^2)^3}$ extension field to accelerate arithmetic operation in $\mathbb{F}_{p^{18}}$. The main focus of this thesis was to present the new towering technique along with its implementation procedure that can be used for performing operation efficiently in the context of pairing based cryptography. As our future work, we would like to reduce the number of arithmetic operation by applying subfield isomorphic rational point group technique in the proposed towering approach along with some pairing algorithms implementation in practical case.

Chapter 8

Efficient Optimal-Ate Pairing at 128-bit Security

8.1 Introduction

This chapter tries to efficiently carry out the basic operation of a specific type of pairing calculation over KSS-16 pairing-friendly curves.

8.1.1 Notation Overview

In this section we recall the notations for reference. Generally, a pairing is a bilinear map e typically defined as $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive cyclic sub-groups of order r on a certain elliptic curve E over a finite extension field $\mathbb{F}_{p^k}$ and $\mathbb{G}_3$ is a multiplicative cyclic group of order r in $\mathbb{F}^*_{p^k}$. Let $E(\mathbb{F}_p)$ be the set of rational points over the prime field $\mathbb{F}_p$ which forms an additive Abelian group together with the point at infinity $\mathcal{O}$. The total number of rational points is denoted as $\#E(\mathbb{F}_p)$. Here, the order r is a large prime number such that $r|\#E(\mathbb{F}_p)$ and $\gcd(r,p) = 1$. The embedding degree k is the smallest positive integer such that $r|(p^k - 1)$. Two basic properties of pairing are bilinearity and non-degeneration.

As aforementioned in **??** Galbraith et al. [GPS08] have classified pairings as three major categories based on the underlying group's structure as This chapter chooses one of the Type 3 variants of pairing named as Optimal-Ate [Ver10] with Kachisa-Schaefer-Scott (KSS) [KSS07] pairing-friendly curve of embedding degree $k = 16$. Few previous works have been done on this curve.

8.1.2 Related Works

Zhang et al. [ZL12] have shown the computational estimation of the Miller's loop and proposed efficient final exponentiation for 192-bit security level in the context of Optimal-Ate pairing over KSS-16 curve. A few years later Ghammam et al. [GF16a] have shown that KSS-16 is the best suited for multi-pairing (i.e., the product and/or the quotient) when the number of pairing is more than two. Ghammam et al. [GF16a] also corrected the flaws of proposed final exponentiation algorithm by Zhang et al. [ZL12] and proposed a new one and showed the vulnerability of Zhang's parameter settings against small subgroup attack.

8.1.3 Motivation

The recent development of NFS by Kim and Barbulescu [KB16] requires updating the parameter selection for all the existing pairings over the well known pairing-friendly curve families such as BN [BN06], BLS [FST06] and KSS [KSS07]. The most recent study by Barbulescu et al. [BD18] have shown the security estimation of the current parameter settings used in well-studied curves and proposed new parameters, resistant to small subgroup attack.

Barbulescu and Duquesne's study finds that the current parameter settings for 128-bit security level on BN-curve studied in literature can withstand for 100-bit security. Moreover, they proposed that BLS-12 and surprisingly KSS-16 are the most efficient choice for Optimal-Ate pairing at the 128-bit security level. Therefore, the authors focus on the efficient implementation of the less studied KSS-16 curve for Optimal-Ate pairing by applying the most recent parameters. Mori et al. [Mor+14] and Khandaker et al. [Kha+17a] have shown a specific type of sparse multiplication for BN and KSS-18 curve respectively where both of the curves supports sextic twist. The authors have extended the previous works for quartic twisted KSS-16 curve and derived pseudo-8 sparse multiplication for line evaluation step in the Miller's algorithm. As a consequence, the authors made the choice to concentrate on Miller's algorithm's execution time and computational complexity to verify the claim of [BD18]. The implementation shows that Miller's algorithm time has a tiny difference between KSS-16 and BLS-12 curves. However, they both are more efficient and faster than BN curve.

8.1.4 Contribution Summary

Following the emergence of Kim and Barbulescu's new number field sieve (exTNFS) algorithm at CRYPTO'16 [KB16] for solving discrete logarithm problem (DLP) over the finite field; pairing-based cryptography researchers are intrigued to find new parameters that confirm standard security levels against exTNFS. Recently, Barbulescu and Duquesne have suggested new parameters [BD18] for well-studied pairing-friendly curves i.e., Barreto-Naehrig (BN) [BN06], Barreto-Lynn-Scott (BLS-12) [BLS03] and Kachisa-Schaefer-Scott (KSS-16) [KSS07] curves at 128-bit security level (twist and sub-group attack secure). They have also concluded that in the context of Optimal-Ate pairing with their suggested parameters, BLS-12 and KSS-16 curves are more efficient choices than BN curves. Therefore, this chapter selects the atypical and less studied pairing-friendly curve in literature, i.e., KSS-16 which offers quartic twist, while BN and BLS-12 curves have sextic twist. In this chapter, the authors optimize Miller's algorithm of Optimal-Ate pairing for the KSS-16 curve by deriving efficient sparse multiplication and implement them. Furthermore, this chapter concentrates on the Miller's algorithm to experimentally verify Barbulescu et al.'s estimation. The result shows that

Miller's algorithm time with the derived pseudo 8-sparse multiplication is most efficient for KSS-16 than other two curves. Therefore, this chapter defends Barbulescu and Duquesne's conclusion for 128-bit security.

8.2 Fundamentals of Elliptic Curve and Pairing

8.2.1 Kachisa-Schaefer-Scott (KSS) Curve of Embedding Degree $k = 16$

In [KSS07], Kachisa, Schaefer, and Scott proposed a family of non supersingular pairing-friendly elliptic curves of embedding degree $k = \{16, 18, 32, 36, 40\}$, using elements in the cyclotomic field. In what follows, this chapter considers the curve of embedding degree $k = 16$, named as *KSS-16*, defined over extension field $\mathbb{F}_{p^{16}}$ as follows:

$$E/\mathbb{F}_{p^{16}} : Y^2 = X^3 + aX, \quad (a \in \mathbb{F}_p) \text{ and } a \neq 0, \tag{8.1}$$

where $X, Y \in \mathbb{F}_{p^{16}}$. Similar to other pairing-friendly curves, *characteristic p*, *Frobenius trace t* and *order r* of this curve are given by the following polynomials of integer variable u.

$$\begin{aligned} p(u) &= (u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 + 625u^2 \\ &\quad +2398u + 3125)/980, && (8.2a) \\ r(u) &= (u^8 + 48u^4 + 625)/61255, && (8.2b) \\ t(u) &= (2u^5 + 41u + 35)/35, && (8.2c) \end{aligned}$$

where u is such that $u \equiv 25$ or $45 \pmod{70}$ and the ρ value is $\rho = (\log_2 p / \log_2 r) \approx 1.25$. The total number of rational points $\#E(\mathbb{F}_p)$ is given by Hasse's theorem as, $\#E(\mathbb{F}_p) = p + 1 - t$. When the definition field is the k-th degree extension field $\mathbb{F}_{p^k}$, rational points on the curve E also form an additive Abelian group denoted as $E(\mathbb{F}_{p^k})$. Total number of rational points $\#E(\mathbb{F}_{p^k})$ is given by Weil's theorem [Wei+49] as $\#E(\mathbb{F}_{p^k}) = p^k + 1 - t_k$, where $t_k = \alpha^k + \beta^k$. α and β are complex conjugate numbers.

8.2.2 Extension Field Arithmetic and Towering

Pairing-based cryptography requires performing the arithmetic operation in extension fields of degree $k \geq 6$ [SCA86]. Consequently, such higher degree extension field needs to be constructed as a tower of sub-fields [BS09] to perform arithmetic operation cost efficiently. Bailey et al. [BP01] have explained optimal extension field by towering by using irreducible binomials.

TABLE 8.1: Number of arithmetic operations in $\mathbb{F}_{p^{16}}$ based on Eq.(8.3)

$M_{p^2} = 3M_p + 5A_p + 1m_\alpha \longrightarrow 3M_p$	$S_{p^2} = 3S_p + 4A_p + 1m_\alpha \longrightarrow 3S_p$
$M_{p^4} = 3M_{p^2} + 5A_{p^2} + 1m_\beta \longrightarrow 9M_p$	$S_{p^4} = 3S_{p^2} + 4A_p p^2 + 1m_\beta \longrightarrow 9S_p$
$M_{p^8} = 3M_{p^4} + 5A_{p^4} + 1m_\gamma \longrightarrow 27M_p$	$S_{p^8} = 3S_{p^4} + 4A_{p^4} + 1m_\gamma \longrightarrow 27S_p$
$M_{p^{16}} = 3M_{p^8} + 5A_{p^8} + 1m_\omega \longrightarrow 81M_p$	$S_{p^{16}} = 3M_{p^8} + 4A_{p^8} + 1m_\omega \longrightarrow 81S_p$

8.2.2.1 Towering of $\mathbb{F}_{p^{16}}$ Extension Field

For KSS-16 curve, $\mathbb{F}_{p^{16}}$ construction process given as follows using tower of sub-fields.

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 - c), \\ \mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/(\beta^2 - \alpha), \\ \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\gamma]/(\gamma^2 - \beta), \\ \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\omega]/(\omega^2 - \gamma), \end{cases} \tag{8.3}$$

where $p \equiv 5 \bmod 8$ and c is a quadratic non residue in $\mathbb{F}_p$. This chapter considers $c = 2$ along with the value of the parameter u as given in [BD18].

8.2.2.2 Towering of $\mathbb{F}_{p^{12}}$ Extension Field

Let $6|(p-1)$, where p is the characteristics of BN or BLS-12 curve and -1 is a quadratic and cubic non-residue in $\mathbb{F}_p$ since $p \equiv 3 \bmod 4$. In the context of BN or BLS-12, where $k = 12$, $\mathbb{F}_{p^{12}}$ is constructed as a tower of sub-fields with irreducible binomials as follows:

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 + 1), \\ \mathbb{F}_{p^6} = \mathbb{F}_{p^2}[\beta]/(\beta^3 - (\alpha + 1)), \\ \mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}[\gamma]/(\gamma^2 - \beta). \end{cases} \tag{8.4}$$

8.2.2.3 Extension Field Arithmetic of $\mathbb{F}_{p^{16}}$ and $\mathbb{F}_{p^{12}}$

Among the arithmetic operations multiplication, squaring and inversion are regarded as expensive operation than addition/subtraction. The calculation cost, based on number of prime field multiplication M_p and squaring S_p is given in Table 8.1. The arithmetic operations in $\mathbb{F}_p$ are denoted as M_p for a multiplication, S_p for a squaring, I_p for an inversion and m with suffix denotes multiplication with basis element. However, squaring is more optimized by using Devegili et al.'s [Dev+06] complex squaring technique which cost $2M_p + 4A_p + 2m_\alpha$ for one squaring operation in $\mathbb{F}_{p^2}$. In total it costs $54M_p$ for one squaring in $\mathbb{F}_{p^{16}}$. Table 8.1 shows the operation estimation for $\mathbb{F}_{p^{16}}$.

Table 10.1 shows the operation estimation for $\mathbb{F}_{p^{12}}$ according to the towering shown in Eq.(10.3). The algorithms for $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^3}$ multiplication and

TABLE 8.2: Number of arithmetic operations in $\mathbb{F}_{p^{12}}$ based on Eq.(10.3)

$M_{p^2} = 3M_p + 5A_p + 1m_\alpha \rightarrow 3M_p$	$S_{p^2} = 2S_p + 3A_p \rightarrow 2S_p$
$M_{p^6} = 6M_{p^2} + 15A_{p^2} + 2m_\beta \rightarrow 18M_p$	$S_{p^6} = 2M_{p^2} + 3S_{p^2} + 9A_{p^2} + 2m_\beta \rightarrow 12S_p$
$M_{p^{12}} = 3M_{p^6} + 5A_{p^6} + 1m_\gamma \rightarrow 54M_p$	$S_{p^{12}} = 2M_{p^6} + 5A_{p^6} + 2m_\gamma \rightarrow 36S_p$

squaring given in [Duq+15] have be used in this chapter to construct the $\mathbb{F}_{p^{12}}$ extension field arithmetic.

8.2.3 Ate and Optimal-Ate On KSS-16, BN, BLS-12 Curve

A brief of pairing and it's properties are described in Sect.1. In the context of pairing on the targeted pairing-friendly curves, two additive rational point groups $\mathbb{G}_1, \mathbb{G}_2$ and a multiplicative group $\mathbb{G}_3$ of order r are considered. $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\begin{aligned} \mathbb{G}_1 &= E(\mathbb{F}_p)[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ \mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r, \\ e &: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3, \end{aligned} \tag{8.5}$$

where e denotes Ate pairing [Coh+05]. $E(\mathbb{F}_{p^k})[r]$ denotes rational points of order r and $[n]$ denotes n times scalar multiplication for a rational point. π_p denotes the Frobenius endomorphism given as $\pi_p : (x, y) \mapsto (x^p, y^p)$.

KSS-16 Curve:

In what follows, we consider $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ for KSS-16 curves. Ate pairing $e(Q, P)$ is given as follows:

$$e(Q, P) = f_{t-1,Q}(P)^{\frac{p^{16}-1}{r}}, \tag{8.6}$$

where $f_{t-1,Q}(P)$ symbolizes the output of Miller's algorithm and $\lfloor \log_2(t-1) \rfloor$ is the loop length. The bilinearity of Ate pairing is satisfied after calculating the final exponentiation $(p^k - 1)/r$.

Vercauteren proposed more efficient variant of Ate pairing named as Optimal-Ate pairing [Ver10] where the Miller's loop length reduced to $\lfloor \log_2 u \rfloor$. The previous work of Zhang et al. [ZL12] has derived the optimal Ate pairing on the KSS-16 curve which is defined as follows with $f_{u,Q}(P)$ is the Miller function evaluated on P:

$$e_{opt}(Q, P) = ((f_{u,Q}(P) \cdot l_{[u]Q,[p]Q}(P))^{p^3} \cdot l_{Q,Q}(P))^{\frac{p^{16}-1}{r}}. \tag{8.7}$$

TABLE 8.3: Optimal-Ate pairing formulas for target curves

Curve	Miller's Algo.	Final Exp.
KSS-16	$(f_{u,Q}(P) \cdot l_{[u]Q,[p]Q}(P))^{p^3} \cdot l_{Q,Q}(P)$	$(p^{16}-1)/r$
BN	$f_{6u+2,Q}(P) \cdot l_{[6u+2]Q,[p]Q}(P) \cdot l_{[6u+2+p]Q,[-p^2]Q}(P)$	$(p^{12}-1)/r$
BLS-12	$f_{u,Q}(P)$	$(p^{12}-1)/r$

The formulas for Optimal-Ate pairing for the target curves are given in Table 8.3.

The naive calculation procedure of Optimal-Ate pairing is shown in Alg. 16. In what follows, the calculation steps from 1 to 11, shown in Alg.16, is identified as Miller's Algorithm (MA) and step 12 is the final exponentiation (FE). Steps 2-7 are specially named as Miller's loop. Steps 3, 5, 7 are the line evaluation together with elliptic curve doubling (ECD) and addition (ECA) inside the Miller's loop and steps 9, 11 are the line evaluation outside the loop. These line evaluation steps are the key steps to accelerate the loop calculation. The authors extended the work of [Mor+14],[Kha+17a] for KSS-16 curve to calculate *pseudo 8-sparse multiplication* described in Sect. 3. The ECA and ECD are also calculated efficiently in the twisted curve. The $Q_2 \leftarrow [p]Q$ term of step 8 is calculated by applying one skew Frobenius map over $\mathbb{F}_{p^4}$ and $f_1 \leftarrow f^{p^3}$ of step 10 is calculated by applying one Frobenius map in $\mathbb{F}_{p^{16}}$. Step 12, FE is calculated by applying Ghammam et al.'s work for KSS-16 curve [GF16a].

Algorithm 8: Optimal-Ate pairing on KSS-16 curve

```
Input: $u, P \in \mathbb{G}_1, Q \in \mathbb{G}'_2$
Output: $(Q, P)$
1  $f \leftarrow 1, T \leftarrow Q$
2  for $i = \lfloor \log_2(u) \rfloor$ downto 1 do
3      $f \leftarrow f^2 \cdot l_{T,T}(P), T \leftarrow [2]T$
4      if $u[i] = 1$ then
5          $f \leftarrow f \cdot l_{T,Q}(P), T \leftarrow T + Q$
6      if $u[i] = -1$ then
7          $f \leftarrow f \cdot l_{T,-Q}(P), T \leftarrow T - Q$
8  $Q_1 \leftarrow [u]Q, Q_2 \leftarrow [p]Q$
9  $f \leftarrow f \cdot l_{Q_1,Q_2}(P)$
10 $f_1 \leftarrow f^{p^3}, f \leftarrow f \cdot f_1$
11 $f \leftarrow f \cdot l_{Q,Q}(P)$
12 $f \leftarrow f^{\frac{p^{16}-1}{r}}$
13 return $f$
```

TABLE 8.4: Vector representation of $Q = (x_Q, y_Q) \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$

	1	α	β	$\alpha\beta$	γ	$\alpha\gamma$	$\beta\gamma$	$\alpha\beta\gamma$	ω	$\alpha\omega$	$\beta\omega$	$\alpha\beta\omega$	$\gamma\omega$	$\alpha\gamma\omega$	$\beta\gamma\omega$	$\alpha\beta\gamma\omega$
x_Q	0	0	0	0	b_4	b_5	b_6	b_7	0	0	0	0	0	0	0	0
y_Q	0	0	0	0	0	0	0	0	0	0	0	0	b_{12}	b_{13}	b_{14}	b_{15}

8.2.4 Twist of KSS-16 Curves

In the context of Type 3 pairing, there exists a *twisted curve* with a group of rational points of order r, isomorphic to the group where rational point $Q \in E(\mathbb{F}_{p^k})[r] \cap \text{Ker}(\pi_p - [p])$ belongs to. This subfield isomorphic rational point group includes a twisted isomorphic point of Q, typically denoted as $Q' \in E'(\mathbb{F}_{p^{k/d}})$, where k is the embedding degree and d is the twist degree.

Since points on the twisted curve are defined over a smaller field than $\mathbb{F}_{p^k}$, therefore ECA and ECD become faster. However, when required in the Miller's algorithm's line evaluation, the points can be quickly mapped to points on $E(\mathbb{F}_{p^k})$. Since the pairing-friendly KSS-16 [KSS07] curve has CM discriminant of $D = 1$ and $4|k$; therefore, quartic twist is available.

8.2.4.1 Quartic Twist

Let β be a certain quadratic non-residue in $\mathbb{F}_{p^4}$. The quartic twisted curve E' of KSS-16 curve E defined in Eq.(12.1) and their isomorphic mapping ψ_4 are given as follows:

$$\begin{aligned} E' \ &: \ y^2 = x^3 + ax\beta^{-1}, \quad a \in \mathbb{F}_p, \\ \psi_4 \ &: \ E'(\mathbb{F}_{p^4})[r] \longmapsto E(\mathbb{F}_{p^{16}})[r] \cap \text{Ker}(\pi_p - [p]), \\ & \quad (x, y) \longmapsto (\beta^{1/2}x, \beta^{3/4}y), \end{aligned} \tag{8.8}$$

where Ker(·) denotes the kernel of the mapping and π_p denotes Frobenius mapping for rational point.

Table 11.1 shows the vector representation of $Q = (x_Q, y_Q) = (\beta^{1/2}x_{Q'}, \beta^{3/4}y_{Q'}) \in \mathbb{F}_{p^{16}}$ according to the given towering in Eq.(8.3). Here, $x_{Q'}$ and $y_{Q'}$ are the coordinates of rational point Q' on quartic twisted curve E'.

8.3 Proposal

8.3.1 Overview: Sparse and Pseudo-Sparse Multiplication

Aranha et al. [Ara+11, Section 4] and Costello et al. [CLN10] have well optimized the Miller's algorithm in Jacobian coordinates by 6-sparse multiplication [1] for BN curve. Mori et al. [Mor+14] have shown the pseudo 8-sparse

[1] 6-Sparse refers the state when in a vector (multiplier/multiplicand), among the 12 coefficients 6 of them are zero.

multiplication [2] for BN curve by adapting affine coordinates where the sextic twist is available. It is found that pseudo 8-sparse was efficient than 7-sparse and 6-sparse in Jacobian coordinates.

Let us consider $T = (\gamma x_{T'}, \gamma\omega y_{T'})$, $Q = (\gamma x_{Q'}, \gamma\omega y_{Q'})$ and $P = (x_P, y_P)$, where $x_p, y_p \in \mathbb{F}_p$ given in affine coordinates on the curve $E(\mathbb{F}_{p^{16}})$ such that $T' = (x_{T'}, y_{T'})$, $Q' = (x_{Q'}, y_{Q'})$ are in the twisted curve E' defined over $\mathbb{F}_{p^4}$. Let the elliptic curve doubling of $T + T = R(x_R, y_R)$. The 7-sparse multiplication for KSS-16 can be derived as follows.

$$\begin{gathered}
l_{T,T}(P) = (y_p - y_{T'}\gamma\omega) - \lambda_{T,T}(x_P - x_{T'}\gamma), \quad \text{when } T = Q, \\
\lambda_{T,T} = \tfrac{3x_{T'}^2\gamma^2 + a}{2y_{T'}\gamma\omega} = \tfrac{3x_{T'}^2\gamma\omega^{-1} + a(\gamma\omega)^{-1}}{2y_{T'}} = \tfrac{(3x_{T'}^2 + ac^{-1}\alpha\beta)\omega}{2y_{T'}} = \lambda'_{T,T}\omega, \\
\text{since } \gamma\omega^{-1} = \omega, (\gamma\omega)^{-1} = \omega\beta^{-1}, \quad \text{and} \\
a\beta^{-1} = (a + 0\alpha + 0\beta + 0\alpha\beta)\beta^{-1} = a\beta^{-1} = ac^{-1}\alpha\beta, \quad \text{where } \alpha^2 = c.
\end{gathered}$$

Now the line evaluation and ECD are obtained as follows:

$$\begin{gathered}
l_{T,T}(P) = y_p - x_p\lambda'_{T,T}\omega + (x_{T'}\lambda'_{T,T} - y_{T'})\gamma\omega, \\
x_{2T'} = (\lambda'_{T,T})^2\omega^2 \ - 2x_{T'}\gamma = ((\lambda'_{T,T})^2 \ - 2x_{T'})\gamma \\
y_{2T'} = (x_{T'}\gamma - x_{2T'}\gamma)\lambda'_{T,T}\omega - y_{T'}\gamma\omega = (x_{T'}\lambda'_{T,T} - x_{2T'}\lambda'_{T,T} - y_{T'})\gamma\omega.
\end{gathered}$$

The above calculations can be optimized as follows:

$$\begin{gathered}
A = \tfrac{1}{2y_{T'}}, B = 3x_{T'}^2 + ac^{-1}, C = AB, D = 2x_{T'}, x_{2T'} = C^2 - D, \\
E = Cx_{T'} - y_{T'}, y_{2T'} = E - Cx_{2T'}, \\
l_{T,T}(P) = y_P + E\gamma\omega - Cx_P\omega = y_P + F\omega + E\gamma\omega, \qquad (8.9)
\end{gathered}$$

where $F = -Cx_P$.

The elliptic curve addition phase ($T \neq Q$) and line evaluation of $l_{T,Q}(P)$ can also be optimized similar to the above procedure. Let the elliptic curve addition of $T + Q = R(x_R, y_R)$.

$$\begin{gathered}
l_{T,Q}(P) = (y_p - y_{T'}\gamma\omega) - \lambda_{T,Q}(x_P - x_{T'}\gamma), \quad T \neq Q, \\
\lambda_{T,Q} = \tfrac{(y_{Q'} - y_{T'})\gamma\omega}{(x_{Q'} - x_{T'})\gamma} = \tfrac{(y_{Q'} - y_{T'})\omega}{x_{Q'} - x_{T'}} = \lambda'_{T,Q}\omega, \\
x_R = (\lambda'_{T,Q})^2\omega^2 \ - x_{T'}\gamma - x_{Q'}\gamma = ((\lambda'_{T,Q})^2 \ - x_{T'} - x_{Q'})\gamma \\
y_R = (x_{T'}\gamma - x_R\gamma)\lambda'_{T,Q}\omega - y_{T'}\gamma\omega = (x_{T'}\lambda'_{T,Q} - x_{R'}\lambda'_{T,Q} - y_{T'})\gamma\omega.
\end{gathered}$$

[2] Pseudo 8-sparse refers to a certain length of vector's coefficients where instead of 8 zero coefficients, there are seven 0's and one 1 as coefficients.

TABLE 8.5: Vector representation of $Q = (x_Q, y_Q) \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{12}})$

	1	α	β	$\alpha\beta$	β^2	$\alpha\beta^2$	γ	$\alpha\gamma$	$\beta\gamma$	$\alpha\beta\gamma$	$\beta^2\gamma$	$\alpha\beta^2\gamma$
x_Q	0	0	0	0	b_4	b_5	0	0	0	0	0	0
y_Q	0	0	0	0	0	0	0	0	b_8	b_9	0	0

Representing the above line equations using variables as following :

$$\begin{aligned}
A &= \tfrac{1}{x_{Q'}-x_{T'}}, B = y_{Q'} - y_{T'}, C = AB, D = x_{T'} + x_{Q'},\\
x_{R'} &= C^2 - D, E = Cx_{T'} - y_{T'}, y_{R'} = E - Cx_{R'},\\
l_{T,Q}(P) &= y_P + E\gamma\omega - Cx_P\omega = y_P + F\omega + E\gamma\omega, \qquad (8.10)\\
F &= -Cx_P,
\end{aligned}$$

Here all the variables (A, B, C, D, E, F) are calculated as $\mathbb{F}_{p^4}$ elements. The position of the y_P, E and F in $\mathbb{F}_{p^{16}}$ vector representation is defined by the basis element 1, $\gamma\omega$ and ω as shown in Table 11.1. Therefore, among the 16 coefficients of $l_{T,T}(P)$ and $l_{T,Q}(P) \in \mathbb{F}_{p^{16}}$, only 9 coefficients $y_P \in \mathbb{F}_p$, $Cx_P \in \mathbb{F}_{p^4}$ and $E \in \mathbb{F}_{p^4}$ are non-zero. The remaining 7 zero coefficients leads to an efficient multiplication, usually called sparse multiplication. This particular instance in KSS-16 curve is named as 7-sparse multiplication.

8.3.2 Pseudo 8-Sparse Multiplication for BN and BLS-12 Curve

Here we have followed Mori et al.'s [Mor+14] procedure to derive pseudo 8-sparse multiplication for the parameter settings of [BD18] for BN and BLS-12 curves. For the new parameter settings, the towering is given as Eq.(10.3) for both BN and BLS-12 curve. However, the curve form $E : y^2 = x^3 + b,\ b \in \mathbb{F}_p$ is identical for both BN and BLS-12 curve. The sextic twist obtained for these curves are also identical. Therefore, in what follows this chapter will denote both of them as E_b defined over $\mathbb{F}_{p^{12}}$.

8.3.2.1 Sextic twist of BN and BLS-12 curve:

Let $(\alpha + 1)$ be a certain quadratic and cubic non-residue in $\mathbb{F}_{p^2}$. The sextic twisted curve E'_b of curve E_b and their isomorphic mapping ψ_6 are given as follows:

$$\begin{aligned}
E'_b &: y^2 = x^3 + b(\alpha + 1), \quad b \in \mathbb{F}_p,\\
\psi_6 &: E'_b(\mathbb{F}_{p^2})[r] \longmapsto E_b(\mathbb{F}_{p^{12}})[r] \cap \mathrm{Ker}(\pi_p - [p]),\\
&\quad (x, y) \longmapsto ((\alpha + 1)^{-1}x\beta^2, (\alpha + 1)^{-1}y\beta\gamma). \qquad (8.11)
\end{aligned}$$

The line evaluation and ECD/ECA can be obtained in affine coordinate for the rational point P and $Q', T' \in E'_b(\mathbb{F}_{p^2})$ as follows:

Elliptic curve addition when $T' \neq Q'$ **and** $T' + Q' = R'(x_{R'}, y_{R'})$

$$\begin{aligned} A = \tfrac{1}{x_{Q'}-x_{T'}}, B = y_{Q'} - y_{T'}, C = AB, D = x_{T'} + x_{Q'}, \\ x_{R'} = C^2 - D, E = Cx_{T'} - y_{T'}, y_{R'} = E - Cx_{R'}, \\ l_{T',Q'}(P) = y_P + (\alpha+1)^{-1}E\beta\gamma - (\alpha+1)^{-1}Cx_P\beta^2\gamma, \quad (8.12a) \\ y_P^{-1}l_{T',Q'}(P) = 1 + (\alpha+1)^{-1}Ey_P^{-1}\beta\gamma - (\alpha+1)^{-1}Cx_Py_P^{-1}\beta^2\gamma, \quad (8.12b) \end{aligned}$$

Elliptic curve doubling when $T' = Q'$

$$\begin{aligned} A = \tfrac{1}{2y_{T'}}, B = 3x_{T'}^2,, C = AB, D = 2x_{T'}, x_{2T'} = C^2 - D, \\ E = Cx_{T'} - y_{T'}, y_{2T'} = E - Cx_{2T'}, \\ l_{T',T'}(P) = y_P + (\alpha+1)^{-1}E\beta\gamma - (\alpha+1)^{-1}Cx_P\beta^2\gamma, \quad (8.13a) \\ y_P^{-1}l_{T',T'}(P) = 1 + (\alpha+1)^{-1}Ey_P^{-1}\beta\gamma - (\alpha+1)^{-1}Cx_Py_P^{-1}\beta^2\gamma, \quad (8.13b) \end{aligned}$$

The line evaluations of Eq.(10.9b) and Eq.(10.8b) are identical and more sparse than Eq.(10.9a) and Eq.(10.8a). Such sparse form comes with a cost of computation overhead. But such overhead can be minimized by the following isomorphic mapping, which also accelerates the Miller's loop iteration.

Isomorphic mapping of $P \in \mathbb{G}_1 \mapsto \hat{P} \in \mathbb{G}_1'$:

$$\begin{aligned} \hat{E} \;:\; & y^2 = x^3 + b\hat{z}, \\ & \hat{E}(\mathbb{F}_p)[r] \longmapsto E(\mathbb{F}_p)[r], \\ & (x, y) \longmapsto (\hat{z}^{-1}x, \hat{z}^{-3/2}y), \quad (8.14) \end{aligned}$$

where $\hat{z} \in \mathbb{F}_p$ is a quadratic and cubic residue in $\mathbb{F}_p$. Eq.(10.10) maps rational point P to $\hat{P}(x_{\hat{P}}, y_{\hat{P}})$ such that $(x_{\hat{P}}, y_{\hat{P}}^{-1}) = 1$. The twist parameter $\hat{z}$ is obtained as:

$$\hat{z} = (x_P y_P^{-1})^6. \quad (8.15)$$

From the Eq.(10.11) $\hat{P}$ and $\hat{Q}'$ is given as

$$\begin{aligned} \hat{P}(x_{\hat{P}}, y_{\hat{P}}) &= (x_P z^{-1}, y_P z^{-3/2}) = (x_P^3 y_P^{-2}, x_P^3 y_P^{-2}), \quad (8.16a) \\ \hat{Q}'(x_{\hat{Q}'}, y_{\hat{Q}'}) &= (x_P^2 y_P^{-2} x_{Q'}, x_P^3 y_P^{-3} y_{Q'}). \quad (8.16b) \end{aligned}$$

Using Eq.(10.12a) and Eq.(10.12b) the line evaluation of Eq.(10.8b) becomes

$$\begin{aligned} y_{\hat{P}}^{-1} l_{\hat{T}',\hat{T}'}(\hat{P}) &= 1 + (\alpha+1)^{-1}Ey_{\hat{P}}^{-1}\beta\gamma - (\alpha+1)^{-1}Cx_{\hat{P}}y_{\hat{P}}^{-1}\beta^2\gamma, \\ \hat{l}_{\hat{T}',\hat{T}'}(\hat{P}) &= 1 + (\alpha+1)^{-1}Ey_{\hat{P}}^{-1}\beta\gamma - (\alpha+1)^{-1}C\beta^2\gamma. \quad (8.17a) \end{aligned}$$

The Eq.(10.9b) becomes similar to Eq.(10.13a). The calculation overhead can be reduced by pre-computation of $(\alpha+1)^{-1}$, $y_{\hat{P}}^{-1}$ and $\hat{P}$, $\hat{Q}'$ mapping using x_P^{-1} and y_P^{-1} as shown by Mori et al. [Mor+14].

