2016

Efficient Computation For Hyper Elliptic Curve Based Cryptography

Raqib Ahmed Asif
University of Windsor

EFFICIENT COMPUTATION FOR HYPER ELLIPTIC CURVE BASED CRYPTOGRAPHY

by

RAQIB AHMED ASIF

A Thesis
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering in
Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science
at the University of Windsor

Windsor, Ontario, Canada
2016

EFFICIENT COMPUTATION FOR HYPER ELLIPTIC CURVE BASED CRYPTOGRAPHY

by

RAQIB AHMED ASIF

APPROVED BY:

_____

Dr. Dan Wu, Outside Department Reader
School of Computer Science

_____

Dr. Rashid Rashidzadeh, Department Reader
Department of Electrical and Computer Engineering

_____

Dr. Huapeng Wu, Advisor
Department of Electrical and Computer Engineering

25 April, 2016

Author’s Declaration of Originality

I hereby certify that I am sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any propriety rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

In this thesis we have proposed explicit formulae for group operation such as addition and doubling on the Jacobians of Hyper Elliptic Curves genus 2, 3 and 4. The Cantor Algorithm generally involves to perform arithmetic operations in the polynomial ring $\mathbb{F}_q[x]$. The explicit method performs the arithmetic operation in the integer ring of $\mathbb{F}_q$. Significant improvement has been made in the explicit formulae algorithm proposed here. Other explicit formulae used Montgomery trick to derive efficient formulae for faster group computation. The method used in this thesis to develop an efficient explicit formula was inspired by the geometric properties in the hyper elliptic curves of genus and by keeping the Jacobian variety curve constant. This formulae take Mumford coordinates as input. The explicit formulae here performs the computation in affine space of genus 2, 3 and 4 of Hyper Elliptic Curves in general form, which can be used to develop Hyper Elliptic Curve Cryptosystem.

Key Words: Hyper Elliptic Curve, Hyper Elliptic Curve Cryptosystem, Jacobian Curve, genus 2, genus 3, genus 4.

Dedication

To my mother (Mrs. Israt Jahan) and father (Mr. Iqbal Ahmed), for nurturing and raising me to the place where I stand.

To my wife (Mrs. Minnatul Fatema), for her companionship and love.

To my sister (Ms, Raqima Tahseen Raisa), for her support.

Acknowledgements

I would like to thank my supervisor Dr. Huapeng Wu, Professor of Electrical and Computer Engineering at University of Windsor, for introducing me to the world of cyber – security and cryptography. His broad knowledge and logical way of thinking have been of great value; without his detailed and constructive comments on my research, none of this thesis would be possible.

Finally, I wish to extend my gratitude to everyone at the UWindsor's Faculty of ECE, for their efforts during my study in the MASc. Program.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

GF	Finite Field or Galois Field
EC	Elliptic Curve
ECC	Elliptic Curve Cryptography
HEC	Hyper Elliptic Curve
RSA	Rivest, Shamir, Adleman
DLP	Discrete Logarithm Problem
ECDLP	Elliptic Curve Discrete Logarithm Problem
HECDLP	Hyper Elliptic Curve Discrete Logarithm Problem
DH	Diffie – Hellman
PKI	Public Key Infrastructure
CA	Cantor Algorithm
SA	Subexpression Algorithm
JC	Jacobian Curve
$\mathbb{R}$	Real Field
$\mathbb{F}_q$	Prime field

Chapter 1

Introduction

Cybersecurity plays a very important role in our daily lives. For example, we would like to protect our data in our personal electronic devices, such as laptop and smart phone. To have a secure communication over the unsafe internet, using communication application such as Skype, WhatsApp and many more. To protect out data in cloud storage services offered by Dropbox, Google Drive and etc. or we would like to do online banking services and paying bills, or we purchase online and enjoy Electronic Commerce services offered by Amazon, EBay, Alibaba and many more. All those are made possible with the implementation and improvement of cybersecurity. The field of cryptography, as the core technology to achieve cybersecurity for the things mentioned above, can provide crucial security services such as Privacy, Authentication, Key Establishment and Data Integrity.

Higher security strength, faster implementation and low power consumption, this is what we are after in the area of cryptography engineering. There are already many cryptographic algorithms available which are able to satisfy these requirements. However, the new communication gadgets are smaller in size, which has very limited processing power and storage. The key size of the RSA [9] is quite large and RSA implementation in small devices takes long processing time and consumes a lot power. So cryptosystems that use smaller key size are favored in practice, such like those rely on the discrete logarithm problem over multiplicative group of elliptic curve defined in finite fields.

Koblitz [1] introduced Elliptic Curve Cryptography (ECC) in 1987. It is based on the discrete logarithm problem over the abelian group of points of the curve. The group law over the curve makes the operation fast and easy to compute. The advantages of using ECC is its key size are smaller than RSA and also there is no sub exponential algorithms for Elliptic Curve Discreet Logarithm Problem (ECDLP).

Elliptic Curve Cryptography (ECC) can provide the same level of security as RSA or discreet logarithm problem (DLP) based systems such as Diffie Hellman Key Exchange (DHKE) and ElGamal public key cryptosystem at much smaller key size. On the hand, the complexity of the mathematics of the elliptic curves are more involved than those of the RSA and DLP based systems. Hyper Elliptic Curve Cryptography (HECC) is one of the late members of the established public-key algorithms: DHKE, RSA, ElGamal and ECC [6], [7], [8] [10].

In 1989, Koblitz [2] introduced discrete logarithm problem on hyper elliptic curves (HEC) and the cryptosystem constructed over the Jacobian of Hyper elliptic curves and based on this hard problem. This research subject is called hyper elliptic curve cryptography (HECC). Note that hyper elliptic curves can also be viewed as a special type of elliptic curves with genus ≥ 2. The advantage of using HECC is the smaller key size for the same level of security, even compared to ECC. Moreover, it has no sub exponential algorithms to solve HEC DLP, similar to that for ECC. The smaller size of the base field also makes hyper elliptic curves a good choice for the light weight cryptosystems.

Hyper Elliptic Curve Cryptography (HECC) offers theoretically higher level of security than all the established public key cryptosystem [5]. This is due to the high level of mathematical complexity even compare to Elliptic Curve Cryptosystems with the same key lengths size. In this thesis the mathematical background of HECC is discussed in detail and efficient methods for performing group operation are studied.

In hyper elliptic curve cryptosystem, the group law includes addition and doubling in the Jacobian of the curve. The algorithm for the group operation was given by the Cantor [3]. Since then there have many improvements on efficient computation of group operations and also very active research works in the field of HECC. One of the earliest attempt made to efficient algorithm for group operation for HECC was obtained by Harley [11]. The Harley's algorithm is an explicit representation of the Cantor Algorithm [3]. Later works presented more efficient algorithm for performing group operations we done by Lange [12], Matsuo, Chao and Tsujii [13], Miyamoto, Doi, Matsuo, Chao and Tsujii [14] and Takahashi [15].

In hyper elliptic curve the algorithms for group operation is not very fast for high genus compared to that for elliptic curve. There are faster algorithms for the elliptic curves or hyper elliptic curves with genus =

1. For the curves with larger genus, the existing algorithms for group operation are still relatively difficult to perform. It is a challenging task to develop faster algorithms for the group operation, which makes the study of HECC interesting.

1.1 Objective

The objective of the thesis can be listed as follow:

1. Understand the group laws of Hyper Elliptic Curve over the finite field.
2. Explain and discuss Mumford Representation of the intersecting Cartesian points between the Jacobian Variety Curve and Hyper Elliptic Curve over the finite field.
3. Discuss the group operations in Cantor Algorithm and Subexpression Algorithm for point addition and doubling.
4. Develop an explicit formulae algorithm for efficient group operation such as addition and doubling.

1.2 Overview

The following is an outline of the rest of the thesis.

Chapter 2: Mathematical Fundamentals.

In this chapter we will discuss the basic abstract algebra, such as the definitions of groups, Abelian group, subgroup, Homomorphism, Kernel, Rings, Polynomial Rings, Fields, Field Extension and etc. Later in this chapter, we will discuss the basic properties of Elliptic and Hyper Elliptic Curves. The fundamental difference between Elliptic and Hyper Elliptic Curve with pictorial examples.

Chapter 3: Hyper Elliptic Curve Cryptography.

In this chapter we present the necessary definitions and methods required in the later chapters. It provides a description of group operations, group order in Elliptic Curve Cryptography and Hyper Elliptic Curve

Cryptography. The group operation such as addition and scalar multiplication (generally better known as doubling) is very fundamental. Here we discuss in detail with example on how to convert the Cartesian points on the hyper elliptic curve over a field to Mumford Representation, which is polynomial representation of the co-ordinates. Definitions like Divisor Class, Divisor Class Group would be discussed.

Chapter 4: An overview of Hyper Elliptic Curve Computational Method.

In this chapter, we present how we can perform group operation such as addition and doubling by applying Cantor Algorithm. Later in the same section we intended to make it clear by presenting an example on how to apply this algorithm, the advantages and disadvantages of using Cantor Algorithm. Here we also discuss another method for performing group operation: Subexpression Algorithm. We try to make it clear with an example and its limitation (such as it advantages and disadvantages). Later in the same chapter we introduces efficient method for computing group operation with Explicit Formulae Algorithm. Its benefit and its limitation.

Chapter 5: Proposed efficient computation for Hyper Elliptic Curve Cryptography.

In this chapter, we will present with an efficient algorithm for group operation. Later in this chapter we discuss theorems and proposition used to build an efficient explicit formulae algorithm. Separate explicit formula algorithm for addition and doubling for specific hyper elliptic curve with different number of genus.

Chapter 6: Discussions and possible future works.

In this thesis we have proposed efficient explicit formulae with less complexity for group operations for the Hyper Elliptic Curves of genus 2, 3 and 4. The same procedure used here can be expanded to the hyper elliptic curves of great number of genus. In the future works, hardware implementation of HEC cryptographic system with the proposed efficient explicit algorithm for Hyper Elliptic Curve – Public Key Infrastructure (HEC – PKI) and Hyper Elliptic Curve – Digital Signature (HEC - DS).

Chapter 2

Mathematical Background

2.1 Elementary Algebraic Background

There are good reference book for the study of basic abstract algebra. The two books used for references are by Gallian [16], Herstein [17] and for the field theory is by Roman [18]. The area of study for abstract algebra is vast and to give a concise background is very difficult. The three books I have mentioned above is a good place to start. The definitions I give in abstract algebra which will be useful for the study of hyperelliptic curves.

2.1.1 Groups

Definition 1 (Law of Composition)

A law of composition on a set G is a rule for performing an operation between any two elements in the set G, let it be a and b. The result of the operation, let it be p. Where p is also an element of the set G.

Definition 2 (Group)

A group is a set G together with the law of composition under this operation if the following three properties are satisfied.

1. Associativity: The operation is said to be associative; that is $(a \circ b) \circ c = a \circ (b \circ c)$ for all $a, b, c \in G$.
2. Identity: There is an element $e \in G$ (called the identity element) such that $a \circ e = a$ for all $a \in G$.

3. Inverse: For every element $a \in G$, there is an element $b \in G$ (called an inverse of a) such that $a \circ b = b \circ a = e$

Definition 3 (Abelian/Commutative Group)

A group G is called an Abelian group if and only if $a, b \in G$ where $a \circ b = b \circ a$ for all elements in the group G.

Definition 4 (Subgroup)

A subset H of a group G is called a subgroup of G containing the identity element e and such that it all satisfies the all the properties of a group.

1. For all $a, b \in H$, $a \circ b = c$, where $c \in H$
2. If $c \in H$ then $c^{-1} \in H$

Definition 5 (Cyclic Group)

A group G is called cyclic if there is an element $a \in G$ such that $G = \{a^n | \, n \in Z\}$. Such an element is called a generator of G. We can represent the cyclic nature of G as $G =< a >$.

Definition 6 (Order of a Group)

The number of elements of a group is called the order. If the group is finite, then the group is called a finite group. $|G|$ is denoted as the order of the group G.

Definition 7 (Order of an Element)

The order of an element a in a group is the smallest positive integer n such that $a^n = e$. The order of an element a is denoted as $|g|$. The element g has infinite order if no such integer n exists.

Definition 8 (Equivalence Relation)

An equivalence relation on a set G is a set R of ordered pairs of elements of G such that

1. $(a, a) \in R$ for all $a \in G$ (reflexive property).
2. $(a, b) \in R$ implies $(b, a) \in R$ (symmetric property).
3. $(a, b) \in R$ and $(b, c) \in R$ imply $(a, c) \in R$ (transitive property).

Definition 9 (Cosets)

If H is a subgroup of the group G, and an element g of the group G. Then $gH = \{gh|h$ is an element of $H\}$ is called the left coset of H in G, $Hg = \{hg|h$ is an element of $H\}$ is called the right coset of H in G.

Properties of Cosets:

Let H be a subgroup of G, where the element a and $b \in G$. Then,

1. $a \in aH$.
2. $aH = H$ iff $a \in H$.
3. $(ab)H = a(bH)$ and $H(ab) = (Ha)b$.
4. $aH = bH$ iff $a \in bH$.
5. $aH = bH$ or$aH \cap bH = \emptyset$.
6. $aH = bH$ iff $a^{-1}b \in H$.
7. $|aH| = |bH|$.
8. $aH = Ha$ iff $H = aHa^{-1}$.
9. $aH \subset G$, iff $a \in H$.

Theorem 10 (Lagrange)

If the group G is a finite group and H be a subgroup of G. Then the order of H divides the order of G. Which makes the order of every element also divide the order of G.

Definition 11 (Homomorphism)

If $(G,\cdot)$ and $(H,\circ)$ are two groups, a homomorphism from G to H is a function $\varphi: G \to H$ that satisfies, for all $a, b \in G$.

$$\varphi(a \cdot b) = \varphi(a) \circ \varphi(b)$$

If φ is a one – to – one (INJECTIVE) → monomorphism.
If φ is a onto (SURJECTIVE) → epimorphism.
If φ is a bijective (BIJECTIVE) → isomorphism.

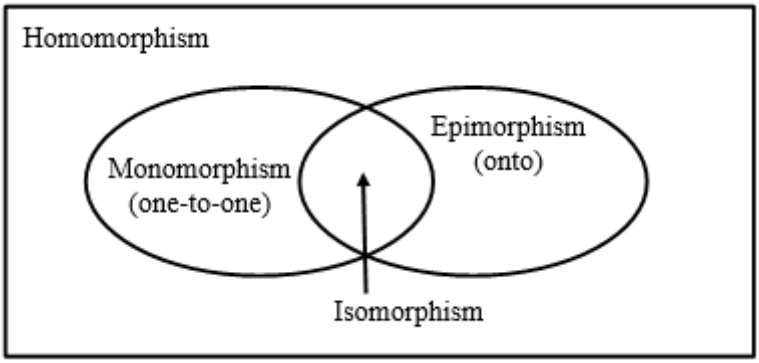

Figure 2.1: Venn diagram Representation on types of Homomorphism

Definition 12 (Kernel)

If φ is a homomorphism from the group $G \to G'$, then the kernel of φ is defined by $k\varphi = \{a \in G | \varphi(a) = e'\}$, and e' is the identity element of the group G'.

Definition 13 (Normal Subgroup)

A subgroup H of G is normal in G iff $aHa^{-1} \subseteq H$ for all the element $a \in G$.

Definition 14 (Quotient Group)

The subgroup H of G is normal, then the set of left (or right) cosets of H in G is by itself is a group – called the factor group (or quotient group) of G by H. Let G be a group and let H be a normal subgroup of G. The set $G/H = \{aH \mid a \in G\}$ is a group under the operation $(aH)(bH) = abH$.

2.1.2 Rings

Definition 15 (Rings)

A set R is said to form a ring with respect to the binary operations addition (+) and multiplication (·) provided the elements $a, b, c \in R$ holds the following properties.

Properties 16 (Rings)

1. Associative law of addition: $(a + b) + c = a + (b + c)$
2. Commutative law of addition: $a + b = b + a$
3. Presence of additive identity: there exists $z \in R$ such that $a + z = a$
4. Presence of additive inverse: for every element $a \in R$, there exists $-a \in R$ such that $a + (-a) = z$.
5. Associative law of multiplication: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
6. Distributive law: $a \cdot (b + c) = a \cdot b + a \cdot c$ or $(b + c) \cdot a = b \cdot a + c \cdot a$

Definition 17 (Subrings)

Let R be a ring and S is the non – empty subset of R, which itself is a ring with respect to the binary operations on R, is called a subring of R.

Definition 18 (Commutative Ring)

A ring for which multiplication is commutative is called a commutative ring.

For all the elements in the ring R. $a, b \in R$.

$a \cdot b = b \cdot a$ (Commutative law of multiplication)

Definition 19 (Ring with identity element or ring with unity)

A ring with a multiplicative identity is called a ring with identity element i or ring with unity

For all the elements in the ring R. $a, a^{-1} \in R$.

$a.a^{-1} = i$ (Multiplicative identity element)

Definition 20 (Zero - Divisors)

Let R be a ring and let $a, b \in R$ such that $a \neq 0$ and $b \neq 0$. If $ab = 0$, then a and b are called Zero – Divisors.

Definition 21 (Integral Domain)

An integral domain is a commutative ring R with unity 1 (assuming $1 \neq 0$) and no zero – divisor.

Definition 22 (Ideals)

Let R be a ring, a non – empty subset A of R is called an Ideal ring. For a ring to be an Ideal the conditions below has to be fulfilled.

Conditions:

1. A is an additive subgroup of R.
2. For every elements, $r \in R$ and $a \in A$. $rA = \{ra | a \in A\} \subseteq A$ and $Ar = \{ar | a \in A\} \subseteq A$ for all $r \in R$.

Definition 23 (Principal Ideals)

If I is an ideal of the ring R such that the ring I is generated by one element. That is $I =< a >$ for some $a \in R$, then I is said to be a principle ideal of R.

Definition 24 (Prime Ideals)

An ideal I if a ring R is called a prime ideal if $ab \in I$ implies either $a \in I$ or $b \in I$.

Definition 25 (Maximal Ideals)

Let I be an ideal of a ring R with $I \neq R$. Then I is called a maximal ideal of R if there exists an ideal J of R with $I \subset J \subset R$ and $I \neq J \neq R$.

Definition 26 (Ring Homomorphism)

A ring homomorphism φ from a ring R to a ring S is a mapping from R to S that preserves the two ring operations; that is, for all a, b in R

$$\varphi(a+b) = \varphi(a) + \varphi(b) \quad \text{and} \quad \varphi(ab) = \varphi(a)\varphi(b)$$

A ring homomorphism that is both one – to – one and onto is called a ring isomorphism. [16]

Properties 27 (Ring Homomorphism) [16]

Let φ be a ring homomorphism from a ring R to a ring S. Let A be a subring of R and let B be an ideal of S.

1. For any $r \in R$ and any positive integer n, $\varphi(nr) = n\varphi(r)$ and $\varphi(r^n) = (\varphi(r))^n$.
2. $\varphi(A) = \{\varphi(a) | a \in A\}$ is a subring of S.

3. If A is an ideal and φ is onto S, then $\varphi(A)$ is an ideal.
4. $\varphi^{-1}(B) = \{r \in R | \varphi(r) \in B\}$ is an ideal of R.
5. If R is a commutative, then $\varphi(R)$ is commutative.
6. If R has a unity 1, $S \neq \{0\}$, and φ is onto, then $\varphi(1)$ is the unity of S.
7. φ is an isomorphism if and only if φ is onto and $Ker\ \varphi = \{r \in R | \varphi(r) = 0\} = \{0\}$.
8. If φ is an isomorphism from R onto S, then φ^{-1} is an isomorphism from S onto R.

2.1.3 Fields

Definition 28 (Fields)

A ring is a field, let F be the field. It forms an Abelian group under addition and multiplication and satisfies the distributive laws under addition:

$$\text{(i)} \quad a(b+c) = as + ac \quad \text{and} \quad \text{(ii)} \quad (a+b)c = ac + bc$$

The field F as satisfies the following three conditions below:

1. Multiplicative identity, unity e (or 1), which is defined by $ea = ae = a$ for every $a \in F$.
2. Multiplicative inverse, a^{-1}, exists for every $a \in F$, where $a \neq 0$, such that $aa^{-1} = a^{-1}a = e$.
3. Multiplicative commutativity, $ab = ba$ for every $a, b \in F$.

2.1.4 Polynomial Rings

Definition 29 (Polynomial Rings over R)

Let R be a commutative ring. Where $R[x] = \{a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x^1 + a_0 | a_i \in R\}$ is called the polynomial ring over R.

Definition 30 (Addition and Multiplication in $R[x]$)

Let R be a commutative ring and let $f(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ and $g(x) = b_m x^m + b_{m-1}x^{m-1} + \cdots + b_1 x + b_0$ belong to $R[x]$. So, $f(x) + g(x) = (a_s + b_s)x^s + (a_{s-1} + b_{s-1})x^{s-1} + \cdots + (a_1 + b_1)x + a_0 + b_0$. $a_i = 0$ for $i > n$, and $b_i = 0$ for $i > m$.

Also, $f(x)g(x) = c_{m+n}x^{m+n} + c_{m+n-1}x^{m+n-1} + \cdots + c_1 x + c_0$, where $c_k = a_k b_0 + a_{k-1}b_1 + \cdots + a_1 b_{k-1} + a_0 b_k$ for $k = 0, \ldots, m+n$.

Theorem 31 (Integral Domain)

If the ring D is an integral domain, then $D[x]$ is an integral domain.

Theorem 32 (Division Algorithm)

If F is a field and $f(x)$ and $g(x) \in F[x]$ with $g(x) \neq 0$. Then there exists unique polynomial $q(x)$ and $r(x)$ in $F[x]$ such that $f(x) = g(x)q(x) + r(x)$ and either $r(x) = 0$ or $\deg r(x) < \deg g(x)$.

Theorem 33 (Remainder Theorem)

If F is a field, $a \in F$, and $f(x) \in F[x]$. Then f(a) is the remainder in the division of $f(x)$ by $x - a$.

Theorem 34 (Factor Theorem)

If F is a field, $a \in F$, and $f(x) \in F[x]$. Then a is a zero of f(x) if and only if $x - a$ is a factor of $f(x)$

Definition 35 (Principal Ideal Domain)

A principal ideal domain is an integral domain R in which every ideal has the form $< a > = \{ra | r \in R\}$ for some a in R.

Theorem 36 ($I =< g(x) >$)

If F is a field, I a non zero ideal in $F[x]$, and $g(x)$ is a nonzero polynomial of minimum degree in I.

Definition 37 (Irreducible or Prime Polynomial)

Let D be an integral domain. A polynomial $f(x) \in D[x]$ which is neither a zero polynomial nor a unit in $D[x]$ is an irreducible polynomial if the expression $f(x)$ cannot be represented as factor of two or more polynomials such as $f(x) = g(x)h(x)$.

Definition 38 (Reducible Polynomial)

Let D be an integral domain. A polynomial $f(x) \in D[x]$, where the expression $f(x)$ can be represented as factor of two or more polynomials such as $f(x) = g(x)h(x)$.

2.1.5 Extension Fields

Definition (Extension Field)

A field E is an extension field of a field F, if the field F is the subset of the field E such that the operations of F are the operations of E confined to F.

2.2 Basics of Hyper Elliptic Curve

In the area of cryptology, hyperelliptic curves are eagerly studied. Since it gives the same level of security with a smaller key length as compared to cryptosystems using elliptic curves. In 1987 Koblitz proposed that Jacobians of the hyperelliptic curves can produce abelian groups which will be suitable for cryptography. Hyperelliptic curves are a special group of algebraic curves and can be seen as a generalization of the elliptic curves. A hyperelliptic curve of genus $g = 1$ is also called an elliptic curves. Therefore, all the curves of every genus $g \geq 1$ are hyperelliptic curves. In this chapter, we will briefly introduce hyperelliptic curve cryptography and provide an overview of the parameters involved. In the following section, hyperelliptic curve cryptography or cryptosystem will always be abbreviated as HECC.

Definition 1 (Hyperelliptic Curves)

Let K be a field and let $\overline{K}$ be the algebraic closure of the field K. A hyperelliptic curve C of genus g ($g \geq 1$) over K is an equation of the form

$$C: y^2 + h(x)y = f(x) \text{ in } K[x, y]$$

Where $h(x) \in K[x]$ is a polynomial of degree at most g, $f(x) \in K[x]$ and a monic polynomial of degree $2g + 1$ and there is no solution $(x, y) \in \overline{K} \times \overline{K}$, which simultaneously satisfies the equations

$$y^2 + h(x)y = f(x)$$
$$2y + h(x) = 0$$
$$h'(x)y - f'(x) = 0$$

A singular point on the curve C is a solution $(x, y) \in \overline{K} \times \overline{K}$ which satisfies all the above three equations. Thus a hyperelliptic curve does not have any singular points by definition.

Definition 2 (Extension field of K)

Let L be an extension field of K. The set of L is the rational points on the curve C, denoted by $C(L) = \{(x, y) \in L \times L: y^2 + h(x)y = f(x)\} \cup \{O\}$, where O is a special point, called the point at infinity.

Definition 3 (Rational Points, Points at infinity, Finite Points)

To make the definition 2 to be clearer, the set of points in $C(L)$ are the rational points $P = (x, y) \in L \times L$ which satisfies the main general expression of the hyperelliptic curve. The point O is a special point, called the point at infinity. All the points in the curve C except O are finite points.

Definition 4 (Opposite, Special Points)

The opposite of a finite point $P(x, y)$ on the curve C is defined to be the point $\tilde{P} = (x, -y - h(x))$. The opposite of O is itself. A special point is a point if it is equal to its opposite. Like the point O, it is a special point. Otherwise P is an ordinary point.

2.2.1 Hyper Elliptic Curve – An example

2.2.1.1 Graphical representation of an elliptic curve over the real field

$$E = \{(x, y): y^2 = x^3 + x \ (mod\ 17)\}.$$

To represent the curve E in the graph, the curve has to be drawn over the real number field $\mathbb{R}$. Therefore,

$$E = \{(x, y): y^2 = x^3 - 16x\} \text{ over } \mathbb{R}$$

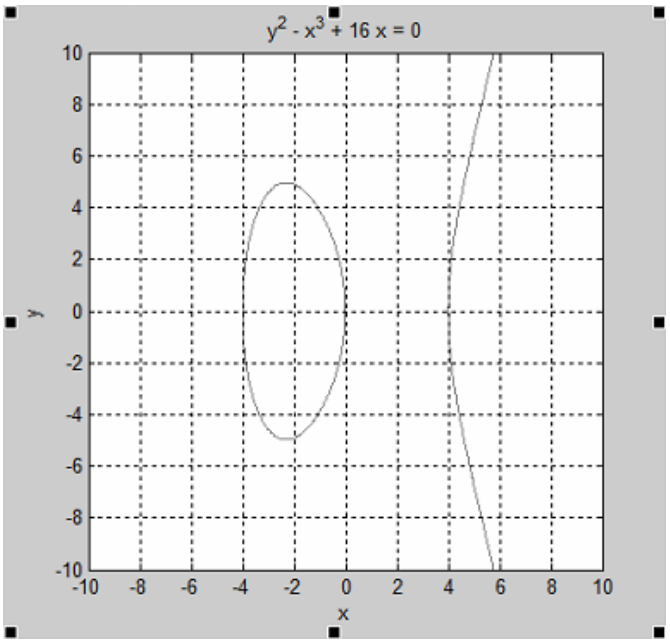

Figure 2.2: Elliptic Curve over $\mathbb{R}$

2.2.1.2 Determining the Cartesian points in an elliptic curve over the real field

When $F = Z_p$ (or more generally, when F is a finite field), the elliptic curves over Z_p will be a finite set. Here we take an equation of an elliptic curve with $F = Z_p$ and consider

$$E = \{(x, y): y^2 = x^3 + x \ (mod\ 17)\} \cup \{O\}$$

Now we want to know what points are on the curve E. To do that, we first compute the square table over F, which tells us what element in F can have a square root. This can be done by using power and mod function in MATLAB. Below shows the pseudo code for calculating $y^2 mod\ 17$.

Table 2.1 **PSEUDO CODE FOR CALCULATING: $y^2 mod\ p$**
$E = \{(x, y): y^2 = x^3 + x \ (\text{mod } p)\}$
INPUT: Range of y; Mod of p;
OUTPUT: $A = y^2 \text{ mod } p$;

Output in the tabular form:

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$y^2 mod\ p$	0	1	4	9	16	8	2	15	13	13	15	2	8	16	9	4	1

Then, we compute $x = 0,1,2,\ldots,16$ to solve the equation $y^2 = x^3 + x$ in Z_{17}. First, we compute the $x^3 + x$ in Z_{17} table over F.

Table 2.2 **PSEUDO CODE FOR CALCULATING: $(x^3 + x)$ mod p**
$E = \{(x, y): y^2 = x^3 + x \pmod p\}$
INPUT: Range of x; Mod of p;
OUTPUT: $B = (x^3 + x) \bmod p$;

Output in the tabular form:

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$x^3 + x\ mod\ p$	0	2	10	13	0	11	1	10	10	7	7	16	6	0	4	7	15

For $x = 1$, $y^2 = 1 + 1$ and so the square root table gives $y = \pm 6$. Hence $(1, \pm 6) \in E$, For $x = 2$, we have $y^2 = 8 + 2 = 10$, the square root table tell us that there is no solution, and so we move to the case $x = 3$. The following MATLAB code computes all the needed information.

Table 2.3 **PSEUDO CODE TO DETERMINE THE POINTS ON EC**
$E = \{(x, y): y^2 = x^3 + x \pmod p\}$
INPUT: Range of x; Range of y; Mod of p;

`A = y.^2 mod p;`
`B = x.^3 + x mod p;`
`for b = [1:p]`
`  for a = [1:p]`
`  if B(b) == A(a)`

```
            print "Valid Points";
        end
    end
end
```

OUTPUT: Points (x, y);

In this way we have all the valid points of the curve: $E = \{(x, y): y^2 = x^3 + x \ (mod\ 17)\} \cup \{O\}$

$$E = \{(0,0), (1,6), (1,11), (3,8), (3,9), (4,0), (6,1), (6.16), (11,4), (11,13), (13,0)$$
$$(14,2), (14,15), (16,7), (16,10), O\}$$

For the curve with the equation $E = \{(x, y): y^2 = x^3 + x \ (mod\ 17)\}$ has 16 valid points. If we perform the calculation in a larger finite field we will be able to work with greater number of valid points.

For the curve with equation $E = \{(x, y): y^2 = x^3 + x\ \}$.