Finally, pseudo 8-sparse multiplication for BN and BLS-12 is given in

```
Algorithm 9: Pseudo 8-sparse multiplication for BN and BLS-12 curves
Input: $a, b \in \mathbb{F}_{p^{12}}$
$a = (a_0 + a_1\beta + a_2\beta^2) + (a_3 + a_4\beta + a_5\beta^2)\gamma$, $b = 1 + b_4\beta\gamma + b_5\beta^2\gamma$
where $a_i, b_j, c_i \in \mathbb{F}_{p^2}(i = 0, \cdots, 5, j = 4, 5)$
Output: $c = ab = (c_0 + c_1\beta + c_2\beta^2) + (c_3 + c_4\beta + c_5\beta^2)\gamma \in \mathbb{F}_{p^{12}}$
$c_4 \longleftarrow a_0 \times b_4$, $t_1 \longleftarrow a_1 \times b_5$, $t_2 \longleftarrow a_0 + a_1$, $S_0 \longleftarrow b_4 + b_5$
$c_5 \longleftarrow t_2 \times S_0 - (c_4 + t_1)$, $t_2 \longleftarrow a_2 \times b_5$, $t_2 \longleftarrow t_2 \times (\alpha + 1)$
$c_4 \longleftarrow c_4 + t_2$, $t_0 \longleftarrow a_2 \times b_4$, $t_0 \longleftarrow t_0 + t_1$
$c_3 \longleftarrow t_0 \times (\alpha + 1)$, $t_0 \longleftarrow a_3 \times b_4$, $t_1 \longleftarrow a_4 \times b_5$, $t_2 \longleftarrow a_3 + a_4$
$t_2 \longleftarrow t_2 \times S_0 - (t_0 + t_1)$
$c_0 \longleftarrow t_2 \times (\alpha + 1)$, $t_2 \longleftarrow a_5 \times b_4$, $t_2 \longleftarrow t_1 + t_2$
$c_1 \longleftarrow t_2 \times (\alpha + 1)$, $t_1 \longleftarrow a_5 \times b_5$, $t_1 \longleftarrow t_1 \times (\alpha + 1)$
$c_2 \longleftarrow t_0 + t_1$
$c \longleftarrow c + a$
return $c = (c_0 + c_1\beta + c_2\beta^2) + (c_3 + c_4\beta + c_5\beta^2)\gamma$
```

8.3.3 Pseudo 8-sparse Multiplication for KSS-16 Curve

The main idea of *pseudo 8-sparse multiplication* is finding more sparse form of Eq.(9.14) and Eq.(9.16), which allows to reduce the number of multiplication of $\mathbb{F}_{p^{16}}$ vector during Miller's algorithm evaluation. To obtains the same, y_P^{-1} is multiplied to both side of Eq.(9.14) and Eq.(9.16), since y_P remains the same through the Miller's algorithms loop calculation.

$$y_P^{-1} l_{T,T}(P) = 1 - Cx_P y_P^{-1}\omega + Ey_P^{-1}\gamma\omega, \tag{8.18a}$$

$$y_P^{-1} l_{T,Q}(P) = 1 - Cx_P y_P^{-1}\omega + Ey_P^{-1}\gamma\omega, \tag{8.18b}$$

Although the Eq.(9.17a) and Eq.(9.17b) do not get more sparse, but 1st coefficient becomes 1. Such vector is titled as *pseudo sparse form* in this chapter. This form realizes more efficient $\mathbb{F}_{p^{16}}$ vectors multiplication in Miller's loop. However, the Eq.(9.17b) creates more computation overhead than Eq.(9.16), i.e., computing $y_P^{-1} l_{T,Q}(P)$ in the left side and $x_P y_P^{-1}$, Ey_P^{-1} on the right. The same goes between Eq.(9.17a) and Eq.(9.14). Since the computation of Eq.(9.17a) and Eq.(9.17b) are almost identical, therefore the rest of the chapter shows the optimization technique for Eq.(9.17a). To overcome these overhead computations, the following techniques can be applied.

- $x_P y_P^{-1}$ is omitted by applying further isomorphic mapping of $P \in \mathbb{G}_1$.
- y_P^{-1} can be pre-computed. Therefore, the overhead calculation of Ey_P^{-1} will cost only 2 $\mathbb{F}_p$ multiplication.
- $y_P^{-1} l_{T,T}(P)$ doesn't effect the pairing calculation cost since the final exponentiation cancels this multiplication by $y_P^{-1} \in \mathbb{F}_p$.

To overcome the $Cx_Py_P^{-1}$ calculation cost, $x_Py_P^{-1} = 1$ is expected. To obtain $x_Py_P^{-1} = 1$, the following isomorphic mapping of $P = (x_P, y_P) \in \mathbb{G}_1$ is introduced.

8.3.3.1 Isomorphic map of $P = (x_P, y_P) \rightarrow \bar{P} = (x_{\bar{P}}, y_{\bar{P}})$.

Although the KSS-16 curve is typically defined over $\mathbb{F}_{p^{16}}$ as $E(\mathbb{F}_{p^{16}})$, but for efficient implementation of Optimal-Ate pairing, certain operations are carried out in a quartic twisted isomorphic curve E' defined over $\mathbb{F}_{p^4}$ as shown in Sec. 12.3.1. For the same, let us consider $\bar{E}(\mathbb{F}_{p^4})$ is isomorphic to $E(\mathbb{F}_{p^4})$ and certain $z \in \mathbb{F}_p$ as a quadratic residue (QR) in $\mathbb{F}_{p^4}$. A generalized mapping between $E(\mathbb{F}_{p^4})$ and $\bar{E}(\mathbb{F}_{p^4})$ can be given as follows:

$$\begin{aligned} \bar{E} \ &: \ y^2 = x^3 + az^{-2}x, \\ &\bar{E}(\mathbb{F}_{p^4})[r] \longmapsto E(\mathbb{F}_{p^4})[r], \\ &(x, y) \longmapsto (z^{-1}x, z^{-3/2}y), \end{aligned} \tag{8.19}$$

where

$$z, z^{-1}, z^{-3/2} \in \mathbb{F}_p$$

. The mapping considers $z \in \mathbb{F}_p$ is a quadratic residue over $\mathbb{F}_{p^4}$ which can be shown by the fact that $z^{(p^4-1)/2} = 1$ as follows:

$$\begin{aligned} z^{(p^4-1)/2} &= z^{(p-1)(p^3+p^2+p+1)/2} \\ &= 1^{(p^3+p^2+p+1)/2} \\ &= 1 \quad \text{QR} \in \mathbb{F}_{p^4}. \end{aligned} \tag{8.20}$$

Therefore, z is a quadratic residue over $\mathbb{F}_{p^4}$.
Now based on $P = (x_P, y_P)$ be the rational point on curve E, the considered isomorphic mapping of Eq.(9.18) can find a certain isomorphic rational point $\bar{P} = (x_{\bar{P}}, y_{\bar{P}})$ on curve $\bar{E}$ as follows:

$$\begin{aligned} y_P^2 &= x_P^3 + ax_P, \\ y_P^2z^{-3} &= x_P^3z^{-3} + ax_Pz^{-3}, \\ (y_Pz^{-3/2})^2 &= (x_Pz^{-1})^3 + az^{-2}x_Pz^{-1}, \end{aligned} \tag{8.21}$$

where $\bar{P} = (x_{\bar{P}}, y_{\bar{P}}) = (x_Pz^{-1}, y_Pz^{-3/2})$ and the general form of the curve $\bar{E}$ is given as follows:

$$y^2 = x^3 + az^{-2}x. \tag{8.22}$$

To obtain the target relation $x_{\bar{P}}y_{\bar{P}}^{-1} = 1$ from above isomorphic map and rational point $\bar{P}$, let us find isomorphic twist parameter z as follows:

$$\begin{aligned} x_{\bar{P}}y_{\bar{P}}^{-1} &= 1 \\ z^{-1}x_P(z^{-3/2}y_P)^{-1} &= 1 \\ z^{1/2}(x_P.y_P^{-1}) &= 1 \\ z &= (x_P^{-1}y_P)^2. \end{aligned} \tag{8.23}$$

Now using $z = (x_P^{-1}y_P)^2$ and Eq.(9.19), $\bar{P}$ can be obtained as

$$\bar{P}(x_{\bar{P}}, y_{\bar{P}}) = (x_Pz^{-1}, y_Pz^{-3/2}) = (x_P^3y_P^{-2}, x_P^3y_P^{-2}), \tag{8.24}$$

where the x and y coordinates of $\bar{P}$ are equal. For the same isomorphic map we can obtain $\bar{Q}$ on curve $\bar{E}$ defined over $\mathbb{F}_{p^{16}}$ as follows:

$$\bar{Q}(x_{\bar{Q}}, y_{\bar{Q}}) = (z^{-1}x_{Q'}\gamma, z^{-3/2}y_{Q'}\gamma\omega), \tag{8.25}$$

where from Eq.(12.10), $Q'(x_{Q'}, y_{Q'})$ is obtained in quartic twisted curve E'.

At this point, to use $\bar{Q}$ with $\bar{P}$ in line evaluation we need to find another isomorphic map that will map $\bar{Q} \mapsto \bar{Q}'$, where $\bar{Q}'$ is the rational point on curve $\bar{E}'$ defined over $\mathbb{F}_{p^4}$. Such $\bar{Q}'$ and $\bar{E}'$ can be obtained from $\bar{Q}$ of Eq.(9.23) and curve $\bar{E}$ from Eq.(9.20) as follows:

$$\begin{aligned} (z^{-3/2}y_{Q'}\gamma\omega)^2 &= (z^{-1}x_{Q'}\gamma)^3 + az^{-2}z^{-1}x_{Q'}\gamma, \\ (z^{-3/2}y_{Q'})^2\gamma^2\omega^2 &= (z^{-1}x_{Q'})^3\gamma^3 + az^{-2}z^{-1}x_{Q'}\gamma, \\ (z^{-3/2}y_{Q'})^2\beta\gamma &= (z^{-1}x_{Q'})^3\beta\gamma + az^{-2}z^{-1}x_{Q'}\gamma, \\ (z^{-3/2}y_{Q'})^2 &= (z^{-1}x_{Q'})^3 + az^{-2}\beta^{-1}z^{-1}x_{Q'}. \end{aligned}$$

From the above equations, $\bar{E}'$ and $\bar{Q}'$ are given as,

$$\bar{E}' : \; y_{\bar{Q}'}^2 = x_{\bar{Q}'}^3 + a(z^2\beta)^{-1}x_{\bar{Q}'}. \tag{8.26}$$

$$\begin{aligned} \bar{Q}'(x_{\bar{Q}'}, y_{\bar{Q}'}) &= (z^{-1}x_{Q'}, z^{-3/2}y_{Q'}), \\ &= (x_{Q'}x_P^2y_P^{-2}, y_{Q'}x_P^3y_P^{-3}). \end{aligned} \tag{8.27}$$

Now, applying $\bar{P}$ and $\bar{Q}'$, the line evaluation of Eq.(9.17b) becomes as follows:

$$\begin{aligned} y_{\bar{P}}^{-1}l_{\bar{T}',\bar{Q}'}(\bar{P}) &= 1 - C(x_{\bar{P}}y_{\bar{P}}^{-1})\gamma + Ey_{\bar{P}}^{-1}\gamma\omega, \\ \bar{l}_{\bar{T}',\bar{Q}'}(\bar{P}) &= 1 - C\gamma + E(x_P^{-3}y_P^2)\gamma\omega, \end{aligned} \tag{8.28}$$

where $x_{\bar{P}}y_{\bar{P}}^{-1} = 1$ and $y_{\bar{P}}^{-1} = z^{3/2}y_P^{-1} = (x_P^{-3}y_P^2)$. The Eq.(9.17a) becomes the same as Eq.(9.26). Compared to Eq.(9.17b), the Eq.(9.26) will be faster while using in Miller's loop in combination of the pseudo 8-sparse multiplication shown in Alg.18. However, to get the above form, we need the following pre-computations once in every Miller's Algorithm execution.

- Computing $\bar{P}$ and $\bar{Q}'$,

- $(x_P^{-3} y_P^2)$ and
- z^{-2} term from curve $\bar{E}'$ of Eq.(9.24).

The above terms can be computed from x_P^{-1} and y_P^{-1} by utilizing Montgomery trick [Mon87], as shown in Alg. 17. The pre-computation requires 21 multiplication, 2 squaring and 1 inversion in $\mathbb{F}_p$ and 2 multiplication, 3 squaring in $\mathbb{F}_{p^4}$.

Algorithm 10: Pre-calculation and mapping $P \mapsto \bar{P}$ and $Q' \mapsto \bar{Q}'$

```
Input: $P = (x_P, y_P) \in \mathbb{G}_1, Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2'$
Output: $\bar{Q}', \bar{P}, y_P^{-1}, (z)^{-2}$
$A \leftarrow (x_P y_P)^{-1}$
$B \leftarrow A x_P^2$
$C \leftarrow A y_P$
$D \leftarrow B^2$
$x_{\bar{Q}'} \leftarrow D x_{Q'}$
$y_{\bar{Q}'} \leftarrow BD y_{Q'}$
$x_{\bar{P}}, y_{\bar{P}} \leftarrow D x_P$
$y_P^{-1} \leftarrow C^3 y_P^2$
$z^{-2} \leftarrow D^2$
return $\bar{Q}' = (x_{\bar{Q}'}, y_{\bar{Q}'}), \bar{P} = (x_{\bar{P}}, y_{\bar{P}}), y_P^{-1}, z^{-2}$
```

The overall mapping and the curve obtained in the twisting process is shown in the Fig. 8.1.

Finally the Alg.13 shows the derived pseudo 8-sparse multiplication.

Algorithm 11: Pseudo 8-sparse multiplication for KSS-16 curve

```
Input: $a, b \in \mathbb{F}_{p^{16}}$
$a = (a_0 + a_1\gamma) + (a_2 + a_3\gamma)\omega,\ b = 1 + (b_2 + b_3\gamma)\omega$
$a = (a_0 + a_1\omega + a_2\omega^2 + a_3\omega^3),\ b = 1 + b_2\omega + b_3\omega^3$
Output: $c = ab = (c_0 + c_1\gamma) + (c_3 + c_4\gamma)\omega \in \mathbb{F}_{p^{16}}$
$t_0 \leftarrow a_3 \times b_3 \times \beta, t_1 \leftarrow a_2 \times b_2, t_4 \leftarrow b_2 + b_3, c_0 \leftarrow (a_2 + a_3) \times t_4 - t_1 - t_0$
$c_1 \leftarrow t_1 + t_0 \times \beta$
$t_2 \leftarrow a_1 \times b_3, t_3 \leftarrow a_0 \times b_2, c_2 \leftarrow t_3 + t_2 \times \beta$
$t_4 \leftarrow (b_2 + b_3), c_3 \leftarrow (a_0 + a_1) \times t_4 - t_3 - t_2$
$c \leftarrow c + a$
return $c = (c_0 + c_1\gamma) + (c_3 + c_4\gamma)\omega$
```

8.3.4 Final Exponentiation

Scott et al. [Sco+09] show the process of efficient final exponentiation (FE) $f^{p^k - 1/r}$ by decomposing the exponent using cyclotomic polynomial Φ_k as

$$(p^k - 1)/r = (p^{k/2} - 1) \cdot (p^{k/2} + 1)/\Phi_k(p) \cdot \Phi_k(p)/r. \tag{8.29}$$

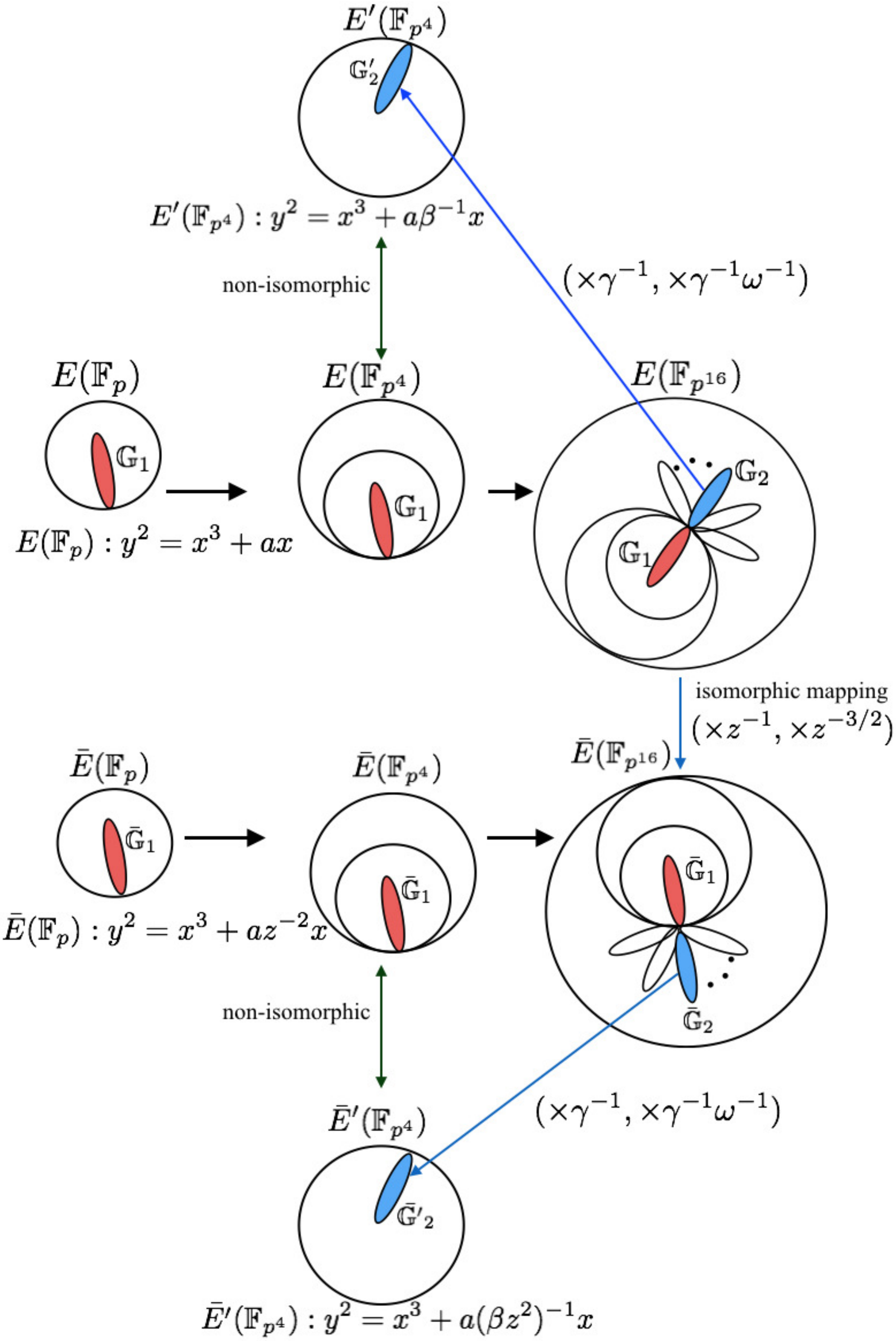

$E'(\mathbb{F}_{p^4})$
$\mathbb{G}'_2$
$E'(\mathbb{F}_{p^4}) : y^2 = x^3 + a\beta^{-1}x$
non-isomorphic
$(\times\gamma^{-1}, \times\gamma^{-1}\omega^{-1})$
$E(\mathbb{F}_p)$
$\mathbb{G}_1$
$E(\mathbb{F}_{p^4})$
$\mathbb{G}_1$
$E(\mathbb{F}_{p^{16}})$
$\mathbb{G}_2$
$\mathbb{G}_1$
$E(\mathbb{F}_p) : y^2 = x^3 + ax$
isomorphic mapping
$(\times z^{-1}, \times z^{-3/2})$
$\bar{E}(\mathbb{F}_p)$
$\bar{\mathbb{G}}_1$
$\bar{E}(\mathbb{F}_{p^4})$
$\bar{\mathbb{G}}_1$
$\bar{E}(\mathbb{F}_{p^{16}})$
$\bar{\mathbb{G}}_1$
$\bar{\mathbb{G}}_2$
$\bar{E}(\mathbb{F}_p) : y^2 = x^3 + az^{-2}x$
non-isomorphic
$(\times\gamma^{-1}, \times\gamma^{-1}\omega^{-1})$
$\bar{E}'(\mathbb{F}_{p^4})$
$\bar{\mathbb{G}}'_2$
$\bar{E}'(\mathbb{F}_{p^4}) : y^2 = x^3 + a(\beta z^2)^{-1}x$

FIGURE 8.1: Overview of the twisting process to get pseudo sparse form in KSS-16 curve.

The 1st two terms of the right part are denoted as easy part since it can be easily calculated by Frobenius mapping and one inversion in affine coordinates. The last term is called hard part which mostly affects the computation performance. According to Eq.(10.14), the exponent decomposition of the target curves is shown in Table 8.6.

TABLE 8.6: Exponents of final exponentiation in pairing

Curve	Final exponent	Easy part	Hard part
KSS-16	$\frac{p^{16}-1}{r}$	p^8-1	$\frac{p^8+1}{r}$
BN, BLS-12	$\frac{p^{12}-1}{r}$	$(p^6-1)(p^2+1)$	$\frac{p^4-p^2+1}{r}$

This chapter carefully concentrates on Miller's algorithm for comparison and making pairing efficient. However, to verify the correctness of the bilinearity property, the authors made a "not state-of-art" implementation of Fuentes et al.'s work [FKR12] for BN curve case and Ghammam's et al.'s works [GF16a; GF16b] for KSS-16 and BLS-12 curves. For scalar multiplication by prime p, i.e., $p[Q]$ or $[p^2]Q$, skew Frobenius map technique by Sakemi et al. [Sak+08] is adapted.

8.4 Experimental Result Evaluation

This section gives details of the experimental implementation. The source code can be found in Github[3]. The code is not an optimal code, and the sole purpose of it compare the Miller's algorithm among the curve families and validate the estimation of [BD18]. Table 10.3 shows implementation environment. Parameters chosen from [BD18] is shown in Table 10.4. Table 10.5

TABLE 8.7: Computational Environment

CPU*	Memory	Compiler	OS	Language	Library
Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz	4GB	GCC 5.4.0	Ubuntu 16.04 LTS	C	GMP v 6.1.0 [Gt15]

*Only single core is used from two cores.

TABLE 8.8: Selected parameters for 128-bit security level [BD18]

Curve	u	HW(u)	$\lfloor \log_2 u \rfloor$	$\lfloor \log_2 p(u) \rfloor$	$\lfloor \log_2 r(u) \rfloor$	$\lfloor \log_2 p^k \rfloor$
KSS-16	$u = 2^{35} - 2^{32} - 2^{18} + 2^8 + 1$	5	35	339	263	5424
BN	$u = 2^{114} + 2^{101} - 2^{14} - 1$	4	115	462	462	5535
BLS-12	$u = -2^{77} + 2^{50} + 2^{33}$	3	77	461	308	5532

[3] https://github.com/eNipu/pairingma128.git

shows execution time for Miller's algorithm implementation in millisecond for a single Optimal-Ate pairing. Results here are the average of 10 pairing operation. From the result, we find that Miller's algorithm took the least time for KSS-16. And the time is almost closer to BLS-12. The Miller's algorithm is about 1.7 times faster in KSS-16 than BN curve. Table 8.12 shows that the complexity of this implementation concerning the number of $\mathbb{F}_p$ multiplication and squaring and the estimation of [BD18] are almost coherent for Miller's algorithm. Table 8.12 also show that our derived pseudo 8-sparse multiplication for KSS-16 takes fewer $\mathbb{F}_p$ multiplication than Zhang et al.'s estimation [ZL12]. The execution time of Miller's algorithm also goes with this estimation [BD18], that means KSS-16 and BLS-12 are more efficient than BN curve. Table 10.6 shows the complexity of Miller's algorithm for the target curves in$\mathbb{F}_p$ operations count.

TABLE 8.9: Comparative results of Miller's Algorithm in [ms].

	KSS-16	BN	BLS-12
Miller's Algorithm	4.41	7.53	4.91

The operation counted in Table 10.6 are based on the counter in implementation code. For the implementation of big integer arithmetic `mpz_t` data type of GMP [Gt15] library has been used. For example, multiplication between 2 `mpz_t` variables are counted as $\mathbb{F}_p$ multiplication and multiplication between one `mpz_t` and one "unsigned long" integer can also be treated as $\mathbb{F}_p$ multiplication. Basis multiplication refers to the vector multiplication such as $(a_o + a_1\alpha)\alpha$ where $a_0, a_1 \in \mathbb{F}_p$ and α is the basis element in $\mathbb{F}_{p^2}$.

TABLE 8.10: Complexity of this implementation in $\mathbb{F}_p$ for Miller's algorithm [single pairing operation]

	Multiplication		Squaring	Addition/ Subtraction	Basis Multiplication	Inversion
	mpz_t * mpz_t	mpz_t * ui				
KSS-16	6162	144	903	23956	3174	43
BN	10725	232	157	35424	3132	125
BLS-12	6935	154	113	23062	2030	80

As said before, this work is focused on Miller's algorithm. However, the authors made a "not state-of-art" implementation of some final exponentiation algorithms [GF16a; FKR12; GF16b]. Table 8.11 shows the total final exponentiation time in [ms]. Here final exponentiation of KSS-16 is slower than BN and BLS-12. We have applied square and multiply technique for exponentiation by integer u in the hard part since the integer u given in the sparse form. However, Barbulescu et al. [BD18] mentioned that availability of compressed squaring [Ara+11] for KSS-16 will lead a fair comparison using final exponentiation.

TABLE 8.11: Final exponentiation time (not state-of-art) in [ms]

	KSS-16	BN	BLS-12
Final exponentiation	17.32	11.65	12.03

TABLE 8.12: Complexity comparison of Miller's algorithm between this implementation and Barbulescu et al.'s [BD18] estimation [Multiplication + Squaring in $\mathbb{F}_p$]

	KSS-16	BN	BLS-12
Barbulescu et al. [BD18]	$7534M_p$	$12068M_p$	$7708M_p$
This implementation	$7209M_p$	$11114M_p$	$7202M_p$

8.5 Conclusion and Future Work

This chapter has presented two major ideas.

- Finding efficient Miller's algorithm implementation technique for Optimal-Ate pairing for the less studied KSS-16 curve. The author's presented pseudo 8-sparse multiplication technique for KSS-16. They also extended such multiplication for BN and BLS-12 according to [Mor+14] for the new parameter.
- Verifying Barbulescu and Duquesne's conclusion [BD18] for calculating Optimal-Ate pairing at 128-bit security level; that is, BLS-12 and less studied KSS-16 curves are more efficient choices than well studied BN curves for new parameters. This chapter finds that Barbulescu and Duquesne's conclusion on BLS-12 is correct as it takes the less time for Miller's algorithm. Applying the derived pseudo 8-sparse multiplication, Miller's algorithm in KSS-16 is also more efficient than BN.

As a prospective work authors would like to evaluate the performance by finding compressed squaring for KSS-16's final exponentiation along with scalar multiplication of $\mathbb{G}_1$, $\mathbb{G}_2$ and exponentiation of $\mathbb{G}_3$. The execution time for the target environment can be improved by a careful implementation using assembly language for prime field arithmetic.

Chapter 9

INDOCRYPT Revisited Journal 2017

Finding efficiently computable underlying finite field arithmetic is one of the major bottlenecks for faster pairing operation. In this thesis, the authors exhibit efficiently computable extension field operation for Optimal-Ate pairing in Kachisa-Schaefer-Scott curve of embedding degree 16. The recent suggestion of escalating parameter's size by Barbulescu and Duquesne due to improved Kim and Barbulescu's new number field sieve (exTNFS) have taken into account while selecting the parameter for 128-bit level AES security. The authors revisited their idea of *pseudo 8-sparse multiplication* for line evaluation in Miller's algorithm presented in IndoCrypt'2017, with more efficient base field arithmetic by applying cyclic vector multiplication algorithm (CVMA) in the $\mathbb{F}_{p^4}$ extension field. To compare the complexity of this work with the previous one, the base extension field $\mathbb{F}_{p^4}$ is constructed in two different bases. Moreover, the state-of-the-art final exponentiation algorithm is optimized with cyclotomic squaring technique. The comparative results find that the CVMA has a clear advantage over Karatsuba based operation.

9.1 Introduction

Pairing-Based Cryptography (PBC) provides several protocols, for example, short signature protocols and hierarchical encryption, [Sha84], making it a promising tool for the Internet of things (IoT) or cloud computing. The inception of pairing-based cryptography by the independent work of Sakai et al. [Sak00], and Joux [Jou04] has begun a new era in cryptographic protocol innovation. A major breakthrough came when the parameterized pairing-friendly curves are given as polynomial formulas by Barreto et al. [BN06]. Over the years, distinct families of pairing-friendly elliptic curves are introduced e.g. Barreto-Lynn-Scott (BLS) [BLS03] and Kachisa-Schaefer-Scott (KSS) [KSS07] curves. At the same time, the pairing has also evolved towards a more methodical direction bringing several variants of Weil's pairing i.e. Ate [Coh+05], R-ate[LLP09], χ-ate [Nog+08] pairings. In 2010 Vercauteren proposed the Optimal-Ate pairing [Ver10] as the best pairing in terms of efficiently. In this work, we are interested in improving the Optimal-Ate pairing for the KSS-16 elliptic curve.

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two additive cyclic sub-groups and $\mathbb{G}_3$ is a multiplicative cyclic group of prime order r. Also assuming that a large set of points of an elliptic curve E of order r is defined over a finite extension field $\mathbb{F}_{p^k}$, where, p is the base field characteristic and k is called the embedding degree. The embedding degree k is the most significant complexity parameter of a pairing-friendly elliptic curve, which is defined as the smallest integer such that $r|p^k - 1$. By definition the pairing is a non-degenerate bilinear map e with $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$.

The bilinearity property allows many novel protocols with provable security such as ID-based encryption [BLS01], group signature authentication [BBS04]. For any cryptographic protocols, we have to check its security, in this context, we present the three main problems on which the security of pairing-based protocols depends.

- Infeasibility of solving the elliptic curve discrete logarithm problem (ECDLP) in the groups of order r over $\mathbb{F}_p$.
- The difficulty of solving discrete logarithm problem (DLP) in the multiplicative group $\mathbb{G}_3 \in \mathbb{F}^*_{p^k}$,
- and the difficulty of pairing inversion.

For a security level λ, $\mathbb{G}_1$ should have order of size $\log_2 r \geq 2\lambda$. In the case of parameterized curves, to balance the security and efficiency of pairing implementation a ratio index denoted as $\rho = \log_2 p / \log_2 r$ is often used. It's value ranges $1 \leq \rho \leq 2$, yet $\rho = 1$ is sought after for efficiency purpose. In practice, elliptic curves with small embedding degrees k and highest twist degree d are desired. For the case of a KSS-16 elliptic curve, the curve that we study in this thesis, ρ is equal to ≈ 1.25.
In general, to obtain 128-bit AES level security it is expected that the order r of $\mathbb{G}_1$ should be equal to 2λ (256-bit prime). Then the field size of $\mathbb{G}_1$ should be at least $\rho * 256 = 320$-bit and the lower limit of extension field size of $\mathbb{G}_3$ should be about $\rho * k * 256 = 5120$-bit. Since, $d = 4$ is the maximum twist degree for KSS-16, hence the field size of $\mathbb{G}_2 \subset E'(\mathbb{F}_{p^{k/d}})$ after twist is equal to $5120/d = 1280$-bit, where, E' is the twist curve of E.