F_x	Number of Valid Points on the curve over the field of F_x	F_x	Number of Valid Points on the curve over the field of F_x
$x = 2$	3	$x = 3$	4
$x = 5$	4	$x = 7$	8
$x = 11$	12	$x = 13$	20
$x = 17$	16	$x = 19$	20
$x = 23$	24	$x = 29$	20
$x = 31$	32	$x = 37$	36
$x = 41$	32	$x = 43$	44
$x = 47$	48	$x = 53$	68
$x = 59$	60	$x = 61$	52
$x = 67$	68	$x = 71$	72
$x = 73$	80	$x = 79$	80

$x = 83$	84	$x = 89$	80
$x = 97$	80	$x = 101$	100
$x = 103$	104	$x = 107$	108
$x = 109$	116	$x = 113$	128
$x = 127$	128	$x = 131$	132
$x = 137$	160	$x = 139$	140
$x = 149$	164	$x = 151$	152
$x = 157$	180	$x = 163$	164
$x = 167$	168	$x = 173$	148
$x = 179$	180	$x = 181$	164
$x = 191$	192	$x = 193$	208
$x = 197$	196	$x = 199$	200

Table 2.4: Finite Field and its corresponding number of valid points

2.2.1.3 Graphical representation of Hyper Elliptic Curve over the real field

$$E = \{(x, y): y^2 = x(x-2)(x-1)(x+1)(x+2)\ (mod\ p)\}.$$

To represent the curve E in the graph, the curve has to be drawn over the real number field $\mathbb{R}$. Therefore,

$$E = \{(x, y): y^2 = x(x-2)(x-1)(x+1)(x+2)\} \text{ over } \mathbb{R}$$

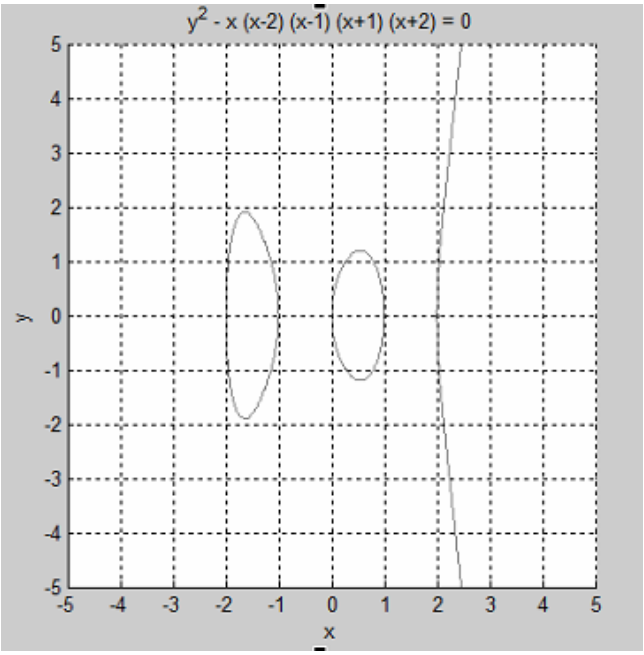

Figure 2.3: Hyper Elliptic Curve over $\mathbb{R}$

2.2.1.4 Determining the Cartesian points in a hyper elliptic curve over the real field

When $F = Z_p$ (or more generally, when F is a finite field), the elliptic curves over Z_p will be a finite set. Here we take an equation of an elliptic curve with $F = Z_p$ and consider

$$E = \{(x, y): y^2 + xy = x^5 + 2x^4 + x^3 - 5x^2 + 10\ (mod\ 11)\} \cup \{O\}$$

Now we want to know what points are on the curve E. To do that, we first compute the square table over F, which tells us what element in F can have a square root. This can be done by using power and mod function in MATLAB.

Table 2.5 **PSEUDO CODE TO DETERMINE THE POINTS ON HEC**
$E = \{(x, y): y^2 + xy = x^5 + 2x^4 + x^3 - 5x^2 + 10 \mod p\}$
INPUT: Range of x; Range of y; Mod of p;

STEPS:
`B = x.^5 + 2*x.^4 + x.^3 – 5*x.^2 + 10 mod p;`
`for b = [1:p]`
`  for a = [1:p]`
`  A = y.^2 + x(b).*y mod p;`
`  if B(b) == A(a)`
`  print “Valid Points”;`
`  end`
`  end`
`end`

OUTPUT: Points (x, y);

In this way we have all the valid points of the curve:

$$E = \{(x, y): y^2 + xy = x^5 + 2x^4 + x^3 - 5x^2 + 10(mod\ 11)\} \cup \{O\}$$

$$E = \{(1,4), (1,6), (4,2), (4,5), (5,7), (5,10), (8,0), (8.3), (9,5), (9,8), O\}$$

For the curve with the equation $E = \{(x, y): y^2 + xy = x^5 + 2x^4 + x^3 - 5x^2 + 10\ (mod\ 11)\}$ has 11 valid points.

2.2.1.5 Comparing the curves: Elliptic Curves and Hyper Elliptic Curve

Elliptic curve: $E_E = \{(x, y): y^2 = x^3 - 16x\}$ over $\mathbb{R}$

Hyper Elliptic curve: $E_{HEC} = \{(x, y): y^2 = x(x-2)(x-1)(x+1)(x+2)\}$ over $\mathbb{R}$

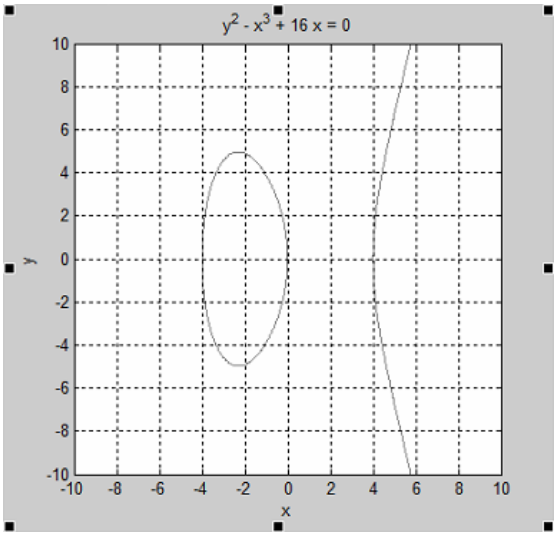

Figure 2.4: Elliptic curve E_{EC}

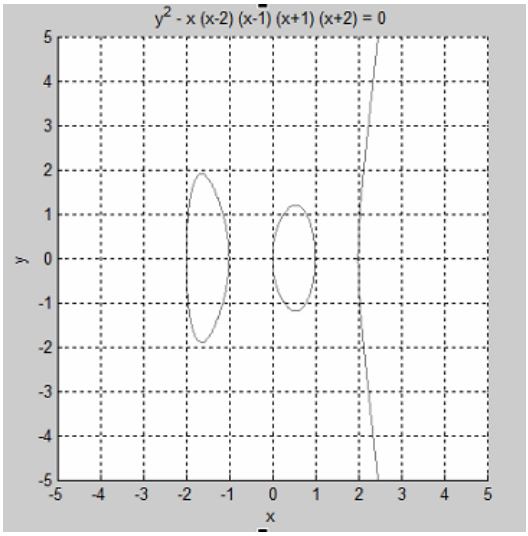

Figure 2.5: Hyper Elliptic curve E_{HEC}

By comparing the table 1 and 2, the figure 5 and 6. Hyper elliptic curves to have more number of valid points compare to Elliptic curve. For the field F_{197}, the curve E_{EC} has 196 valid points, whereas for the field F_{197}, , the curve E_{HEC} has 206 valid points. Approximate we can comment that E_{HEC} have more 10 valid points to E_{EC}. Figure 5 represents an elliptic curve of genus – 1. Genus in plain English means number of loops or holes. Figure 6 represents a hyperelliptic curve. Hyperelliptic curve is also a type elliptic curve with genus – 2, a curve with two holes or two loops.

2.2.1.6 Finding the points on the Hyper Elliptic Curve over large prime field

Let's take a hyperelliptic curve over a field. The curve $E: y^2 = x^5 + 1184x^3 + 1846x^2 + 956x + 560$ over $F_p,\ p = 2003$.

Number of valid points on the curve: 1867

Here are some of the valid points on the curve: (4,1712), (8,168), (257,1895), (258,783), (1000,1529), (1002,519), (1502,1293), (1505,388), (1999,1232), (2000,818).

2.2.2 Polynomial and Rational Function

Definition 5 (Coordinate Ring, Quotient Ring and Polynomial Function)

If I be the ideal of $K[x, y]$ which is generated by the polynomial $x^2 + h(x)y - f(x)$, that is $I = x^2 + h(x)y - f(x)$. The quotient ring of $K[x, y]/I$ is also called the coordinate ring of C over $\overline{K}$, denoted by $K[C]$. Elements in the $\overline{K}[C]$ is also called the polynomial functions of C over K.

It is easy to check that every polynomial, let it be $G(x, y) \in K[x, y]$ can be uniquely represented in the form of

$$G(x, y) = u(x) - v(x)y$$

where the polynomial $u(x),\ v(x) \in K[x]$.

Definition 6 (Function Field, Rational Functions)

The function field $K(C)$ of C over $K$is the field of all fractions of polynomial functions in $K[C]$. An element of $K(C)$ is called a rational function on C. A polynomial function is also a rational function. We have to make a note that $\overline{K}[C]$ is a subring of $K(C)$.

Definition 7 (Degree of a Polynomial Function)

Let $G(x, y) = u(x) - v(x)y$ be a polynomial and also a non – zero one in $\overline{K}[C]$. The degree of the polynomial G is defined to be

$$deg(G) = max[2deg_x(u), 2g + 1 + 2deg_x(v)].$$

Properties 8 (Degree of a Polynomial Function)

Let $G, H \in K[C]$.

1. $\deg(G) = deg_x(N(G))$
2. $\deg(GH) = \deg(G) + \deg(H)$
3. $\deg(G) = \deg(\bar{G})$

2.2.3 Zeroes and Poles

Definition 12 (Zeros and Poles)

Let $R \in \overline{K}(C)$ and $P \in C$. If $R(P) = 0$, then R is a zero at P. If R is not defined at P, then R has a pole at P. Where we write it as $R(P) = \infty$.

Definition 13 (Special Point, Zeros)

Let $P = (x, y)$ be a point on the curve C. Let us suppose that the polynomial function $G(u, v) = a(u) - b(u)v \in \overline{K}(C)$ has a zero at P and x is not a root of both $a(u)$ and $b(u)$. The $\bar{G}(P) = 0$ iff P is a special point.

Definition 14 (Ordinary Point, Zeros and Poles)

Let $P = (x, y)$ be an ordinary point on the curve C, and $G(u, v) = a(u) - b(u)v \in \overline{K}(C)$. Assuming that $G(P) = 0$ and x is not a root of both of the polynomials $a(u)$ and $b(u)$. Then G can be written in the form of $(u - x)^s S$, where s is the highest power of $(u - x)$ which divides $N(G)$, and $S \in \overline{K}(C)$ does not have a zero nor a pole at P.

Definition 15 (Special Point, Zeros and Pole)

Let $P = (x, y)$ be a special point on the curve C. Then $(u - x)$ can be written in the form $(v - y)^2$. $S(u, v)$, where $S(u, v) \in \overline{K}(C)$ has neither a zero nor a pole at P.

Chapter 3

Hyper Elliptic Curve Cryptography

To realize secure communication over the unsafe internet, cyber security techniques are indispensable such as privacy and authentication. Among these techniques, public key cryptography is an essential in our daily life. This cyber – security technology supports one of the fundamental aspect, like electronic payment infrastructure.

Public key cryptography was first introduced by Diffie and Hellman in 1976 [6]. However the first practical application was made by Rivest, Shamir and Adleman in 1977. Now the most widely used public key cryptosystem called RSA [8]. Although RSA is widely popular but the best public key cryptography currently available is the elliptic curve cryptosystem. Elliptic curve was first proposed by Koblitz [1] and Miller [19] independently. The security of the RSA depends on the difficulty of solving the integer factorization problem and Elliptic Curve Cryptography (ECC) depends on solving the discrete logarithm problem on an elliptic curve. The Cartesian points on the Elliptic Curve over a specific field can form a group based on the concepts generalized by Diffie – Hellman. This group is later used to develop ECC. Just like ECC, Hyper Elliptic Curve (HECC) can also form group structure over Jacobian of a hyper elliptic curve defined over a finite field. Koblitz [20] proposed this for the first time in 1988 . In this chapter we will discuss the group operations in HECC such as addition and multiplication (generally knows as doubling operation).

3.1 Elliptic and Hyper Elliptic Curve Group Law

3.1.1 Arithmetic of Elliptic Curve

Definition 1 (Properties – Elliptic Curve)

An elliptic curve E over the field $\mathbb{Z}_p$, denoted by $E/\mathbb{Z}_p$ where $p > 3$, is given by the Weierstrass equation.

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

Where the coefficients $a_1, a_2, a_3, a_4, a_6 \in Z_p$ and for the each point (x, y) on the curves, the coordinate $(x, y) \in Z_p$ together with an imaginary point O. All the points on the curve must also satisfy the partial derivatives $2y_1 + a_1x_1 + a_3$ and $3x_1^2 + 2a_2x_1 + a_4 - a_1y_1$ equals to zero at the same time.

The partial derivative conditions says whether the elliptic curve is non-singular or singular. A point on a curve is called singular if both of the partial derivatives equals to zero.

Definition 2 (Discriminant – Elliptic Curve) [34]

Smoothness of the curves can also be figured out by finding the discriminant of the curve. Let expressions.

$$b_2 = a_1^2 + 4a_2$$
$$b_4 = a_1a_3 + 2a_4$$
$$b_6 = a_3^2 + 4a_6$$
$$b_8 = a_1^2a_6 - a_1a_3a_4 + 4a_2a_6 + a_2a_3^2 - a_4^2$$

Let E be a curve defined over $\mathbb{Z}_p$ and let b_2, b_4, b_6 and b_8. The discriminant of the curve E denoted by Δ satisfies. The curve E is nonsingular and an elliptic curve, iff $\Delta \neq 0$.

$$\Delta = -b_2^2 b_8 - 8b_4^3 - 27b_6^2 + 9b_2 b_4 b_6$$

3.1.1.1 Group Operations on Elliptic Curve

Definition 3 (Point Addition – Elliptic Curve)

Point Addition P + Q. Denoting the group operation with the symbol " + ". "Addition" means that given two points and their coordinates lies in the curve E, say $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. In this this case we computer $R = P + Q$ and $P \neq Q$. A tangent is drawn through the points P and Q and obtain a third point of intersection. The point of intersection R' is reflected on the $x -$ axis to obtain the point R on the curve. The figure 1 below shows the point addition on an elliptic curve over $\mathbb{R}$.

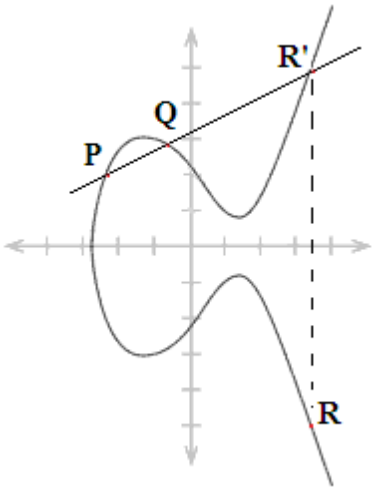

Figure 3.1: Point Addition on an Elliptic Curve over $\mathbb{R}$

It is important to define sum of the two points with the same $x -$ coordinate such as (x_1, y_1) and $(x_1, -y_1)$. In such case it is important to find a neutral element of the group. A further point P_∞ called the point at infinity or O. It can be understood that a point lying far out on the $y -$ axis such that the line $x = x_1 = c$ which is parallel to the $y -$ axis and passed through the point P_∞ or O. This point at the infinity is called the neutral point or element of the group. Therefore we can conclude that the line passing through (x_1, y_1) and $(x_1, -y_1)$ also passes through P_∞ or O.

$$(x_1, y_1) + (x_1, -y_1) = P_\infty, \text{ i.e.. } (x_1, -y_1) = -P$$

Definition 4 (Point Doubling – Elliptic Curve)

Point Doubling P + P. Given two points and their coordinates lies in the curve E, say $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. In this this case we computer $R = P + Q$ and $P = Q$. Making $R = P + P = 2P$. A tangent is drawn through the point P and obtain a second point of intersection. The point of intersection R' is reflected on the $x -$ axis to obtain the point R on the curve. The figure 2 below shows the point addition on an elliptic curve over $\mathbb{R}$.

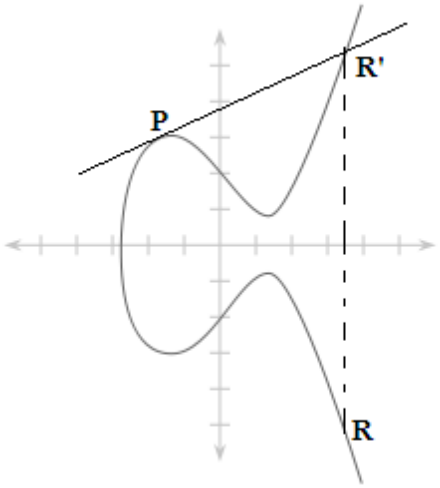

Figure 3.2: Point doubling on an Elliptic Curve over $\mathbb{R}$

3.1.1.2 Point Addition and Doubling in Elliptic Curve

Point Addition

The simplest form of an elliptic curve is given by the equation

$$y^2 = x^3 + ax + b$$

Let $P = (x_1, y_1)$, $Q = (x_2, y_2)$ and $R = (x_3, y_3)$. $P \neq Q$, let the straight line passing through the point P and Q be: $y = \alpha x + \beta$, where, α is the gradient of the line and β is the y – intercept.

The gradient of the line: $\alpha = \frac{y_2 - y_1}{x_2 - x_1}$. The y – intercept: at y – axis, $x = 0$, $y = y_1 - \alpha x_1$ Therefore, $\beta = y_1 - \alpha x_1$

A point $(x, \alpha x + \beta)$ lies in the elliptic curve if and only if: $y^2 = x^3 + ax + b$, $(\alpha x + \beta)^2 = x^3 + ax + b$, $x^3 - (\alpha x + \beta)^2 + ax + b = (x - x_1)(x - x_2)(x - x_3)$, $x^3 - \alpha^2 x^2 - 2\alpha\beta x - \beta^2 + ax + b = x^3 + (-x_2 - x_1 - x_3)x^2 + (x_1 x_2 + x_2 x_3 + x_1 x_3)x + (-x_1 x_2 x_3)$.

Let L.H.S ≡ R.H.S (x^2)

$$-\alpha^2 = -x_2 - x_1 - x_3$$

Or, $x_3 = \alpha^2 - x_2 - x_1$

Substituting x_3 into $y = \alpha x + \beta$, $y_3 = \alpha x_3 + (y_1 - \alpha x_1) = \alpha x_3 + y_1 - \alpha x_1 = \alpha(x_3 - x_1) + y_1$.
$-R \equiv (x_3, y_3), -R: x_3 = \alpha^2 - x_2 - x_1. y_3 = \alpha(x_3 - x_1) + y_1$

Reflecting the co-ordinate $-R$ on the x – axis. Therefore, $R \equiv (x_3, -y_3) = (\alpha^2 - x_2 - x_1, \alpha(x_1 - x_3) - y_1)$. Therefore, $x_3 = \alpha^2 - x_2 - x_1. y_3 = \alpha(x_1 - x_3) - y_1.$

where, $\alpha = \frac{y_2 - y_1}{x_2 - x_1} \ mod\ p$; if $P \neq Q$

Point Doubling

The simplest form of an elliptic curve is given by the equation

$$y^2 = x^3 + ax + b$$

Let $P = (x_1, y_1)$ and $R = (x_3, y_3)$, $\frac{dy}{dx} = \frac{3{x_1}^2 + a}{2y_1}$ at $P = (x_1, y_1)$

Let the line tangent to the curve at P be: $y = \alpha x + \beta$ where, α is the gradient of the line tangent and β is the y – intercept. The gradient of the line: $\alpha = \frac{y_2 - y_1}{x_2 - x_1}$. The y – intercept: at y – axis, $x = 0$. Or, $y - y_1 = \alpha(x - x_1)$, $y - y_1 = -\alpha x_1$, $y = y_1 - \alpha x_1$. Therefore, $\beta = y_1 - \alpha x_1$

A point $(x, \alpha x + \beta)$ lies in the elliptic curve if and only if: $y^2 = x^3 + ax + b$, $(\alpha x + \beta)^2 = x^3 + ax + b$, $x^3 - (\alpha x + \beta)^2 + ax + b = (x - x_1)^2(x - x_3)$, $x^3 - \alpha^2 x^2 - 2\alpha\beta x - \beta^2 + ax + b = x^3 + (-2x_1 x - x_3)x^2 + ({x_1}^2 + 2x_1 x_3)x - {x_1}^2 x_3$

Let L.H.S ≡ R.H.S (x^2): $-\alpha^2 = 2x_1 - x_3$, $x_3 = \alpha^2 - 2x_1$

Substituting x_3 into $y = \alpha x + \beta$, $y_3 = \alpha x_3 + \beta = \alpha x_3 + y_1 - \alpha x_1 = \alpha(x_3 - x_1) + y_1$. $-R \equiv (x_3, y_3)$. $-R$: $x_3 = \alpha^2 - 2x_1$, $y_3 = \alpha(x_3 - x_1) + y_1$

Reflecting the co-ordinate $-R$ on the x – axis. Therefore, $R \equiv (x_3, -y_3) = (\alpha^2 - 2x_1, \alpha(x_3 - x_1) + y_1)$. Therefore, $x_3 = \alpha^2 - 2x_1$, $y_3 = \alpha(x_1 - x_3) - y_1$. Where, $\alpha = \frac{3{x_1}^2 + a}{2y_1}$

3.1.2 Arithmetic of Hyper Elliptic Curves

3.1.2.1 Group Operations on Hyper Elliptic Curves

In elliptic curves we can take the points on the curve with the point of infinity to form a group. However for the hyper elliptic curves, if we take the points on the curve and with the points of infinity cannot no longer form a group. To form a group with respect to the points of hyper elliptic curve, we need to take sum of points as group elements and then we can perform addition like$(P_1 + P_2) \oplus (Q_1 + Q_2) = (R_1' + R_2')$. The symbol $+$ and $\oplus$ doesn't refers to addition and XOR operation respectively. The symbol $\oplus$ refers to group operation. We will discuss this specific operation between the two Cartesian points on the curve later in this chapter.

If we start to form group by this expression$(P_1 + P_2) \oplus (Q_1 + Q_2) = (R_1' + R_2')$, then we would end up with an infinite group and larger and larger representation of the group elements. In this case we use the quotient group of the group based on the all sum of points that lie on the curve.

Below we give a graphical representation of a hyper elliptic curve for a genus 2 over the finite field $\mathbb{F}_q$ given by the equation $y^2 + h(x)y = f(x)$. This equation of the curve must fulfill the five conditions before we can perform group operation.

Hyper elliptic curve of genus g over the finite field $\mathbb{F}_q$ in the set of points in $\mathbb{F}_q \times \mathbb{F}_q$ such that:

$$\text{C: } y^2 + h(x)y = f(x).$$

Conditions:

1. $h(x), f(x) \in \mathbb{F}_q[x]$.
2. $f(x) \rightarrow$ monic, and $\deg(f) = 2g + 1$ is odd.
3. $\deg(h) \leq g$, if $char(\mathbb{F}_q) = 2$; $h(x) = 0$ if $char(\mathbb{F}_q) \neq 2$.
4. The curve HEC doesn't have any singular point over $\mathbb{F}_q \times \mathbb{F}_q$.
5. $char(\mathbb{F}_q) \neq 2$: $y^2 = f(x)$, where $f(x)$ is monic, odd – degree and square free.

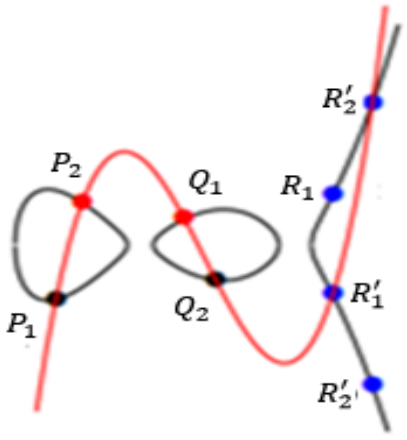

Figure 3.3: Group operation on the HEC of genus 2 over $\mathbb{R}$, $y^5 = f(x)$, $\deg f(x) = 5$ and $f(x)$ is monic for $(P_1 + P_2) \oplus (Q_1 + Q_2) = (R_1' + R_2')$.

As discussed before that the chord and tangent method in the elliptic curve cannot be used. The curve which intersects with the hyper elliptic curve of genus 2 shown above is called Jacobian variety curve. The Jacobian curve intersects in 5 points instead of 3 points unlike chord and tangent method in elliptic curve. In order to build a group we take the quotient group; which is the sum of the intersecting points of the Jacobian variety curve with the hyper elliptic curve by the subset of the points which lie on the HEC.

The six points P_1, P_2, Q_1, Q_2 R_1 and R_2 on the HE curve adds upto zero in the quotient group. The point $R_1' = (x_{R1}, y_{R1})$ and $-R_1' = R_1 = (x_{R1}, -y_{R1})$ lie on the curve. Similarly, $R_1' \oplus R_1 = 0$. The points R_1 and R_2 are the reflection of the points R_1' and R_2' on the HE curve respectively. And the resulting group operation $(P_1 + P_2) \oplus (Q_1 + Q_2) = (R_1' + R_2')$.

3.1.2.2 Divisor and Divisor Class Group. [37], [38]

Definition 5 (Divisor)

The rational points of a hyper elliptic curve do not form a group, unlike the points on an elliptic curve. The group which provides by the hyper elliptic curve for cryptography is a subgroup of the random group D generated by the set of points on the curve. If the curve C is the hyper elliptic curve of genus g over the finite field $\mathbb{F}_q$. The elements of D is known as divisors.

$$D = \sum m_p P, \qquad m_P \in \mathbb{F}_q \text{ and } P \in C$$

Definition 6 (Group of divisors)

For the hyper elliptic curve C of genus g over the finite field $\mathbb{F}_q$ given by an equation of the form C: $y^2 + h(x)y = f(x)$. The group of divisors of the curve C of degree 0 is given by

$$Div_C^0 = \sum_{P \in C} m_P P \mid m_P \in \mathbb{F}_q, m_P = 0, \text{ for most of the points on the curve } P \in C.$$

The group which describes before as the quotient group is also known as the divisor class group Pic_C^0 of C. In order to formally define this quotient group we need to take the point P_∞ called the point of infinity

into the divisor class group. Since in this thesis we are considering the hyper elliptic curve of $\deg f(x) =$ odd. Therefore, there is only a single point at the infinity. However, if we were working on the hyper elliptic curve of $\text{def}\, f(x) =$ even, then there would have been two point of infinity. To visualize this we can imagine a point far on the $y-$ axis such that any line which is parallel to it passes through the point P_∞.

Definition 7 (Divisor Class Group)

The divisor class group Pic_C^0 of C is the quotient group of the group of divisors Div_C^0. In the divisor class group, each divisor class can be represented by

$$D = \sum_{i=1}^{r} P_i - rP_\infty,\ P_i \in C \text{ including the point } \{P_\infty\},\ r \leq g.$$

By using the definition above. The individual divisor class can be represented for implementation purpose. The divisor class group Pic_C^0 of C is isomorphic to the finite field of $\mathbb{F}_q$ of the Jacobian J_C of the hyper elliptic curve C.

3.1.2.3 Jacobian variety of Hyper Elliptic Curve

Definition 8 (Jacobian) [39], [40]

The Jacobian of the curve C is defined by the quotient group:

$$J = J(C) = Div_C^0/P$$

Hence, $D_1, D_2 \in Div_C^0$ are equivalent if $D_1 - D_2 \in P$. In every equivalence class there's only one divisor D, called the reduced divisor:

$$D = \sum m_P P - (\sum m_P)P_\infty, \text{ such that } \sum m_p \leq g.$$

Jacobian variety curve in a specific curve based on the Jacobian $J(C)$. In simple form, the intersection points between the Jacobian variety curve with the hyper elliptic curve forms a group including the point at the infinity P_∞. Where the sum of the all the intersecting point sums up to zero. The figure below is the graphical representation of the Jacobian variety and the hyper elliptic curve of genus 2.

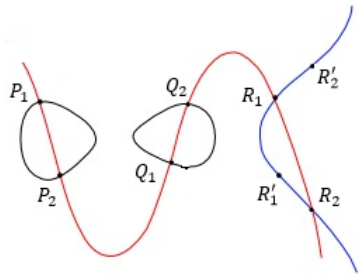

Figure 3.4: Hyper Elliptic Curve of genus 2 and Jacobian Variety Curve.

Since the intersection points between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$[P_1] + [P_2] + [Q_1] + [Q_2] + [R_1] + [R_2] = 0.$$

$$[P_1] + [P_2] + [Q_1] + [Q_2] = -[R_1] - [R_2] = [R_1'] + [R_2'].$$

The Cartesian or affine space points P_1 and P_2 could be transformed to individual divisor class group based on Mumford Representation, which is to be discussed in a separate section.

The divisor class, $D_1 = div\{P_1, P_2\} \cup \{P_\infty\} = [u_1(x), v_1(x)]$. Similarly $D_2 = div\{Q_1, Q_2\} \cup \{P_\infty\} = [u_2(x), v_2(x)]$ and $D_3 = div\{R_1', R_2'\} \cup \{P_\infty\} = [u_3(x), v_3(x)]$. The expression $u(x)$ and $v(x)$ are the polynomial representation of the affine space points on the curve. These are covered in the section detailing Mumford representation.

3.2 Point Representation - Divisor

The definition of the divisor group is the simplest form of representation. However, we can represent the divisors just as the sum of points with the order of the points m_P.

$$Div_C^0 = \sum_{P \in C} m_P P_P$$

The disadvantage of representing the divisor is that we cannot use this for computational purposes. To represent the points in the form of divisor the best option in the Mumford Representation.

3.2.1 Mumford Representation [35], [36]

Mumford representation is the clearest representation of the Cartesian points into polynomial divisor form. The divisor can be represented with two polynomial as $u(x)$ and $v(x)$. Let D be the individual reduced divisor of the divisor class group Pic_C^0 of C.

$$D = \sum m_P P - (\sum m_P) P_\infty$$

One fundamental reason for using the Mumford Representation is that this representation can be used for computing purpose. Let consider a hyper elliptic curve C of genus g, where the curve C is represented as:

$$y^2 + h(x)y = f(x)$$

where the polynomial expressions $h(x)$ and $f(x) \in$ the polynomial field $\mathbb{F}_q[x]$, the $\deg f(x) = 2g + 1$ and the $\deg h \leq g$. As discussed before that the divisor class over the field $\mathbb{F}_q$ can be represented by a pair of polynomials $u(x)$ and $v(x)$, where this polynomials $u(x)$, $v(x) \in \mathbb{F}_q[x]$.

Although the polynomials $u(x)$ and $v(x)$ belongs to the polynomial field of $\mathbb{F}_q[x]$. However this polynomials must fulfill the three conditions below:

Conditions:

1. $u(x)$ must a monic polynomial.
2. $\deg v(x) < \deg u(x) \leq g$.
3. $u(x)|v(x)^2 + v(x)h(x) - f$.

The polynomial expression of $u(x)$ of the divisor class D is represented by:

$$u(x) = \prod_{i=1}^{r}(x - x_i)$$

Where the divisor class D is represented as shown below.

$$D = \sum_{i=1}^{r} P_i - rP_\infty$$

The point $P_i \neq P_\infty, r \leq g$ and the points $P_i = (x_i, y_i)$ lies on the curve. If the points P_i on the curve occurs n_i number of times then

$$\left(\frac{d}{dt}\right)^j [v(x)^2 + v(x)h(x) - f(x)] = 0$$

Where $x = a_i$ and $0 \leq j \leq n_j - 1$. In the hyper elliptic curve of genus 2, each divisor class can be represented by the 4 coefficients u_1, u_0, v_1, v_0 of the polynomials $u(x)$ and $v(x)$. The divisor class D represented by the polynomials $u(x)$ and $v(x)$ as $D = [u(x),v(x)]$.