As the parameterized pairing-friendly curve gives advantage on optimization of Miller's algorithm (MA) and final exponentiation (FE), it also comes with a cost of security. In [Sch10], Schirokauer mentioned that the Number Field Sieve (NFS) for solving DLP in $\mathbb{G}_3$ would be easier for parameterized form prime. At CRYPTO'16, Kim and Barbulescu proposed extended tower number field sieve (SexTNFS) algorithm[KB16]. Their optimization on resolving the discrete logarithm problem in $\mathbb{F}_{p^k}$ is based on the fact that the base field characteristic is presented as a polynomial. Their results intrigued researchers to find new parameters for pairing-friendly elliptic curves since the security level has changed. In response, Barbulescu and Duquesne have analyzed the security of popular pairing-friendly curve families against the NFS variants and suggested new parameters [BD18] holding twist security and immune to sub-group attack for standard security levels. In the context of Optimal-Ate pairing, they concluded that holding existing parameters, BN curve, that is the most used in practice, can endure at most 100-bit security against the exTNFS. Using their recommended new parameters, they found BLS-12 and KSS-16 curves are efficient choices over BN curve. As both BLS-12 and BN curves have the same embedding degree and both support sextic twist; therefore competitiveness between these two can be determinable from the length of integer parameter. However, the KSS-16 seems an atypical choice since the highest embedding degree supported is 4 and hasn't studied much as BN or BLS curves.

In [Kha+17b] the authors showed that Miller's loop for KSS-16 with the suggested parameter proposed in [BD18] is faster than for BN and BLS-12 with their proposed pseudo 8-sparse multiplication in Karatsuba based implementation [Kha+17b]. In this thesis, we explored to find a more efficient implementation of Optimal-Ate pairing. Therefore, we revisited the pseudo 8-sparse multiplication with cyclic vector multiplication algorithm (CVMA) [Kat+07]. This thesis adopts two different approaches of towering to construct $\mathbb{F}_{p^{16}}$ extension field. In what follows let's denote them as Type-I $\mathbb{F}_{(((p^2)^2)^2)^2}$ and Type-II $\mathbb{F}_{((p^4)^2)^2}$. The Type-I is also characterized as optimal extension field (OEF) [BP01]. Since OEF uses Karatsuba based polynomial multiplication and irreducible binomial as the modular polynomial; multiplications are efficiently carried out in OEF. In Type-II, the base extension field $\mathbb{F}_{p^4}$ is constructed with the optimal normal basis to employ cyclic vector multiplication where the modular polynomial is a degree 5 cyclotomic polynomial. We also applied Ghammam et al's [GF16a] final exponentiation algorithm with cyclotomic squaring [Kar13a] for a fair comparison. We found that Optimal-Ate in KSS-16 curve pairing using CVMA is about 30% faster than Karatsuba based implementation.

The thesis is organized into 5 sections with relevant subsections. Section 9.1 surveys the pairing in brief with related background works. Section 12.2 overviews the related fundamentals. In section 9.3 we present the main contribution. Section 9.4 and 9.5 gives the result evaluation and final words respectively.
In the rest of this thesis, we use the following notations.

- M_{p^k} is a multiplication in $\mathbb{F}_{p^k}$.
- S_{p^k} is a squaring in $\mathbb{F}_{p^k}$.
- F_{p^k} is a Frobenius map application in $\mathbb{F}_{p^k}$.
- I_{p_k} is an inversion in $\mathbb{F}_{p^k}$.

Without any additional explanation, lower and upper case letters show elements in prime field and extension field, respectively, and a lower case Greek alphabet denotes a zero of a modular polynomial.
For simplicity, we use M_p, S_p, I_p, A_p instead of M_1, S_1 and I_1 and the m with lower case Greek suffix denotes multiplication with basis element.

9.2 Fundamentals of Elliptic Curve and Pairing

9.2.1 Kachisa-Schaefer-Scott (KSS) Curve [KSS07]

Kachisa, Schaefer, and Scott proposed a new family of parameterized non super-singular pairing-friendly elliptic curves of embedding degree $k = \{16, 18, 32, 36, 40\}$. Unlike BN and BLS pairing-friendly curve families, usually embraced with the sextic twist, the KSS family holds the quartic twist. In what follows, this thesis considers the KSS curve of embedding degree $k = 16$, denoted as *KSS-16*, defined over extension field $\mathbb{F}_{p^{16}}$ as follows:

$$E/\mathbb{F}_{p^{16}} : Y^2 = X^3 + aX, \quad (a \in \mathbb{F}_p) \text{ and } a \neq 0, \tag{9.1}$$

where $X, Y \in \mathbb{F}_{p^{16}}$. As a typical feature of pairing-friendly curves, it's properties are given by the polynomial formulas of integer u as follows:

$$\begin{aligned} p(u) &= (u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 \\ &\quad +625u^2 + 2398u + 3125)/980, && (9.2a) \\ r(u) &= (u^8 + 48u^4 + 625)/61255, && (9.2b) \\ t(u) &= (2u^5 + 41u + 35)/35, && (9.2c) \end{aligned}$$

where the tuple (p, r, t) are *characteristic*, *group order* and *Frobenius trace* respectively. The integer u, denoted the pairing parameter, is abide by the condition $u \equiv 25$ or $45 \pmod{70}$.

9.2.2 Extension Field Arithmetic for Pairing

While implementing pairing, a major speedup comes from the efficient finite field implementation. Calculation of pairing requires executing the arithmetic operation in the extension field of degree greater than 6[BS09]. In what follows, the aforementioned towering procedure of $\mathbb{F}_{p^{16}}$ extension field is given with the irreducible polynomials.

9.2.2.1 Type-I towering

Efficient extension field $\mathbb{F}_{p^4}$ with the Karatsuba-based method is constructed by a towering technique such as $\mathbb{F}_{(p^2)^2}$. For such construction, in addition with $4|p-1$, p satisfies $p \equiv 3, 5 \bmod 8$.

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 - c_0), \\ \mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/(\beta^2 - \alpha), \\ \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\gamma]/(\gamma^2 - \beta), \\ \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\omega]/(\omega^2 - \gamma), \end{cases} \tag{9.3}$$

where c_0 is a quadratic non-residue (QNR) in $\mathbb{F}_p$. This thesis considers $c_0 = 2$, where $X^{16} - 2$ is irreducible in $\mathbb{F}_{p^{16}}$.

9.2.2.2 Type-II towering

An additional condition $p \equiv 2, 3 \bmod 5$ is required to construct this towering.

$$\begin{cases} \mathbb{F}_{p^4} = \mathbb{F}_p[\alpha]/(\alpha^4 + \alpha^3 + \alpha^2 + \alpha + 1), \\ \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\beta]/(\beta^2 - (\alpha \pm c_1)), \\ \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\gamma]/(\gamma^2 - \beta). \end{cases} \tag{9.4}$$

Here the $\Phi_5(x) = (x^5 - 1)/(x - 1)$ is irreducible over $\mathbb{F}_{p^4}$ and $(\alpha \pm c_1)$ should be the QNR in $\mathbb{F}_{p^4}$. In what follows, when the basis elements are implicitly known, the vector representation $A = (a_0, a_1, a_2, a_3) \in \mathbb{F}_{p^4}$ refers to the same element represented as $A = a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4$.

9.2.2.3 Field Arithmetic of $\mathbb{F}_{p^{16}}$

For any platform, multiplication, squaring and inversion are regarded as computationally expensive than addition or subtraction. For convenient estimation of the total pairing cost, we count operations in $\mathbb{F}_p$ for extension field arithmetic. The following table, Table 12.1 shows operation count for Karatsuba based multiplication and squaring. The squaring is optimized by using Devegili et al.'s [Dev+06] complex squaring technique which costs $2M_p + 4A_p + 2m_\alpha$ for one squaring operation in $\mathbb{F}_{p^2}$. Since, $c_0 = 2$ in Eq.(12.5), therefore, the multiplication by the basis element α is carried out by 1 addition in $\mathbb{F}_p$.

Multiplication	Squaring
$M_{p^2} = 3M_p + 5A_p + 1m_\alpha \longrightarrow 3M_p$	$S_{p^2} = 2M_p + 6A_p + \longrightarrow 2M_p$
$M_{p^4} = 2M_{p^2} + 5A_{p^2} + 1m_\beta \longrightarrow 9M_p$	$S_{p^4} = 2M_{p^2} + 5A_{p^2} + 2m_\beta \longrightarrow 6M_p$
$M_{p^8} = 3M_{p^4} + 5A_{p^4} + 1m_\gamma \longrightarrow 27M_p$	$S_{p^8} = 2M_{p^4} + 5A_{p^4} + 2m_\gamma \longrightarrow 18M_p$
$M_{p^{16}} = 3M_{p^8} + 5A_{p^8} + 1m_\omega \longrightarrow 81M_p$	$S_{p^{16}} = 2M_{p^8} + 5A_{p^8} + 2m_\omega \longrightarrow 54M_p$

TABLE 9.1: Number of arithmetic operations in $\mathbb{F}_{p^{16}}$ based on Type-I towering Eq.(12.5).

9.2.3 Optimal-Ate Pairing on KSS-16 Curve

In the context of pairing on the KSS-16 curves, the valid bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ takes input from two additive rational point groups $\mathbb{G}_1, \mathbb{G}_2$ and output an element in the multiplicative group $\mathbb{G}_3$ of order r. $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\begin{aligned} \mathbb{G}_1 &= E(\mathbb{F}_p)[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ \mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r, \end{aligned}$$

where $E(\mathbb{F}_{p^k})[r]$ denotes rational points of order r and $[n]$ is scalar multiplication for a rational point. Let π_p denotes the Frobenius endomorphism given as $\pi_p : (x, y) \mapsto (x^p, y^p)$.

Unless otherwise stated, rest of the thesis considers $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$. The map e involves two major steps named Miller's loop followed by the final exponentiation. The Optimal-Ate pairing [Ver10] proposed by Vercauteren reduces the Miller's loop length to $\lfloor \log_2 u \rfloor = \frac{\lfloor \log_2 r \rfloor}{\varphi(k)}$, where φ is the Euler's totient function. The choice of the parameter u is an important factor for efficient Miller's algorithm since the smaller hamming weight of u adds advantage by reducing elliptic curve doubling (ECD) inside the loop.

The Optimal-Ate pairing on KSS-16 elliptic curve is given by Zhang et al. [ZL12] and presented by the following map.

$$\begin{array}{rcl} e_{opt} : \mathbb{G}_1 \times \mathbb{G}_2 & \longrightarrow & \mathbb{G}_3 \\ (P, Q) & \longmapsto & \left((f_{u,Q}(P) l_{[u]Q,[p]Q}(P))^{p^3} l_{Q,Q}(P) \right)^{\frac{p^{16}-1}{r}} \end{array}$$

The rational function $f_{u,Q}(P)$ is computed thanks to Miller algorithm which is included in the first step of computing the Optimal-Ate pairing. Then, we have the second step which is the computation of the exponent $\frac{p^{16}-1}{r}$ named the Final Exponentiation.
The calculation of the Optimal-Ate pairing in KSS-16 elliptic curve is given by the following algorithm 16.

Algorithm 12: The Optimal-Ate pairing algorithm for KSS-16 curve

```
Input: $u, P \in \mathbb{G}_1, Q \in \mathbb{G}'_2$
Output: $(Q, P)$
$f \leftarrow 1, T \leftarrow Q$
for $i = \lfloor \log_2(u) \rfloor$ downto $1$ do
    $f \leftarrow f^2 \cdot l_{T,T}(P), T \leftarrow [2]T$          ▷ (see Eq.(9.14))
    if $u[i] = 1$ then
        $f \leftarrow f \cdot l_{T,Q}(P), T \leftarrow T + Q$       ▷ (see Eq.(9.16))
    if $u[i] = -1$ then
        $f \leftarrow f \cdot l_{T,-Q}(P), T \leftarrow T - Q$      ▷ (see Eq.(9.16))
$Q_1 \leftarrow [u]Q, Q_2 \leftarrow [p]Q$
$f \leftarrow f \cdot l_{Q_1,Q_2}(P)$
$f_1 \leftarrow f^{p^3}, f \leftarrow f \cdot f_1$
$f \leftarrow f \cdot l_{Q,Q}(P)$
$f \leftarrow f^{\frac{p^{16}-1}{r}}$
return $f$
```

Steps between 1 to 11 are identified as Miller's algorithm and step 12 is the FE. Optimization scopes of the thesis are the line evaluation of steps 3, 5, 7, 9, 11 together with ECD and ECA. These line evaluation steps are the key steps to accelerate the Miller loop calculation.

In [Kha+17b], the authors showed an efficient technique for the above steps by *pseudo 8-sparse multiplication* in the optimal extension field. The calculations were carried out in affine coordinates using Karatsuba based multiplications in Type-I towering.

In the next sections, we will show the revision of *pseudo 8-sparse multiplication* by using CVMA based multiplication. In addition authors also optimize the step 12 calculation: the final exponentiation by cyclotomic squaring [GS10] in Ghammam et al.'s [GF16a] final exponentiation algorithm.

9.3 Finding Efficient Line Evaluation in Type-II Towering and Sparse Multiplication

This section describes the main idea of obtaining efficient line evaluation for the proposed towering Eq.(9.4) with the combination of *pseudo 8-sparse multiplication*. In [Kha+17b], the authors showed the *pseudo 8-sparse multiplication* for towering Eq.(12.5). In this thesis, the parameter and consequently the settings of KSS-16 curve is different than [Kha+17b]. Most importantly the basis representation and underlying finite field arithmetic are also changed. Therefore, in this section, the authors will revisit [Kha+17b] by using CVMA. The overall process is as follows:

1. Finding efficient finite field operation in $\mathbb{F}_{p^4}$.
   - efficient inversion, multiplication, squaring and Frobenius map using CVMA.
2. Finding the quartic twisted curve $E'(\mathbb{F}_{p^4})$ of $E(\mathbb{F}_{p^{16}})$ and define the isomorphic mapping $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}}) \mapsto \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$ between the rational points.
3. Obtaining the line equation in $E(\mathbb{F}_{p^{16}})$, nevertheless, the actual calculation is in $\mathbb{F}_{p^4}$.
4. Finding the more sparse line representation by:
   - using isomorphic map of $\mathbb{G}_1 \mapsto \bar{\mathbb{G}_1}' \subset \bar{E}(\mathbb{F}_p)$ and $\mathbb{G}_2 \mapsto \bar{\mathbb{G}_2}$.
   - Finding another twisted map $\bar{\mathbb{G}_2} \mapsto \bar{\mathbb{G}}_2'$.
   - Rational points from the $\bar{\mathbb{G}}_2' \subset \bar{E}'(\mathbb{F}_{p^4})$ and $\bar{\mathbb{G}}_1' \subset \bar{E}(\mathbb{F}_p)$ act as the input of the Miller's algorithm.
5. Deriving *pseudo 8-sparse multiplication* using the sparse form obtained in step 4.
6. Computing the final exponentiation by using algorithm in [GF16a] together with cyclotomic squaring [GS10].

7. Finally, we compare the proposed implementation with [Kha+17b]'s approach.

9.3.1 $\mathbb{F}_{p^4}$ arithmetic in Type-II towering

In [San+16] (Japanese), Sanada et al. primarily focus on the $\mathbb{F}_{p^4}$ finite field operation. They reduced 5 and 3 prime field additions for a single $\mathbb{F}_{p^4}$ multiplication and squaring respectively than Karatsuba method. However, $\mathbb{F}_{p^4}$ inversion in [San+16] requires $(31M_p + 66A_p + 1I_p)$. In contrast, the authors applied Karatsuba based $\mathbb{F}_{p^4}$ inversion in [Kha+17b] which costs $(14M_p + 29A_p + 1I_p)$. In this thesis, the authors derived a better $\mathbb{F}_{p^4}$ inversion than [San+16] that reduces the cost to $(16M_p + 26A_p + 1I_p)$. The comparative operation count is shown in Table 9.2.

$\mathbb{F}_{p^4}$ operations	Karatsuba method	CVMA method
Multiplication	$9M_p + 29A_p$	$9M_p + 22A_p$
Squaring	$6M_p + 24A_p$	$6M_p + 14A_p$
Inversion	$14M_p + 29A_p + 1I_p$	$16M_p + 26A_p + 1I_p$

TABLE 9.2: Number of $\mathbb{F}_p$ operations in the field $\mathbb{F}_{p^4}$ based on Type-I and Type-II towering.

9.3.1.1 Multiplication in $\mathbb{F}_{p^4}$ using CVMA

Let's consider A, B, two elements in $\mathbb{F}_{p^4}$ based on Eq.(9.4) as follows:

$$\begin{aligned} A &= a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4, \\ B &= b_0\alpha + b_1\alpha^2 + b_2\alpha^3 + b_3\alpha^4, \end{aligned}$$

where $a_i, b_i \in \mathbb{F}_p$ and $i = 0, 1, 2, 3$.

$$\begin{aligned} A{\times}B = \; & (a_2b_2 + a_1b_3 + a_3b_1 - a_0b_3 - a_1b_2 - a_2b_1 - a_3b_0)\alpha \\ & +(a_0b_0 + a_2b_3 + a_3b_2 - a_0b_3 - a_1b_2 - a_2b_1 - a_3b_0)\alpha^2 \\ & +(a_3b_3 + a_0b_1 + a_1b_0 - a_0b_3 - a_1b_2 - a_2b_1 - a_3b_0)\alpha^3 \\ & +(a_1b_1 + a_0b_2 + a_2b_0 - a_0b_3 - a_1b_2 - a_2b_1 - a_3b_0)\alpha^4. \end{aligned} \tag{9.5}$$

By noticing that each term of Eq.(9.5) shares the common term $-a_0b_3 - a_1b_2 - a_2b_1 - a_3b_0$; we can consider this fact in the following expression U_1:

$$U_1 = (a_0 - a_3)(b_0 - b_3) + (a_1 - a_2)(b_1 - b_2). \tag{9.6}$$

By using the Eq.(9.6), Eq.(9.5) can be expressed as follows:

$$\begin{aligned} A\times B &= \{U_1-(a_1-a_3)(b_1-b_3)-a_0b_0\}\alpha \\ &\quad +\{U_1-(a_2-a_3)(b_2-b_3)-a_1b_1\}\alpha^2 \\ &\quad +\{U_1-(a_0-a_1)(b_0-b_1)-a_2b_2\}\alpha^3 \\ &\quad +\{U_1-(a_0-a_2)(b_0-b_2)-a_3b_3\}\alpha^4. \end{aligned} \tag{9.7}$$

Here, the Eq.(9.6) can be optimized more and expressed as U_2:

$$\begin{aligned} U_2 &= (a_0-a_3)(b_0-b_3)+(a_1-a_2)(b_1-b_2), \\ &= (a_0+a_1-a_2-a_3)(b_0+b_1-b_2-b_3)\{(a_0-a_3)(b_1-b_2)+(b_0-b_3)(a_1-a_2)\}, \\ &= (a_0+a_1-a_2-a_3)(b_0+b_1-b_2-b_3)+(a_0-a_1)(b_0-b_1)-(a_0-a_2)(b_0-b_2) \\ &\quad -(a_1-a_3)(b_1-b_3)+(a_2-a_3)(b_2-b_3). \end{aligned}$$

Now let us replace U_1 in Eq.(9.7) with U_2 and express $A\times B = S_1\alpha + S_2\alpha^2 + S_3\alpha^3 + S_4\alpha^4$, where S_1, S_2, S_3, S_4 coefficients are given as follows:

$$\begin{aligned} S_1 &= U_2-T_5-a_0b_0, \quad S_2 = U_2-T_8-a_1b_1, \\ S_3 &= U_2-T_7-a_2b_2, \quad S_4 = U_2-T_6-a_3b_3, \end{aligned}$$

With

$$\begin{aligned} U_2 &= (T_1+T_2)(T_3+T_4)-T_5-T_6+T_7+T_8, \quad T_1 = a_0-a_2, \quad T_2 = a_1-a_3, \quad T_3 = b_0-b_2, \\ T_4 &= b_1-b_3, \quad T_5 = T_2T_4, \quad T_6 = T_1T_3, \quad T_7 = (a_0-a_1)(b_0-b_1), \ T_8 = (a_2-a_3)(b_2-b_3). \end{aligned}$$

The cost of each computed term is given in the following Table 9.3. In total the multiplication in $\mathbb{F}_{p^4}$ costs $9M_p+22A_p$, which saves $5A_p$ compared to Karatsuba based multiplication for elements in $\mathbb{F}_{p^4}$.

Computed Terms	Cost of each term
T_1, T_2, T_3, T_4	A_p
T_5, T_6	M_p
T_7, T_8	M_p+2A_p
U_2	M_p+6A_p
S_1, S_2, S_3, S_4	M_p+2A_p

TABLE 9.3: The detailed cost of a multiplication in $\mathbb{F}_{p^4}$ using CVMA technique.

9.3.1.2 Squaring in $\mathbb{F}_{p^4}$ using CVMA

To compute the squaring of $A \in \mathbb{F}_{p^4}$, we will replace the b_i terms in Eq.(9.5) by a_i, with $i \in \{0, 1, 2, 3\}$ obtaining A^2 as follows:

$$\begin{aligned}
A^2 &= (2a_1a_3 - 2a_0a_3 - 2a_1a_2 + a_2^2)\alpha + (2a_2a_3 - 2a_0a_3 - 2a_1a_2 + a_0^2)\alpha^2 \\
&\quad +(2a_0a_1 - 2a_0a_3 - 2a_1a_2 + a_3^2)\alpha^3 + (2a_0a_2 - 2a_0a_3 - 2a_1a_2 + a_1^2)\alpha^4, \\
&= \{2(a_0 - a_1)(a_2 - a_3) - 2a_0a_2 + a_2^2\}\alpha + \{2(a_0 - a_2)(a_1 - a_3) - 2a_0a_1 + a_0^2\}\alpha^2 \\
&\quad +\{2(a_0 - a_2)(a_1 - a_3) - 2a_2a_3 + a_3^2\}\alpha^3 + \{2(a_0 - a_1)(a_2 - a_3) - 2a_1a_3 + a_1^2\}\alpha^4, \\
&= \{2(a_0 - a_1)(a_2 - a_3) - a_2(2a_0 - a_2)\}\alpha + \{2(a_0 - a_2)(a_1 - a_3) - a_0(2a_1 - a_0)\}\alpha^2 \\
&\quad +\{2(a_0 - a_2)(a_1 - a_3) - a_3(2a_2 - a_3)\}\alpha^3 + \{2(a_0 - a_1)(a_2 - a_3) \\
&\quad -a_1(2a_3 - a_1)\}\alpha^4. \qquad (9.8)
\end{aligned}$$

Let $A^2 = S_1\alpha + S_2\alpha^2 + S_3\alpha^3 + S_4\alpha^4$. From Eq.(9.8), S_1, S_2, S_3, S_4 can be obtained as follows.

$$\begin{aligned}
S_1 &= T_5 - a_2(a_0 + T_1), S_2 = T_6 - a_0(a_1 - T_2), \\
S_3 &= T_6 - a_3(a_2 + T_3), S_4 = T_5 - a_1(a_3 - T_4).
\end{aligned}$$

With

$$T_1 = a_0 - a_2,\ T_2 = a_0 - a_1,\ T_3 = a_2 - a_3,\ T_4 = a_1 - a_3,\ T_5 = 2T_2T_3,\ T_6 = 2T_1T_4.$$

The cost of each computed term is given in the following Table 9.4. The overall cost for computing a squaring by CVMA is then $6M_p + 14A_p$. It saves $10A_p$ than Karatsuba based squaring for $\mathbb{F}_{p^4}$ elements.

Computed Terms	Cost
T_1, T_2, T_3, T_4	A_p
T_5, T_6	$M_p + A_p$
S_1, S_2, S_3, S_4	$M_p + 2A_p$

TABLE 9.4: The detailed cost of a squaring in $\mathbb{F}_{p^4}$ using CVMA.

9.3.1.3 Frobenius mapping in $\mathbb{F}_{p^4}$ using CVMA

Since, $\alpha^5 = 1$, then, $\alpha^p = (\alpha^5)^{\frac{p-2}{5}}\alpha^2 = \alpha^2$. Recall that the Frobenius map, denoted as $\pi_p : (A) = (a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4)^p$, is the p-th power of the vector which can be derived as follows:

$$\begin{aligned}
A^p &= (a_0\alpha + a_1\alpha^2 + a_2\alpha^3 + a_3\alpha^4)^p \\
&= a_0^p\alpha^p + a_1^p\alpha^{2p} + a_2^p\alpha^{3p} + a_3^p\alpha^{4p} \\
&= a_0\alpha^2 + a_1\alpha^4 + a_2\alpha + a_3\alpha^3 \\
&= a_2\alpha + a_0\alpha^2 + a_3\alpha^3 + a_1\alpha^4 \\
&= (a_2, a_0, a_3, a_1). \qquad (9.9)
\end{aligned}$$

From the above procedure it is clear that the Frobenius map on an $\mathbb{F}_{p^4}$ element by applying CVMA is free of cost.

9.3.1.4 Inversion in $\mathbb{F}_{p^4}$ used in [San+16]

Let L be an $\mathbb{F}_{p^4}$ element, which is the result of the product of the Frobenius maps A^p, A^{p^2}, A^{p^3}. The inversion of A can be obtained as follows.

$$L = A^p A^{p^2} A^{p^3},\ s = AL \in \mathbb{F}_p,$$
$$A^{-1} = s^{-1}L,$$

where $s \in \mathbb{F}_p$ element represented as $(-s, -s, -s, -s)$ in normal basis. The calculation cost becomes $((9M_p + 22A_p) \times 3M_p) + 4M_p + I_p = 31M_p + 66A_p + I_p$.

9.3.1.5 Optimized $\mathbb{F}_{p^4}$ Inversion using CVMA

Let $A = (a_0, a_1, a_2, a_3)$ be an element in $\mathbb{F}_{p^4}$. The proposed optimized method applies subfield calculation in $\mathbb{F}_{p^2}$ as

$$\begin{aligned} B &= AA^{p^2} \in \mathbb{F}_{p^2}, \\ A^{-1} &= B^{-1}A^{p^2}, \end{aligned}$$

where, $B \in \mathbb{F}_{p^2} = (b_0, b_1, b_1, b_0)$ in the normal basis. While $p \equiv 2 \pmod 5$, Frobenius mapping A^{p^2} is equal to (a_3, a_2, a_1, a_0), i.e. coefficients only change the basis position without costing any $\mathbb{F}_p$ operation. Therefore, b_0 and b_1 are given as follows:

$$b_0 = -(a_0 + a_1 - a_2 - a_3)^2 + 3(a_0 - a_2)(a_1 - a_3) - 2(a_0 - a_1)(a_2 - a_3) - a_0a_3,$$
$$b_1 = -(a_0 + a_1 - a_2 - a_3)^2 + 2(a_0 - a_2)(a_1 - a_3) - (a_0 - a_1)(a_2 - a_3) - a_1a_2,$$

which costs $(4M_P + S_p + 12A_p)$. Then, B^{-1} can be calculated as follows:

$$\begin{aligned} s &= BB^p \in \mathbb{F}_p, \\ B^{-1} &= s^{-1}B^p, \end{aligned}$$

where $s = (-s, -s, -s, -s)$ in the normal basis defined in Eq.(9.4). The Frobenius mapping B^p becomes (b_1, b_0, b_0, b_1) and s can be expressed as $s = -(b_0 - b_1)^2 + b_0b_1$. Therefore, one inversion cost over $\mathbb{F}_{p^2}$ is $3M_p + S_p + 2A_p + I_p$. If B^{-1} is represented as (b'_0, b'_1, b'_1, b'_0), $A^{-1} = B^{-1}A^{p^2} = (a'_0, a'_1, a'_2, a'_3)$ is calculated as follows with a cost $(7M_p + 12A_p)$.

$$a'_0 = (b'_0 - b'_1)(a_1 - a_0) - b'_0a_0 + (b'_0 - b'_1)(a_0 - a_3),$$
$$a'_1 = (b'_0 - b'_1)(a_1 - a_0) - b'_1a_1 + (b'_0 - b'_1)(a_0 - a_3) + (b'_0 - b'_1)(a_2 - a_1),$$
$$a'_2 = (b'_0 - b'_1)(a_1 - a_0) - b'_1a_2,$$
$$a'_3 = (b'_0 - b'_1)(a_1 - a_0) - b'_0a_3 + (b'_0 - b'_1)(a_2 - a_1).$$

Then, by applying this method, inversion cost over $\mathbb{F}_{p^4}$ becomes $14M_p + 2S_p + 26A_p + I_p$. In what follows, this thesis considers the cost of one $\mathbb{F}_p$ squaring, as a similar cost of one $\mathbb{F}_p$ multiplication. The details of CVMA based operations in $\mathbb{F}_{p^2}$ for the above inversion is described in following sections.

9.3.1.6 Calculation over $\mathbb{F}_{p^2}$ based on towering Eq.(9.4)

Let $X = (x_0, x_1, x_1, x_0)$ and $Y = (y_0, y_1, y_1, y_0)$ be two $\mathbb{F}_{p^2}$ elements. In this paragraph, we present the cost of the multiplication of X and Y, the squaring of X and its Frobenius.

9.3.1.6.1 Multiplication: Let R be the result of computing the multiplication XY, $R = (r_0, r_1, r_1, r_0)$ is calculated as follows:

$$\begin{aligned} r_0 &= -(x_0 - x_1)(y_0 - y_1) - x_0 y_0, \\ r_1 &= -(x_0 - x_1)(y_0 - y_1) - x_1 y_1. \end{aligned}$$

It is simple to verify that the cost of computing $R = XY$ is $(3M_p + 4A_p)$.

9.3.1.6.2 Squaring: Let R be the result of computing the squaring of X. $R = X^2 = (r_0, r_1, r_1, r_0)$ can be computed as follows.

$$\begin{aligned} r_0 &= -(x_0 - x_1)^2 - {x_0}^2, \\ r_1 &= -(x_0 - x_1)^2 - {x_1}^2. \end{aligned}$$

This calculation costs $(3S_p + 5A_p)$.

9.3.1.6.3 Frobenius map: According to Eq.(9.9), Frobenius mapping X^p is calculated with no-cost. It consists only in changing the positions of the X_i as $X^p = (x_1, x_0, x_0, x_1)$.

9.3.1.6.4 Inversion: The inversion of X denoted $R = X^{-1} = (r_0, r_1, r_1, r_0)$ is calculated using the following steps.

$$\begin{aligned} u &= XX^p, \\ X^{-1} &= u^{-1}X^p, \end{aligned}$$

where $u = (-u, -u, -u, -u)$ is given by $u = -(x_0 - x_1)^2 + x_0 x_1$ Therefore, the inversion in $\mathbb{F}_{p^2}$ requires $(3M_p + S_p + 2A_p + I_p)$.

9.3.1.7 Frobenius mapping in $\mathbb{F}_{p^{16}}$ using CVMA

Let $A = (a_0 + a_1\beta + a_2\gamma + a_3\beta\gamma)$ be certain vector in $\mathbb{F}_{p^{16}}$ where $a_0, a_1, a_2, a_3 \in \mathbb{F}_{p^4}$. By the definition, Frobenius map of A, i.e. $\pi_p : (A) = (a_0 + a_1\beta + a_2\gamma + a_3\beta\gamma)^p$, can be computed as Frobenius map of each $\mathbb{F}_{p^4}$ vector separately according to Eq.(9.9). The Frobenius map of a_0 is obtained as $(x_0\alpha + x_1\alpha^2 + x_2\alpha^3 + x_3\alpha^4)^p = (x_2\alpha + x_0\alpha^2 + x_3\alpha^3 + x_1\alpha^4)$, where $x_i \in \mathbb{F}_p$. Similarly, for a_1, a_2 and a_3, it will be obtained by swapping the coefficients position. The Frobenius map of the basis elements $\beta^p, \gamma^p, (\beta\gamma)^p$ can be obtained as follows:

$$\begin{aligned}\beta^p &= (\beta^2)^{\frac{p-1}{2}}\beta \\ &= (\alpha-1)^{\frac{p-1}{2}}\beta,\end{aligned} \qquad \begin{aligned}\gamma^p &= (\gamma^2)^{\frac{p-1}{2}}\gamma \\ &= (\beta)^{\frac{p-1}{2}}\gamma \\ &= (\beta^2)^{\frac{p-1}{4}}\gamma \\ &= (\alpha-1)^{\frac{p-1}{4}}\gamma,\end{aligned} \qquad \begin{aligned}\beta^p\gamma^p &= (\alpha-1)^{\frac{p-1}{2}}\beta(\alpha-1)^{\frac{p-1}{4}}\gamma \\ &= (\alpha-1)^{\frac{3(p-1)}{4}}\beta\gamma.\end{aligned}$$

Using the above calculations, the Frobenius map for A^p is obtained as follows:

$$\begin{aligned}A^p \;=\; & (x_2\alpha + x_0\alpha^2 + x_3\alpha^3 + x_1\alpha^4) \\ & +(x_6\alpha + x_4\alpha^2 + x_7\alpha^3 + x_5\alpha^4)(\alpha-1)^{\frac{(p-1)}{2}}\beta \\ & +(x_{10}\alpha + x_8\alpha^2 + x_{11}\alpha^3 + x_9\alpha^4)(\alpha-1)^{\frac{(p-1)}{4}}\gamma \\ & +(x_{14}\alpha + x_{12}\alpha^2 + x_{15}\alpha^3 + x_{13}\alpha^4)(\alpha-1)^{\frac{3(p-1)}{4}}\beta\gamma. \qquad (9.10)\end{aligned}$$

Here, it requires 3 multiplication of $\mathbb{F}_{p^4}$ elements $(\alpha-1)^{\frac{(p-1)}{2}}, (\alpha-1)^{\frac{(p-1)}{4}}, (\alpha-1)^{\frac{3(p-1)}{4}}$, with the 2nd, 3rd and 4th term of Eq.(9.10) respectively; costing 27 $\mathbb{F}_p$ multiplication, whereas in Karatsuba case it is just 14 $\mathbb{F}_p$ multiplication.