However, the divisor class group Pic_C^0 of C is the quotient group of the group of divisors Div_C^0. So the identify or neutral elements, in this case its neutral divisor class of the group is represented as [1,0].

An example in the section 3.2.2 will present an example.

3.2.2 Mumford Representation – An example

In this example we consider the hyper elliptic curve $C: y^2 = x^5 + 3x^3 + 2x^2 + 3$ of genus $g = 2$ over the field $\mathbb{F}_q$. The Cartesian points $P_1 = (3,0)$, $P_2 = (1,2)$, $Q_1 = (4,1)$ and $Q_2 = (3,0)$. The divisor class group Pic_C^0 of C is the quotient group of the group of divisors Div_C^0. Each divisor class can be represented by D including the point $\{P_\infty\}$, $r \leq g$.

$$D = \sum_{i=1}^{r} P_i - rP_\infty$$

Taking the points $P_1 = (3,0)$, $P_2 = (1,2)$, where $x_1 = 3$ and $x_2 = 1$. The polynomial expression of $u(x)$ of the divisor class D is represented by:

$$u(x) = \prod_{i=1}^{r} (x - x_i)$$

Therefore, $u(x) = \prod_{i=1}^{r}(x - x_i)$ and $u(x) = \prod_{i=1}^{2}(x - x_i) = (x - x_1)(x - x_2) = (x - 3)(x - 1) = x^2 - 4x + 3 = x^2 + x + 3$ over the polynomial field of $\mathbb{F}_5[x]$. The polynomial $u(x) = x^2 + x + 3 \in \mathbb{F}_5[x]$.

The condition for finding the polynomial expression $v(x)$ must satisfy the second and the third condition (2). $\deg v(x) < \deg u(x) \leq g$ and (3). $u(x) | v(x)^2 + v(x)h(x) - f$ respectively. Since the degree of $v(x)$ is less than the degree of $u(x)$, the polynomial expression of $v(x)$ would appear as $v(x) = v_{11}x + v_{10}$.

The number of combinations of $v_{11}x + v_{10}$, where $\mathbb{F}_5 \in \{0.1.2.3.4\}$. The possible combinations we can get for (v_{11}, v_{10}) are:

$$(0,0)\ (1,0)\ (2,0)\ (3,0)\ (4,0)$$
$$(0,1)\ (1,1)\ (2,1)\ (3,1)\ (4,1)$$
$$(0,2)\ (1,2)\ (2,2)\ (3,2)\ (4,2)$$

$$(0,3)\ (1,3)\ (2,3)\ (3,3)\ (4,3)$$
$$(0,4)\ (1,4)\ (2,4)\ (3,4)\ (4,4)$$

Any of the combination of (v_{11}, v_{10}) will satisfy the third condition $u(x)|v(x)^2 + v(x)h(x) - f$. In this case the combination $(v_{11}, v_{10}) = (4,3)$ satisfies the condition mentioned above.

Therefore, the Mumford representation of the point $P_1 = (3,0)$ and $P_2 = (1,2)$ on the hyper elliptic curve $C: y^2 = x^5 + 3x^3 + 2x^2 + 3$ of genus $g = 2$ over the field $\mathbb{F}_5$ is:

$$D_1 = [x^2 + x + 3, 4x + 3]$$

Similarly, the Cartesian points $Q_1 = (4,1)$ and $Q_2 = (3,0)$ can be represented in Mumford form.

$$D_2 = [x^2 + 3x + 2, x + 2]$$

Chapter 4

An Overview of Hyper Elliptic Curve Computation Method

In this chapter we will discuss Hyper Elliptic Curve Computation Methods for performing group operations, such as addition and doubling of the divisor classes of hyper elliptic curves discussed in the previous chapter. The purpose of this chapter is to discuss in detail on how we perform the group operations of the divisor class group obtained from the Jacobians of the hyper elliptic curves. The intersecting points of the Jacobian variety curve with the hyper elliptic curves seems to form a group [19].

However, the arithmetic operations of the divisor classes in the hyper elliptic curve was usually performed by using Cantor Algorithm. Cantor algorithm has been optimized by Harley, and the first to obtain subexpression and explicit formulas for the hyper elliptic curves of genus 2 and later in was extended by Lange and others.

Here we concentrate on the hyper elliptic curves of genus 2, 3 and 4 and provide an efficient explicit formulae for performing the arithmetic operations such as addition and doubling in HEC. The first explicit formula for genus 4 curves in to be found in this chapter.

4.1 Cantor Algorithm

Before the adverse of Cantor Algorithm [3] many explicit formulas for the addition of divisor classes has appeared, such as Montgomery [21] and Chudnosky [22]. Cantor Algorithm presents a formula for addition by using the divisor class in Mumford form. The same algorithm can also be used for scalar multiplication by using it in repeated manner. The sections below discusses the algorithm in more detail.

4.1.1 Composition and Reduction Stage

Let consider a hyper elliptic curve C of genus g, where the curve C is represented as [1]:

$$y^2 + h(x)y = f(x)$$

where the polynomial expressions $h(x)$ and $f(x) \in$ the polynomial field $\mathbb{F}_q[x]$, the $\deg f(x) = 2g + 1$ and the $\deg h \leq g$. As discussed before that the divisor class over the field $\mathbb{F}_q$ can be represented by a pair of polynomials $u(x)$ and $v(x)$, where this polynomials $u(x)$, $v(x) \in \mathbb{F}_q[x]$.

Here the polynomials $u(x)$ and $v(x)$ of the divisor class are the representation of the intersection points between the Jacobian variety curve and the hyper elliptic curve in Mumford form. The divisor class $D = [u(x), v(x)]$.

In the composition section of the Cantor Algorithm, the algorithm takes the polynomial expression $h(x)$ and $(x) \in \mathbb{F}_q[x]$. The Mumford representation of the points P_1, P_2 and Q_1, Q_2 in the divisor class: $D_P = [u_P(x), v_P(x)]$ & $D_Q = [u_Q(x), v_Q(x)]$. Here the polynomials $u_P(x), u_Q(x), v_P(x), v_Q(x) \in \mathbb{F}_q[x]$. The algorithm below performs the calculation for: $D_R = D_P + D_Q$.

	Cantor Algorithm (Composition)
INPUT	HEC: $y^2 + h(x)y = f(x)$. $D_P = [u_P, v_P]$, $D_Q = [u_Q, v_Q]$.
OUTPUT	$D_R = [u_R, v_R]$, Semi reduced divisor $D_R = D_P + D_Q$
Steps	Expressions
1	Compute $d_1 = \mathrm{GCD}(u_P, u_Q) = e_1 u_1 + e_2 u_2$;
2	Compute $d = \mathrm{GCD}(d_1, v_P + v_Q + h) = c_1 d_1 + c_2(v_P + v_Q + h)$;
3	Where $d = s_1 u_P + s_2 u_Q + s_3(v_P + v_Q + h)$, $s_1 = c_1 e_1$, $s_2 = c_1 e_2$, $s_3 = c_2$;
4	$u_R = u_P u_Q / d^2$, $v_R = s_1 u_P v_Q + s_2 u_Q v_P + s_3(v_P v_Q + f)/d \bmod u_R$;

Table 4.1: Cantor Algorithm (Composition)

In the step 1, $d_1 = e_1u_1 + e_2u_2$ is the resultant polynomial expression found by calculating the greatest common divisor GCD of the two polynomials u_P and u_Q. In the step 2, $d = c_1d_1 + c_2(v_P + v_Q + h)$ resultant polynomial expression found by calculating the GCD of the two polynomials d_1 and the sum of the polynomials $v_P + v_Q + h$. The expression d in the step 2 can be represented as s_1, s_2 and s_3. The step 4 calculates the expression for u_R and reduced expression of v_R mod u_R.

	Cantor Algorithm (Reduction)
INPUT	$D_R = [u_R, v_R]$ semi – reduced.
OUTPUT	$D'_R = [u'_R, v'_R]$ reduced $D_R \equiv D'_R$
Steps	Expressions
1	Calculate $u'_R = f - vh - v_R^2/u_R$, $v'_R = (-h - v_R) \bmod u'_R$;
2	If $\deg u'_R > g$ put $u_R = u'_R$, $v_R = v'_R$, goto step 1;
3	Make u'_R monic.

Table 4.2: Cantor Algorithm (Reduction)

The divisor $D_R = D_P + D_Q$ is known as semi reduced as calculated in the composition section. That means it is possible for further reduction. The second part of the Cantor Algorithm (Reduction) can be used to further reduce the polynomials expression of u_R and v_R. The result of the composition section can be used to perform further calculation. However it is better in practice to reduce the two polynomial expressions. In the section below, we have presented an example which will clear the mathematical steps in the Cantor Algorithm.

4.1.2 Cantor Algorithm – An example

In this example we consider the hyper elliptic curve $C: y^2 = x^5 + 3x^3 + 7x^2 + x + 2$ of genus 2 over the field $\mathbb{F}_{11}$. The divisor $D_P = [u_P, v_P] = [x^2 + 7x + 10,\ x + 9]$ and $D_Q = [u_Q, v_Q] = [x^2 + 10,\ 7x + 9]$. Here the polynomial expression $f(x) = x^5 + 3x^3 + 7x^2 + x + 2$, u_P, u_Q, v_P and $v_Q \in \mathbb{F}_{11}[x]$. The step 1 and 2 of the composition section calculates the GCD of the two polynomials. In step 1, we need to

compute $d_1 = \text{GCD}(u_P, u_Q) = e_1 u_1 + e_2 u_2$. So we can rewrite this expression as $d_1 = \text{GCD}(x^2 + 7x + 10, x^2 + 10)$.

Here, the GCD is calculated by using Extended Euclidean Algorithm as shown below:

$$a(x^2 + 7x + 10) + b(x^2 + 10) = \text{GCD}(x^2 + 7x + 10, x^2 + 10)$$

a	b	d	k
1	0	$x^2 + 7x + 10$	
0	1	$x^2 + 10$	1
1	-1	$7x$	$8x$
$3x$	$8x + 1$	10	$4x$
		0	

Therefore, $d_1 = e_1 u_P + e_2 u_Q = (3x)(x^2 + 7x + 10) + (8x + 1)(x^2 + 10)$. In step 2, we need to compute $d = \text{GCD}(d_1, v_P + v_Q + h) = c_1 d_1 + c_2(v_P + v_Q + h)$. So we can rewrite this expression as $d_1 = \text{GCD}(10, 8x + 10)$. Here, the GCD is calculated by using Extended Euclidean Algorithm as shown below:

$$a(8x + 7) + b(10) = \text{gcd}(10,\ 8x + 7)$$

a	b	d	k
1	0	$8x + 7$	
0	1	10	$3x + 4$
		0	

Therefore, $d = c_1 d_1 + c_2(v_1 + v_2 + h) = 1(10) + (0)(8x + 7)$. In the step 3, we need to represent the result of step 3 as $d = s_1 u_P + s_2 u_Q + s_3(v_P + v_Q + h)$. Where we need to calculate $s_1 = c_1 e_1 = 1 \times 3x = 3x$, $s_2 = c_1 e_2 = 1 \times (8x + 1) = 8x + 1$ and $s_3 = c_2 = 0$. In the step 4, $u_R = u_P u_Q / d^2 = (x^2 + 7x + 10)(x^2 + 10)/10^2 = x^4 + 7x^3 + 9x^2 + 4x + 1$.

$v_R = s_1 u_P v_Q + s_2 u_Q v_P + s_3(v_P v_Q + f)/d \bmod u_R = 4x^2 + 7x + 5$. So, the semi reduced divisor $D = [x^4 + 7x^3 + 9x^2 + 4x + 1,\ 4x^2 + 7x + 5]$. In the step 1 of the Cantor Algorithm (Reduction), we calculate $u'_R = f - vh - v_R^2/u_R = x + 10$ and $v'_R = (-h - v_R) \bmod u'_R = 6$. Working on the step 2 and 3 of the algorithm the reduced divisor D_R. $D_R = D_P + D_Q = [x + 10,6]$ in Mumford Representation.

4.1.3 Advantages and Disadvantages of using Cantor Algorithm

Cantor Algorithm was the first solid algorithm to perform the computations in the Jacobian groups of hyper elliptic curves over the fields of odd characteristics. The biggest advantage of the Cantor Algorithm is that we can apply this algorithm for any hyper elliptic curve of any genus over any field. Although the Cantor Algorithm is very computationally intensive, it can perform divisor class operations on hyper elliptic curves of any properties. The disadvantage lies in its computationally intensiveness. In the step 1 and 2 of its composition section, both of the steps uses Extended Euclidean Algorithm to calculate the GCD, which is computationally very intensive. Calculating GCD requires polynomial multiplication and especially polynomial inverses, which is computationally intensive. Other steps also requires polynomial multiplication and inverses. The Cantor Algorithm only offers the addition operation. For the scalar multiplication or doubling, the algorithm needs to be repeated. The table below shows the complexity of the Cantor Algorithm for genus 4 hyper elliptic curve over the field $\mathbb{F}_q$.

Algorithm	Inversion (I)	Addition Operation	
		Multiplication (M)	Squaring (S)
Cantor [23]	6	386 M/S	

Table 4.3: Complexity of the Cantor Algorithm of the hyper elliptic curve of genus 4

4.2 Subexpression Algorithm

Similarly like the Cantor Algorithm, Subexpression Algorithm [11] considers a hyper elliptic curve C of genus g, where the curve C is represented as: $y^2 + h(x)y = f(x)$ where the polynomial expressions $h(x)$ and $f(x) \in$ the polynomial field $\mathbb{F}_q[x]$, the $\deg f(x) = 2g + 1$ and the $\deg h \leq g$. As discussed before that the divisor class over the field $\mathbb{F}_q$ can be represented by a pair of polynomials $u(x)$ and $v(x)$, where this polynomials $u(x)$, $v(x) \in \mathbb{F}_q[x]$. The divisor class $D = [u(x), v(x)]$. The algorithm below performs the calculation for: $D_3 = D_1 + D_2$.

	Subexpression Algorithm [24]
INPUT	Genus = 2, HEC: $y^2 + h(x)y = f(x)$. $D_1 = [u_1, v_1]$, $D_2 = [u_2, v_2]$. $u_1 = x^2 + u_{11}x + u_{10}$, $u_2 = x^2 + u_{21}x + u_{20}$; $v_1 = v_{11}x + v_{10}$, $v_2 = v_{21}x + v_{20}$;
OUTPUT	$D_3 = [u_3, v_3] = [u_1, v_1] + [u_2, v_2]$;
Steps	Expressions
1	$k = (f - v_2 h - v_2^2)/u_2$;
2	$s = (v_1 - v_2)/u_2 \mod u_1$;
3	$l = s \cdot u_2$;
4	$u = (k - s(l + h + sv_2))/u_1$;
5	$u_3 = u$ made monic;
6	$v_3 = -h - (l + v_2) \mod u_3$;

Table 4.4: Subexpression Algorithm (Addition)

The algorithm above performs addition operation between the two divisor classes. Similar algorithm below performs the doubling operation for: $D' = 2D_1$.

	Subexpression Algorithm [24]		
INPUT	Genus = 2, HEC: $y^2 + h(x)y = f(x)$. $D = [u, v]$, $u = x^2 + u_1x + u_0$, $v = v_1x + v_0$;		
OUTPUT	$D' = 2D = [u', v']$;		
Steps	Expressions	Steps	Expression
1	$k = (f - hv - v^2)/u$	4	$u_1 = s^2 - (hs + 2vs - k)/u$
2	$k/(h + 2v) \bmod u$	5	$u' = u_1$ made monic
3	$l = s \cdot u$	6	$v' \equiv -h - (l + v) \bmod u'$

Table 4.5: Subexpression Algorithm (Doubling)

4.2.1 Subexpression Algorithm – An example

Using the same example used in the section of Cantor Algorithm – An example. Considering the hyper elliptic curve $C: y^2 = x^5 + 3x^3 + 7x^2 + x + 2$ of genus 2 over the field $\mathbb{F}_{11}$. The divisor $D_1 = [u_1, v_1] = [x^2 + 7x + 10,\ x + 9]$ and $D_2 = [u_2, v_2] = [x^2 + 10,\ 7x + 9]$. Here the polynomial expression $f(x) = x^5 + 3x^3 + 7x^2 + x + 2$, u_1, u_2, v_1 and $v_2 \in \mathbb{F}_{11}[x]$.
In the step 1, we calculate the expression, $k = (f - v_2h - v_2^2)/u_2$;

$$k = ((x^5 + 3x^3 + 7x^2 + x + 2) - (7x + 9) \cdot (0) - (7x + 9)^2)/x^2 + 10$$

Therefore, $k = x^3 + 4x + 2$.

In the step 2, we calculate the expression, $s = (v_1 - v_2)/u_2 \bmod u_1$;

$$s = ((x + 9) - (7x + 9))/x^2 + 10 \bmod x^2 + 7x + 10$$

Therefore, $s = 4$.

In the step 3,4 and 5. We calculate the expression, $l = s \cdot u_2 = 4x^2 + 7$, the expression $u = (k - s(l + h + sv_2))/u_1 = x + 10$ and the expression $u_3 = u$ made monic $= x + 10$.
Finally in the step 6, we calculate the expression, $v_3 = -h - (l + v_2) \bmod u_3 = 6$.

$$D_3 = D_1 + D_2 = [x + 10,6] \text{ in Mumford Representation.}$$

4.2.2 Advantages and Disadvantages of using Subexpression Algorithm

The algorithms takes the polynomial representation of the divisor class of Cartesian points in Mumford Representation and also the polynomial expression of $h(x)$ and $f(x)$. The biggest advantage of the Subexpression Algorithm is that, unlike the Cantor Algorithm which uses Extended Euclidean Algorithm twice to calculate GCD. In this algorithm, we don't have to compute GCD, which saves a lot of computationally intensive calculations such as polynomial inverses and multiplication.

The disadvantage lies in its computationally intensiveness. All the steps requires polynomial multiplication and especially polynomial inverses, which is computationally intensive. Unlike Cantor Algorithm, which can be applied to any hyper elliptic curve of any number of genus's, this algorithm is limited to the hyper elliptic curve of genus 2.

4.3 Explicit Formulae Algorithm

As discussed before that the disadvantages of using Cantor Algorithm is the computational intensity in the steps and the GCD calculation of polynomial by using Extended Euclidean Algorithm. Similarly, for the Subexpression Algorithm, where we still need to perform polynomial multiplications and inverses. However, in the Subexpression we do not need to perform GCD calculation of polynomials using Extended Euclidean Algorithm. In or der to avoid Cantor and Subexpression Algorithm, deriving an

explicit formula for genus 2 and for odd characteristics was made by Harley [11] and later was derived to have an explicit formula for even characteristics by Lange [9].
Matsuo, Chao and Tsujii [13] has already presented an explicit formulae for addition and doubling operation. To reduce the number of inversions to 1, Miyamoto, Doi, Matsuo, Chao and Tsujii [14] and the work by Takahashi [15] had obtained by using Montgomery trick.

4.3.1 Advantages and Disadvantages on Explicit Formulae Algorithm

Unlike Cantor Algorithm and Subexpression, which uses computationally intensive polynomial multiplication and inverses. Explicit formula only takes the co-efficient of the input polynomial and perform integer multiplication, inverses and squaring.

The only disadvantage of Explicit Formulae over the Cantor Algorithm is we need to derive separate explicit formula for the hyper elliptic curve of genus 2, 3, 4 and further. Unlike Cantor Algorithm, where we can use the same algorithm for performing group operation such as addition and doubling. Explicit formula has separate algorithms for addition and doubling operation.

Chapter 5

Proposed Efficient Computation for Hyper Elliptic Curve Cryptography

In this chapter, we will discuss the proposed efficient explicit formulae algorithm for group operation. Also the theorems and proposition used to build an efficient explicit formulae algorithm. Separate explicit formula algorithm for addition and doubling for specific hyper elliptic curve with different number of genus.

5.1 Explicit Formulae Algorithm for Hyper Elliptic Curve for genus 2

As discuss before that the group law operations in the Jacobian. The intersecting points of the Jacobian variety curve with the hyper elliptic curve form a group. In the section 3.1.2.3 of Jacobian variety of Hyper Elliptic Curve, we have mentioned that that Jacobian variety curve is a specific curve and its intersecting points with the hyper elliptic curve for a group including the point at the infinity P_∞. Where the intersecting points sums to zero. The figure below is the graphical representation of the Jacobian variety curve and the hyper elliptic curve of genus 2.

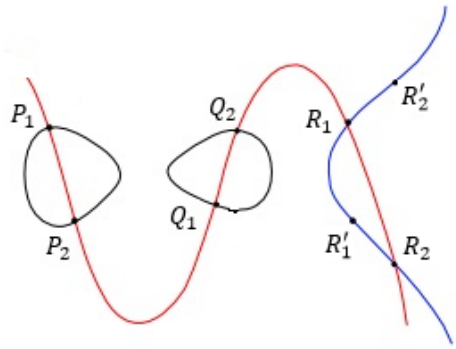

Figure 5.1: Hyper Elliptic Curve of genus 2 and Jacobian Variety Curve.

Since the intersection points between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$[P_1] + [P_2] + [Q_1] + [Q_2] + [R_1] + [R_2] = 0.$$

$$[P_1] + [P_2] + [Q_1] + [Q_2] = -[R_1] - [R_2] = [R_1'] + [R_2'].$$

The Cartesian or affine space points P_1 and P_2 could be transformed to individual divisor class group based on Mumford Representation, which is to be discussed in a separate section.

In the section 3.2.2 Mumford Representation – An example, we have shown how we can convert the Cartesian points on the curve into polynomial expression based on Mumford. By applying Mumford Representation, we can convert all the Cartesian points into divisors, after the conversion we can obtain the equation for the Jacobian Variety curve. Here we denote the Jacobian curve as $y = l(x)$.

5.1.1 Generating General Addition Explicit Formula for HEC of genus 2

Let's consider a general Hyper Elliptic Curve C of genus $g = 2$ over the finite field $\mathbb{F}_q$:

HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

The intersecting coordinates $P_1 = (x_{P1}, y_{P1})$ and $P_2 = (x_{P2}, y_{P2})$ would be converted to polynomial expression using Mumford.

The divisor class group, D_1 for the point P_1, P_2, D_2 for the point Q_1, Q_2 and D_3 for the point R_1, R_2 as shown below:

$$D_1 = \prod_{r=1}^{2}(x - x_{P1})(x - x_{P2}) - 2P_\infty = [x^2 + u_{11}x + u_{10}, v_{11}x + v_{10}]$$

$$D_2 = \prod_{r=1}^{2}(x - x_{Q1})(x - x_{Q2}) - 2P_\infty = [x^2 + u_{21}x + u_{20}, v_{21}x + v_{20}]$$

$$D_3 = \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2}) - 2P_\infty = [x^2 + u_{31}x + u_{30}, v_{31}x + v_{30}]$$

From the figure 1 above we can assert the polynomial expression of $l(x)$ to be $l(x) = l_3x^3 + l_2x^2 + l_1x + l_0$. The Jacobian curve $y = l(x)$ is a cubic function since we can see in the graph that the function has two extreme points and intersecting with the hyper elliptic curve with six Cartesian points.

At the intersecting points the y-coordinates are same. Therefore we can write, at the intersection points $l(x) = v(x)$ or $l(x) - v(x) \equiv 0\ mod\ u(x)$ since we have to perform polynomial reduction.

For the intersecting points P_1 and P_2, we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$(l_3x^3 + l_2x^2 + l_1x + l_0) - (v_{11}x + v_{10}) \equiv 0 \bmod x^2 + u_{11}x + u_{10}$$

Or, $(l_3x^3 + l_2x^2 + l_1x + l_0) \equiv (v_{11}x + v_{10}) \bmod x^2 + u_{11}x + u_{10}$

By reducing the L.H.S with the polynomial expression $x^2 + u_{11}x + u_{10}$ and comparing with the R.H.S, we get four simultaneous equations.

$$l_0 - u_{10}l_2 + u_{11}u_{10}l_3 \equiv v_{10} \qquad \text{EQN 1}$$

$$l_1 - u_{11}l_2 + (u_{11}^2 - u_{10})l_3 \equiv v_{11} \qquad \text{EQN 2}$$

$$l_0 - u_{20}l_2 + u_{21}u_{20}l_3 \equiv v_{20} \qquad \text{EQN 3}$$

$$l_1 - u_{21}l_2 + (u_{21}^2 - u_{20})l_3 \equiv v_{21} \qquad \text{EQN 4}$$

Subtracting the EQN 1 from EQN 3, we get:

$$(u_{10} - u_{20})l_2 + (u_{21}u_{20} - u_{11}u_{10})l_3 = v_{20} - v_{10} \qquad \text{EQN 5}$$

Subtracting the EQN 4 from EQN 2, we get:

$$(u_{11} - u_{21})l_2 + [(u_{21}^2 - u_{20}) - (u_{11}^2 - u_{10})]l_3 = v_{21} - v_{11} \qquad \text{EQN 6}$$

Generating an explicit formulae with too many variables may cause error or the final result of the algorithm may look tedious. At the same time in order to reduce repetitive computation which may consume processing power, we can denote the variables such as u_{11}, u_{21}, v_{11} and v_{20} with other notifications, as shown below:

$$A_0 = u_{11}u_{10} \qquad \Delta_0 = u_{10} - u_{20} \qquad B_0 = u_{11}^2 \qquad \mathrm{C}_0 = v_{20} - v_{10}$$

$$A_1 = u_{21}u_{20} \qquad \Delta_1 = u_{11} - u_{21} \qquad B_1 = u_{21}^2 \qquad \mathrm{C}_1 = v_{21} - v_{11}$$

$$A_1 - A_0 = \theta_1 \qquad \epsilon_0 = -u_{20} + u_{10} \qquad B_1 - B_0 = \theta_2 \qquad \theta_2 + \epsilon_0 = \theta_3$$

We can re – write the EQN 5 & 6 in a simpler way as shown below:

$$\Delta_0 l_2 + \theta_1 l_3 = C_0 \qquad \text{EQN 5}$$

$$\Delta_1 l_2 + (\theta_2 + \epsilon_0) l_3 = C_1 \qquad \text{EQN 6}$$

Solving the EQN 5 & 6 simultaneously we will get expressions for l_2 and l_3 as shown below:

$$l_2 = \frac{C_0\theta_3 - C_1\theta_1}{\Delta_0\theta_3 - \Delta_1\theta_1}$$

$$l_3 = \frac{\Delta_0 C_1 - \Delta_1 C_0}{\Delta_0\theta_3 - \Delta_1\theta_1}$$

By substituting l_2 and l_3 into the EQN 1 and in EQN 2 respectively, we will get the expression for l_0 and l_1, as shown below:

$$l_0 = u_{10}(l_2 - u_{11}l_3) + v_{10}$$

$$l_1 = u_{11}l_2 + v_{11} + (B_0 - u_{10})l_3$$

At intersecting points of the two curves, in this case it is the Jacobian Variety and the Hyper Elliptic Curve of genus 2, the value of the y on both curve is the same. So we can replace the y expression of the hyper elliptic curve by the Jacobian variety curve $y = l_3x^3 + l_2x^2 + l_1x + l_0$.

HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
The Jacobian Variety Curve: $y_{JC} = l_3x^3 + l_2x^2 + l_1x + l_0$

Substituting y in HEC with y_{JC}:

$$y_{JC}{}^2 + (h_2x^2 + h_1x + h_0)y_{JC} = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\text{Or, } (l_3x^3 + l_2x^2 + l_1x + l_0)^2 + (h_2x^2 + h_1x + h_0) \cdot (l_3x^3 + l_2x^2 + l_1x + l_0) = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\text{Or, } (l_3x^3 + l_2x^2 + l_1x + l_0)^2 + (h_2x^2 + h_1x + h_0) \cdot (l_3x^3 + l_2x^2 + l_1x + l_0) - (x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0) = 0$$

Expanding the L.H.S and comparing its coefficients with the R.H.S we will get as show below:

x^6	$l_3^2.$
x^5	$2l_2l_3 + h_2l_3 + 1.$
x^4	$2l_1l_3 + l_2^2 + h_2l_2 + h_1l_3 - f_4.$
x^3	$2l_1l_2 + l_1l_3 + l_0l_3 + h_2l_1 + h_2l_2 + h_0l_3 - f_3.$
x^2	$2l_0l_2 + l_1^2 + h_2l_0 + h_1l_1 + h_0l_2 - f_2.$
x^1	$2l_0l_1 + h_1l_0 + h_0l_1 - f_1.$
x^0	$h_0l_0 + l_0^2 - f_0.$

After computing the Jacobian Variety Curve $y_{JC} = l_3x^3 + l_2x^2 + l_1x + l_0$. However, if we intended to solve the y_{JC} and the HEC to find the Cartesian coordinates of the intersecting points. We can find the remaining two intersecting points by solving the expression below:

$$\prod_{r=1}^{2}(x-x_{P1})(x-x_{P2})\cdot\prod_{r=1}^{2}(x-x_{Q1})(x-x_{Q2})\cdot\prod_{r=1}^{2}(x-x_{R1})(x-x_{R2})$$

$$\equiv$$

$$(l_3x^3+l_2x^2+l_1x+l_0)^2+(h_2x^2+h_1x+h_0)\cdot(l_3x^3+l_2x^2+l_1x+l_0)$$

$$-(x^5+f_4x^4+f_3x^3+f_2x^2+f_1x+f_0)$$

In order to avoid tedious work and for simplification, we have introduced four new variable as s_3, s_2, s_1 and s_0. The expression of the variable is shown below:

$$s_3=u_{21}+u_{11}$$

$$s_2=u_{20}+u_{11}u_{21}+u_{10}$$

$$s_3=u_{11}u_{20}+u_{10}u_{20}$$

$$s_0=u_{10}u_{20}$$

Expanding the L.H.S of the equation and compare with the coefficients of x on the R.H.S. The expansion of the L.H.S is shown below:

$$(x^2+u_{11}x+u_{10})\cdot(x^2+u_{21}x+u_{20})\cdot(x^2+u_{31}x+u_{30})$$

x^6	$1.$
x^5	$u_{31}+s_3.$
x^4	$u_{30}+s_3u_{31}+s_2.$
x^3	$s_3u_{30}+s_2u_{31}+s_1.$
x^2	$s_2u_{30}+s_1u_{31}+s_0.$
x^1	$s_1u_{30}+s_0u_{31}.$
x^0	$s_0u_{30}.$

Comparing the coefficients of the L.H.S with the R.H.S we get:

$$u_{31} = 2l_2l_3 + h_2l_3 + 1 - s_3$$

$$u_{30} = 2l_1l_3 + l_2^2 + h_2l_2 + h_1l_3 - f_4 - u_{31}s_3 - s_2$$

Similarly, we can get the result for v_{31} and v_{30} by solving the equation:

$$l(x) \bmod (x^2 + u_{31}x + u_{30}) \equiv v_{31}x + v_{30}$$

Or, $l_3x^3 + l_2x^2 + l_1x + l_0 \bmod (x^2 + u_{31}x + u_{30}) \equiv v_{31}x + v_{30}$

After expanding the equation and comparing the coefficients L.H.S ≡ R.H.S we get:

$$v_{31} = -\{u_{31}^2l_3 - UL_0 + UL\}$$

$$v_{30} = -\{u_{31}UL_0 + UL\}$$

For simplification reason we presented with two variables:

$$UL_0 = u_{30}l_3$$

$$UL = -u_{31}l_2 + l_1$$

5.1.2 Computational Complexity of General Addition Explicit Formula for HEC of $g = 2$

In the section above, we have proposed and derived a General Explicit Formula for Addition on a HEC of genus = 2. For practical purpose one can eliminate the function of $h(x)$ and the co-efficient of f_4 from the function $f(x)$. The computational complexity for the General Addition Explicit Formulae is defined as (no. of Inverses, no. of Multiplication, no. of Squaring).