9.3.2 Quartic Twist of KSS-16 Curves

The KSS-16 elliptic curve has CM discriminant of $D = 1$ and it's embedding degree $k = 16$ is a multiple of 4. Therefore, the maximum twist available for KSS-16 is the quartic twist or degree $d = 4$ twist. Let $(\alpha - 1)$ has no square root in $\mathbb{F}_{p^4}$. Then, the quartic twisted curve E' of curve E and their isomorphic mapping ψ_4 can be given as follows:

$$\begin{aligned}\psi_4 :& E'(\mathbb{F}_{p^4})[r] \longmapsto E(\mathbb{F}_{p^{16}})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ & (x, y) \longmapsto ((\alpha-1)^{1/2}x, (\alpha-1)^{3/4}y), \qquad (9.11)\end{aligned}$$

recall that E is defined in Eq.(12.1) and E' is the twisted elliptic curve defined as $y^2 = x^3 + ax(\alpha-1)^{-1}, \quad a \in \mathbb{F}_p$. Since points on the twisted curve are defined over a smaller field than $\mathbb{F}_{p^{16}}$, therefore, their vector representation becomes shorter, resulting in faster ECA and ECD during Miller's loop.

9.3.2.0.1 Rational points: Let, $Q' = (x', y')$ be a rational point in $E'(\mathbb{F}_{p^4})$. From Eq.(9.4), we have $(\alpha-1)^{1/2} = \beta$ and $(\alpha-1)^{3/4} = \beta\gamma$. Therefore, the map given in Eq.(9.11) enables toll free mapping and remapping between $Q = (x, y)$ and $Q' = (x', y')$. Table 11.1 shows the vector representation of $Q = (x_Q, y_Q) = ((\alpha-1)^{1/2}x_{Q'}, (\alpha-1)^{3/4}y_{Q'}) \in \mathbb{F}_{p^{16}}$ according to Eq.(9.4).

It's important here to show that $(\alpha - 1)$ is a QNR in $\mathbb{F}_{p^4}$. From the definition of Eq.(9.4), α is one of the zeros of $\Phi_5(x)$, therefore $\alpha^5 = 1$. As a result, Frobenius

Type-I	1	α	β	$\alpha\beta$	γ	$\alpha\gamma$	$\beta\gamma$	$\alpha\beta\gamma$	ω	$\alpha\omega$	$\beta\omega$	$\alpha\beta\omega$	$\gamma\omega$	$\alpha\gamma\omega$	$\beta\gamma\omega$	$\alpha\beta\gamma\omega$
x_Q	0	0	0	0	a_4	a_5	a_6	a_7	0	0	0	0	0	0	0	0
y_Q	0	0	0	0	0	0	0	0	0	0	0	0	a_{12}	a_{13}	a_{14}	a_{15}
Type-II	α	α^2	α^3	α^4	$\alpha\beta$	$\alpha^2\beta$	$\alpha^3\beta$	$\alpha^4\beta$	$\alpha\gamma$	$\alpha^2\gamma$	$\alpha^3\gamma$	$\alpha^4\gamma$	$\alpha\beta\gamma$	$\alpha^2\beta\gamma$	$\alpha^3\beta\gamma$	$\alpha^4\beta\gamma$

TABLE 9.5: Vector representation of $Q = (x_Q, y_Q) \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$.

map $\alpha^p = \alpha^2(\alpha^5)^{(\frac{p-2}{5})} = \alpha^2$, since $p \equiv 2 \bmod 5$.

$$\begin{aligned}
(\alpha-1)^{\frac{p^4-1}{2}} &= (\alpha-1)^{(p^2+1)(\frac{p^2-1}{2})} \\
&= ((\alpha-1)(\alpha-1)^{p^2})^{(\frac{p^2-1}{2})} \\
&= ((\alpha-1)(\alpha^4-1))^{(\frac{p^2-1}{2})} \\
&= ((\alpha^5-\alpha^4-\alpha+1)^{(\frac{p^2-1}{2})} \\
&= ((-\alpha^4-\alpha+2)^{(p+1)(\frac{p-1}{2})} \\
&= ((-\alpha^4-\alpha+2)(-\alpha^4-\alpha+2)^p)^{(\frac{p-1}{2})} \\
&= (-\alpha-\alpha^2-\alpha^3-\alpha^4+4)^{(\frac{p-1}{2})} \\
&= 5^{(\frac{p-1}{2})},
\end{aligned}$$

where, $5^{(\frac{p-1}{2})}$ is the Legendre symbol $(5/p) = -1$, which refers $(\alpha - 1)$ is a QNR in $\mathbb{F}_{p^4}$.

9.3.3 Overview: Sparse and Pseudo-Sparse Multiplication

Pseudo 8-sparse refers to a certain length of vector's coefficients where instead of 8 zero coefficients, there are seven 0's and one 1 as coefficients. Mori et al. [Mor+14] shown the pseudo 8-sparse multiplication for BN curve in affine coordinates where the sextic twist is available. In [Mor+14], pseudo 8-sparse is found a little more efficient than 7-sparse in similar coordinates and 6-sparse in Jacobian coordinates.

Let us consider $T = (x_{T'}\beta, y_{T'}\beta\gamma)$, $Q = (x_{Q'}\beta, y_{Q'}\beta\gamma)$ and $P = (x_P, y_P)$, where $x_p, y_p \in \mathbb{F}_p$ given in affine coordinates on the curve $E(\mathbb{F}_{p^{16}})$ such that $T' = (x_{T'}, y_{T'})$, $Q' = (x_{Q'}, y_{Q'})$ are in the twisted curve E' defined over $\mathbb{F}_{p^4}$.

9.3.3.0.1 7-Sparse Multiplication: We start this paragraph by presenting the 7-sparse multiplication of the elliptic curve doubling of $T + T = R(x_R, y_R)$ given in [Ara+11; Gre+13].

$$l_{T,T}(P) = (y_p - y_{T'}\beta\gamma) - \lambda(x_P - x_{T'}\beta),$$

$$\lambda_{T,T} = \frac{3x_{T'}^2\beta^2 + a}{2y_{T'}\beta\gamma} = \frac{3x_{T'}^2\beta\gamma^{-1} + a(\beta\gamma)^{-1}}{2y_{T'}} = \frac{(3x_{T'}^2 + a(\alpha-1)^{-1})\gamma}{2y_{T'}} = \lambda'\gamma \quad (9.12)$$

Here $\lambda_{T,T}$ is the gradient of the line going through the rational points T, P. Let, $a(\alpha-1)^{-1} = \delta \in \mathbb{F}_{p^4}$. Since a and $(\alpha - 1)$ is already know at this stage, therefore, $a(\alpha-1)^{-1}$ can be pre-calculated. It will save calculation cost during ECD inside the Miller's loop. Now the line evaluation and ECD are obtained as follows:

$$\begin{cases} l_{T,T}(P) & = y_p - x_p\lambda'_{T,T}\gamma + (x_{T'}\lambda'_{T,T} - y_{T'})\beta\gamma, \\ x_{2T'} & = (\lambda'_{T,T})^2\gamma^2 - 2x_{T'}\beta = ((\lambda'_{T,T})^2 - 2x_{T'})\beta \\ y_{2T'} & = (x_{T'}\beta - x_{2T'}\beta)\lambda'_{T,T}\gamma - y_{T'}\beta\gamma = (x_{T'}\lambda'_{T,T} - x_{2T'}\lambda'_{T,T} - y_{T'})\beta\gamma \end{cases} \tag{9.13}$$

Calculations of Eq.(9.12) and Eq.(9.13) can be optimized as follows:

$$\begin{aligned} & A = \frac{1}{2y_{T'}}, B = 3x^2_{T'}, + \delta, C = AB, D = 2x_{T'}, \\ & x_{2T'} = C^2 - D, E = Cx_{T'} - y_{T'}, y_{2T'} = E - Cx_{2T'}, F = -Cx_P \\ & l_{T,T}(P) = y_P + F\beta + E\beta\gamma \end{aligned} \tag{9.14}$$

The elliptic curve addition phase ($T \neq Q$) and line evaluation of $l_{T,Q}(P)$ can also be optimized similarly to the above procedure. Let the elliptic curve addition of $T + Q = R(x_R, y_R)$ computed as follows.

$$\begin{cases} l_{T,Q}(P) & = (y_p - y_{T'}\beta\gamma) - \lambda_{T,Q}(x_P - x_{T'}\beta), \\ \lambda_{T,Q} & = \frac{(y_{Q'}-y_{T'})\beta\gamma}{(x_{Q'}-x_{T'})\gamma} = \frac{(y_{Q'}-y_{T'})\gamma}{x_{Q'}-x_{T'}} = \lambda'_{T,Q}\gamma, \\ x_R & = ((\lambda'_{T,Q})^2 - x_{T'} - x_{Q'})\beta \\ y_R & = (x_{T'}\lambda'_{T,Q} - x_{R'}\lambda'_{T,Q} - y_{T'})\beta\gamma. \end{cases} \tag{9.15}$$

The common calculations in Eq.(9.15) can be reduced as follows:

$$\begin{aligned} & A = \frac{1}{x_{Q'} - x_{T'}}, B = y_{Q'} - y_{T'}, C = AB, D = x_{T'} + x_{Q'}, \\ & x_{R'} = C^2 - D, E = Cx_{T'} - y_{T'}, y_{R'} = E - Cx_{R'}, F = -Cx_P \\ & l_{T,Q}(P) = y_P - Cx_P\gamma + E\beta\gamma = y_P + F\beta + E\beta\gamma. \end{aligned} \tag{9.16}$$

Comparing with Table 11.1, it can be noticed that y_P, F and E in Eq.(9.14) and Eq.(9.16) are coefficients in the basis position of α, β, and $\beta\gamma$ of an $\mathbb{F}_{p^{16}}$ vector. Therefore, among the 16 coefficients of $l_{T,T}(P)$ and $l_{T,Q}(P) \in \mathbb{F}_{p^{16}}$, only 9 coefficients $y_P \in \mathbb{F}_p$, $Cx_P \in \mathbb{F}_{p^4}$ and $E \in \mathbb{F}_{p^4}$ are non-zero. The remaining 7 zero coefficients leads to an efficient multiplication, which we call 7-sparse multiplication in KSS-16 curve. Another important thing is, vectors A, B, C, D, E, F are calculated in $\mathbb{F}_{p^4}$ extension field while performing operations in $\mathbb{F}_{p^{16}}$.

9.3.4 Pseudo 8-sparse Multiplication for KSS-16 Curve using Type-II Towering

The main idea of *pseudo 8-sparse multiplication* is finding a more sparse form of Eq.(9.14) and Eq.(9.16), which allows reducing the number of multiplication of $\mathbb{F}_{p^{16}}$ vector during Miller's algorithm evaluation. To simplify both of Eq.(9.14) and Eq.(9.16), y_P^{-1} is multiplied to both side of these two equations since y_P remains the same through the Miller's algorithms loop calculation. We get the following equations.

$$y_P^{-1} l_{T,T}(P) = 1 - C x_P y_P^{-1} \gamma + E y_P^{-1} \beta\gamma, \tag{9.17a}$$

$$y_P^{-1} l_{T,Q}(P) = 1 - C x_P y_P^{-1} \gamma + E y_P^{-1} \beta\gamma, \tag{9.17b}$$

Although the Eq.(9.17a) and Eq.(9.17b) do not get more sparse, but 1st coefficient becomes 1. Such vector is defined as *pseudo sparse form* in this thesis. This form realizes more efficient $\mathbb{F}_{p^{16}}$ vectors multiplication in Miller's loop. However, it is clear that the Eq.(9.17b) creates computation overhead than Eq.(9.16). We have to compute $y_P^{-1} l_{T,Q}(P)$ in the left side and $x_P y_P^{-1}$, $E y_P^{-1}$ on the right. The same goes between Eq.(9.17a) and Eq.(9.14). Since the computation of Eq.(9.17a) and Eq.(9.17b) are almost identical, therefore the rest of the thesis shows the optimization technique for Eq.(9.17a). To overcome these overhead computations, the following techniques can be applied.

- $x_P y_P^{-1}$ is omitted by applying further isomorphic mapping of $P \in \mathbb{G}_1$.
- y_P^{-1} can be pre-computed. Therefore, the overhead calculation of $E y_P^{-1}$ will cost only 4 $\mathbb{F}_p$ multiplication.
- $y_P^{-1} l_{T,T}(P)$ doesn't effect the pairing calculation cost since the final exponentiation cancels this multiplication by $y_P^{-1} \in \mathbb{F}_p$.

To overcome the $C x_P y_P^{-1}$ calculation cost, $x_P y_P^{-1} = 1$ is expected. To obtain $x_P y_P^{-1} = 1$, the following isomorphic mapping of $P = (x_P, y_P) \in \mathbb{G}_1$ is introduced.

9.3.4.1 Isomorphic map of $P = (x_P, y_P) \longrightarrow \bar{P} = (x_{\bar{P}}, y_{\bar{P}})$.

Although the KSS-16 curve is typically defined over $\mathbb{F}_{p^{16}}$ as $E(\mathbb{F}_{p^{16}})$, for efficient implementation of Optimal-Ate pairing, certain operations are carried out in a quartic twisted isomorphic curve E' defined over $\mathbb{F}_{p^4}$ as shown in Sec. 12.3.1. For the same, let us consider $\bar{E}(\mathbb{F}_{p^4})$ is isomorphic to $E(\mathbb{F}_{p^4})$ and certain $z \in \mathbb{F}_p$ as a quadratic residue (QR) in $\mathbb{F}_{p^4}$. A generalized mapping between $E(\mathbb{F}_{p^4})$ and $\bar{E}(\mathbb{F}_{p^4})$ can be given as follows:

$$\bar{E}(\mathbb{F}_{p^4})[r] \longmapsto E(\mathbb{F}_{p^4})[r],$$
$$(x, y) \longmapsto (z^{-1}x, z^{-3/2}y),$$

where, $\bar{E}$ is the elliptic curve defined by $y^2 = x^3 + az^{-2}x$, and $z, z^{-1}, z^{-3/2} \in \mathbb{F}_p$. The mapping considers $z \in \mathbb{F}_p$ is a quadratic residue over $\mathbb{F}_{p^4}$ which can be shown by the fact that $z^{(p^4-1)/2} = 1$ as follows:

$$\begin{aligned} z^{(p^4-1)/2} &= z^{(p-1)(p^3+p^2+p+1)/2} \\ &= 1^{(p^3+p^2+p+1)/2} \\ &= 1 \quad \text{QR} \in \mathbb{F}_{p^4}. \end{aligned} \tag{9.18}$$

Therefore, z is a quadratic residue over $\mathbb{F}_{p^4}$.
Now based on $P = (x_P, y_P)$ be the rational point on curve E, the considered isomorphic mapping of Eq.(9.18) can find a certain isomorphic rational point $\bar{P} = (x_{\bar{P}}, y_{\bar{P}})$ on the curve $\bar{E}$ as follows:

$$\begin{aligned} y_P^2 &= x_P^3 + ax_P, \\ y_P^2 z^{-3} &= x_P^3 z^{-3} + ax_P z^{-3}, \\ (y_P z^{-3/2})^2 &= (x_P z^{-1})^3 + az^{-2} x_P z^{-1}, \end{aligned} \tag{9.19}$$

where $\bar{P} = (x_{\bar{P}}, y_{\bar{P}}) = (x_P z^{-1}, y_P z^{-3/2})$ and recall that the general form of the curve $\bar{E}$ is given as follows:

$$y^2 = x^3 + az^{-2}x. \tag{9.20}$$

To obtain the target relation $x_{\bar{P}} y_{\bar{P}}^{-1} = 1$ from above isomorphic map and rational point $\bar{P}$, let us find twist parameter z as follows:

$$\begin{aligned} x_{\bar{P}} y_{\bar{P}}^{-1} &= 1 \\ z^{-1} x_P (z^{-3/2} y_P)^{-1} &= 1 \\ z^{1/2} (x_P . y_P^{-1}) &= 1 \\ \text{So, } z &= (x_P^{-1} y_P)^2. \end{aligned} \tag{9.21}$$

Now using $z = (x_P^{-1} y_P)^2$ and Eq.(9.19), $\bar{P}$ can be obtained as

$$\bar{P}(x_{\bar{P}}, y_{\bar{P}}) = (x_P z^{-1}, y_P z^{-3/2}) = (x_P^3 y_P^{-2}, x_P^3 y_P^{-2}), \tag{9.22}$$

For the same isomorphic map we can obtain $\bar{Q}$ on curve $\bar{E}$ defined over $\mathbb{F}_{p^{16}}$ as follows:

$$\bar{Q}(x_{\bar{Q}}, y_{\bar{Q}}) = (z^{-1} x_{Q'} \beta, z^{-3/2} y_{Q'} \beta \gamma), \tag{9.23}$$

where from Eq.(9.11), $Q'(x_{Q'}, y_{Q'}) \in E'$.

At this point, to use $\bar{Q}$ with $\bar{P}$ in line evaluation we need to find another isomorphic map that will map $\bar{Q} \mapsto \bar{Q}'$, where $\bar{Q}'$ is the rational point on curve $\bar{E}'$ defined over $\mathbb{F}_{p^4}$. Such $\bar{Q}'$ and $\bar{E}'$ can be obtained from $\bar{Q}$ of Eq.(9.23) and curve $\bar{E}$ from Eq.(9.20) as follows:

$$\begin{aligned}(z^{-3/2}y_{Q'}\beta\gamma)^2 &= (z^{-1}x_{Q'}\beta)^3 + az^{-2}z^{-1}x_{Q'}\beta,\\ (z^{-3/2}y_{Q'})^2\beta^2\gamma^2 &= (z^{-1}x_{Q'})^3\beta^3 + az^{-2}z^{-1}x_{Q'}\beta,\\ (z^{-3/2}y_{Q'})^2 &= (z^{-1}x_{Q'})^3 + z^{-1}x_{Q'}a(z\beta)^{-2}.\end{aligned}$$

From the above equations, $\bar{E}'$ and $\bar{Q}'$ are given as,

$$\bar{E}' : y_{\bar{Q}'}^2 = x_{\bar{Q}'}^3 + a(z\beta)^{-2}x_{\bar{Q}'}. \tag{9.24}$$

$$\bar{Q}'(x_{\bar{Q}'}, y_{\bar{Q}'}) = (z^{-1}x_{Q'}, z^{-3/2}y_{Q'}) = (x_{Q'}x_P^2y_P^{-2}, y_{Q'}x_P^3y_P^{-3}). \tag{9.25}$$

Now, by applying $\bar{P}$ and $\bar{Q}'$, the line evaluation of Eq.(9.17b) becomes:

$$\begin{aligned}y_{\bar{P}}^{-1}l_{\bar{T}',\bar{Q}'}(\bar{P}) &= 1 - C(x_{\bar{P}}y_{\bar{P}}^{-1})\gamma + Ey_{\bar{P}}^{-1}\beta\gamma,\\ \bar{l}_{\bar{T}',\bar{Q}'}(\bar{P}) &= 1 - C\gamma + E(x_P^{-3}y_P^2)\beta\gamma, \end{aligned}\tag{9.26}$$

where $x_{\bar{P}}y_{\bar{P}}^{-1} = 1$ and $y_{\bar{P}}^{-1} = z^{3/2}y_P^{-1} = (x_P^{-3}y_P^2)$. The Eq.(9.17a) becomes the same as Eq.(9.26). Compared to Eq.(9.17b), the Eq.(9.26) will be faster while using in Miller's loop in combination of the pseudo 8-sparse multiplication recalled in Alg. 13.

Algorithm 13: Pseudo 8-sparse multiplication for KSS-16 curve

```
Input: $A, B \in \mathbb{F}_{p^{16}}$
    $A = (a_0 + a_1\beta) + (a_2 + a_3\beta)\gamma$, $B = 1 + (b_2 + b_3\beta)\gamma$
    $A = a_0 + a_2\gamma + a_1\gamma^2 + a_3\gamma^3$, $B = 1 + b_2\gamma + b_3\gamma^3$
    $a_i, b_i \in \mathbb{F}_{p^4}$ where $i = 0, 1, 2, 3$
Output: $C = AB = (c_0 + c_1\beta) + (c_3 + c_4\beta)\gamma \in \mathbb{F}_{p^{16}}$
$t_0 \leftarrow a_3 \times b_3, t_1 \leftarrow a_2 \times b_2, t_4 \leftarrow b_2 + b_3$        ▷ ($18M_p$)
$c_0 \leftarrow (a_2 + a_3) \times t_4 - t_1 - t_0, c_0 \leftarrow c_0 \times (\alpha - 1)$        ▷ ($9M_p$)
$c_1 \leftarrow t_1 + t_0 \times (\alpha - 1)$
$t_2 \leftarrow a_1 \times b_3, t_3 \leftarrow a_0 \times b_2, c_2 \leftarrow t_3 + t_2 \times (\alpha - 1)$        ▷ ($18M_p$)
$c_3 \leftarrow (a_0 + a_1) \times t_4 - t_3 - t_2$        ▷ ($9M_p$)
$C \leftarrow C + A$
return $C = (c_0 + c_1\gamma) + (c_3 + c_4\gamma)\beta$        ▷ (Total $54M_p$)
```

However, to apply Eq.(9.26) in Miller's algorithm, we need the following pre-computations once in every Miller's Algorithm execution.

- Computing $\bar{P}$ and $\bar{Q}'$,
- Computing $y_{\bar{P}}^{-1} = (x_P^{-3}y_P^2)$ and

- Deducing the z^{-2} term from curve $\bar{E}'$ of Eq.(9.24).
- Calculating $az^{-2}(\alpha - 1)^{-1} = z^{-2}\delta$ used during ECD of curve $\bar{E}'$.

Among the above terms $a = 1$ and $\delta = (\alpha - 1)^{-1}$ is pre-calculated during parameter setup. Rest of the operations are calculated as follows using Alg. 17. The remaining part of the Miller's algorithm i.e. the multiplication

Algorithm 14: Pre-calculation and mapping $P \mapsto \bar{P}$ and $Q' \mapsto \bar{Q}'$

```
Input: $P = (x_P, y_P) \in \mathbb{G}_1, Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2'$
Output: $\bar{Q}', \bar{P}, y_P^{-1}, z^{-2}, z^{-2}\delta$
$A \leftarrow x_P y_P^{-1}$                                  ▷ $(1I_{p^4} + 1M_{p^4})$
$B \leftarrow A^2$                                           ▷ $(1S_{p^4})$
$x_{\bar{P}}, y_{\bar{P}} \leftarrow Bx_P$                   ▷ $(1M_{p^4})$
$x_{\bar{Q}'} \leftarrow Bx_{Q'}$                            ▷ $(1M_{p^4})$
$y_{\bar{Q}'} \leftarrow ABy_{Q'}$                           ▷ $(2M_{p^4})$
$y_P^{-1} \leftarrow y_{\bar{P}}^{-1}$                       ▷ $(1I_{p^4})$
$z^{-2} \leftarrow B^2$                                      ▷ $(1S_{p^4})$
$z^{-2} \leftarrow z^{-2}\delta$          ▷ (used during ECD in Eq.(9.24); $1M_{p^4}$)
return $\bar{Q}' = (x_{\bar{Q}'}, y_{\bar{Q}'}), \bar{P} = (x_{\bar{P}}, y_{\bar{P}}), y_P^{-1}, z^{-2}, z^{-2}\delta$
```

by prime $p[Q]$ or $[p^2]Q$ can be evaluated by applying skew Frobenius map [Sak+08].

9.3.4.2 Skew Frobenius Map to Compute $[p]\bar{Q}'$

From the definition of $Q \in \mathbb{G}_2$ we recall that Q satisfies $[\pi_p - p]Q = \mathcal{O}$ or $\pi_p(Q) = [p]Q$, which is also applicable for $\bar{Q}'$. Applying skew Frobenius map we can optimize $[p]\bar{Q}'$ calculation in Miller's algorithm as follows:

$$(x_{\bar{Q}'}\beta)^p = (x_{\bar{Q}'})^p\beta^p, \qquad (y_{\bar{Q}'}\beta\gamma)^p = (y_{\bar{Q}'})^p\beta^p\gamma^p.$$

After remapping the above terms tern as follows:

$$(x_{\bar{Q}'})^p\beta^{p-1} = (x_{\bar{Q}'})^p(\beta^2)^{\frac{p-1}{2}}, \qquad (y_{\bar{Q}'})^p\beta^{p-1}\gamma^{p-1} = (y_{\bar{Q}'})^p(\beta^2)^{\frac{p-1}{2}}(\gamma^2)^{\frac{p-1}{2}}.$$

The above $(x_{\bar{Q}'})^p$ and $(y_{\bar{Q}'})^p$ terms can be computed using Eq.(9.9) without any costs. The rest can be done similar to Sect. 9.3.1.7 with a cost of $18M_p$.

9.3.5 Final Exponentiation

Thanks to the cyclotomic polynomial and the definitions of r and k, the exponent $\frac{p^{16}-1}{r}$ broken down into two parts. We have,

$$\frac{p^{16}-1}{r} = (p^8-1)\frac{(p^8+1)}{r}.$$

The first part, (p^8-1) is the simple part of the final exponentiation because it is easy to be performed thanks to a Frobenius operation, an inversion and a multiplication (in $\mathbb{F}_{p^{16}}$. However, it has an important consequence for the computation of the second part of the final exponentiation. Indeed, powering f, the result of Miller loop, to the p^8-1 makes the result unitary [SB04]. So during the hard part of the final exponentiation, which consists on computing $f^{\frac{p^8+1}{r}}$), all the elements involved are unitary. This simplifies computations, for example, any future inversion can be implemented as a Frobenius operator, more precisely $f^{-1} = f^{p^8}$ which is just a conjugation [SB04], [SL03]. The hard part $\frac{(p^8+1)}{r}$ can be efficiently calculated using Ghammam's et al.'s works [GF16a] addition chain algorithm.
In this thesis, we reduce the number of temporary variables used in the [GF16a] to calculate $f_1^{857500\frac{(p^8+1)}{r}}$, where f_1 is the result of computing the first part of the final exponentiation. The number $d = 857500$, chosen in [GF16a] results efficient addition chain calculation that ultimately helps efficient hard part evaluation. Alg. 9.6 shows the space-optimized final exponentiation.
The squaring during hard part computation is the most operation used, it can be efficiently carried out using Granger et Scott [GS10] cyclotomic squaring. Their method consists of: Let A be a $\mathbb{G}_3$ element that is actually in a cyclotomic subfield. So $A = (a_0 + a_1\gamma) \in \mathbb{F}^*_{p^{16}}$, it verifies $A^{(p^8+1)} = 1$. Therefore, $(a_0 + a_1\gamma)(a_0 - a_1\gamma) = 1$ or $a_0^2 = 1 + a_1^2\gamma^2 = 1 + a_1^2\beta$ can be obtained, where $\bar{A} = (a_0 - a_1\gamma)$ is a conjugate of A. By using this relation we can obtain the cyclotomic squaring as follows:

$$\begin{aligned}
A^2 &= a_0^2 + a_1^2\beta + 2a_0a_1\gamma \\
&= a_0^2 + a_1^2\beta + ((a_0+a_1)^2 - a_0^2 - a_1^2)\gamma \\
&= 1 + a_1^2\beta + a_1^2 + ((a_0+a_1)^2 - 1 - a_1^2\beta - a_1^2)\gamma \\
&= (1 + 2a_1^2\beta) + ((a_0+a_1)^2 - 1 - a_1^2(1+\beta))\gamma
\end{aligned}$$

Here, only two squaring in $\mathbb{F}_{p^8}$ where in normal $\mathbb{F}_{p^{16}}$ squaring requires 2 multiplications in $\mathbb{F}_{p^8}$.

Instead of computing the cyclotomic squaring, Karabina has proposed in

Algorithm 4:	Operation	Cost
Input: f, u, p, r		
Output: $f_1^{d\frac{(p^8+1)}{r}}$		
Temp.Var:$t, t_0, t_1, \cdots, t_{14}$		
$f_1 \leftarrow f^{p^8}, f_1 \leftarrow f_1 * f^{-1}$		
$t_0 \leftarrow f_1^2, t_1 \leftarrow t_0^2$	f_1^2, f_1^4	$2S_{c16}$
$t_2 \leftarrow f_1^{(u+1)}, t_3 \leftarrow t_2^{(u+1)}$	$f_1^{(u+1)}, f_1^{(u+1)^2}$	$2E_u$
$t_4 \leftarrow t_3 * t_1$	$f_1^{(u+1)^2+4} = f_1^B$	$1M_{p^{16}}$
$t_5 \leftarrow t_4^u, t_6 \leftarrow t_4^5$	f_1^{uB}, f_1^{5B}	$1E_u + 1M_{p^{16}} + 2S_{c16}$
$t_7 \leftarrow t_1^8, t_8 \leftarrow t_7^2$	f_1^{32}, f_1^{64}	$4S_{c16}$
$t_9 \leftarrow t7 * t_1^{-1}, t_{10} \leftarrow t_9^2$	f_1^{28}, f_1^{56}	$1M_{p^{16}} + 1S_{c16}$
$t_{11} \leftarrow t_5^u, t_{12} \leftarrow t_{11}^u$	$f_1^{u^2B}, f_1^{u^3B}$	$2E_u$
$t_{13} \leftarrow t_{12} * t_9$	$f_1^{(u^3B+56)} = f_1^A$	$1M_{p^{16}}$
$t_9 \leftarrow t_{13}^u, t_2 \leftarrow t_9^{-2}$	f_1^{uA}, f_1^{-2uA}	$1E_u + 1S_{c16}$
$t_{10} \leftarrow t_6^5, t_{10} \leftarrow t_{10}^5$	f_1^{25B}, f_1^{125B}	$2M_{p^{16}} + 2S_{c16}$
$t_0 \leftarrow t_2 * t_{10}^{-1}$	$f_1^{-2uA-125B} = f_1^{c_2}$	$1M_{p^{16}}$
$t_3 \leftarrow t_0^2, t_2 \leftarrow t_2^4$	$f_1^{2c_2}; f_1^{-8uA}$	$3S_{c16}$
$t_2 \leftarrow t_2 * t_9$	f_1^{-7uA}	$1M_p^{16}$
$t2 \leftarrow t_2 * t_3$	$f_1^{2c_2-7uA} = f_1^{c_6}$	$1M_p^{16}$
$t_3 \leftarrow t_9^u, t_6 \leftarrow t_3^u$	$f_1^{u^22A}; f_1^{u^3A}$	$2E_u$
$t_7 \leftarrow t_6^u, t_{10} \leftarrow t_3^2$	$f_1^{u^4}; f_1^{2u^2A}$	$1E_u + 1S_{c16}$
$t_9 \leftarrow t_5^5, t_9 \leftarrow t_9^5$	$f_1^{5uB};f_1^{25uB}$	$2M_p^{16} + 4S_{c16}$
$t_4 \leftarrow t_9^3, t_9 \leftarrow t_4 * t_9$	$f_1^{75uB};f_1^{100uB}$	$1C_{16} + 1M_{p^{16}}$
$t_{10} \leftarrow t_{10}^2$	$f_1^{4u^2A}$	$1S_{c16}$
$t_{14} \leftarrow (t_{10} * t_4)^{-1}$	$f_1^{-4u^2A-75uB} = f_1^{c_1}$	$1M_{p^{16}}$
$t_3 \leftarrow t_{10} * t_3^{-1}$	$f_1^{3u^2A}$	$1M_{p^{16}}$
$t_3 \leftarrow t_3 * t_9$	$f_1^{3u^2A+100xB} = f_1^{c_5}$	$1M_{p^{16}}$
$t_{11} \leftarrow t_{11}^5, t_9 \leftarrow t_{11}^2$	$f_1^{5u^2B}; f_1^{10u^2B}$	$1M_{p^{16}} + 3S_{c16}$
$t_4 \leftarrow t_9 * t_6$	$f_1^{u^3A+10u^2B} = f_1^{c_4}$	$1M_{p^{16}}$
$t_6 \leftarrow t_6^2, t_9 \leftarrow t_9^5$	$f_1^{2u^3A}\ f_1^{50u^2B}$	$1M_p^{16} + 3S_{c16}$
$t_9 \leftarrow t_9 * t_{11}, t_9 \leftarrow t_9 * t_6$	$f_1^{55u^2B};f_1^{2u^3A-55u^2B} = f_1^{c_0}$	$2M_p^{16}$
$t_{12} \leftarrow t_{12}^{24}$	$f_1^{24u^3B}$	$1C_{16} + 3S_{c16}$
$t_5 \leftarrow t_7^{-1} * t_{12}^{-1}$	$f_1^{-u^4A-24u^3B}$	$1M_p^{16}$
$t_8 \leftarrow t_8^3, t_6 \leftarrow t_8 * t_1$	f_1^{196}	$1C_{16} + 1M_{p^{16}}$
$t_7 \leftarrow t_5 * t_6$	$f_1^{-u^4A-24u^3B+196} = F_1^{c_3}$	$1M_{p^{16}}$
$t_8 \leftarrow t_{13}^7$	$f_1^{7A} = f_1^{c_7}$	$2M_{p^{16}} + 2S_{c16}$
$t_1 \leftarrow t_{14}^p * t_7^{p^3} * t_3^{p^5} * t_8^{p^7}$	$f_1^{c_1p+c_3p^3+c_5p^5+c_7p^7}$	$3M_{p^{16}} + 4(15M)$
$t_2 \leftarrow t_0^{p^2} * t_2^{p^6}$	$f_1^{c_2p^2+c_6p^6}$	$1M_{p^{16}} + 2(12M)$
$t \leftarrow t_9 * t_2 * t_1 * t_4^{p^4}$	$f_1^{d\frac{(p^8+1)}{r}}$	$3M_{p^{16}} + 1(8M)$
return t		

TABLE 9.6: Final Exponentiation with reduced temporary variables of [GF16a]

[Kar13b] a new method for computing the squaring in the cyclotomic subgroup. This method is called the compressed squaring. It contains two steps, compression where we compute the squaring of the compressed form of an element in the cyclotomic subgroup of $\mathbb{F}_{p^k}$. Then, before performing another operation except the squaring, we have to use the decompression form of the element in question. In his thesis, Karabina proved that his method is applicable when the extension degree $k = 2^a 3^b$ with $a, b \in \mathbb{N}$ and $a, b > 0$ and he presented the example of computing the compressed squaring in the cyclotomic subgroup of $\mathbb{F}_{p^{12}}$. In this thesis, we are interested in generalizing Karabina's method for $k = 2^a$ and $k = 3^b$. In this context, we find the following result.

Proposition 9.3.1 *For the case of $k = 16$, it is not possible to compute the compressed squaring in the cyclotomic subgroup of $\mathbb{F}_{p^{16}}$. Then, this result is generalized when the extension degree $k = 2^a$ and $k = 3^b$.*

Our proof is based on the fact that . . . HERE THE PROOF

For this reason, in our work, we consider only the cyclotomic squaring.

The overall optimizations can be seen as the following Alg. 15.

Algorithm 15: The improved Optimal-Ate pairing algorithm for KSS-16 curve using CVMA

```
Input: $u, P \in \mathbb{G}_1 \subset E(\mathbb{F}_{p^4}), Q' \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$
Output: $e(\bar{Q}', \bar{P})$
Pre-compute $\bar{Q}', \bar{P}, y_P^{-1}, z^{-2}, z^{-2}\delta$          ▷ (see Alg. 17)
$f \leftarrow 1, \bar{T}' \leftarrow \bar{Q}'$
for $i = \lfloor \log_2(u) \rfloor$ downto 1 do
    $f \leftarrow f^2 \cdot \bar{l}_{\bar{T}',\bar{T}'}(\bar{P}), \bar{T}' \leftarrow [2]\bar{T}'$          ▷ (apply Alg. 13 )
    if $u[i] = 1$ then
        $f \leftarrow f \cdot \bar{l}_{\bar{T}',\bar{Q}'}(\bar{P}), \bar{T}' \leftarrow \bar{T}' + \bar{Q}'$ ▷ (apply Alg.13 to solve Eq.(9.26))
    if $u[i] = -1$ then
        $f \leftarrow f \cdot \bar{l}_{\bar{T}',\bar{Q}'}(\bar{P}), \bar{T}' \leftarrow \bar{T}' - \bar{Q}'$ ▷ (apply Alg.13 to solve Eq.(9.26))
$Q_1 \leftarrow [u]\bar{Q}'$          ▷ (here $Q_1 = \bar{T}'$)
$Q_2 \leftarrow [p]\bar{Q}'$          ▷ (Skew Frobenius map Sec. 12.3.6)
$f \leftarrow f \cdot l_{Q_1,Q_2}(\bar{P})$          ▷ (Alg.13)
$f_t \leftarrow f^{p^3}$          ▷ (Forbenius map of $p^3$)
$f \leftarrow f \cdot f_t$          ▷ (Alg.13)
$f \leftarrow f \cdot l_{\bar{Q}',\bar{Q}'}(\bar{P})$          ▷ (Alg.13)
$f_1 \leftarrow f^{(p^8-1)}$          ▷ $(1I_{p^{16}} + 1M_{p^{16}})$
$f \leftarrow f_1^{d\frac{p^8+1}{r}}$          ▷ (Alg.9.6)
return $f$
```

9.4 Experimental Result Evaluation

This section gives details of the experimental implementation. The source code can be found in Github[1]. The implemented code is not optimized for any specific platform, rather it is written keeping in mind of scalability with the change of parameters. The sole purpose of the piece of code is to compare the Optimal-Ate pairing operations between CVMA (this work) and Karatsuba based implementations [Kha+17b] while applying state-of-art algorithms.

9.4.1 Experiment Environment and Assumptions

Table 10.3 shows the implementation environment used to evaluate the proposal.

CPU*	Memory	Compiler	OS	Language	Library
Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz	4GB	GCC 5.4.0	Ubuntu 16.04 LTS	C	GMP v 6.1.0 [Gt15]

TABLE 9.7: Computational Environment

The authors made no attempts to utilize multiple cores of the CPU. The data type of mpz_t of GMP is used to define the big integer in $\mathbb{F}_p$. The code is compiled with -O3 flag in gcc. To compare the prime field operations of pairing, the authors assumed that 8 prime field addition A_p in the above environment is almost equivalent to 1 multiplication(M_p) in $\mathbb{F}_p$ with respect of time. The assumption is based on the average time of 1 million iterations of A_p and M_p of operand size ≈ 334-bit. The authors also found that for the above settings, the assumptions hold in other environments. TODO The authors also compare the cycles count of the operations, obtained from CPU's Time Stamp Counter. It's worth mentioning that none of the time and cycles promise constant output for certain operation in a certain environment due to several operating system factors.

The parameter is chosen according to [BD18]'s suggestion for to make DLP size secure enough against exTNFS [KB16] as is shown in Table 10.4. The chosen parameter is twist secure but doesn't guarantee subgroup security. However, finding both twist secure and subgroup secure parameters with lowest hamming weight can be a matter of time.

Curve	Integer u	HW(u)	$\lfloor \log_2 u \rfloor$	$\lfloor \log_2 p(u) \rfloor$	$\lfloor \log_2 r(u) \rfloor$	$\lfloor \log_2 p^k \rfloor$
KSS-16	$u = -2^{33} - 2^{32} - 2^{13} - 2^{11} + 2^6 + 1$	6	34	334	259	5344

TABLE 9.8: Selected parameters for 128-bit security level according to [BD18]

[1] https://github.com/alaminkhandaker/KSS16-opt-ate

9.4.2 Result and Analysis

Table 9.9 shows the total number of operations in $\mathbb{F}_p$ for notable finite field operation applied in pairing calculation. The negative value refers to the decrements of operations after applying CVMA technique. As aforementioned, CVMA reduces the number of A_p for multiplications and squaring over the extension field. Although the Frobenius map in $\mathbb{F}_{p^4}$ is free of cost; however, the Frobenius map in $\mathbb{F}_{p^{16}}$ in CVMA costs more than Karatsuba based constructions. The inversion in $\mathbb{F}_{p^4}$ is costlier in CVMA. But in terms of total operation, the CVMA approach shows better performance than Karatsuba approach.

	CVMA			Karatsuba			Increment of A_p	approx %
	M_p	A_p	I_p	M_p	A_p	I_p	$[8A_p \simeq 1M_p$ in $\mathbb{F}_p]$	[-ve is decrement]
$\mathbb{F}_{p^4}$ inversion	16	26	1	14	29	1	13	9.2
$\mathbb{F}_{p^4}$ multiplication	9	22		9	29		-7	-6.9
$\mathbb{F}_{p^4}$ squaring	6	14		6	24		-10	-13.9
$\mathbb{F}_{p^8}$ inversion	46	109	1	44	140	1	-15	-3
$\mathbb{F}_{p^8}$ multiplication	27	93		27	108		-15	-4.6
$\mathbb{F}_{p^8}$ squaring	18	78		18	80		-2	-0.9
$\mathbb{F}_{p^{16}}$ inversion	136	466	1	134	525	1	-43	-2.7
$\mathbb{F}_{p^{16}}$ multiplication	81	326		81	365		-39	-3.8
$\mathbb{F}_{p^{16}}$ squaring	54	240		54	258		-18	-2.6
$\mathbb{F}_{p^{16}}$ Frobenius	27	66		14			170	151.7
$\mathbb{F}_{p^{16}}$ skew Frob.	18	44		8			124	193.8

TABLE 9.9: Operation count in $\mathbb{F}_p$ for extension field operations used in pairing

Then, in Table 9.10 we compare Miller algorithm with CVMA with Miller algorithm with Karatsuba with respect to operation count.

	CVMA			Karatsuba			Increment	approx %
Operations	M_p	A_p	I_p	M_p	A_p	I_p	of A_p	
MA	6679	23663	41	6578	27194	41	-2723	-3.4
MA pre-com	98	212	2	94	280	2	-36	-3.5

TABLE 9.10: Miller's algorithm (MA) operation comparison with respect to $\mathbb{F}_p$ addition

In the following Table 9.4.2 we compare the final exponentiation with CVMA with Miller algorithm with Karatsuba with respect to operation count.

	CVMA		Karatsuba		Increment of A_p	approx %
	M_p	A_p	M_p	A_p		
Pseudo 8-sparse multiplication	54	205	54	229	-24	-3.6

TABLE 9.11: Comparison in terms of operation count for Final exponentiation (FE)

The Miller's algorithms proposed pre-computation cost is negligible compared to the rest of the computation. The Karatsuba based implementation takes 101 less $\mathbb{F}_p$ multiplication than CVMA in Miller's algorithm. However, such advantage is overtaken by the number of reduced addition in CVMA compared to Karatsuba. The 3.4% improvement is seemingly very insignificant in terms of 1 pairing. However, a real pairing-based protocol requiring multiple pairings can be benefited from it. **WHY?? Explain**

Table 9.13 shows execution time in millisecond (rounded 2 decimal places) and cycle counts for Optimal-Ate pairing implementation for the Table 10.3 settings. The main purpose of this execution time comparison is to show that the theoretic optimization also reflects in the real implementation. However, the implementation doesn't guarantee constant time operation which is crucial in the context of the side-channel attack. The negative value refers to CVMA's efficiency over Karatsuba based implementation. The cycle counts are almost coherent with the time performances. The execution time also binds with the respective operation counts of Table 9.10, 9.12. The total pairing time is significantly influenced by the hard part of final exponentiation. It may seem confusing that 0.7% reduction of operation count for the FE hard part in CVMA, results in relatively more faster execution time. However, the authors relate this irregularity to cyclotomic squaring operation. Since towering is involved, therefore, the extension field operations are implemented in top-down order. Therefore, in CVMA, the $\mathbb{F}_{p^8}$ squaring for cyclotomic squaring operation, calls $\mathbb{F}_{p^4}$ squaring; which is more efficient than the Karatsuba counterpart (Table 9.9). The further time-profile investigation finds that the number of times GMP library calls its memory allocation/reallocation impacts in the execution time.

	CVMA			Karatsuba			Increment	approx %
Operations	M_p	A_p	A_{ui}	M_p	A_p	A_{ui}	of A_p	
Final exp. [hard]	19134	93933	2744	19102	96129	686	-1796	-0.7
Final exp. [easy]	217	792		215	890		-82	-3.1

TABLE 9.12: Comparison in terms of operation count for Final exponentiation (FE)

	CVMA		Karatsuba		Increment in % [-ve refers decrement]	
	≈ Time [ms]	Cycles	≈ Time [ms]	Cycles	Time	Cycles
Pairing pre-computation	0.05	159161	0.05	156660	0	1.6
Miller's algo.	2.23	7125491	3.45	11010338	-35.4	-35.3
FE [easy]	0.12	378786	0.13	413408	-7.7	-8.4
FE [hard]	7.13	22765766	10.18	32507719	-30.0	-30.0
Total	9.53	30429204	13.81	44088125	-31.0	-31.0

TABLE 9.13: Time comparison in millisecond [ms] of CVMA vs Karatsuba based implementation of Pseudo 8-sparse Optimal-Ate

9.5 Conclusion and Future Work

This thesis shows several improvement ideas for Optimal-Ate pairing in the less studied KSS-16 curve while revisiting [Kha+17b] to find more efficient Miller's algorithm implementation technique for Optimal-Ate pairing

- applied combination of normal basis and polynomial basis for $\mathbb{F}_{p^{16}}$ extension field operation.
- The selling point for of CVMA in this work is $\mathbb{F}_{p^4}$ extension field operation. It requires fewer $\mathbb{F}_p$ additions than its Karatsuba counterparts. However, Inversion and Frobenius map for the $\mathbb{F}_{p^{16}}$ is still expensive for the applied towering.
- The authors optimized inversion operation cost for CVMA approach.
- Optimized the pseudo 8-sparse multiplication for CVMA, which becomes 3.6% efficient than the similar method presented in IndoCrypt'17 [Kha+17b].
- The final exponentiation by Ghammam et al [GF16a] is more memory-optimized now.

The main drawback of this CVMA setting is the inversion in $\mathbb{F}_{p^4}$ and Frobenius map in $\mathbb{F}_{p^{16}}$. As a future improvement, the authors would like to find settings which can overcome these obstacles. The implementation and execution time given here is a comparative purpose. It can be more optimized by careful low-level prime field implementation.

Chapter 10

CSS 2017

This thesis shows an efficient Miller's algorithm implementation technique by applying pseudo 8-sparse multiplication over Barreto-Lynn-Scott (BLS12) curve of embedding degree 12. The recent development of exTNFS algorithm for solving discrete logarithm problem urges researchers to update parameter for pairing-based cryptography. Therefore, this papers applies the most recent parameters and also shows a comparative implementation of optimal-Ate pairing between BLS12 curve and Kachisa-Schaefer-Scott (KSS16) curve. The result finds that pairing in BLS12 curve is faster than KSS16 although the BLS12's Miller loop parameter is twice larger than the KSS16.

10.1 Introduction

At the beginning of this century, Sakai et al. [Sak00] and Joux [Jou04] independently proposed a cryptosystem that has unlocked many novel ideas to cryptography researchers. Many researchers tried to find out security protocol that exploits pairings to remove the need of certification by a trusted authority. In this consequence, several ingenious pairing based encryption scheme such as ID-based encryption scheme by Boneh and Franklin [Sak00] and group signature authentication by Nakanishi et al. [NF05] has come into the focus. In such outcome, Ate-based pairings such as Ate [Coh+05], Optimal-ate [Ver10], twisted Ate [Mat+07] and χ-Ate [Nog+08] pairings and their applications in cryptosystems have caught much attention since they have achieved quite efficient pairing calculation. But it has always been a challenge for researchers to make pairing calculation more efficient for being used practically as pairing calculation is regarded as quite time consuming operation.

Generally, a pairing is a bilinear map e typically defined as $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive cyclic sub-groups of order r on a certain elliptic curve E over a finite extension field $\mathbb{F}_{p^k}$ and $\mathbb{G}_3$ is a multiplicative cyclic group of order r over $\mathbb{F}^*_{p^k}$. This thesis chooses an asymmetric variants of pairing named as Optimal-Ate [Ver10] with Barreto-Lynn-Scott (BLS) [BLS03] pairing friendly curve of embedding degree $k = 12$ named as BLS-12.

Acceleration of Optimal-Ate pairing depends not only on the optimization of Miller algorithm's loop parameter but also on efficient elliptic curve arithmetic operation and efficient final exponentiation. This thesis has proposed a *pseudo 8-sparse multiplication* to accelerate Miller's loop calculation in BLS-12 curve by utilizing the property of rational point groups. In addition, this papers has showed an enhancement of the elliptic curve addition and doubling calculation in Miller's algorithm by applying implicit mapping of its sextic twisted isomorphic group.

The recent development of NFS by Kim and Barbulescu [KB16] requires to update the parameter selection for all the existing pairings over the well know pairing friendly curve families such as BN [BN06], BLS [BLS03] and KSS [KSS07]. Barbulescu and Sylvain [BD18] has proposed new parameters that for 128-bit security level and found BLS-12 is most efficient choice for Optimal-Ate pairing than well studied BN curve. Therefore the authors focuses on efficient implementation of BLS-12 curve for Optimal-Ate pairing by applying most recent parameters. The authors also applied final exponentiation algorithm of [GF16b] and compared the simulation result with BN with similar implementation technique.

The simulation result shows that the given *pseudo 8-sparse multiplication* gives more efficient Miller's loop calculation of an Optimal-Ate pairing operation along with recommended parameters than pairing calculation without sparse multiplication.

Related works.

Aranha et al. [Ara+11, Section 4] and Costello et al. [CLN10] have well optimized the Miller's algorithm in Jacobian coordinates by 6-sparse multiplication [1] for BN curve. Mori et al. [Mor+14] and Khandaker et al. [Kha+17a] have shown specific type of sparse multiplication for BN curve and KSS-18 curve respectively where both of the curves supports sextic twist. It is found that pseudo 8-sparse was clearly efficient than 7-sparse and 6-sparse in Jacobian coordinates. The authors have extended the previous works for sextic twisted BLS-12 curve.

10.2 Fundamentals

10.2.1 BLS-12 curve

Barreto, Lynn and Scott propose polynomial parameterizations by a integer variable u for certain complete pairing-friendly curve families for specific embedding degrees [BLS03]. The target curve of this thesis is such pairing-friendly curve, usually called BLS-12 of embedding degree $k - 12$, defined

[1] 6-Sparse refers the state when in a vector (multiplier/multiplicand), among the 12 coefficients 6 of them are zero.

over extension field $\mathbb{F}_{p^{16}}$ as follows:

$$E/\mathbb{F}_{p^{12}} : y^2 = x^3 + b, \quad (b \in \mathbb{F}_p) \text{ and } b \neq 0, \tag{10.1}$$

where $x, y \in \mathbb{F}_{p^{12}}$. Similar to other pairing-friendly curves, *characteristic p*, *Frobenius trace t* and *order r* of this curve are given by the following polynomials of integer variable u also known as *mother parameter*.

$$\begin{aligned} p(u) &= (u-1)^2(u^4-u^2+1)/3+u, && (10.2a)\\ r(u) &= (u^4-u^2+1) && (10.2b)\\ t(u) &= u+1, && (10.2c) \end{aligned}$$

where u is such that $6|(p-1)$ and the ρ value is $\rho = (\log_2 p/\log_2 r) \approx 1.25$. The total number of rational points $\#E(\mathbb{F}_p)$ is given by Hasse's theorem as, $\#E(\mathbb{F}_p) = p + 1 - t$. When the definition field is the k-th degree extension field $\mathbb{F}_{p^k}$, rational points on the curve E also forms an additive Abelian group denoted as $E(\mathbb{F}_{p^k})$.

10.2.2 Extension Field Arithmetic and Towering

In extension field arithmetic, higher level computations can be improved by towering. In towering, higher degree extension field is constructed as a polynomial of lower degree extension fields. In some previous works, such as Bailey et al. [BP01] explained tower of extension by using irreducible binomials. In what follows, Let $6|(p-1)$, where p is the characteristics of BLS-12 curve and -1 is a quadratic and cubic non residue in $\mathbb{F}_p$. Since BLS-12 curve is defined over $\mathbb{F}_{p^{12}}$, this thesis has represented extension field $\mathbb{F}_{p^{12}}$ as a tower of sub-fields to improve arithmetic operations.

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2+1), \\ \mathbb{F}_{p^6} = \mathbb{F}_{p^2}[\beta]/(\beta^3-(\alpha+1)), \\ \mathbb{F}_{p^{12}} = \mathbb{F}_{p^4}[\gamma]/(\gamma^2-\beta). \end{cases} \tag{10.3}$$

Extension Field Arithmetic of $\mathbb{F}_{p^{12}}$

Among the arithmetic operations multiplication, squaring and inversion are regarded as expensive operation than addition/subtraction. The calculation cost, based on number of prime field multiplication M_p and squaring S_p is given in Table 10.1. The algorithms for extension field operation are implemented from [Duq+15]. The arithmetic operations in $\mathbb{F}_p$ are denoted as M_p for a multiplication, S_p for a squaring, I_p for an inversion and m with suffix denotes multiplication with basis element.

10.2.3 Optimal-Ate pairing on BLS-12 Curve

In the context of pairing on the targeted pairing-friendly curves, two additive rational point groups $\mathbb{G}_1, \mathbb{G}_2$ and a multiplicative group $\mathbb{G}_3$ of order r are

TABLE 10.1: Number of arithmetic operations in $\mathbb{F}_{p^{12}}$ based on Eq.(10.3)

$M_{p^2} = 3M_p + 5A_p + 1m_\alpha \longrightarrow 3M_p$	$S_{p^2} = 2S_p + 3A_p \longrightarrow 2S_p$
$M_{p^6} = 6M_{p^2} + 15A_{p^2} + 2m_\beta \longrightarrow 18M_p$	$S_{p^6} = 2M_{p^2} + 3S_{p^2} + 9A_{p^2} + 2m_\beta \longrightarrow 12S_p$
$M_{p^{12}} = 3M_{p^6} + 5A_{p^6} + 1m_\gamma \longrightarrow 54M_p$	$S_{p^{12}} = 2M_{p^6} + 5A_{p^6} + 2m_\gamma \longrightarrow 36S_p$

considered. $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\begin{aligned} \mathbb{G}_1 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [1]), \\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ \mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r, \\ e &: \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3, \end{aligned} \tag{10.4}$$

here e denotes Optimal-Ate pairing [Ver10]. $E(\mathbb{F}_{p^k})[r]$ denotes rational points of order r and $[i]$ denotes i times scalar multiplication for a rational point. π_p denotes the Frobenius map given as $\pi_p : (x, y) \mapsto (x^p, y^p)$.

In the case of BLS-12, the above $\mathbb{G}_1$ is just $E(\mathbb{F}_p)$. In what follows, rest of this thesis considers $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{12}})$ for BLS-12 curve. Optimal-Ate pairing $e(Q, P)$ is given as follows:

$$e(Q, P) = f_{u,Q}(P)^{\frac{p^{12}-1}{r}}, \tag{10.5}$$

where $f_{u,Q}(P)$ is the Miller's algorithm's result and $\lfloor \log_2(u) \rfloor$ is the loop length. The bilinearity of Ate pairing is satisfied after calculating the final exponentiation $\frac{p^{12}-1}{r}$.

The generalized calculation procedure of Opt-Ate pairing is shown in Alg. 16. In what follows, the calculation steps from 1 to 7, shown in Alg. 16, is identified as Miller's Algorithm and step 8 is the final exponentiation. Steps 3, 5 and 7 are the line evaluation together with elliptic curve doubling (ECD) and addition (ECA) inside the Miller's loop. These line evaluation steps are the focus point of this thesis for acceleration. The authors extended the work of [Mor+14],[Kha+17a] for BLS-12 curve to calculate *pseudo 8-sparse multiplication* described in Sect. 3. The ECA and ECD are also calculated efficiently in the twisted curve. Step 8, FE is calculated by applying Ghammam et al.'s final exponentiation algorithm [GF16b].

Algorithm 16: Optimal-Ate pairing on BLS-12 curve

```
Input: $u, P \in \mathbb{G}_1, Q' \in \mathbb{G}'_2$
Output: $(Q, P)$
$f \leftarrow 1, T \leftarrow Q'$
for $i = \lfloor \log_2(u) \rfloor$ downto 1 do
    $f \leftarrow f^2 \cdot l_{T,T}(P), T \leftarrow [2]T$
    if $u[i] = 1$ then
        $f \leftarrow f \cdot l_{T,Q'}(P), T \leftarrow T + Q'$
    if $u[i] = -1$ then
        $f \leftarrow f \cdot l_{T,-Q'}(P), T \leftarrow T - Q'$
$f \leftarrow f^{\frac{p^{12}-1}{r}}$
return $f$
```

10.2.4 Sextic Twist of BLS-12 Curve

In the context of Optimal-Ate, there exists a *twisted curve* with a group of rational points of order r, isomorphic to the group where rational point $Q \in E(\mathbb{F}_{p^k})[r] \cap \text{Ker}(\pi_p - [p])$ belongs to. This subfield isomorphic rational point group includes a twisted isomorphic point of Q, typically denoted as $Q' \in E'(\mathbb{F}_{p^{k/d}})$, where k is the embedding degree and d is the twist degree.

Since points on the twisted curve are defined over a smaller field than $\mathbb{F}_{p^k}$, therefore ECA and ECD becomes faster. However, when required in the Miller's algorithm's line evaluation, the points can be quickly mapped to points on $E(\mathbb{F}_{p^k})$. Since the pairing-friendly BLS-12 [BLS03] curve has CM discriminant of $D = 3$ and $6|k$, therefore sextic twist is available. Let $(\alpha + 1)$ be a certain quadratic and cubic non residue in $\mathbb{F}_{p^2}$. The sextic twisted curve E'_b of curve E_b and their isomorphic mapping ψ_6 are given as follows:

$$\begin{aligned} E'_b &: \; y^2 = x^3 + b(\alpha + 1), \quad b \in \mathbb{F}_p, \\ \psi_6 &: \; E'_b(\mathbb{F}_{p^2})[r] \longmapsto E_b(\mathbb{F}_{p^{12}})[r] \cap \text{Ker}(\pi_p - [p]), \\ & \quad (x, y) \longmapsto ((\alpha + 1)^{-1} x \beta^2, (\alpha + 1)^{-1} y \beta \gamma). \end{aligned} \tag{10.6}$$

where Ker(·) denotes the kernel of the mapping and π_p denotes Frobenius mapping for rational point.

Table 10.2 shows a the vector representation of $Q = (x_Q, y_Q) = (\alpha + 1)^{-1} x_{Q'} \beta^2, (\alpha + 1)^{-1} y_{Q'} \beta \gamma \in \mathbb{F}_{p^{12}}$ according to the given towering in Eq.(10.3). Here, $x_{Q'}$ and $y_{Q'}$ are the coordinates of rational point Q' on sextic twisted curve E' defined over $\mathbb{F}_{p^2}$.

10.3 Proposal Overview

Before going to the details, the overall procedure can be described as follows:

TABLE 10.2: $\mathbb{G}_2$ rational point $Q = (x_Q, y_Q) \in \mathbb{F}_{p^{12}}$ vector representation

	1	α	β	$\alpha\beta$	β^2	$\alpha\beta^2$	γ	$\alpha\gamma$	$\beta\gamma$	$\alpha\beta\gamma$	$\beta^2\gamma$	$\alpha\beta^2\gamma$
x_Q	0	0	0	0	b_4	b_5	0	0	0	0	0	0
y_Q	0	0	0	0	0	0	0	0	b_8	b_9	0	0

1. First we define the line equation for rational point $P \in E(\mathbb{F}_p)$ and Q', T' of sextic twisted curve $E'(\mathbb{F}_{p^2})$.
2. Next we obtain more sparse form by multiplying y_P^{-1} with line equations obtained at step 1.
3. To reduce the computational overhead introduced in step 2, we obtain an isomorphic map of $P \mapsto \bar{P}$ and same map for $Q \mapsto \bar{Q}$ defined over curve $\bar{E}$.
4. $\bar{Q} \in \bar{E}(\mathbb{F}_{p^{12}})$ is isomorphic to E, however it's sextic twisted $\bar{Q}$ defined over the curve $\bar{E}(\mathbb{F}_{p^2})$ is not isomorphic. Therefore, we again obtain the twisted map of $\bar{Q} \in \bar{E}(\mathbb{F}_{p^{12}})$ to $\bar{Q}'$, defined over $\bar{E}'(\mathbb{F}_{p^2})$.
5. The mapping of step 2 and 3 reduces the overhead computation and help us to achieve pseudo 8-sparse multiplication.

Obtaining line equations

Let us consider $T = (\gamma x_{T'}, \gamma\omega y_{T'})$, $Q = (\gamma x_{Q'}, \gamma\omega y_{Q'})$ and $P = (x_P, y_P)$, where $x_p, y_p \in \mathbb{F}_p$ be given in affine coordinates on the curve $E(\mathbb{F}_{p^{12}})$ such that $T' = (x_{T'}, y_{T'})$, $Q' = (x_{Q'}, y_{Q'})$ are in the twisted curve E' defined over $\mathbb{F}_{p^2}$. Let the elliptic curve doubling of $T + T = R(x_R, y_R)$. The 7-sparse multiplication for BLS-12 can be derived as follows.

$$\begin{aligned} l_{T,T}(P) &= (y_p - y_{T'}(\alpha+1)^{-1}\beta\gamma) - \lambda_{T,T}(x_P - x_{T'}(\alpha+1)^{-1}\beta^2), \quad \text{when } T = Q, \\ \lambda_{T,T} &= \frac{3x_{T'}^2\beta\gamma}{2y_{T'}\beta^2} = \lambda'_{T,T}\frac{\gamma}{\beta} = \lambda'_{T,T}(\alpha+1)^{-1}\beta^2\gamma \end{aligned} \tag{10.7}$$

The line evaluation and ECD are obtained as follows:

$$\begin{aligned} l_{T,T}(P) &= y_p + (\lambda'_{T,T}x_{T'} - y_{T'})(\alpha+1)^{-1}\beta\gamma - \lambda'_{T,T}x_P(\alpha+1)^{-1}\beta^2\gamma, \\ x_{2T'} &= ((\lambda'_{T,T})^2 - 2x_{T'})(\alpha+1)^{-1}\beta^2 \\ y_{2T'} &= ((x_{T'} - x_{2T'})\lambda'_{T,T} - y_{T'})(\alpha+1)^{-1}\beta\gamma. \end{aligned}$$

The above calculations can be optimized as follows:

Elliptic curve doubling when $T' = Q'$

$$
\begin{aligned}
A = \tfrac{1}{2y_{T'}}, B = 3x_{T'}^2, C = AB, D = 2x_{T'}, x_{2T'} = C^2 - D,\\
E = Cx_{T'} - y_{T'}, y_{2T'} = E - Cx_{2T'},\\
l_{T',T'}(P) = y_P + (\alpha+1)^{-1}E\beta\gamma - (\alpha+1)^{-1}Cx_P\beta^2\gamma, \qquad (10.8a)\\
y_P^{-1}l_{T',T'}(P) = 1 + (\alpha+1)^{-1}Ey_P^{-1}\beta\gamma - (\alpha+1)^{-1}Cx_Py_P^{-1}\beta^2\gamma, \qquad (10.8b)
\end{aligned}
$$

The elliptic curve addition phase ($T \neq Q$) and line evaluation of $l_{T,Q}(P)$ can also be optimized similar to the above procedure. Let the elliptic curve addition of $T + Q = R(x_R, y_R)$.

$$
\begin{aligned}
l_{T,Q}(P) = (y_p - y_{T'})(\alpha+1)^{-1}\beta\gamma - \lambda_{T,Q}(x_P - x_{T'})(\alpha+1)^{-1}\beta^2, \quad T \neq Q,\\
\lambda_{T,Q} = \tfrac{(y_{Q'}-y_{T'})(\alpha+1)^{-1}\beta\gamma}{(x_{Q'}-x_{T'})(\alpha+1)^{-1}\beta^2} = \lambda'_{T,Q}(\alpha+1)^{-1}\beta^2\gamma,\\
x_R = ((\lambda'_{T,Q})^2 - x_{T'} - x_{Q'})(\alpha+1)^{-1}\beta^2\\
y_R = (x_{T'}\lambda'_{T,Q} - x_{R'}\lambda'_{T,Q} - y_{T'})(\alpha+1)^{-1}\beta\gamma.
\end{aligned}
$$

Representing the above line equations using variables as following :

Elliptic curve addition when $T' \neq Q'$ **and** $T' + Q' = R'(x_{R'}, y_{R'})$

$$
\begin{aligned}
A = \tfrac{1}{x_{Q'}-x_{T'}}, B = y_{Q'} - y_{T'}, C = AB, D = x_{T'} + x_{Q'},\\
x_{R'} = C^2 - D, E = Cx_{T'} - y_{T'}, y_{R'} = E - Cx_{R'},\\
l_{T',Q'}(P) = y_P + (\alpha+1)^{-1}E\beta\gamma - (\alpha+1)^{-1}Cx_P\beta^2\gamma, \qquad (10.9a)\\
y_P^{-1}l_{T',Q'}(P) = 1 + (\alpha+1)^{-1}Ey_P^{-1}\beta\gamma - (\alpha+1)^{-1}Cx_Py_P^{-1}\beta^2\gamma, \qquad (10.9b)
\end{aligned}
$$

Here all the variables (A, B, C, D, E) are calculated as $\mathbb{F}_{p^2}$ elements. The position of the y_P, E and C in $\mathbb{F}_{p^{12}}$ vector representation is defined by the basis element 1, $\beta\gamma$ and $\beta^2\gamma$ as shown in Table 10.2. Therefore, among the 12 coefficients of $l_{T,T}(P)$ and $l_{T,Q}(P) \in \mathbb{F}_{p^{12}}$, only 5 coefficients $y_P \in \mathbb{F}_p$, $Cx_Py_P^{-1} \in \mathbb{F}_{p^2}$ and $Ey_P^{-1} \in \mathbb{F}_{p^2}$ are non-zero other 7 coefficients are zero. These zero coefficients leads to an efficient multiplication in Miller's loop usually called sparse multiplication.

10.3.1 Pseudo 8-sparse Multiplication

The line evaluations of Eq.(10.9b) and Eq.(10.8b) are identical and more sparse than Eq.(10.9a) and Eq.(10.8a). Such sparse form comes with a cost of computation overhead i.e., computing $y_P^{-1}l_{T,Q}(P)$ in the left side and $x_Py_P^{-1}$, Ey_P^{-1} on the right. But such overhead can be minimized by the following isomorphic mapping, which also accelerates the Miller's loop iteration.

Isomorphic mapping of $P \in \mathbb{G}_1 \mapsto \bar{P} \in \mathbb{G}_1'$:

$$\begin{aligned}\bar{E} \ : \ & y^2 = x^3 + b\bar{z}, \\ & \bar{E}(\mathbb{F}_p)[r] \longmapsto E(\mathbb{F}_p)[r], \\ & (x, y) \longmapsto (\bar{z}^{-1}x, \bar{z}^{-3/2}y), \qquad (10.10)\end{aligned}$$

where $\bar{z} \in \mathbb{F}_p$ is a quadratic and cubic residue in $\mathbb{F}_p$. The Eq.(10.10) maps rational point P to $\bar{P}(x_{\bar{P}}, y_{\bar{P}})$ such that $(x_{\bar{P}}, y_{\bar{P}}^{-1}) = 1$. The twist parameter $\bar{z}$ is obtained as:

$$\bar{z} = (x_P y_P^{-1})^6 \qquad (10.11)$$

From the Eq.(10.11) $\bar{P}$ and $\bar{Q}'$ is given as

$$\begin{aligned}\bar{P}(x_{\bar{P}}, y_{\bar{P}}) &= (x_P z^{-1}, y_P z^{-3/2}) = (x_P^3 y_P^{-2}, x_P^3 y_P^{-2}) \qquad (10.12a) \\ \bar{Q}'(x_{\bar{Q}'}, y_{\bar{Q}'}) &= (x_P^2 y_P^{-2} x_{Q'}, x_P^3 y_P^{-3} y_{Q'}) \qquad (10.12b)\end{aligned}$$

Using Eq.(10.12a) and Eq.(10.12b) the line evaluation of Eq.(10.8b) becomes

$$\begin{aligned}y_{\bar{P}}^{-1} l_{\bar{T}',\bar{T}'}(\bar{P}) &= 1 + (\alpha+1)^{-1} E y_{\bar{P}}^{-1} \beta\gamma - (\alpha+1)^{-1} C x_{\bar{P}} y_{\bar{P}}^{-1} \beta^2\gamma, \\ \bar{l}_{\bar{T}',\bar{T}'}(\bar{P}) &= 1 + (\alpha+1)^{-1} E(x_P^{-3} y_P^2)\beta\gamma - (\alpha+1)^{-1} C\beta^2\gamma. \qquad (10.13a)\end{aligned}$$

The Eq.(10.9b) becomes similar to Eq.(10.13a). However, the to get the above form we need the following pre-computations once in every Miller's Algorithm execution.

- Computing $\bar{P}$ and $\bar{Q}'$,
- $(x_P^{-3} y_P^2)$

The $(\alpha + 1)^{-1}$ can precomputed once since it is just inversion of the basis element. The above terms can be computed from x_P^{-1} and y_P^{-1} by utilizing Montgomery trick [Mon87], as shown in **Alg.** 17. The pre-computation requires 21 multiplication, 1 squaring and 1 inversion in $\mathbb{F}_p$ and 2 multiplication, 3 squaring in $\mathbb{F}_{p^4}$.

Algorithm 17: Pre-calculation and mapping $P \mapsto \bar{P}$ and $Q' \mapsto \bar{Q}'$

```
Input: $P = (x_P, y_P) \in \mathbb{G}_1, Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2'$
Output: $\bar{Q}', \bar{P}, y_P^{-1}$
$A \leftarrow (x_P y_P^{-1})$
$B \leftarrow A x_P^2$
$C \leftarrow A y_P$
$D \leftarrow D x_{Q'}$
$x_{\bar{Q}'} \leftarrow D x_{Q'}$
$y_{\bar{Q}'} \leftarrow B D y_{Q'}$
$x_{\bar{P}}, y_{\bar{P}} \leftarrow D x_P$
$y_P^{-1} \leftarrow C^3 y_P^2$
return $\bar{Q}' = (x_{\bar{Q}'}, y_{\bar{Q}'}), \bar{P} = (x_{\bar{P}}, y_{\bar{P}}), y_P^{-1}$
```

Finally, pseudo 8-sparse multiplication for BLS-12 is given in

Algorithm 18: Pseudo 8-sparse multiplication for BLS-12 curves

```
Input: $a, b \in \mathbb{F}_{p^{12}}$
    $a = (a_0 + a_1\beta + a_2\beta^2) + (a_3 + a_4\beta + a_5\beta^2)\gamma,\ b = 1 + b_4\beta\gamma + b_5\beta^2\gamma$
    where $a_i, b_j, c_i \in \mathbb{F}_{p^2} (i = 0, \cdots, 5, j = 4, 5)$
Output: $c = ab = (c_0 + c_1\beta + c_2\beta^2) + (c_3 + c_4\beta + c_5\beta^2)\gamma \in \mathbb{F}_{p^{12}}$
$c_4 \leftarrow a_0 \times b_4,\ t_1 \leftarrow a_1 \times b_5,\ t_2 \leftarrow a_0 + a_1,\ S_0 \leftarrow b_4 + b_5$
$c_5 \leftarrow t_2 \times S_0 - (c_4 + t_1),\ t_2 \leftarrow a_2 \times b_5,\ t_2 \leftarrow t_2 \times (\alpha + 1)$
$c_4 \leftarrow c_4 + t_2,\ t_0 \leftarrow a_2 \times b_4,\ t_0 \leftarrow t_0 + t_1$
$c_3 \leftarrow t_0 \times (\alpha + 1),\ t_0 \leftarrow a_3 \times b_4,\ t_1 \leftarrow a_4 \times b_5,\ t_2 \leftarrow a_3 + a_4$
$t_2 \leftarrow t_2 \times S_0 - (t_0 + t_1)$
$c_0 \leftarrow t_2 \times (\alpha + 1),\ t_2 \leftarrow a_5 \times b_4,\ t_2 \leftarrow t_1 + t_2$
$c_1 \leftarrow t_2 \times (\alpha + 1),\ t_1 \leftarrow a_5 \times b_5,\ t_1 \leftarrow t_1 \times (\alpha + 1)$
$c_2 \leftarrow t_0 + t_1$
$c \leftarrow c + a$
return $c = (c_0 + c_1\beta + c_2\beta^2) + (c_3 + c_4\beta + c_5\beta^2)\gamma$
```

10.3.2 Final Exponentiation

Scott et al. [Sco+09] shows efficient final exponentiation $f^{p^k - 1/r}$ by decomposing it using cyclotomic polynomial Φ_k as

$$(p^k - 1)/r = (p^{k/2} - 1) \cdot (p^{k/2} + 1)/\Phi_k(p) \cdot \Phi_k(p)/r \tag{10.14}$$

Here, the 1st 2 terms of the right part is denoted as easy part, since it can be easily calculated by Frobenius mapping and 1 inversion in affine coordinates. The last term is called hard part which mostly effects the computation performance. According to Eq.(10.14), the exponent decomposition of the BLS-12 curve is shown in Eq.(10.15).

$$(p^{12} - 1)/r = (p^6 - 1) \cdot (p^2 + 1) \cdot (p^4 - p^2 + 1)/r \tag{10.15}$$

To efficiently carry out FE for the target curves we applied p-adic representation as shown in [GF16b]. For scalar multiplication by prime p, i.e., $p[Q]$ or $[p^2]Q$, skew Frobenius map technique by Sakemi et al. [Sak+08] has been adapted.

10.4 Experimental result evaluation

This gives details of the experimental implementation. Table 10.3 shows implementation environment. Parameters chosen from [BD18] is shown in Table 10.4. Table 10.5 shows execution time in millisecond for a single Opt-Ate pairing. Results here are the average of 10 pairing. Table 10.6 shows complexity of Miller's algorithm and final exponentiation. From the results we find that Miller's algorithm took least time for BN curve and Most for BLS-12. However, the time differences for the Miller's algo. among the

TABLE 10.3: Computational Environment

CPU*	Memory	Compiler	OS	Language	Library
Intel(R) Core(TM) i5-6500 CPU @ 3.20GHz	4GB	GCC 5.4.0	Ubuntu 16.04 LTS	C	GMP v 6.1.0 [Gt15]

*Only single core is used from two cores.

TABLE 10.4: Selected parameters for 128-bit security level [BD18]

Curve	u	HW(u)	$\lfloor \log_2 u \rfloor$	$\lfloor \log_2 p(u) \rfloor$	$\lfloor \log_2 r(u) \rfloor$	$\lfloor \log_2 p^k \rfloor$
BN	$u = 2^{114} + 2^{101} - 2^{14} - 1$	4	115	462	462	5535
BLS-12	$u = -2^{77} + 2^{50} + 2^{33}$	3	77	461	308	5532

curves are not significant as final exponentiation. The major difference is made by the calculation of hard part of the final exp.

TABLE 10.5: Comparative results of Miller's Algorithm and Final Exp. in [ms]

	Pairing		
	Miller Algo.	Final Exp.	Total time [ms]
BN	7.53	20.63	**28.16**
BLS-12	9.93	37.05	46.98

TABLE 10.6: Operation count in $\mathbb{F}_p$ for 1 single pairing operation

		Multiplication	Squaring	Addition/ Subtraction	Basis multiplication	Inversion
BN	Miller's Algo.	10957	157	35424	3132	125
	Final exp.	29445	25	126308	9808	1
	Total	40402	182	161732	12940	**126**
BLS-12	Miller's Algo.	7178	183	23768	857	81
	Final exp.	25708	2	111157	3832	1
	Total	32886	185	134925	4689	82

Chapter 11

ITC CSCC 2017

In pairing-based cryptography, scalar multiplication is often regarded as one of the major bottlenecks for faster pairing calculations. Frobenius map and skew Frobenius map over the twisted curve, are common techniques to speed up scalar multiplication in a pairing calculation. This thesis explicitly shows the detailed procedure to calculate the Frobenius map and skew Frobenius map and their computational complexity in the context of Ate-based pairing over Kachisa-Schaefer-Scott (KSS) curve of embedding degree 16.

11.1 Introduction

Pairing-based cryptography is regarded as the basis of next generation security protocols. From the very beginning, it attracts many researchers which offered us many innovative security protocols till this date. But still there exist several major challenges such as efficiently carry out Miller's algorithm, final exponentiation, efficient scalar multiplication and so on, to practically use pairing in cryptography. Among several optimization techniques, the Frobenius mapping is well-known for efficient scalar multiplication. Sakemi et al. [Sak+08] have shown a technique named as skew Frobenius map in a twisted curve for efficiently calculating scalar multiplication.

The main focus of this thesis is to explicitly show the implementation procedure of Frobenius map and skew Frobenius map for KSS curve of embedding degree 16 (KSS16) in the context of optimal Ate pairing. This thesis also gives some comparative study between this two procedures. Recently Ghammam et al. [GF16a] have proposed that KSS16 curve is a strong candidate to implement pairing-based cryptography at 192-bit security level. Therefore the authors selected KSS16 curve to obtain the skew Frobenius map over the quartic twisted curve. Moreover, to our knowledge, till this date, no work has been proposed for efficiently calculating scalar multiplication over KSS16 curve using skew Frobenius map. This thesis will give a clear outline to utilize skew Frobenius map for efficient scalar multiplication.

11.2 Preliminaries

Fundamentals of KSS curve and optimal Ate pairing are briefly given in this section.

11.2.1 Kachisa-Schaefer-Scott (KSS) curve [KSS07]

In [KSS07], Kachisa, Schaefer, and Scott proposed a family of non super-singular Brezing-Weng pairing friendly elliptic curves using the elements in the cyclotomic field. In what follows, this papers considers *KSS16* curve of embedding degree $k = 16$, defined over $\mathbb{F}_{p^{16}}$ as follows:

$$E/\mathbb{F}_{p^{16}} : Y^2 = X^3 + aX, \quad (a \neq 0 \in \mathbb{F}_p)\,, \tag{11.1}$$

where $X, Y \in \mathbb{F}_{p^{16}}$. Its characteristic p, Frobenius trace t and order r are given by the integer variable u as follows:

$$\begin{aligned} p(u) &= (u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240 \\ &\quad u^4 + 625u^2 + 2398u + 3125)/980, && (11.2a) \\ r(u) &= u^8 + 48u^4 + 625, && (11.2b) \\ t(u) &= (2u^5 + 41u + 35)/35, && (11.2c) \end{aligned}$$

where u is such that $u \equiv 25$ or $45 \pmod{70}$.

11.2.1.1 Towering of $\mathbb{F}_{p^{16}}$ extension field

Let the characteristics p of KSS16 is such that $p \equiv 5 \bmod 8$ and c is a quadratic non-residue in $\mathbb{F}_p$. By using irreducible binomials, $\mathbb{F}_{p^{16}}$ is constructed for KSS16 curve as follows:

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 - c), \\ \mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/(\beta^2 - \alpha), \\ \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\gamma]/(\gamma^2 - \beta), \\ \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\omega]/(\omega^2 - \gamma), \end{cases} \tag{11.3}$$

Here $c = 2$ will be the most efficient if chosen along with the value of mother parameter u.

11.2.2 Pairings

Asymmetric bilinear pairing requires two rational point groups to be mapped to a multiplicative group. In what follows, optimal Ate pairing over KSS curve of embedding degree $k = 16$ can be described as follows.

11.2.2.1 Optimal-Ate pairing

Let us consider the following two additive groups as $\mathbb{G}_1$ and $\mathbb{G}_2$ and a multiplicative group as $\mathbb{G}_3$ of the same order r. The Ate pairing α is defined as follows:

$$\begin{aligned}\mathbb{G}_1 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [1]),\\ \mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]).\end{aligned}$$

$$\alpha : \mathbb{G}_2 \times \mathbb{G}_1 \longrightarrow \mathbb{F}_{p^k}/(\mathbb{F}^*_{p^k})^r. \tag{11.4}$$

where $\mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $\mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ in the case of KSS16 curve.

Let $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$, Ate pairing $\alpha(Q, P)$ is given as follows.

$$\alpha(Q, P) = f_{t-1,Q}(P)^{\frac{p^k-1}{r}}, \tag{11.5}$$

where $f_{t-1,Q}(P)$ symbolizes the output of Miller's algorithm.

The optimal Ate pairing over the KSS16 curve is represented as,

$$(Q, P) = ((f_{u,Q} \cdot l_{[u]Q,[p]Q})^{p^3} \cdot l_{Q,Q})^{\frac{p^{16}-1}{r}}, \tag{11.6}$$

by Zhang et al. [ZL12] utilizing $p^8 + 1 \equiv 0 \bmod r$, where u is the mother parameter. In Eq.(11.6), line evaluation $l_{[u]Q,[p]Q}$ requires scalar multiplication of Q by p. The multiplication of the 1st two terms requires exponentiation by p^3. This two calculation can be efficiently carried by Frobenius map and skew Frobenius map which is the major focus of this thesis.

11.3 Proposal

This section describes the Frobenius map for the rational points of KSS16 curve and skew Frobenius map for the rational points of quartic twisted curve of KSS16 curve defined over $\mathbb{F}_{p^4}$.

11.3.1 Frobenius mapping in $E(\mathbb{F}_{p^{16}})$

Let (x, y) be certain rational point in $E(\mathbb{F}_{p^{16}})$. By the definition, Frobenius map, denoted as $\pi_p : (x, y) \mapsto (x^p, y^p)$, is the p-th power of the rational point defined over $\mathbb{F}_{p^{16}}$.

Since towering is applied to construct the extension field arithmetic for KSS16 curve, therefore a top-down approach can be applied to calculate the Frobenius map. Let $Q \in E(\mathbb{F}_{p^{16}})$ be a rational point of KSS16 curve E, whose Frobenius map (FM) is given as $\pi_p(Q) = (x_Q^p, y_Q^p)$. Now the FM of $x_Q^p = (x_0 + x_1\omega)^p$, where $x_0, x_1 \in \mathbb{F}_{p^8}$ can be calculated as follows:

$$x_Q^p = x_0^p + x_1^p \omega^p,$$

where x_0^p, x_1^p are the Frobenius maps in $\mathbb{F}_{p^8}$. The ω^p term can be simplified as follows:

$$\begin{aligned}
\omega^p &= (\omega^2)^{\frac{p-1}{2}} \omega \\
&= (\gamma^2)^{\frac{p-1}{4}} \omega, \quad \text{since } p \equiv 5 \bmod 8, \\
&= (\beta)^{\frac{p-1}{4}-1} \beta\omega \\
&= (\beta^2)^{\frac{p-5}{8}} \beta\omega \\
&= (\alpha)^{\frac{p-5}{8}-1} \alpha\beta\omega \\
&= (\alpha^2)^{\frac{p-13}{16}} \alpha\beta\omega \\
&= c^{\frac{p-13}{16}} \alpha\beta\omega.
\end{aligned}$$

Therefore, FM of x_Q in $\mathbb{F}_{p^{16}}$ requires FM of x_0, x_1 in $\mathbb{F}_{p^8}$. The simplified ω^p shows that 8 $\mathbb{F}_p$ multiplications by the pre-computed $c^{\frac{p-13}{16}}$ is required with FM of x_1^p. Multiplication by the basis element $\alpha\beta$ will change the position of the coefficients. The appearance of $\alpha^2 = c$ during the basis multiplication can also be pre-calculated together with $c^{\frac{p-13}{16}}$. Therefore, the number of $\mathbb{F}_p$ multiplication will not increase in this context.

FM of $x_0^p = (n_0 + n_1\gamma)^p \in \mathbb{F}_{p^8}$, $n_0, n_1 \in \mathbb{F}_{p^4}$, can be obtained as follows:

$$x_0{}^p = n_0^p + n_1^p \gamma^p,$$

where n_0^p, n_1^p are FM in $\mathbb{F}_{p^4}$ and γ^p is simplified as,

$$\begin{aligned}
\gamma^p &= (\gamma^2)^{\frac{p-1}{2}} \gamma \\
&= (\beta^2)^{\frac{p-1}{4}} \gamma \\
&= (\alpha)^{\frac{p-1}{4}-1} \alpha\gamma \\
&= (\alpha^2)^{\frac{p-5}{8}} \alpha\gamma \\
&= c^{\frac{p-13}{8}} \alpha\gamma. \qquad (11.7)
\end{aligned}$$

The same procedure is also applicable for $x_1^p \in \mathbb{F}_{p^8}$. From the above simplification of γ^p, it is clear that 4 $\mathbb{F}_p$ multiplications by pre-computed $c^{\frac{p-5}{8}}$ and a multiplication by the basis element α is required. Since they are also part of $\mathbb{F}_{p^8}$ vector, therefore the multiplication of $c^{\frac{p-5}{8}}$ can be combined with $c^{\frac{p-13}{16}}$ during FM of $\mathbb{F}_{p^{16}}$.

FM of $n_0^p = (m_0 + m_1\beta)^p \in \mathbb{F}_{p^4}$ where $m_0, m_1 \in \mathbb{F}_{p^2}$ is calculated as follows:

$$n_0^p = m_0^p + m_1^p \beta^p, \qquad (11.8)$$

where m_0^p and m_1^p are FM in $\mathbb{F}_{p^2}$. The β^p is calculated as,

$$\begin{aligned}\beta^p &= (\beta^2)^{\frac{p-1}{2}}\beta \\ &= (\alpha^2)^{\frac{p-1}{4}}\beta \\ &= c^{\frac{p-1}{4}}\beta.\end{aligned}$$

It implies that FM in $\mathbb{F}_{p^4}$ requires 2 FM in $\mathbb{F}_{p^2}$ and 2 $\mathbb{F}_p$ multiplication by pre-calculated $c^{\frac{p-1}{4}}$. This 2 $\mathbb{F}_p$ multiplications can also be combined with previous pre-calculated multiplications.

And finally FM of $m_0^p = (b_0 + b_1\alpha)^p \in \mathbb{F}_{p^2}$, $b_0, b_1 \in \mathbb{F}_p$, is given as follows:

$$\begin{aligned}m_0^p &= b_0^p + b_1^p\alpha^p \\ &= b_0 + b_1(\alpha^2)^{\frac{p-1}{12}}\alpha \\ &= b_0 + b_1 c^{\frac{p-1}{2}}\alpha \\ &= b_0 - b_1\alpha,\end{aligned}$$

where except changing the sign, no operations are required since c is quadratic no-residue in $\mathbb{F}_p$. Therefore, during the FM of x_Q, the 1st half ($x_0 \in \mathbb{F}_{p^8}$) of 16 coefficients, it only takes 6 $\mathbb{F}_p$ multiplications and for the 2nd half (x_1) it requires 8 $\mathbb{F}_p$ multiplications. The total number of operation in $\mathbb{F}_p$ for a single FM of $Q \in \mathbb{F}_{p^{16}}$ is given in Table 11.3.

11.3.2 Skew Frobenius map

Similar to Frobenius mapping, skew Frobenius map (SFM) is the p-th power of the rational points over the twisted curve. In the context of KSS16 curve, there exists a quartic twisted curve E' of order r defined over $\mathbb{F}_{p^4}$. Let $Q' = (x', y')$ be a point on the twisted curve E'. Then SFM of Q' is given as $\pi' : (x', y') \mapsto (x'^p, y'^p)$. To calculate the SFM, at first let us find the quartic twisted curve of KSS16.

11.3.2.1 Quartic twisted mapping

For quartic twisted mapping first we need to obtain certain ration point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ of subgroup order r. In what follows, let us consider the rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ and its quartic twisted rational point $Q' \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$. Rational point Q has a special vector representation given in Table 11.1. From the Table 11.1, coordinates of $Q = (x_Q, y_Q) \in \mathbb{F}_{p^{18}}$ are obtained as $Q = (x_Q, y_Q) = (\gamma x_{Q'}, \omega\gamma y_{Q'})$, where $x_{Q'}, y_{Q'}$ are the coordinates of the rational point Q' in the twisted curve. Now let's find the twisted curve of Eq.(12.1) in

TABLE 11.1: Vector representation of $Q = (x_Q, y_Q) \in \mathbb{F}_{p^{16}}$

-	1	α	β	$\alpha\beta$	γ	$\alpha\gamma$	$\beta\gamma$	$\alpha\beta\gamma$	ω	$\alpha\omega$	$\beta\omega$	$\alpha\beta\omega$	$\gamma\omega$	$\alpha\gamma\omega$	$\beta\gamma\omega$	$\alpha\beta\gamma\omega$
x_Q	0	0	0	0	b_4	b_5	b_6	b_7	0	0	0	0	0	0	0	0
y_Q	0	0	0	0	0	0	0	0	0	0	0	0	b_{12}	b_{13}	b_{14}	b_{15}

$\mathbb{F}_{p^4}$ as follows:

$$\begin{aligned}
(\omega\gamma y_{Q'})^2 &= (\gamma x_{Q'})^3 + a(\gamma x_{Q'}), \\
\gamma\beta y_{Q'}^2 &= \gamma\beta x_{Q'}^3 + a\gamma x_{Q'}, \\
& \text{multiplying } (\gamma\beta)^{-1} \text{ both sides.} \\
y_{Q'}^2 &= x_{Q'}^3 + a\beta^{-1} x_{Q'},
\end{aligned} \tag{11.9}$$

The twisted curve of E is obtained as $E' : y^2 = x^3 + a\beta^{-1}x$, where β is the basis element in $\mathbb{F}_{p^4}$. Therefore the quartic mapping can be represented as follows:

$$\begin{aligned}
Q &= (x_Q, y_Q) = (\gamma x_{Q'}, \omega\gamma y_{Q'}) \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}}) \\
&\longmapsto Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})
\end{aligned}$$

For mapping and remapping between Q to Q' and no extra calculation is required. By picking the non-zero coefficients of Q and placing it to the corresponding basis position is enough to get Q'.

Moreover, in the case of KSS16 curve, it is known that Q satisfies the following relations:

$$\begin{aligned}
[\pi_p - p]Q &= \mathcal{O} \\
\pi_p(Q) &= [p]Q.
\end{aligned} \tag{11.10}$$

which can be accelerated scalar multiplication in $\mathbb{G}_2$.

11.3.2.2 SFM calculation

The detailed procedure to obtain the skew Frobenius map of $Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$ is given bellow:

$$(x_{Q'}\gamma)^p = (x_{Q'})^p \gamma^p.$$

After remapping

$$(x_{Q'})^p \gamma^{p-1} = (x_{Q'})^p (\gamma^2)^{\frac{p-1}{2}},$$

where $(x_{Q'})^p \in \mathbb{F}_{p^4}$ can be calculated as Frobenius map in $\mathbb{F}_{p^4}$ same as Eq.(11.8). The $(\gamma^2)^{\frac{p-1}{2}}$ term can be simplified as follows:

$$\begin{aligned}
(\gamma^2)^{\frac{p-1}{2}} &= (\beta^2)^{\frac{p-1}{4}}, \quad \text{since } p \equiv 5 \bmod 8, \\
&= (\alpha)^{\frac{p-1}{4}-1}\alpha \\
&= (\alpha^2)^{\frac{p-5}{8}}\alpha \\
&= c^{\frac{p-5}{8}}\alpha. \qquad (11.12\text{a})
\end{aligned}$$

SFM of $y_{Q'}$ is given as,

$$(y_{Q'}\gamma\omega)^p = (y_{Q'})^p\gamma^p\omega^p.$$

After remapping

$$(y_{Q'})^p\gamma^{p-1}\omega^{p-1} = (y_{Q'})^p(\gamma^2)^{\frac{p-1}{2}}(\omega^2)^{\frac{p-1}{2}},$$

$(y_{Q'})^p$ is calculated as same of Eq.(11.8) in $\mathbb{F}_{p^4}$ and $(\gamma^2)^{\frac{p-1}{2}}$ is calculated same as Eq.(12.26a). The $(\omega^2)^{\frac{p-1}{2}}$ term is calculated as follows:

$$\begin{aligned}
(\omega^2)^{\frac{p-1}{2}} &= (\gamma^2)^{\frac{p-1}{4}}, \quad \text{since } p \equiv 5 \bmod 8, \\
&= \beta^{\frac{p-1}{4}-1}\beta \\
&= (\beta^2)^{\frac{p-5}{8}}\beta \\
&= (\alpha)^{\frac{p-5}{8}}\beta \\
&= (\alpha)^{\frac{p-5}{8}-1}\alpha\beta \\
&= (\alpha^2)^{\frac{p-13}{16}}\alpha\beta \\
&= c^{\frac{p-13}{16}}\alpha\beta.
\end{aligned}$$

Here the multiplications by $c^{\frac{p-13}{16}}$ and $c^{\frac{p-5}{8}}$ together with the basis elements α and $\alpha\beta$ will generate scalars, basically exponents of c. Therefore they can be pre-computed since c is known during extension field construction. Finally, it requires 8 $\mathbb{F}_p$ multiplications by pre-computed values to calculate SFM of $Q' \in \mathbb{G}_2'$.

11.4 Results evaluation

This section gives the computational cost comparison of Frobenius map and skew Frobenius map with respect to operation count and execution time while it has been implemented for calculating optimal Ate pairing over KSS16 curve. Recently, Barbulescu et al. [BD17] have presented new parameters for pairing friendly curves. This thesis has considered their proposed KSS16 curve as $y^2 = x^3 + x \in \mathbb{F}_{p^{16}}$. The mother parameter $u = 2^{35} - 2^{32} - 2^{18} + 2^8 + 1$ and the quadratic non-residue $c = 2$ in $\mathbb{F}_p$ of Eq.(11.3) is considered accordingly.

Table 11.2 shows the experiment environment used to implement the techniques. Table 11.3 shows the execution time for calculating the p-th power, FM and SFM for rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ where m denotes multiplication in $\mathbb{F}_p$. It is apparent that skew Frobenius map over $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^4})$ in the twisted curve is about four times faster than Frobenius mapping in $Q \in \mathbb{G}_2$. In [Sak+08], Sakemi et al. have shown an efficient scalar multiplication by applying skew Frobenius mapping in the context of Ate-based pairing in BN curve of embedding degree $k = 12$. Such technique can also be applied in KSS16 curve for the same. Moreover, multi-scalar multiplication technique can also be obtained using the proposed skew Frobenius map.

TABLE 11.2: Computational Environment

•	PC
CPU [*]	2.7 GHz Intel Core i5
Memory	16 GB
OS	Mac OS X 10.12.3
Compiler	gcc 4.2.1
Programming Language	C
Library	GNU MP 6.1.0 [Gt15]

[*] Only single core is used from two cores.

TABLE 11.3: Computational cost

Operation	Execution time [ms]	$\mathbb{F}_p$ operations
p-th power	343.21	-
Frobenius map	0.054	28 m
Skew Frobenius map	0.014	8 m

11.5 Conclusion and future work

This thesis shows the detailed procedure to efficiently carry out Frobenius map and skew Frobenius map in a quartic twisted KSS16 curve in the context of optimal Ate pairing. It is evident from the experimental implementation that, skew Frobenius map is about 4 times faster than Frobenius map for $\mathbb{G}_2$ rational points. As a future work, we would like to extend this work for efficient scalar multiplication together with some pairing-based protocol implementation.

Chapter 12

Efficient $\mathbb{G}_2$ Scalar Multiplican in KSS-16 Curve

Pairing-based protocols are getting popular in many cryptographic applications. Pairing algorithms involve computations on elements in all three pairing groups, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$; however, most protocols usually require additional scalar multiplication and exponentiation in any of these three groups. The Gallant-Lambert-Vanstone (GLV) method is an elegant technique to accelerate the scalar multiplication which can reduce the number of elliptic curve doubling by using Straus-Shamir simultaneous multi-scalar multiplication technique. However, efficiently computable endomorphisms are required to apply GLV for the elliptic curves. This thesis shows the GLV technique by deriving efficiently computable endomorphism for Kachisa-Schaefer-Scott (KSS) curve defined over degree 16 extension field. In addition, the authors show explicit formulas to compute the GLV method together with Straus-Shamir simultaneous multi-scalar multiplication technique for 2, 4 and 8 dimensions in $\mathbb{G}_2$ group. The comparative implementation shows that dimension 4 gives faster computational time than dimension 8 and 2.

12.1 Introduction

The independent works of Sakai et al. [Sak00] and Joux et al. [Jou04] set up a new type of public key cryptography based on bilinear pairing over the elliptic curve. Since then it has encouraged to invent several innovative pairing-based cryptographic applications such as Boneh et al.'s short signature [BLS01] and short group signature authentication [BBS04], those have increased the popularity of pairing-based cryptographic research. However, most pairing-based protocols require additional scalar multiplication (SCM) or exponentiation in the pairing groups beside the pairing. Moreover, some protocols [Gro10] require only a single pairing and several scalar multiplications. Therefore, we are interested in such peripheral operation that is the scalar multiplication of pairing-based protocols.

In general, pairing is a bilinear map of two rational point groups $\mathbb{G}_1$ and $\mathbb{G}_2$ to a multiplicative group $\mathbb{G}_3$ [SCA86]. The typical notation of pairing is $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3$. Pairings are often defined over a certain extension field $\mathbb{F}_{p^k}$, where p is the prime number, also known as characteristics and k is the minimum extension degree for pairing also called *embedding* degree. More importantly, pairing is performed over the special form of elliptic curves typically known as pairing-friendly curves. There are several widely studied pairing-friendly curve families i.e. Barreto-Naehrig (BN), Barreto-Lynn-Scott (BLS) curves[FST10]. The set of rational points $E(\mathbb{F}_{p^k})$ are defined over a certain pairing-friendly curve of embedded extension field of degree k. This thesis has considered Kachisa-Schaefer-Scott (KSS) [KSS07] pairing-friendly curves of embedding degree $k = 16$ (KSS-16) in the context of Optimal-ate pairing. The motivation to work on KSS-16 curve came from the recent work of Barbulescu et al. [BD18] and Khandaker et al. [Kha+17b], where they concluded that with the recent parameters for pairing-based protocols, KSS-16 curve is a better choice for Optimal-ate pairing over BN curve.

Scalar multiplication dominates the execution time of any elliptic curve cryptography (ECC) algorithms. The common approach to accelerate scalar multiplication are log-step algorithm such as binary and non-adjacent form (NAF) methods. However, in the context of asymmetric pairing where there exists no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$, more efficient approach is to use GLV [Sak+08; KN17]. In order to accelerate scalar multiplication, Gallant-Lambert-Vanstone [GLV01] proposed a technique for rational points of prime order known as GLV method. Fundamentally, it divides the scalar into half of the bit length of the original one that reduces the number of doubling. The critical point of this technique is, there should have to be an efficiently computable endomorphism. Otherwise, the advantage obtained from reduced doubling will have no effect on the acceleration.

There is a vast literature on GLV decomposition in pairing-friendly curves i.e. Barreto-Naehrig [BN06], Kachisa-Schaefer-Scott (KSS) curve of embedding degree 18, [Sak+08; KN17; Nog+09; FLS15; GLS11]. The common fact of in such literature is, they all applied GLV on sextic twisted curves. However, in our knowledge till date, there is no literature on GLV decomposition for KSS curve of embedding degree 16 where at most degree 4 twist is available.

The major contributions of this thesis are (I) obtaining the endomorphism to enable GLV decomposition for $\mathbb{G}_2$ rational point in KSS-16 curve. (II) Deriving the dimension 2, 4 and 8 GLV decomposition along with finding efficiently computable Frobenius maps. (III) Implementation of the derived techniques and their comparison. This thesis shows that increasing the dimension of decomposition not necessarily accelerate the scalar multiplication. In the case of $\mathbb{G}_2$ points of KSS-16 curve, our experiment finds that dimension 4 is the fastest.

Throughout this thesis, p and k denote characteristic and embedding extension degree respectively. $\mathbb{F}_{p^k}$ denotes k-th extension field over prime field $\mathbb{F}_p$ and $\mathbb{F}^*_{p^k}$ denotes the multiplicative group in $\mathbb{F}_{p^k}$.

12.2 Fundamentals of Elliptic Curve and Pairing

12.2.1 Kachisa-Schaefer-Scott (KSS) Curve [KSS07]

Kachisa, Schaefer, and Scott proposed a new family of parameterized non super-singular pairing-friendly elliptic curves of embedding degree $k = \{16, 18, 32, 36, 40\}$. In this thesis, we are interested in the KSS curve whose embedding degree is 16. Unlike mostly used degree 6 twist, this curve offers at most degree 4 twist often named as the *quartic twist*.

In what follows, this thesis considers the KSS curve of embedding degree $k = 16$, denoted as *KSS-16*, defined over extension field $\mathbb{F}_{p^{16}}$ as follows:

$$E/\mathbb{F}_{p^{16}} : y^2 = x^3 + ax, \quad a \neq 0, \tag{12.1}$$

where $x, y \in \mathbb{F}_{p^{16}}$. As a typical feature, its properties are parameterized by the polynomial formulas of integer u as follows:

$$\begin{aligned}
p(u) &= (u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 \\
&\quad +625u^2 + 2398u + 3125)/980, && (12.2a)\\
r(u) &= (u^8 + 48u^4 + 625)/61255, && (12.2b)\\
t(u) &= (2u^5 + 41u + 35)/35, && (12.2c)
\end{aligned}$$

where the tuple (p, r, t) are *characteristic*, *group order* and *Frobenius trace* respectively. The integer u denotes the pairing parameter conform to the condition $u \equiv 25$ or $45 \pmod{70}$.

12.2.2 Point Addition and Doubling

Let $E(\mathbb{F}_{p^k})$ be the set of all rational points on the curve E including the point at infinity $\mathcal{O}$. $\#E(\mathbb{F}_{p^k})$ denotes the total number of point in $E(\mathbb{F}_{p^k})$. Let us consider two rational points using the affine coordinates as $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, and their addition $R = P_1 + P_2$, where $R = (x_3, y_3)$ and $P_1, P_2, R \in E(\mathbb{F}_{p^k})$. Then the x and y coordinates of R are calculated as follows:

$$\begin{aligned}
x_3 &= \lambda^2 - x_1 - x_2, && (12.3a)\\
y_3 &= (x_1 - x_3)\lambda - y_1, && (12.3b)
\end{aligned}$$

where λ is the tangent at the point on the curve given as follows:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1}; & P_1 \neq P_2, \\ \frac{3x_1^2 + a}{2y_1}; & P_1 = P_2, \end{cases} \tag{12.3c}$$

and $\mathcal{O}$ is the additive unity in $E(\mathbb{F}_{p^k})$. If $P_1 \neq P_2$ then $P_1 + P_2$ is called elliptic curve addition (ECA). If $P_1 = P_2$ then $P_1 + P_2 = 2P_1$, which is known as elliptic curve doubling (ECD).

12.2.3 Elliptic Curve Scalar Multiplication

Let scalar s is $0 \leq s < r$, where r is the order of the target rational point group. Scalar multiplication of rational points P_1, denoted as $[s]P_1$ is calculated by $(s-1)$-times additions of P_1 as,

$$[s]P_1 = \sum_{i=0}^{s-1} P_1, \quad 0 \leq s < r. \tag{12.4}$$

When $s = r$, then $[r]P_1 = \mathcal{O}$, where r is the order of the curve. Let $[s]P_1 = P_2$, and value of s is not obtained, then the solving s from P_1 and P_2 is known as the elliptic curve discrete logarithm problem (ECDLP). The difficulty level of solving ECDLP defines the security strength of elliptic curve cryptography.

12.2.4 Extension Field Arithmetic for Pairing

While implementing pairing-based protocols, a major speedup comes from the efficient finite field implementation. Therefore, it is common to apply the towering technique to efficiently carry out the extension field operations. In what follows, the $\mathbb{F}_{p^{16}}$ extension field is constructed using the following towering with help of the irreducible binomials given in Eq.(12.5). For such construction, in addition with $4|p-1$, p satisfies $p \equiv 3, 5 \bmod 8$.

$$\begin{cases} \mathbb{F}_{p^2} = \mathbb{F}_p[\alpha]/(\alpha^2 - c), \\ \mathbb{F}_{p^4} = \mathbb{F}_{p^2}[\beta]/(\beta^2 - \alpha), \\ \mathbb{F}_{p^8} = \mathbb{F}_{p^4}[\gamma]/(\gamma^2 - \beta), \\ \mathbb{F}_{p^{16}} = \mathbb{F}_{p^8}[\omega]/(\omega^2 - \gamma), \end{cases} \tag{12.5}$$

where c is a quadratic non-residue (QNR) in $\mathbb{F}_p$. This thesis considers $c = 2$, where $X^{16} - 2$ is irreducible in $\mathbb{F}_{p^{16}}$.

12.2.5 Extension Field Arithmetic of $\mathbb{F}_{p^{16}}$

Towering allows to efficiently carry out multiplication and squaring operation in the extension fields. **Table** 12.1 shows operation count for Karatsuba based multiplication and Devegili et al.'s [Dev+06] complex squaring technique for squaring. The arithmetic operations in $\mathbb{F}_p$ are denoted as M_p for a multiplication, S_p for a squaring, I_p for an inversion. The m with Greek alphabet and M with numeric suffix denote multiplication with basis element and multiplication in extension field respectively. Since, $c = 2$ in Eq.(12.5), therefore, the multiplication by the basis element α is carried out by 1 addition in $\mathbb{F}_p$.

12.2.6 Optimal-Ate Pairing on KSS-16 Curve

In the context of pairing on the KSS-16 curves, the valid bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_3$ takes input from two additive rational point groups $\mathbb{G}_1, \mathbb{G}_2$

TABLE 12.1: Number of arithmetic operations in $\mathbb{F}_{p^{16}}$ based on Type-I towering Eq.(12.5).

Multiplication	Squaring
$M_2 = 3M_p + 5A_p + 1m_\alpha \longrightarrow 3M_p$	$S_2 = 2M_p + 6A_p + \longrightarrow 2M_p$
$M_4 = 2M_2 + 5A_2 + 1m_\beta \longrightarrow 9M_p$	$S_4 = 2M_2 + 5A_2 + 2m_\beta \longrightarrow 6M_p$
$M_8 = 3M_4 + 5A_4 + 1m_\gamma \longrightarrow 27M_p$	$S_8 = 2M_4 + 5A_4 + 2m_\gamma \longrightarrow 18M_p$
$M_{16} = 3M_8 + 5A_8 + 1m_\omega \longrightarrow 81M_p$	$S_{16} = 2M_8 + 5A_8 + 2m_\omega \longrightarrow 54M_p$

and output an element in the multiplicative group $\mathbb{G}_3$ of order r. $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_3$ are defined as follows:

$$\begin{aligned}
\mathbb{G}_1 &= E(\mathbb{F}_p)[r] \cap \mathrm{Ker}(\pi_p - [1]),\\
\mathbb{G}_2 &= E(\mathbb{F}_{p^k})[r] \cap \mathrm{Ker}(\pi_p - [p]),\\
\mathbb{G}_3 &= \mathbb{F}^*_{p^k}/(\mathbb{F}^*_{p^k})^r,
\end{aligned}$$

where $E(\mathbb{F}_{p^k})[r]$ denotes rational points of order r and $[n]$ is scalar multiplication for a rational point. Let π_p denotes the Frobenius endomorphism given as $\pi_p : (x, y) \mapsto (x^p, y^p)$.

Unless otherwise stated, the rest of the thesis considers $P \in \mathbb{G}_1 \subset E(\mathbb{F}_p)$ and $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$, where, Q satisfies $[\pi_p - p]Q = \mathcal{O}$. The map e involves two major steps named Miller's loop followed by the final exponentiation. The Optimal-ate pairing [Ver10] on KSS-16 elliptic curve is given by Zhang et al. [ZL12] and presented by the following map.

$$e_{opt} : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}_3, \tag{12.6}$$

$$(P, Q) \longmapsto \left((f_{u,Q}(P) l_{[u]Q,[p]Q}(P))^3 l_{Q,Q}(P) \right)^{\frac{16-1}{r}}. \tag{12.7}$$

The rational function $f_{u,Q}(P)$, line evaluation $l_{[u]Q,[p]Q}$ have computed thanks to the Miller algorithm. Then, we have the second step which is the computation of the exponent $\frac{16-1}{r}$ named the final exponentiation.

12.2.7 Gallant, Lambert, and Vanstone (GLV) Decomposition

In CRYPTO 2001 [GLV01], Gallant, Lambert, and Vanstone found that any multiple $[s]Q$ of a point Q of prime order r lying on an elliptic curve with a low-degree endomorphism Φ over $\mathbb{F}_p$ can be calculated as follows:

$$[s]Q = s_1 Q + s_2 \Phi(Q), \tag{12.8}$$

where $max|s_1|, |s_2| \leq C_1\sqrt{r}$ for some explicit constant $C_1 > 0$. The main idea of the GLV trick is it exists essentially in an algorithm that finds a decomposition of an arbitrary scalar multiplication $[s]$ for $s \in [1, r]$ into two scalar multiplications, while the new scalars having only about half the bit length of the original scalar. This immediately enables the elimination of half the doubling by employing the Straus-Shamir simultaneous multi-scalar point multiplication. Later on Galbraith-Lin-Scott (GLS) have shown that over $\mathbb{F}_{p^2}$. This thesis focuses on such a trick for the KSS-16 curve in the context of Optimal-ate pairing.

12.3 GLV technique for $\mathbb{G}_2$ Rational Point on KSS-16 Curve

As aforementioned, Optimal-ate pairing is computed over a twisted curve. Therefore, the following sections will describe the twist property of KSS-16 curve and the procedure to obtain GLV decomposition in the $\mathbb{G}_2$ group of a KSS-16 curve.

12.3.1 Quartic Twist of KSS-16 Curves

There exists a *twisted curve* with a group of rational points of order r for a KSS-16 curve. This isomorphic rational point group includes a twisted isomorphic point of $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^k})$, typically denoted as $Q' \in E'(\mathbb{F}_{p^{k/d}})$, where k is the embedding degree and d is the twist degree. Since the pairing-friendly KSS-16 [KSS07] curve has CM discriminant of $D = 1$ and $4|k$; therefore, a quartic twist is available.

Let β be a certain quadratic non-residue in $\mathbb{F}_{p^4}$. The quartic twisted curve E' of KSS-16 curve E defined in Eq.(12.1) and their isomorphic mapping ψ_4 are given as follows:

$$E' \; : \; y^2 = x^3 + ax\beta^{-1}, \quad a \in \mathbb{F}_p, \tag{12.9}$$

$$\begin{aligned} \psi_4 \; : \; & E'(\mathbb{F}_{p^4})[r] \longmapsto E(\mathbb{F}_{p^{16}})[r] \cap \mathrm{Ker}(\pi_p - [p]), \\ & (x, y) \longmapsto (\beta^{1/2}x, \beta^{3/4}y), \end{aligned} \tag{12.10}$$

where Ker$(\cdot)$ denotes the kernel of the mapping and π_p denotes Frobenius mapping for rational point.

For the above mapping, the vector representation of $Q = (x_Q, y_Q) = (\beta^{1/2}x_{Q'}, \beta^{3/4}y_{Q'}) \in \mathbb{F}_{p^{16}}$ is obtained according to the given towering in Eq.(12.5). Here, $x_{Q'}$ and $y_{Q'}$ are the coordinates of rational point Q' on quartic twisted curve E'.

12.3.2 Elliptic Curve Operation in Twisted Curve E'

Since E' in Eq.(12.9) is different from E, therefore, the elliptic curve addition and doubling operation slightly changed. Let us consider $T = (\gamma x_{T'}, \gamma\omega y_{T'})$, $Q = (\gamma x_{Q'}, \gamma\omega y_{Q'})$ and $P = (x_P, y_P)$, where $x_p, y_p \in \mathbb{F}_p$ given in affine coordinates on the curve $E(\mathbb{F}_{p^{16}})$ such that $T' = (x_{T'}, y_{T'})$, $Q' = (x_{Q'}, y_{Q'})$ are in the twisted curve E' defined over $\mathbb{F}_{p^4}$. Let the elliptic curve doubling of $T + T = R(x_R, y_R)$.

$$\begin{aligned}\lambda &= \frac{3x_{T'}^2\gamma^2 + a}{2y_{T'}\gamma\omega} = \frac{3x_{T'}^2\gamma\omega^{-1} + a(\gamma\omega)^{-1}}{2y_{T'}},\\ &= \frac{(3x_{T'}^2 + ac^{-1}\alpha\beta)\omega}{2y_{T'}} = \lambda'\omega,\end{aligned}$$

since $\gamma\omega^{-1} = \omega, (\gamma\omega)^{-1} = \omega\beta^{-1}$, and $a\beta^{-1} = (a + 0\alpha + 0\beta + 0\alpha\beta)\beta^{-1} = a\beta^{-1} = ac^{-1}\alpha\beta$, where $\alpha^2 = c$. Now the ECD are obtained as follows:

$$\begin{aligned}x_R &= (\lambda')^2\omega^2 - 2x_{T'}\gamma = ((\lambda')^2 - 2x_{T'})\gamma,\\ y_R &= (x_{T'}\lambda' - x_{2T'}\lambda' - y_{T'})\gamma\omega.\end{aligned}$$

The elliptic curve addition phase (i.e. $T \neq Q$) can be written as $T + Q = R(x_R, y_R)$.

$$\begin{aligned}\lambda &= \frac{(y_{Q'} - y_{T'})\gamma\omega}{(x_{Q'} - x_{T'})\gamma} = \frac{(y_{Q'} - y_{T'})\omega}{x_{Q'} - x_{T'}} = \lambda'\omega,\\ x_R &= ((\lambda')^2 - x_{T'} - x_{Q'})\gamma,\\ y_R &= (x_{T'}\lambda' - x_{R'}\lambda' - y_{T'})\gamma\omega.\end{aligned}$$

12.3.3 Finding Endomorphism between p and u

Let us find an endomorphism between the prime p and the integer u from using the Hasse's theorem

$$p + 1 - t \equiv 0 \bmod r,$$

as follows:

$$\begin{aligned}p &\equiv t - 1 \bmod r,\\ 35p &\equiv 2u^5 + 41u \bmod r. \qquad (12.11)\end{aligned}$$

The modulus of order r defined in Eq.(12.2b) can be expressed as

$$u^8 + 48u^4 + 625 \bmod r \equiv 0. \qquad (12.12)$$

From the above equation we approach to find the relation between p and u as follows:

$$\begin{aligned}
2u^8 + 96u^4 + 2 \cdot 5^4 \bmod r &\equiv 0, \\
35pu^3 - 41u^4 + 96u^4 + 2 \cdot 5^4 \bmod r &\equiv 0, \\
35pu^3 + 55u^4 + 2 \cdot 5^4 \bmod r &\equiv 0, \\
7pu^3 + 11u^4 + 2 \cdot 5^3 \bmod r &\equiv 0, \\
11u^4 + 2 \cdot 5^3 \bmod r &\equiv -7pu^3, \\
11u + 2 \cdot 5^3u^{-3} \bmod r &\equiv -7p. \qquad (12.13)
\end{aligned}$$

Let us take 4-th power of both side of the Eq.(12.13).

$$\begin{aligned}
7^4p^4 &\equiv (11u + 2 \cdot 5^3u^{-3})^4 \bmod r, \\
&\equiv 11^4u^4 + 8 \cdot 5^3 11^3 + 24 \cdot 5^6 11^2 u^{-4} + 32 \cdot 11 \cdot 5^9 u^{-8} \\
&\quad + 2^4 5^{12} u^{-12} \bmod r. \qquad (12.14)
\end{aligned}$$

Multiplying u^{-12} with Eq.(12.12) result in the following relation.

$$u^{-4} + 48u^{-8} + 5^4u^{-12} \bmod r \equiv 0.$$

Afterward multiplying $2^4 5^8$ with the above equation is obtained as follows:

$$2^4 5^8 u^{-4} + 48 \cdot 2^4 5^8 u^{-8} + 2^4 5^{12} u^{-12} \bmod r \equiv 0,$$

which helps to simplify the Eq.(12.14) as

$$\begin{aligned}
7^4p^4 &\equiv 11^4u^4 + 8 \cdot 5^3 11^3 + 24 \cdot 5^6 11^2 u^{-4} + 32 \cdot 11 \cdot 5^9 u^{-8} \\
&\quad - 2^4 5^8 u^{-4} - 48 \cdot 2^4 5^8 u^{-8} \bmod r, \\
&\equiv 11^4u^4 + 8 \cdot 5^3 11^3 + 2504 \cdot 5^6 u^{-4} \\
&\quad + 992 \cdot 5^8 u^{-8} \bmod r. \qquad (12.15)
\end{aligned}$$

At this point let us multiply $992 \cdot 5^4u^{-8}$ with Eq.(12.12) to obtain

$$992 \cdot 5^4 + 992 \cdot 48 \cdot 5^4u^{-4} + 992 \cdot 5^8u^{-8} \bmod r \equiv 0.$$

Using the above relation, Eq.(12.15) can be expressed as

$$\begin{aligned}
7^4p^4 &\equiv 11^4u^4 + 8 \cdot 5^3 11^3 + 2504 \cdot 5^6 u^{-4} - 992 \cdot 48 \cdot 5^4 u^{-4} \\
&\quad - 992 \cdot 5^4 \bmod r, \\
&\equiv 11^4x^4 + 5688 \cdot 5^3 + 14984 \cdot 5^4 u^{-4} \bmod r. \qquad (12.16)
\end{aligned}$$

Now, let us multiply $14984u^{-4}$ with Eq.(12.12) to obtain the following equation as

$$14984u^4 + 14984 \cdot 48 + 14984 \cdot 5^4 \bmod r \equiv 0. \qquad (12.17)$$

Substituting the above equation in Eq.(12.16) the final relation can be obtained as follows:

$$\begin{aligned}
7^4p^4 &\equiv 11^4x^4 + 5688 \cdot 5^3 - 14984u^4 - 14984 \cdot 48 \bmod r,\\
&\equiv (14641 - 14984)u^4 + (711000 - 719232) \bmod r,\\
&\equiv -343u^4 - 8232 \bmod r,\\
7p^4 &\equiv -u^4 - 24 \bmod r. \qquad (12.18)
\end{aligned}$$

Finally, $u^4 \equiv -7p^4 - 24 \bmod r$ is the endomorphism we are interested in. Since the relation is obtained for u^4, therefore, we can apply it for 2 dimension GLV decomposition. The reason can be anticipated clearly as the order r is a polynomial of degree 8 of the integer u.

12.3.4 GLV for the Group Having Order $r(u)$

We can apply at most $\varphi(16) = 8$ dimension GLV decomposition for $\mathbb{G}_2$ rational point group; since the KSS-16 is a curve defined over an extension field of degree 16. Here φ is the Euler's totient function. However, as discussed in the introduction, there is always a trade-off between the number of precomputation and the dimension of GLV for any curve.

In the context of KSS-16, $p^{16} - 1$ can be divisible by r from the definition of pairing. Therefore, we got the following equations.

$$p^{16} \equiv 1 \pmod r, \qquad (12.19a)$$

$$p^8 \equiv -1 \pmod r, \qquad (12.19b)$$

$$p^4 \equiv \sqrt{-1} \equiv i \pmod r. \qquad (12.19c)$$

Since -1 is a QNR in $\mathbb{F}_p$, therefore, $\sqrt{-1}$ exists in $\mathbb{F}_p$.

12.3.4.1 Dimension 8 GLV decomposition

Since order r of the KSS-16 curve defined in Eq.(12.2b) is a degree 8 polynomial of integer u, therefore, to obtain dimension 8 GLV decomposition of a scalar s as the following form

$$s = s_0 + us_1 + u^2s_2 + u^3s_3 + u^4s_4 + u^5s_5 + u^6s_6 + u^7s_7,$$

we need to find a relation between above degrees of u and prime p. Let us first obtain a relation between degree 1 of u and p as follows:

$$\begin{aligned}
p &\equiv t-1 \pmod r,\\
35p &\equiv 2u^5+41u \pmod r, \text{ (see Eq.(12.11))}\\
35p &\equiv u(2u^4+41) \pmod r,\\
35p &\equiv u(2(-7p^4-24)+41) \pmod r, \text{ (see Eq.(12.18))}\\
35p &\equiv u(-14p^4-7) \pmod r,\\
5p &\equiv u(-2p^4-1) \pmod r,\\
u &\equiv 5p(-2p^4-1)^{-1} \pmod r,\\
u &\equiv 5p(-2i-1)^{-1} \pmod r, \text{ (see Eq.(12.19}c\text{))}\\
u &\equiv 5p(-2i-1)^{-1}(-2i-1)(2i-1)/5 \pmod r,\\
u &\equiv p(2i-1) \pmod r,\\
u &\equiv 2p^5-p \pmod r. \qquad (12.20)
\end{aligned}$$

12.3.4.2 Dimension 4 GLV decomposition

To obtain the dimension 4 decomposition, we derive the relation between degree 2 of u and p as follows:

$$\begin{aligned}
u^2 &\equiv p^2(2p^4-1)^2 \pmod r,\\
u^2 &\equiv p^2(-4-4p^4+1) \pmod r, \text{ (see Eq.(12.19}b\text{))}\\
u^2 &\equiv -4p^6-3p^2 \pmod r. \qquad (12.21)
\end{aligned}$$

12.3.4.3 Dimension 2 GLV decomposition

Modular equation for dimension 2 GLV is already obtained in Eq.(12.18). However, we can verify that as follows:

$$\begin{aligned}
u^4 &\equiv p^4(-4p^4-3)^2 \pmod r,\\
u^4 &\equiv p^4(-16+24p^4+9) \pmod r, \text{ (see Eq.(12.19}b\text{))}\\
u^4 &\equiv -7p^4-24 \pmod r. \text{ (see Eq.(12.19}b\text{))} \qquad (12.22)
\end{aligned}$$

Beside u, u^2 and u^4 we also need to find the endomorphisms for u^3, u^5, u^6 and u^7. Using the above Eq.(12.20), Eq.(12.21) and Eq.(12.22), they can be given as follows:

$$\begin{aligned}
u^3 &\equiv 11p^3-2p^7,\\
u^5 &\equiv 38p-41p^5,\\
u^6 &\equiv 117p^6+44p^2,\\
u^7 &\equiv -278p^3-29p^7.
\end{aligned}$$

12.3.4.4 Dimension 2 GLV with Joint Sparse Form

In [GHP04], Solinas proposed a joint sparse form (JSF) for two integers. Let say the two integers are s_0 and s_1. The JSF representation of s_0 and s_1 will ensure that their joint Hamming weight is minimal among all signed binary representations of the same pair of integers. Therefore, we combined 2-dimensional GLV with JSF to make the scalar multiplication faster.

12.3.5 Applying Straus-Shamir Simultaneous Multi-scalar Multiplication Technique

In what follows let us denote the 2-dimension as 2-Split, 4-dimension as 4-Split and 8-dimension as 8-Split scalar multiplication. In our experimental implementation, we adopted the parameter suggested in [BD18]. Using [BD18]'s settings the integer u is obtained as 35-bit and order r as 263-bit. Therefore, the maximum bit length of an s is ≤ 263-bit.

12.3.5.1 2-Split and 4-Split scalar multiplication

The 2-Split scalar multiplication can be expressed as

$$[s]Q = [s_0]Q + s_1[u^4]Q. \tag{12.23}$$

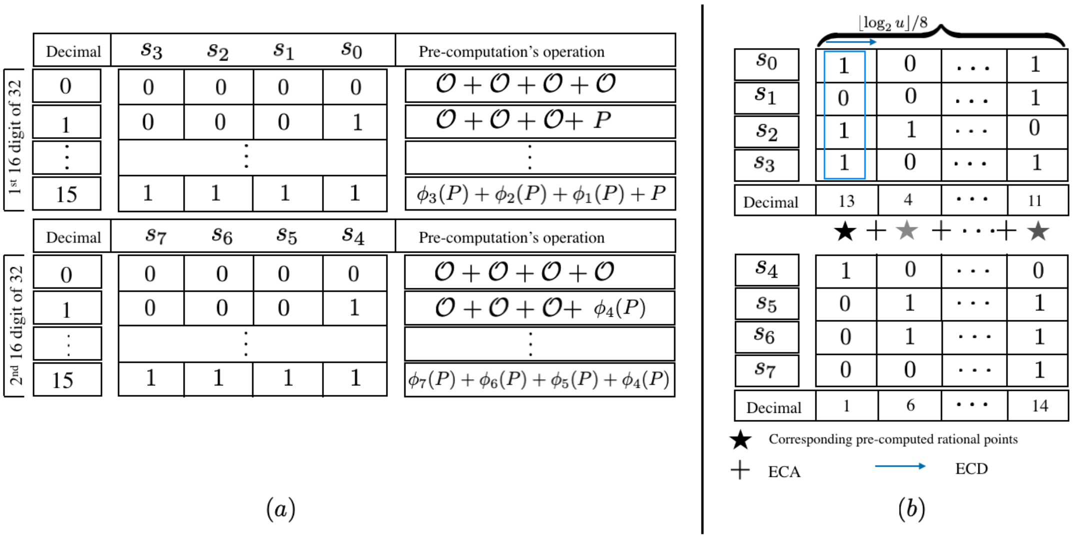
For the above representation, we need at most 2^2 pre-computed points and 2-bit (one for s_0 and another is s_1) simultaneous multi-scalar multiplication. Similarly, 4-Split can be calculated as

$$[s]Q = [s_0]Q + s_1[u^2]Q + s_2[u^4]Q + s_3[u^6]Q, \tag{12.24}$$

using 2^4 pre-computed rational point patterns applied in 4-bit (s_3, s_2, s_1, s_0) simultaneous multi-scalar multiplication.

12.3.5.2 8-Split scalar multiplication

The 8-Split multiplication can be a little bit tricky since the usual way will calculate 2^8 pre-computed points. Since u = 35-bit, the maximum length of the scalar after the dimension 8 decomposition will be ≤ 35-bit. Therefore, at most 35 pre-computed points will be utilized during the multi-scalar multiplication. As a result, we separated the scalar into two groups as (s_3, s_2, s_1, s_0) and (s_7, s_6, s_5, s_4). Then we pre-computed $2^4 + 2^4 = 32$ rational points. **Figure**. 12.1(a) shows the pre-computation steps. Among the 32 pre-computed points each of the points will be utilized at least once during multi-scalar multiplication. Finally, we combined the result of the two separately obtained multi-scalar multiplication by one extra elliptic curve addition. As a result we can save $2^8 - 32 = 224$ pre-computation. **Figure**. 12.1(b) shows the computation of the loop where simultaneous multi-scalar multiplications are carried out.



Decimal	s_3	s_2	s_1	s_0	Pre-computation's operation
0	0	0	0	0	$\mathcal{O}+\mathcal{O}+\mathcal{O}+\mathcal{O}$
1	0	0	0	1	$\mathcal{O}+\mathcal{O}+\mathcal{O}+P$
⋮	⋮				⋮
15	1	1	1	1	$\phi_3(P)+\phi_2(P)+\phi_1(P)+P$

Decimal	s_7	s_6	s_5	s_4	Pre-computation's operation
0	0	0	0	0	$\mathcal{O}+\mathcal{O}+\mathcal{O}+\mathcal{O}$
1	0	0	0	1	$\mathcal{O}+\mathcal{O}+\mathcal{O}+\phi_4(P)$
⋮	⋮				⋮
15	1	1	1	1	$\phi_7(P)+\phi_6(P)+\phi_5(P)+\phi_4(P)$

s_0	1	0	···	1
s_1	0	0	···	1
s_2	1	1	···	0
s_3	1	0	···	1
Decimal	13	4	···	11

s_4	1	0	···	0
s_5	0	1	···	1
s_6	0	1	···	1
s_7	0	0	···	1
Decimal	1	6	···	14

FIGURE 12.1: (a) Pre-computation of rational points for dimension 8 GLV. (b) Computation of SCM for dimension 8 GLV.

To obtain the pre-computed rational points we need to calculate $[p]Q, [p^2]Q, \cdots, [p^7]Q$ as shown in **Figure**. 12.1(a). Thanks to Frobenius map which can be calculated with a few multiplications in $\mathbb{F}_p$. Moreover, since rational points in $\mathbb{G}_2$ have isomorphic twisted points in $\mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$, therefore, skew Frobenius map [Sak+08] can be applied as shown in the **Section**. 12.3.6.

12.3.6 Skew Frobenius Map to Compute $[p]\bar{Q}'$

From the definition of $Q \in \mathbb{G}_2$, we recall that Q satisfies $[\pi_p - p]Q = \mathcal{O}$ or $\pi_p(Q) = [p]Q$, which is also applicable for $\bar{Q}'$. Applying skew Frobenius map we can optimize $[p]\bar{Q}'$ calculation. The detailed procedure to obtain the skew Frobenius map of $Q' = (x_{Q'}, y_{Q'}) \in \mathbb{G}_2' \subset E'(\mathbb{F}_{p^4})$ is given bellow:

$$(x_{Q'}\gamma)^p = (x_{Q'})^p\gamma^p.$$

After remapping

$$(x_{Q'})^p\gamma^{p-1} = (x_{Q'})^p(\gamma^2)^{\frac{p-1}{2}},$$

The $(\gamma^2)^{\frac{p-1}{2}}$ term can be simplified as follows:

$$\begin{aligned}(\gamma^2)^{\frac{p-1}{2}} &= (\beta^2)^{\frac{p-1}{4}}, \quad \text{since } p \equiv 5 \bmod 8,\\ &= (\alpha)^{\frac{p-1}{4}-1}\alpha,\\ &= (\alpha^2)^{\frac{p-5}{8}}\alpha,\\ &= c^{\frac{p-5}{8}}\alpha. \end{aligned} \tag{12.26a}$$

Recall that $c = 2$ in Eq.(12.5).

Similar way the skew Frobenius map of $y_{Q'}$ is given as,

$$(y_{Q'}\gamma\omega)^p = (y_{Q'})^p \gamma^p \omega^p.$$

After remapping

$$(y_{Q'})^p \gamma^{p-1} \omega^{p-1} \;=\; (y_{Q'})^p (\gamma^2)^{\frac{p-1}{2}} (\omega^2)^{\frac{p-1}{2}}.$$

$(\gamma^2)^{\frac{p-1}{2}}$ is calculated same as Eq.(12.26a). The $(\omega^2)^{\frac{p-1}{2}}$ term is calculated as follows:

$$\begin{aligned}
(\omega^2)^{\frac{p-1}{2}} &= (\gamma^2)^{\frac{p-1}{4}}, \quad \text{since } p \equiv 5 \bmod 8,\\
&= \beta^{\frac{p-1}{4}-1}\beta,\\
&= (\alpha)^{\frac{p-5}{8}}\beta,\\
&= (\alpha)^{\frac{p-5}{8}-1}\alpha\beta,\\
&= (\alpha^2)^{\frac{p-13}{16}}\alpha\beta,\\
&= c^{\frac{p-13}{16}}\alpha\beta.
\end{aligned}$$

The above constant terms will be pre-calculated. Now the $x_{Q'})^p, (y_{Q'})^p \in \mathbb{F}_{p^4}$ can be easily calculated where the coefficients will change positions and sign while multiplying with basis elements. For example $(x_{Q'})^p(\gamma^2)^{\frac{p-1}{2}} \in \mathbb{F}_{p^4}$ can be calculated as

$$\begin{aligned}
(x_{Q'})^p(\gamma^2)^{\frac{p-1}{2}} &= (a_0 + a_1\alpha + a_2\beta + a_3\alpha\beta)^p c^{\frac{p-5}{8}}\alpha,\\
&= (-a_1 c + a_0\alpha - a_3 c\beta + a_2\alpha\beta)c^{\frac{3p-7}{8}}.
\end{aligned}$$

Here it costs 4 multiplication in $\mathbb{F}_p$. In the similar way $(y_{Q'})^p(\gamma^2)^{\frac{p-1}{2}}(\omega^2)^{\frac{p-1}{2}}$ can be calculated in costing 4 M_p. Therefore, a single skew Frobenius map will cost 8 multiplications in $\mathbb{F}_p$.

During the pre-computation stage of GLV method we also need to compute $[p^2]Q'$, $[p^3]Q'$, $[p^4]Q'$, $[p^5]Q'$, $[p^6]Q'$, $[p^6]Q'$, and $[p^7]Q'$ skew Frobenius maps. The procedure is similar to computing $[p]Q'$. Interestingly, the coefficients basis positions after the skew Frobenius map is similar for $[p]Q'$ and $[p^5]Q'$ pair; $[p^3]Q$ and $[p^7]Q'$ pair, $[p^2]Q'$ and $[p^6]Q'$ pair. Only the constant multiples will be different.

12.4 Experimental Result Analysis

To determine the advantage of the derived GLV techniques, in one hand we applied the twisted mapping to map rational point $Q \in \mathbb{G}_2 \subset E(\mathbb{F}_{p^{16}})$ to its isomorphic point $Q' \in \mathbb{G}'_2 \subset E'(\mathbb{F}_{p^4})$. After that, we performed the scalar multiplication of Q'. Then the resulted points are re-mapped to $\mathbb{G}_2$ in $\mathbb{F}_{p^{16}}$. On the other hand, we performed scalar multiplication using the GLV techniques derived in **Section**. 12.3. In the experiment, 100 randomly generated scalars

TABLE 12.2: Curve parameters.

u = 35-bit	p	r	t
$2^{35} - 2^{32} - 2^{18} + 2^8 + 1$	339 -bit	263 -bit	270 -bit

TABLE 12.3: Experimental Implementation Environment.

CPU	Memory	Compiler	OS	Language & Library
Intel(R) 2.7 GHz Core(TM) i5	16GB	4.2.1	macOS High Sierra 10.13.6	C GMP v 6.1.0 [Gt15]

of size $\leq r$ (263-bit) are used to calculate SCM for all the cases. Average value of execution time presented in the millisecond is considered for comparison. The source of the experimental implementation can be found in Github [1].

In the experiment, KSS-16 curve over $\mathbb{F}_{p^{16}}$ is obtained as $y^2 = x^3 + 1$ by applying the parameters of Barbulescu et al. [BD18] for 128-bit security level. **Table** 12.3 shows the experiment environment used for comparative evaluation. No optimization is done to execute the program in multithreading.

Table 12.4 shows the maximum bit length after applying the GLV technique on a scalar of length 263-bit. **Table** 12.5 shows the number of operation required to perform single ECA and ECD in $E'(\mathbb{F}_{p^4})$. **Table** 12.6 shows the result with respect to ECA and ECD count and time [ms]. From the results, it is clear that 4-Split is the fastest among the techniques followed by the 8-Split. It is expected that 8-Split should be faster than the 4-Split since it's loop length is half of the 4-Split. In other words, 8-Split requires about less than half of 4-Split's ECD during loop execution. However, combining two 4-Split for one 8-Split increases the number of ECA. As a result, the total ECA count in the loop for 8-Split is almost same a 4-Split. The significant fall back of 8-Split compared to 4-Split comes from its number of precomputed rational points. Moreover, the total number of pre-computation also increases the other overhead calculations such as initialization, memory allocation, padding 0 in MSB of the decomposed scalar smaller than the max length. Which also impacts on the execution time.

12.5 Conclusion

This thesis shows the detailed formula to apply the GLV decomposition together with Straus-Shamir multi-scalar multiplication technique for efficient $\mathbb{G}_2$ scalar multiplication which is a major operation in many pairing-based protocols. The experimental implementation confirms the correctness of the derived technique. The comparative implementations show that dimension 4 is faster than 8 and 2. There is still scope to make the technique better by

[1] `https://github.com/eNipu/candar_glv.git`

TABLE 12.4: Maximum length of scalar s after GLV decomposition in different dimensions.

Max bit length of s after GLV	Normal binary	2-Split	2-Split JSF	4-Split	8-Split
	263-bit	139-bit	139-bit	69-bit	35-bit

TABLE 12.5: ECD and ECA cost in $E'(\mathbb{F}_{p^4})$.

ECD cost in $E'(\mathbb{F}_{p^4})$	ECA cost in $E'(\mathbb{F}_{p^4})$
$3M_4 + 8A_4 + 1I_4 + 1M_p$	$2M_4 + 6A_4 + 1I_4$

optimizing the pre-computation which will reduce the number of ECA and ECD. As a future work, the authors would like to reduce the pre-computation cost by optimizing Frobenius map calculation together with the application of non-adjacent form (NAF) and evaluate the acceleration in a pairing-based protocol.

ß

TABLE 12.6: Comparative result of average execution time in [ms] for scalar multiplication.

	Pre-computation		In SCM Algorithm		Time [ms]
Operation	#ECA	#ECD	#ECA	#ECD	
Normal binary	0	0	120	262	42.81
2-Split	5	6	98	138	28.48
2-Split JSF	8	6	66	138	25.16
4-Split	24	20	64	68	19.09
8-Split	52	47	67	34	21.85

Bibliography

[Ade+16] P. A. R. Ade et al. “Planck 2015 results. XIII. Cosmological parameters”. In: *Astron. Astrophys.* 594 (2016), A13. DOI: 10.1051/0004-6361/201525830. arXiv: 1502.01589 [astro-ph.CO].

[Ara+11] Diego F. Aranha, Koray Karabina, Patrick Longa, Catherine H. Gebotys, and Julio López. “Faster Explicit Formulas for Computing Pairings over Ordinary Curves”. In: *EUROCRYPT 2011*. Ed. by Kenneth G. Paterson. Vol. 6632. LNCS. Springer, Heidelberg, May 2011, pp. 48–68. DOI: 10.1007/978-3-642-20465-4_5.

[Ara+13] Diego F. Aranha, Laura Fuentes-Castañeda, Edward Knapp, Alfred Menezes, and Francisco Rodríguez-Henríquez. “Implementing Pairings at the 192-Bit Security Level”. In: *PAIRING 2012*. Ed. by Michel Abdalla and Tanja Lange. Vol. 7708. LNCS. Springer, Heidelberg, May 2013, pp. 177–195. DOI: 10.1007/978-3-642-36334-4_11.

[Bar+15] Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon. “Subgroup Security in Pairing-Based Cryptography”. In: *LATINCRYPT 2015*. Ed. by Kristin E. Lauter and Francisco Rodríguez-Henríquez. Vol. 9230. LNCS. Springer, Heidelberg, Aug. 2015, pp. 245–265. DOI: 10.1007/978-3-319-22174-8_14.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. “Short Group Signatures”. In: *CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 41–55. DOI: 10.1007/978-3-540-28628-8_3.

[BD17] Razvan Barbulescu and Sylvain Duquesne. *Updating key size estimations for pairings*. Cryptology ePrint Archive, Report 2017/334. http://eprint.iacr.org/2017/334. 2017.

[BD18] Razvan Barbulescu and Sylvain Duquesne. “Updating Key Size Estimations for Pairings”. In: *Journal of Cryptology* (2018). ISSN: 1432-1378. URL: https://doi.org/10.1007/s00145-018-9280-5.

[BF01] Dan Boneh and Matthew K. Franklin. “Identity-Based Encryption from the Weil Pairing”. In: *CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. LNCS. Springer, Heidelberg, Aug. 2001, pp. 213–229. DOI: 10.1007/3-540-44647-8_13.

[BGW05] Dan Boneh, Craig Gentry, and Brent Waters. "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys". In: *CRYPTO 2005*. Ed. by Victor Shoup. Vol. 3621. LNCS. Springer, Heidelberg, Aug. 2005, pp. 258–275. DOI: 10.1007/11535218_16.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. "Short Signatures from the Weil Pairing". In: *ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. LNCS. Springer, Heidelberg, Dec. 2001, pp. 514–532. DOI: 10.1007/3-540-45682-1_30.

[BLS03] Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. "Constructing Elliptic Curves with Prescribed Embedding Degrees". In: *SCN 02*. Ed. by Stelvio Cimato, Clemente Galdi, and Giuseppe Persiano. Vol. 2576. LNCS. Springer, Heidelberg, Sept. 2003, pp. 257–267. DOI: 10.1007/3-540-36413-7_19.

[BN06] Paulo S. L. M. Barreto and Michael Naehrig. "Pairing-Friendly Elliptic Curves of Prime Order". In: *SAC 2005*. Ed. by Bart Preneel and Stafford Tavares. Vol. 3897. LNCS. Springer, Heidelberg, Aug. 2006, pp. 319–331. DOI: 10.1007/11693383_22.

[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 (June 2001), pp. 153–176. DOI: 10.1007/s001450010012.

[BP98] Daniel V. Bailey and Christof Paar. "Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms". In: *CRYPTO'98*. Ed. by Hugo Krawczyk. Vol. 1462. LNCS. Springer, Heidelberg, Aug. 1998, pp. 472–485. DOI: 10.1007/BFb0055748.

[BS09] Naomi Benger and Michael Scott. *Constructing Tower Extensions for the implementation of Pairing-Based Cryptography*. Cryptology ePrint Archive, Report 2009/556. http://eprint.iacr.org/2009/556. 2009.

[CH07] Jaewook Chung and M Anwar Hasan. "Asymmetric squaring formulae". In: *Computer Arithmetic, 2007. ARITH'07. 18th IEEE Symposium on*. IEEE. 2007, pp. 113–122.

[CLN10] Craig Costello, Tanja Lange, and Michael Naehrig. "Faster Pairing Computations on Curves with High-Degree Twists". In: *PKC 2010*. Ed. by Phong Q. Nguyen and David Pointcheval. Vol. 6056. LNCS. Springer, Heidelberg, May 2010, pp. 224–242. DOI: 10.1007/978-3-642-13013-7_14.

[Coh+05] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren, eds. *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman and Hall/CRC, 2005. ISBN: 978-1-58488-518-4. DOI: 10.1201/9781420034981.

[DEM05] Régis Dupont, Andreas Enge, and François Morain. "Building Curves with Arbitrary Small MOV Degree over Finite Prime Fields". In: *Journal of Cryptology* 18.2 (Apr. 2005), pp. 79–89. DOI: 10.1007/s00145-004-0219-7.

[Dev+06] Augusto Jun Devegili, Colm Ó hÉigeartaigh, Michael Scott, and Ricardo Dahab. *Multiplication and Squaring on Pairing-Friendly Fields*. Cryptology ePrint Archive, Report 2006/471. http://eprint.iacr.org/2006/471. 2006.

[DH76] Whitfield Diffie and Martin E. Hellman. "New directions in cryptography". In: *IEEE Trans. Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. ISBN: 3-540-42580-2. DOI: 10.1007/978-3-662-04722-4.

[DSD07] Augusto Jun Devegili, Michael Scott, and Ricardo Dahab. "Implementing Cryptographic Pairings over Barreto-Naehrig Curves (Invited Talk)". In: *PAIRING 2007*. Ed. by Tsuyoshi Takagi, Tatsuaki Okamoto, Eiji Okamoto, and Takeshi Okamoto. Vol. 4575. LNCS. Springer, Heidelberg, July 2007, pp. 197–207. DOI: 10.1007/978-3-540-73489-5_10.

[Duq+15] Sylvain Duquesne, Nadia El Mrabet, Safia Haloui, and Franck Rondepierre. *Choosing and generating parameters for low level pairing implementation on BN curves*. Cryptology ePrint Archive, Report 2015/1212. http://eprint.iacr.org/2015/1212. 2015.

[FKR12] Laura Fuentes-Castañeda, Edward Knapp, and Francisco Rodríguez-Henríquez. "Faster Hashing to $\mathbb{G}_2$". In: *SAC 2011*. Ed. by Ali Miri and Serge Vaudenay. Vol. 7118. LNCS. Springer, Heidelberg, Aug. 2012, pp. 412–430. DOI: 10.1007/978-3-642-28496-0_25.

[FLS15] Armando Faz-Hernández, Patrick Longa, and Ana H. Sánchez. "Efficient and secure algorithms for GLV-based scalar multiplication and their implementation on GLV-GLS curves (extended version)". In: *J. Cryptographic Engineering* 5.1 (2015), pp. 31–52. DOI: 10.1007/s13389-014-0085-7.

[FST06] David Freeman, Michael Scott, and Edlyn Teske. *A taxonomy of pairing-friendly elliptic curves*. Cryptology ePrint Archive, Report 2006/372. http://eprint.iacr.org/2006/372. 2006.

[FST10] David Freeman, Michael Scott, and Edlyn Teske. “A Taxonomy of Pairing-Friendly Elliptic Curves”. In: *Journal of Cryptology* 23.2 (Apr. 2010), pp. 224–280. DOI: 10.1007/s00145-009-9048-z.

[GF16a] Loubna Ghammam and Emmanuel Fouotsa. *Adequate Elliptic Curve for Computing the Product of n Pairings*. Cryptology ePrint Archive, Report 2016/472. http://eprint.iacr.org/2016/472. 2016.

[GF16b] Loubna Ghammam and Emmanuel Fouotsa. *On the Computation of the Optimal Ate Pairing at the 192-bit Security Level*. Cryptology ePrint Archive, Report 2016/130. http://eprint.iacr.org/2016/130. 2016.

[GHP04] Peter J. Grabner, Clemens Heuberger, and Helmut Prodinger. “Distribution results for low-weight binary representations for pairs of integers”. In: *Theor. Comput. Sci.* 319.1-3 (2004), pp. 307–331. DOI: 10.1016/j.tcs.2004.02.012.

[GLS11] Steven D. Galbraith, Xibin Lin, and Michael Scott. “Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves”. In: *Journal of Cryptology* 24.3 (July 2011), pp. 446–469. DOI: 10.1007/s00145-010-9065-y.

[GLV01] Robert P. Gallant, Robert J. Lambert, and Scott A. Vanstone. “Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms”. In: *CRYPTO 2001*. Ed. by Joe Kilian. Vol. 2139. LNCS. Springer, Heidelberg, Aug. 2001, pp. 190–200. DOI: 10.1007/3-540-44647-8_11.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. “Pairings for cryptographers”. In: *Discrete Applied Mathematics* 156.16 (2008), pp. 3113–3121. DOI: 10.1016/j.dam.2007.12.010.

[Gre+13] Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. “Efficient Implementation of Bilinear Pairings on ARM Processors”. In: *SAC 2012*. Ed. by Lars R. Knudsen and Huapeng Wu. Vol. 7707. LNCS. Springer, Heidelberg, Aug. 2013, pp. 149–165. DOI: 10.1007/978-3-642-35999-6_11.

[Gro10] Jens Groth. “Short Pairing-Based Non-interactive Zero-Knowledge Arguments”. In: *ASIACRYPT 2010*. Ed. by Masayuki Abe. Vol. 6477. LNCS. Springer, Heidelberg, Dec. 2010, pp. 321–340. DOI: 10.1007/978-3-642-17373-8_19.

[GS10] Robert Granger and Michael Scott. “Faster Squaring in the Cyclotomic Subgroup of Sixth Degree Extensions”. In: *PKC 2010*. Ed. by Phong Q. Nguyen and David Pointcheval. Vol. 6056. LNCS. Springer, Heidelberg, May 2010, pp. 209–223. DOI: 10.1007/978-3-642-13013-7_13.

[Gt15] Torbjörn Granlund and the GMP development team. *GNU MP: The GNU Multiple Precision Arithmetic Library*. 6.1.0. http://gmplib.org. 2015.

[Hes08] Florian Hess. "Pairing Lattices (Invited Talk)". In: *PAIRING 2008*. Ed. by Steven D. Galbraith and Kenneth G. Paterson. Vol. 5209. LNCS. Springer, Heidelberg, Sept. 2008, pp. 18–38. DOI: 10.1007/978-3-540-85538-5_2.

[HSV06] F. Hess, N. P. Smart, and F. Vercauteren. "The Eta Pairing Revisited". In: *IEEE Transactions on Information Theory* 52.10 (2006), pp. 4595–4602. ISSN: 0018-9448. DOI: 10.1109/TIT.2006.881709.

[Jou04] Antoine Joux. "A One Round Protocol for Tripartite Diffie-Hellman". In: *Journal of Cryptology* 17.4 (Sept. 2004), pp. 263–276. DOI: 10.1007/s00145-004-0312-y.

[Kar13a] Koray Karabina. "Squaring in cyclotomic subgroups". In: *Math. Comput.* 82.281 (2013), pp. 555–579. DOI: 10.1090/S0025-5718-2012-02625-1.

[Kar13b] Koray Karabina. "Squaring in cyclotomic subgroups". In: *Math. Comput.* 82.281 (2013), pp. 555–579. DOI: 10.1090/S0025-5718-2012-02625-1.

[Kat+07] Hidehiro Kato, Yasuyuki Nogami, Tomoki Yoshida, and Yoshitaka Morikawa. "Cyclic Vector Multiplication Algorithm Based on a Special Class of Gauss Period Normal Basis". In: *ETRI Journal* 29.6 (2007), pp. 769–778. DOI: 10.4218/etrij.07.0107.0040.

[KB16] Taechan Kim and Razvan Barbulescu. "Extended Tower Number Field Sieve: A New Complexity for the Medium Prime Case". In: *CRYPTO 2016, Part I*. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9814. LNCS. Springer, Heidelberg, Aug. 2016, pp. 543–571. DOI: 10.1007/978-3-662-53018-4_20.

[Kha+17a] Md. Al-Amin Khandaker, Hirotaka Ono, Yasuyuki Nogami, Masaaki Shirase, and Sylvain Duquesne. "An Improvement of Optimal Ate Pairing on KSS Curve with Pseudo 12-Sparse Multiplication". In: *ICISC 16*. Ed. by Seokhie Hong and Jong Hwan Park. Vol. 10157. LNCS. Springer, Heidelberg, 2017, pp. 208–219. DOI: 10.1007/978-3-319-53177-9_11.

[Kha+17b] Md. Al-Amin Khandaker, Yuki Nanjo, Loubna Ghammam, Sylvain Duquesne, Yasuyuki Nogami, and Yuta Kodera. "Efficient Optimal Ate Pairing at 128-Bit Security Level". In: *INDOCRYPT 2017*. Ed. by Arpita Patra and Nigel P. Smart. Vol. 10698. LNCS. Springer, Heidelberg, Dec. 2017, pp. 186–205.

[KN17] Md. Al-Amin Khandaker and Yasuyuki Nogami. “An Improvement of Scalar Multiplication by Skew Frobenius Map with Multi-Scalar Multiplication for KSS Curve”. In: *IEICE Transactions* 100-A.9 (2017), pp. 1838–1845. DOI: 10.1587/transfun.E100.A.1838.

[KO62] A Karatsuba and Y Ofman. “Multiplication of many-digital numbers by automatic computers”. In: *DOKLADY AKADEMII NAUK SSSR* 145.2 (1962), p. 293.

[Kob87] Neal Koblitz. “Elliptic curve cryptosystems”. In: *Mathematics of computation* 48.177 (1987), pp. 203–209. DOI: 10.1090/S0025-5718-1987-0866109-5.

[Kob92] Neal Koblitz. “CM-Curves with Good Cryptographic Properties”. In: *CRYPTO’91*. Ed. by Joan Feigenbaum. Vol. 576. LNCS. Springer, Heidelberg, Aug. 1992, pp. 279–287. DOI: 10.1007/3-540-46766-1_22.

[Koc96] Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems”. In: *CRYPTO’96*. Ed. by Neal Koblitz. Vol. 1109. LNCS. Springer, Heidelberg, Aug. 1996, pp. 104–113. DOI: 10.1007/3-540-68697-5_9.

[KSS07] Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. *Constructing Brezing-Weng pairing friendly elliptic curves using elements in the cyclotomic field*. Cryptology ePrint Archive, Report 2007/452. http://eprint.iacr.org/2007/452. 2007.

[Lan08] Hoes Lane. “Draft standard for identity-based public key cryptography using pairings”. In: *IEEE P1636* 3 (2008), p. D1.

[LL97] Chae Hoon Lim and Pil Joong Lee. “A Key Recovery Attack on Discrete Log-based Schemes Using a Prime Order Subgroup”. In: *CRYPTO’97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, Heidelberg, Aug. 1997, pp. 249–263. DOI: 10.1007/BFb0052240.

[LLP09] E. Lee, H.-S. Lee, and C.-M. Park. “Efficient and Generalized Pairing Computation on Abelian Varieties”. In: *IEEE Trans. Information Theory* 55.4 (2009), pp. 1793–1803. DOI: 10.1109/TIT.2009.2013048.

[LN96] Rudolf Lidl and Harald Niederreiter. *Finite Fields*. 2nd ed. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1996. DOI: 10.1017/CBO9780511525926.

[Mat+07] Seiichi Matsuda, Naoki Kanayama, Florian Hess, and Eiji Okamoto. *Optimised versions of the Ate and Twisted Ate Pairings*. Cryptology ePrint Archive, Report 2007/013. http://eprint.iacr.org/2007/013. 2007.

[Mil86] Victor S. Miller. "Use of Elliptic Curves in Cryptography". In: *CRYPTO'85*. Ed. by Hugh C. Williams. Vol. 218. LNCS. Springer, Heidelberg, Aug. 1986, pp. 417–426. DOI: 10.1007/3-540-39799-X_31.

[Mon87] Peter L. Montgomery. "Speeding the Pollard and elliptic curve methods of factorization". In: *Math. Comp.* 48.177 (1987), pp. 243–264. ISSN: 0025-5718. DOI: 10.2307/2007888.

[Mor+14] Yuki Mori, Shoichi Akagi, Yasuyuki Nogami, and Masaaki Shirase. "Pseudo 8-Sparse Multiplication for Efficient Ate-Based Pairing on Barreto-Naehrig Curve". In: *PAIRING 2013*. Ed. by Zhenfu Cao and Fangguo Zhang. Vol. 8365. LNCS. Springer, Heidelberg, Nov. 2014, pp. 186–198. DOI: 10.1007/978-3-319-04873-4_11.

[MP13] Gary L. Mullen and Daniel Panario. *Handbook of Finite Fields*. 1st. Chapman & Hall/CRC, 2013. ISBN: 143987378X, 9781439873786.

[NF05] Toru Nakanishi and Nobuo Funabiki. "Verifier-Local Revocation Group Signature Schemes with Backward Unlinkability from Bilinear Maps". In: *ASIACRYPT 2005*. Ed. by Bimal K. Roy. Vol. 3788. LNCS. Springer, Heidelberg, Dec. 2005, pp. 533–548. DOI: 10.1007/11593447_29.

[Nog+08] Yasuyuki Nogami, Masataka Akane, Yumi Sakemi, Hidehiro Katou, and Yoshitaka Morikawa. "Integer Variable chi-Based Ate Pairing". In: *PAIRING 2008*. Ed. by Steven D. Galbraith and Kenneth G. Paterson. Vol. 5209. LNCS. Springer, Heidelberg, Sept. 2008, pp. 178–191. DOI: 10.1007/978-3-540-85538-5_13.

[Nog+09] Yasuyuki Nogami, Yumi Sakemi, Takumi Okimoto, Kenta Nekado, Masataka Akane, and Yoshitaka Morikawa. "Scalar Multiplication Using Frobenius Expansion over Twisted Elliptic Curve for Ate Pairing Based Cryptography". In: *IEICE Transactions* 92-A.1 (2009), pp. 182–189. DOI: 10.1587/transfun.E92.A.182.

[OT10] Tatsuaki Okamoto and Katsuyuki Takashima. "Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption". In: *CRYPTO 2010*. Ed. by Tal Rabin. Vol. 6223. LNCS. Springer, Heidelberg, Aug. 2010, pp. 191–208. DOI: 10.1007/978-3-642-14623-7_11.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems". In: *Commun. ACM* 21.2 (1978), pp. 120–126. DOI: 10.1145/359340.359342.

[Sak00] Ryuichi Sakai. "Cryptosystems based on pairing". In: *The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, Jan.* 2000, pp. 26–28.

[Sak+08] Yumi Sakemi, Yasuyuki Nogami, Katsuyuki Okeya, Hidehiro Katou, and Yoshitaka Morikawa. "Skew Frobenius Map and Efficient Scalar Multiplication for Pairing-Based Cryptography". In: *CANS 08*. Ed. by Matthew K. Franklin, Lucas Chi Kwong Hui, and Duncan S. Wong. Vol. 5339. LNCS. Springer, Heidelberg, Dec. 2008, pp. 226–239.

[San+16] Akihito Sanada, Duquesne Sylvain, Masaaki Shirase, and Yasuyuki Nogami. *A Consideration of an Efficient Calculation over the Extension Field of Degree 4 for Elliptic Curve Pairing Cryptography*. 2016. URL: http://www.ieice.org/ken/paper/20160729yb97/eng/.

[SB04] Michael Scott and Paulo S. L. M. Barreto. "Compressed Pairings". In: *CRYPTO 2004*. Ed. by Matthew Franklin. Vol. 3152. LNCS. Springer, Heidelberg, Aug. 2004, pp. 140–156. DOI: 10.1007/978-3-540-28628-8_9.

[SCA86] Joseph H Silverman, Gary Cornell, and M Artin. *Arithmetic geometry*. Springer, 1986.

[Sch10] Oliver Schirokauer. "The number field sieve for integers of low weight". In: *Math. Comput.* 79.269 (2010), pp. 583–602. DOI: 10.1090/S0025-5718-09-02198-X.

[Sco+09] Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa. "On the Final Exponentiation for Calculating Pairings on Ordinary Elliptic Curves". In: *PAIRING 2009*. Ed. by Hovav Shacham and Brent Waters. Vol. 5671. LNCS. Springer, Heidelberg, Aug. 2009, pp. 78–88. DOI: 10.1007/978-3-642-03298-1_6.

[Sco11] Michael Scott. "On the Efficient Implementation of Pairing-Based Protocols". In: *13th IMA International Conference on Cryptography and Coding*. Ed. by Liqun Chen. Vol. 7089. LNCS. Springer, Heidelberg, Dec. 2011, pp. 296–308.

[Sha84] Adi Shamir. "Identity-Based Cryptosystems and Signature Schemes". In: *CRYPTO'84*. Ed. by G. R. Blakley and David Chaum. Vol. 196. LNCS. Springer, Heidelberg, Aug. 1984, pp. 47–53.

[SK03] Ryuichi Sakai and Masao Kasahara. *ID based Cryptosystems with Pairing on Elliptic Curve*. Cryptology ePrint Archive, Report 2003/054. http://eprint.iacr.org/2003/054. 2003.

[SL03] Martijn Stam and Arjen K. Lenstra. "Efficient Subgroup Exponentiation in Quadratic and Sixth Degree Extensions". In: *CHES 2002*.

Ed. by Burton S. Kaliski Jr., Çetin Kaya Koç, and Christof Paar. Vol. 2523. LNCS. Springer, Heidelberg, Aug. 2003, pp. 318–332. DOI: `10.1007/3-540-36400-5_24`.

[STO06] Masaaki Shirase, Tsuyoshi Takagi, and Eiji Okamoto. *Some Efficient Algorithms for the Final Exponentiation of η_T Pairing*. Cryptology ePrint Archive, Report 2006/431. `http://eprint.iacr.org/2006/431`. 2006.

[Ver10] Frederik Vercauteren. "Optimal pairings". In: *IEEE Trans. Information Theory* 56.1 (2010), pp. 455–461. DOI: `10.1109/TIT.2009.2034881`.

[Was03] Lawrence Washington. *Elliptic curves : number theory and cryptography*. Chapman & Hall/CRC, 2003. ISBN: 9780203484029.

[Wei+49] André Weil et al. "Numbers of solutions of equations in finite fields". In: *Bull. Amer. Math. Soc* 55.5 (1949), pp. 497–508.

[ZL12] Xusheng Zhang and Dongdai Lin. "Analysis of Optimum Pairing Products at High Security Levels". In: *INDOCRYPT 2012*. Ed. by Steven D. Galbraith and Mridul Nandi. Vol. 7668. LNCS. Springer, Heidelberg, Dec. 2012, pp. 412–430. DOI: `10.1007/978-3-642-34931-7_24`.

Index

Biography

Md. Al-Amin Khandaker was born on September 11, 1990, in a beautiful village of Bangladesh. He completed his high school in 2007 and in 2008, admitted to Jahangirnagar University, Bangladesh. In 2012, he graduated majoring in Computer Science and Engineering. After that, he joined in a Holland-based off-shore software development company in Dhaka. In 2015 he awarded Japan Govt. Scholarship (MEXT) to pursue Doctor's course in the field of cryptography in Okayama University under the supervision of Professor Yasuyuki NOGAMI. His main fields of research are optimization and efficient implementation techniques for the elliptic curve, pairing-based cryptography and it's application for IoT security. He is a graduate student member of IEEE.