Finite Field	Curve Properties		Cost	
		Inverses (I)	Multiplication (M)	Squaring (s)
$\mathbb{F}_q$	$h(x), f(x)$	1	23	4
$\mathbb{F}_q$	$h(x) = 0, f_4 = 0.$	1	20	4

Table 5.1: Complexity comparison between the explicit formulae for HEC of genus 2 for different curve property.

5.1.3 Comparison of proposed and existing Explicit Formulae (Addition) for HEC $g = 2$.

The proposed works has been compared with the Explicit Formulae for Addition for the HEC for genus 2 has been compared. The table below presents list the complexity comparison table.

Previous Work	Finite Field	Curve Properties	Cost			Improvement Percentage (%)
			Inverses (I)	Multiplication (M)	Squaring (S)	
Harley [11,25]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	2	24	3	100
Lange [26]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	2	24	3	100
Matsuo [13]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	2	25	-	101.2
Takahashi [15]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	25	-	131.6
Miyamoto [14]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	26	-	130.1
Lange [27]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	22	3	133.4
This work	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	20	4	135.6

Table 5.2: Comparison between the explicit formulas for (genus = 2) curves over $\mathbb{F}_q$ of previous work and the present work.

ALGORITHM I

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 2, HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 6

Input	Genus 2 HEC: $y^2 + h(x)y = f(x)$; $h(x) = h_2x^2 + h_1x + h_0$; $f(x) = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D_1 = [u_1(x), v_1(x)]$, $D_2 = [u_2(x), v_2(x)]$; $u_1(x) = x^2 + u_{11}x + u_{10}$, $v_1(x) = v_{11}x + v_{10}$, $u_2(x) = x^2 + u_{21}x + u_{20}$, $v_1(x) = v_{21}x + v_{20}$;	
Output	$D_3 = [u_3(x), v_3(x)] = D_1 + D_2$, $u_3(x) = x^2 + u_{31}x + u_{30}$, $v_3(x) = v_{31}x + v_{30}$;	Cost (I, M, S)

Step	Expressions	Cost
1	$A_0 = u_{11}u_{10}$, $A_1 = u_{21}u_{20}$; $\Delta_0 = u_{10} - u_{20}$, $\Delta_1 = u_{11} - u_{21}$; $B_0 = u_{11}^2$, $B_1 = u_{21}^2$; $C_0 = v_{20} - v_{10}$, $C_1 = v_{21} - v_{11}$;	(0,2,2)
2	$\epsilon_0 = -u_{20} + u_{10}$; $\theta_1 = A_1 - A_0$, $\theta_2 = B_1 - B_0$, $\theta_3 = \theta_2 + \epsilon_0$;	(0,0,0)
3	$inv = (\Delta_0\theta_3 - \Delta_1\theta_1)^{-1}$;	(1,2,0)
4	$s_2 = u_{20} + u_{11}u_{21} + u_{10}$, $s_3 = u_{21} + u_{11}$;	(0,1,0)
5	$l_3 = inv \cdot (\Delta_0C_1 - \Delta_1C_0)$, $l_2 = inv \cdot (C_o\theta_3 - C_1\theta_1)$, $l_1 = u_{11}l_2 + v_{11} + (B_0 - u_{10})l_3$;	(0,8,0)
6	**Compute $u_3(x) = x^2 + u_{31}x + u_{30}$:** $u_{31} = 2l_2l_3 + h_2l_3 + 1 - s_3$, $u_{30} = 2l_1l_3 + l_2^2 + h_2l_2 + h_1l_3 - f_4 - u_{31}s_3 - s_2$;	(0,6,1)
7	$UL_0 = u_{30}l_3$, $UL = -u_{31}l_2 + l_1$;	(0,2,0)
8	**Compute $v_3(x) = v_{31}x + v_{30}$:** $v_{31} = -\{u_{31}^2l_3 - UL_0 + UL\}$, $v_{30} = -\{u_{31}UL_0 + UL\}$;	(0,2,1)
Sum		(0,23,4)

ALGORITHM II

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 2, HEC: $y^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 6

Input	Genus 3 HEC: $y^2 + h(x)y = f(x)$; $f(x) = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D_1 = [u_1(x), v_1(x)]$, $D_2 = [u_2(x), v_2(x)]$; $u_1(x) = x^2 + u_{11}x + u_{10}$, $v_1(x) = v_{11}x + v_{10}$, $u_2(x) = x^2 + u_{21}x + u_{20}$, $v_1(x) = v_{21}x + v_{20}$;	
Output	$D_3 = [u_3(x), v_3(x)] = D_1 + D_2$, $u_3(x) = x^2 + u_{31}x + u_{30}$, $v_3(x) = v_{31}x + v_{30}$;	Cost (I, M, S)

Step	Expressions	Cost
1	$A_0 = u_{11}u_{10}$, $A_1 = u_{21}u_{20}$; $\Delta_0 = u_{10} - u_{20}$, $\Delta_1 = u_{11} - u_{21}$; $B_0 = u_{11}^2$, $B_1 = u_{21}^2$; $C_0 = v_{20} - v_{10}$, $C_1 = v_{21} - v_{11}$;	(0,2,2)
2	$\epsilon_0 = -u_{20} + u_{10}$; $\theta_1 = A_1 - A_0$, $\theta_2 = B_1 - B_0$, $\theta_3 = \theta_2 + \epsilon_0$;	(0,0,0)
3	$inv = (\Delta_0\theta_3 - \Delta_1\theta_1)^{-1}$;	(1,2,0)
4	$s_2 = u_{20} + u_{11}u_{21} + u_{10}$, $s_3 = u_{21} + u_{11}$;	(0,1,0)
5	$l_3 = inv \cdot (\Delta_0C_1 - \Delta_1C_0)$, $l_2 = inv \cdot (C_o\theta_3 - C_1\theta_1)$, $l_1 = u_{11}l_2 + v_{11} + (B_0 - u_{10})l_3$;	(0,8,0)
6	**Compute $\boldsymbol{u_3(x) = x^2 + u_{31}x + u_{30}}$:** $u_{31} = 2l_2l_3 + 1 - s_3$, $u_{30} = 2l_1l_3 + l_2^2 - u_{31}s_3 - s_2$;	(0,3,1)
7	$UL_0 = u_{30}l_3$, $UL = -u_{31}l_2 + l_1$;	(0,2,0)
8	**Compute $\boldsymbol{v_3(x) = v_{31}x + v_{30}}$:** $v_{31} = -\{u_{31}^2l_3 - UL_0 + UL\}$, $v_{30} = -\{u_{31}UL_0 + UL\}$;	(0,2,1)
Sum		(0,20,4)

5.2 Explicit Formulae Algorithm for Hyper Elliptic Curve for genus 3

As discussed in the section 5.1, the intersecting points of the Jacobian variety curve with the hyper elliptic curve forms a group. We applied this concept to the HEC of genus 2. In this section we apply the same concept to develop a general addition explicit formulae algorithm for genus 3. The figure below is the graphical representation of the Jacobian variety curve and the hyper elliptic curve of genus 3.

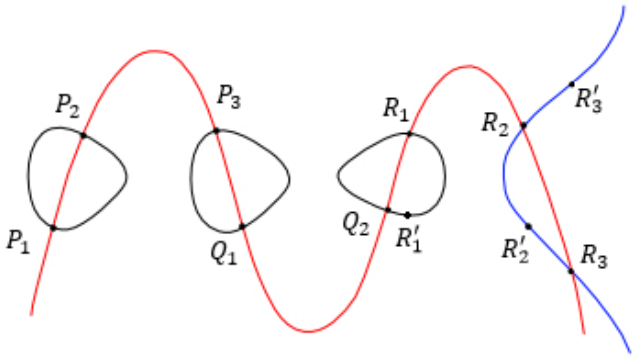

Figure 5.2: Hyper Elliptic Curve of genus 3 and Jacobian Variety Curve.

Since the intersecting points between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$[P_1] + [P_2] + [P_3] + [Q_1] + [Q_2] + [R_1] + [R_2] + [R_3] = 0.$$
$$[P_1] + [P_2] + [P_3] + [Q_1] + [Q_2] = -[R_1] - [R_2] - [R_3] = [R_1'] + [R_2'] + [R_3'].$$

Just like in the previous section, the Cartesian of affine space points P_1, P_2 and P_3 would be transformed to individual divisor class group based on Mumford Representation. By applying the Mumford Representation, we can convert all the Cartesian points into divisor, after the conversion we can obtain the equation for the Jacobian Variety curve. Here we denote the Jacobian curve as $y = l(x)$.

5.2.1 Generating General Addition Explicit Formulae for HEC of genus 3

Let's consider a general Hyper Elliptic Curve C of genus $g = 3$ over the finite field $\mathbb{F}_q$:

HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

The intersecting coordinates $P_1 = (x_{P1}, y_{P1})$, $P_2 = (x_{P2}, y_{P2})$ and $P_3 = (x_{P3}, y_{P3})$ would be converted to polynomial expression using Mumford.

The divisor class group, D_1 for the point P_1, P_2 and P_3. D_2 for the point Q_1, Q_2. D_3 for the point R_1, R_2 and R_3 as shown below:

$D_1 = \prod_{r=1}^{2}(x - x_{P1})(x - x_{P2})(x - x_{P3}) - 3P_\infty =$	$D_1 = [u_1(x), v_1(x)]$ $u_1(x) = x^3 + u_{12}x^2 + u_{11}x + u_{10}$ $v_1(x) = v_{12}x^2 + v_{11}x + v_{10}$
$D_2 = \prod_{r=1}^{2}(x - x_{Q1})(x - x_{Q2}) - 2P_\infty$	$D_2 = [u_2(x), v_2(x)]$ $u_2(x) = x^2 + u_{21}x + u_{20},$ $v_2(x) = v_{21}x + v_{20};$
$D_3 = \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2})(x - x_{R3}) - 3P_\infty$	$D_3 = [u_3(x), v_3(x)]$ $u_3(x) = x^3 + u_{12}x^2 + u_{11}x + u_{10}$ $v_3(x) = v_{12}x^2 + v_{11}x + v_{10}$

Table 5.3: Corresponding conversion of the Cartesian points to Mumford form.

From the figure 2 above we can assert the polynomial expression of $l(x)$ to be $l(x) = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. As shown in the previous section, here there are eight intersecting points with the hyper elliptic curve.

At the intersecting points the y-coordinates are same. Therefore we can write, at the intersection points $l(x) = v(x)$ or $l(x) - v(x) \equiv 0\ mod\ u(x)$ since we have to perform polynomial reduction.

For the intersecting points P_1, P_2 and P_3, we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$((l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (v_{12}x^2 + v_{11}x + v_{10}) \equiv 0 \bmod x^3 + u_{12}x^2 + u_{11}x + u_{10}$$

$$\text{Or, } (l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) \equiv (v_{12}x^2 + v_{11}x + v_{10}) \bmod x^3 + u_{12}x^2 + u_{11}x + u_{10}$$

For the intersecting points Q_1 and Q_2, we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$((l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (v_{21}x + v_{20}) \equiv 0 \bmod x^2 + u_{21}x + u_{20}$$

$$\text{Or, } (l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) \equiv (v_{21}x + v_{20}) \bmod x^2 + u_{21}x + u_{20}$$

By reducing the L.H.S with the polynomial expression $x^3 + u_{12}x^2 + u_{11}x + u_{10}$ and $x^2 + u_{21}x + u_{20}$ comparing with the R.H.S, we get five simultaneous equations.

$$(u_{12}^2 - u_{11})l_4 - u_{12}l_3 + l_2 \equiv v_{12} \qquad \text{EQN 1}$$

$$(u_{12}u_{10})l_4 - u_{10}l_3 + l_0 \equiv v_{10} \qquad \text{EQN 3}$$

$$[-u_{21}(u_{21}^2 - u_{20}) + u_{21}u_{20}]l_4 + (u_{21}^2 - u_{20})l_3 + (-u_{21})l_2 + l_1 \equiv v_{21} \qquad \text{EQN 4}$$

$$[-u_{20}(u_{21}^2 - u_{20})]l_4 + (u_{21}u_{20})l_3 + (-u_{20})l_2 + l_0 \equiv v_{20} \qquad \text{EQN 5}$$

As discussed before, that generating an explicit formulae with too many variable may make the work tedious. In order to reduce repetitive computation, we can denote the variable with other notifications, as shown below:

$$U_{21} = u_{21}^2 \qquad B_0 = u_{11}u_{12} \qquad \Delta_0 = u_{21}U_{12} - U_{21} - B_0 + u_{10} + u_{21}U_{21}$$

$$U_{12} = u_{12}^2 \qquad B_1 = u_{21}u_{12} \qquad \Delta_1 = -B_1 + u_{11} + U_{21} - u_{20}$$

$$A_1 = -(U_{12} - u_{11}) \qquad B_3 = u_{20}u_{12} \qquad \Delta_2 = u_{20}U_{12} - B_4$$

$$A_2 = -B_4 \qquad B_4 = u_{10}u_{12} \qquad \Delta_3 = u_{10} - u_{20}u_{21} - B_3$$

$$A_3 = -(B_0 - u_{10}) \qquad \epsilon_0 = [u_{21}v_{21}] - [v_{11} - v_{21}] \qquad \epsilon_1 = [u_{20}v_{12}] - [v_{10} - v_{20}]$$

Solving the five equations simultaneously, we will get the expression for l_0, l_1, l_2, l_3, and l_4, as shown below:

$$l_0 = u_{10}l_3 + A_2l_4 + v_{10}$$

$$l_1 = u_{11}l_3 + A_3l_4 + v_{11}$$

$$l_2 = u_{12}l_3 + A_1l_4 + v_{12}$$

$$l_3 = \frac{\Delta_2\epsilon_0 - \Delta_0\epsilon_1}{\Delta_1\Delta_2 - \Delta_0\Delta_3} \quad \text{and} \quad l_3 = \frac{\Delta_1\epsilon_1 - \Delta_3\epsilon_0}{\Delta_1\Delta_2 - \Delta_0\Delta_3}$$

At intersecting points of the two curves, in this case it is the Jacobian Variety and the Hyper Elliptic Curve of genus 3, the value of the y on both curve is the same. So we can replace the y expression of the hyper elliptic curve by the Jacobian variety curve $y = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$.

HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
The Jacobian Variety Curve: $y_{JC} = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$

Substituting y in HEC with y_{JC}:

$$y_{JC}{}^2 + (h_2x^2 + h_1x + h_0)y_{JC} = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$$

Or, $(l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2 + (h_2x^2 + h_1x + h_0) \cdot (l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

$$\text{Or, } (l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2 + (h_2x^2 + h_1x + h_0) \cdot (l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0) = 0$$

Expanding the L.H.S and comparing its coefficients with the R.H.S we will get as show below:

$$\begin{aligned}
x^8 &\quad l_4^2. \\
x^7 &\quad 2l_4l_3 - 1. \\
x^6 &\quad 2l_4l_2 + l_3^2 + h_2l_4 - f_6. \\
x^5 &\quad 2l_4l_1 + 2l_3l_2 + h_2l_3 + h_1l_4 - f_5. \\
x^4 &\quad 2l_3l_1 + 2l_4l_0 + l_2^2 + h_2l_2 + h_1l_3 + h_0l_3 - f_4. \\
x^3 &\quad 2l_3l_0 + 2l_2l_1 + h_2l_1 + h_1l_2 + h_0l_3 - f_3. \\
x^2 &\quad 2l_2l_0 + l_1^2 + h_2l_0 + h_1l_1 + h_0l_2 - f_2. \\
x^1 &\quad 2l_1l_0 + h_1l_0 + h_0l_1 - f_1. \\
x^0 &\quad l_0^2 + h_0l_0 - f_0.
\end{aligned}$$

After computing the Jacobian Variety Curve $y_{JC} = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. However, if we intended to solve the y_{JC} and the HEC to find the Cartesian coordinates of the intersecting points. We can find the remaining two intersecting points by solving the expression below:

$$\prod_{r=1}^{3}(x - x_{P1})(x - x_{P2})(x - x_{P3}) \cdot \prod_{r=1}^{2}(x - x_{Q1})(x - x_{Q2}) \cdot \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2})\,(x - x_{R3})$$

$$\equiv$$

$$(l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2 + (h_2x^2 + h_1x + h_0) \cdot (l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0)$$

In order to avoid tedious work and for simplification, we have introduced four new variable as s_4, s_3, s_2, s_1 and s_0. The expression of the variable is shown below:

$$s_0 = u_{21} + u_{12}$$

$$s_1 = u_{20} + B_1 + u_{11}$$

$$s_2 = B_3 + u_{11}u_{21} + u_{10}$$

$$s_3 = u_{11}u_{20} + u_{10}u_{20}$$

$$s_4 = u_{10}u_{20}$$

Expanding the L.H.S of the equation and compare with the coefficients of x on the R.H.S. The expansion of the L.H.S is shown below:

$$(x^3 + u_{12}x^2 + u_{11}x + u_{10}) \cdot (x^2 + u_{21}x + u_{20}) \cdot (x^3 + u_{32}x^2 + u_{31}x + u_{30})$$

x^8	$1.$
x^7	$s_0 + u_{32}.$
x^6	$s_1 + u_{32}s_0 + u_{31}.$
x^5	$s_3 + u_{32}s_1 + u_{31}s_0 + u_{30}.$
x^4	$s_3 + u_{32}s_2 + u_{31}s_1 + u_{30}s_0.$
x^3	$s_4 + u_{32}s_3 + u_{31}s_2 + u_{30}s_1.$
x^2	$u_{32}s_4 + u_{31}s_3 + u_{30}s_1.$
x^1	$u_{31}s_4 + u_{30}s_3.$
x^0	$u_{30}s_4.$

Comparing the coefficients of the L.H.S with the R.H.S we get:

$$u_{32} = 2l_4l_3 - 1 - s_0$$

$$u_{31} = 2l_4l_2 + l_3^2 + h_2l_4 - f_6 - s_1 - u_{32}s_0$$

$$u_{30} = 2l_4l_1 + 2l_3l_2 + h_2l_3 + h_1l_4 - f_5 - s_2 - u_{32}s_1 - u_{31}s_0$$

Similarly, we can get the result for v_{32}, v_{31} and v_{30} by solving the equation:

$$l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0 \equiv (v_{32}x^2 + v_{31}x + v_{30}) \bmod x^3 + u_{32}x^2 + u_{31}x + u_{30}$$

After expanding the equation and comparing the coefficients L.H.S ≡ R.H.S we get:

$$v_{32} = -\{u_{32}^2 l_4 - UL_0 - u_{32} l_3 + l_2\}$$

$$v_{31} = -\{u_{32} UL_0 - UL - u_{31} l_3 + l_1\}$$

$$v_{30} = -\{u_{32} UL - u_{30} l_3 + l_0\}$$

For simplification reason we presented with two variables:

$$UL = u_{30} l_4$$

$$UL_0 = u_{31} l_4$$

5.2.2 Computational Complexity of General Addition Explicit Formula for HEC of $g = 3$

In the section above, we have proposed and derived a General Explicit Formula for Addition on a HEC of genus = 3. For practical purpose one can eliminate the function of $h(x)$ and the co-efficient of f_4 from the function $f(x)$. The computational complexity for the General Addition Explicit Formulae is defined as (no. of Inverses, no. of Multiplication, no. of Squaring).

Finite Field	Curve Properties		Cost	
		Inverses (I)	Multiplication (M)	Squaring (s)
$\mathbb{F}_q$	$h(x), f(x)$	1	44	4
$\mathbb{F}_q$	$h(x) = 0, f_6 = 0.$	1	41	4

Table 5.4: Complexity comparison between the explicit formulae for HEC of genus 3 for different curve property.

5.2.3 Comparison of proposed and existing Explicit Formulae (Addition) for HEC $g = 3$.

The proposed work is compared with the Explicit Formulae for Addition for the HEC for genus 3 has been compared. The table below presents list the complexity comparison table.

Previous Work	Finite Field	Curve Properties	Cost			Improvement Percentage (%)
			Inverse (I)	Multiplication (M)	Squaring (S)	
Kuroki et al [29]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	81 M/S		100
Gonda et al [30]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	70 M/S		110.9
Guyot et al. [31]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	64	6	113.3
Myukai et al. [32]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	67 M/S		113.9
This work	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	41	4	132.2

Table 5.5: Comparison between the explicit formulas for (genus = 3) curves over $\mathbb{F}_q$ of previous work and the present work.

ALGORITHM III

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 3, HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 8

Input	Genus 3 HEC: $y^2 + h(x)y = f(x)$; $h(x) = h_2x^2 + h_1x + h_0$; $f(x) = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D_1 = [u_1(x), v_1(x)]$, $D_2 = [u_2(x), v_2(x)]$; $u_1(x) = x^3 + u_{12}x^2 + u_{11}x + u_{10}$, $v_1(x) = v_{12}x^2 + v_{11}x + v_{10}$, $u_2(x) = x^2 + u_{21}x + u_{20}$, $v_1(x) = v_{21}x + v_{20}$;	
Output	$D_3 = [u_3(x), v_3(x)] = D_1 + D_2$, $u_3(x) = x^3 + u_{32}x^2 + u_{31}x + u_{30}$, $v_3(x) = v_{32}x^2 + v_{31}x + v_{30}$;	Cost (I, M, S)

Step	Expressions	Cost
1	$U_{21} = u_{21}^2$, $U_{12} = u_{12}^2$, $B_0 = u_{11}u_{12}$, $B_1 = u_{21}u_{12}$, $B_3 = u_{20}u_{12}$, $B_4 = u_{10}u_{12}$;	(0,4,2)
2	$\Delta_0 = u_{21}U_{12} - U_{21} - B_0 + u_{10} + u_{21}U_{21} - 2u_{10}u_{11}$, $\Delta_1 = -B_1 + u_{11} + U_{21} - u_{20}$, $\Delta_2 = u_{20}U_{12} - B_4$, $\Delta_3 = u_{10} - u_{20}u_{21} - B_3$;	(0,5,0)
3	$\epsilon_0 = [u_{21}v_{21}] - [v_{11} - v_{21}]$, $\epsilon_1 = [u_{20}v_{12}] - [v_{10} - v_{20}]$;	(0,2,0)
4	$A_1 = -(U_{12} - u_{11})$, $A_2 = -B_4$, $A_3 = -(B_0 - u_{10})$;	(0,0,0)
5	$s_0 = u_{21} + u_{12}$, $s_1 = u_{20} + B_1 + u_{11}$, $s_2 = B_3 + u_{11}u_{21} + u_{10}$; $inv = (\Delta_1\Delta_2 - \Delta_0\Delta_3)^{-1}$;	(1,3,0)
6	$l_4 = inv \cdot (\Delta_1\epsilon_1 - \Delta_3\epsilon_0)$, $l_3 = inv \cdot (\Delta_2\epsilon_0 - \Delta_0\epsilon_1)$, $l_2 = u_{12}l_3 + A_1l_4 + v_{12}$, $l_1 = u_{11}l_3 + A_3l_4 + v_{11}$, $l_0 = u_{10}l_3 + A_2l_4 + v_{10}$;	(0,12,0)
7	**Compute $\boldsymbol{u_3(x) = x^3 + u_{32}x^2 + u_{31}x + u_{30}}$:** $u_{32} = 2l_4l_3 - 1 - s_0$, $u_{31} = 2l_4l_2 + l_3^2 + h_2l_4 - f_6 - s_1 - u_{32}s_0$, $u_{30} = 2l_4l_1 + 2l_3l_2 + h_2l_3 + h_1l_4 - f_5 - s_2 - u_{32}s_1 - u_{31}s_0$;	(0,10,1)
8	$UL = u_{30}l_4$, $UL_0 = u_{31}l_4$;	(0,2,0)
9	**Compute $\boldsymbol{v_3(x) = v_{32}x^2 + v_{31}x + v_{30}}$:** $v_{32} = -\{u_{32}^2l_4 - UL_0 - u_{32}l_3 + l_2\}$, $v_{31} = -\{u_{32}UL_0 - UL - u_{31}l_3 + l_1\}$, $v_{30} = -\{u_{32}UL - u_{30}l_3 + l_0\}$;	(0,6,1)
Sum		(1,44,4)

ALGORITHM IV

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 3, HEC: $y^2 = x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 8.

Input	Genus 3 HEC: $y^2 + h(x)y = f(x)$; $f(x) = x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D_1 = [u_1(x), v_1(x)]$, $D_2 = [u_2(x), v_2(x)]$; $u_1(x) = x^3 + u_{12}x^2 + u_{11}x + u_{10}$, $v_1(x) = v_{12}x^2 + v_{11}x + v_{10}$, $u_2(x) = x^2 + u_{21}x + u_{20}$, $v_1(x) = v_{21}x + v_{20}$;	
Output	$D_3 = [u_3(x), v_3(x)] = D_1 + D_2$, $u_3(x) = x^3 + u_{32}x^2 + u_{31}x + u_{30}$, $v_3(x) = v_{32}x^2 + v_{31}x + v_{30}$;	Cost (I, M, S)

Step	Expressions	Cost
1	$U_{21} = u_{21}^2$, $U_{12} = u_{12}^2$, $B_0 = u_{11}u_{12}$, $B_1 = u_{21}u_{12}$, $B_3 = u_{20}u_{12}$, $B_4 = u_{10}u_{12}$;	(0,4,2)
2	$\Delta_0 = u_{21}U_{12} - U_{21} - B_0 + u_{10} + u_{21}U_{21} - 2u_{10}u_{11}$, $\Delta_1 = -B_1 + u_{11} + U_{21} - u_{20}$, $\Delta_2 = u_{20}U_{12} - B_4$, $\Delta_3 = u_{10} - u_{20}u_{21} - B_3$;	(0,5,0)
3	$\epsilon_0 = [u_{21}v_{21}] - [v_{11} - v_{21}]$, $\epsilon_1 = [u_{20}v_{12}] - [v_{10} - v_{20}]$;	(0,2,0)
4	$A_1 = -(U_{12} - u_{11})$, $A_2 = -B_4$, $A_3 = -(B_0 - u_{10})$;	(0,0,0)
5	$s_0 = u_{21} + u_{12}$, $s_1 = u_{20} + B_1 + u_{11}$, $s_2 = B_3 + u_{11}u_{21} + u_{10}$; $inv = (\Delta_1\Delta_2 - \Delta_0\Delta_3)^{-1}$;	(1,3,0)
6	$l_4 = inv \cdot (\Delta_1\epsilon_1 - \Delta_3\epsilon_0)$, $l_3 = inv \cdot (\Delta_2\epsilon_0 - \Delta_0\epsilon_1)$, $l_2 = u_{12}l_3 + A_1l_4 + v_{12}$, $l_1 = u_{11}l_3 + A_3l_4 + v_{11}$, $l_0 = u_{10}l_3 + A_2l_4 + v_{10}$;	(0,12,0)
7	**Compute $\boldsymbol{u_3(x) = x^3 + u_{32}x^2 + u_{31}x + u_{30}}$:** $u_{32} = 2l_4l_3 - 1 - s_0$, $u_{31} = 2l_4l_2 + l_3^2 - s_1 - u_{32}s_0$, $u_{30} = 2l_4l_1 + 2l_3l_2 - f_5 - s_2 - u_{32}s_1 - u_{31}s_0$;	(0,7,1)
8	$UL = u_{30}l_4$, $UL_0 = u_{31}l_4$;	(0,2,0)
9	**Compute $\boldsymbol{v_3(x) = v_{32}x^2 + v_{31}x + v_{30}}$:** $v_{32} = -\{u_{32}^2l_4 - UL_0 - u_{32}l_3 + l_2\}$, $v_{31} = -\{u_{32}UL_0 - UL - u_{31}l_3 + l_1\}$, $v_{30} = -\{u_{32}UL - u_{30}l_3 + l_0\}$;	(0,6,1)
Sum		(1,41,4)

5.3 Explicit Formulae Algorithm for Hyper Elliptic Curve for genus 4

In this section we apply the same concept to develop a general addition explicit formulae algorithm for genus 3. The figure below is the graphical representation of the Jacobian variety curve and the hyper elliptic curve of genus 4.

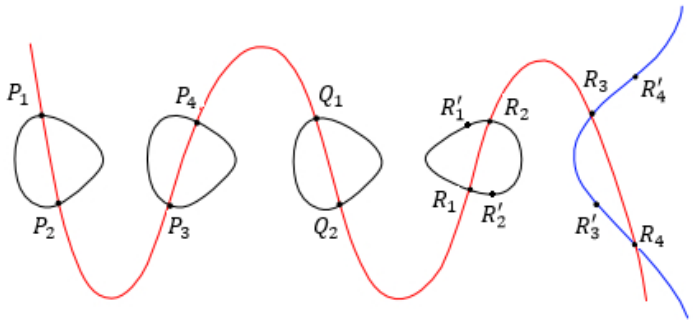

Figure 5.3: Hyper Elliptic Curve of genus 4 and Jacobian Variety Curve.

Since the intersecting points between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$[P_1] + [P_2] + [P_3] + [P_4] + [Q_1] + [Q_2] + [R_1] + [R_2] + [R_3] + [R_4] = 0.$$
$$[P_1] + [P_2] + [P_3] + [P_4] + [Q_1] + [Q_2] = -[R_1] - [R_2] - [R_3] - [R_4]$$
$$= [R_1'] + [R_2'] + [R_3'] + [R_4'].$$

5.3.1 Generating General Addition Explicit Formulae for HEC genus 4

Just like in the previous section, the Cartesian of affine space points to individual divisor class group based on Mumford Representation. After the conversion we can obtain the equation for the Jacobian Variety curve. Here we denote the Jacobian curve as $y = l(x)$.

The divisor class group, D_1 for the point P_1, P_2, P_3 and P_4, for the point as shown below:

$$D_1 = \prod_{r=1}^{2}(x - x_{P1})(x - x_{P2})(x - x_{P3})(x - x_{P4}) - 4P_\infty$$

In Mumford form:

$$D_1 = [u_1(x), v_1(x)] = [x^4 + u_{13}x^3 + u_{12}x^2 + u_{11}x + u_{10}, v_{13}x^3 + v_{12}x^2 + v_{11}x + v_{10}]$$

The divisor class group, D_2 for the point Q_1 and Q_2, for the point as shown below:

$$D_2 = \prod_{r=1}^{2}(x - x_{Q1})(x - x_{Q2}) - 2P_\infty$$

In Mumford form:

$$D_2 = [u_2(x), v_2(x)] = [x^2 + u_{21}x + u_{20}, v_{21}x + v_{20}]$$

The divisor class group, D_1 for the point R_1, R_2, R_3 and R_4, for the point as shown below:

$$D_3 = \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2})(x - x_{R3})(x - x_{R4}) - 4P_\infty$$

In Mumford form:

$$D_3 = [u_3(x), v_3(x)] = [x^4 + u_{33}x^3 + u_{32}x^2 + u_{31}x + u_{30}, v_{33}x^3 + v_{32}x^2 + v_{31}x + v_{30}]$$

From the figure 3 above we can assert the polynomial expression of $l(x)$ to be $l(x) = l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. As shown in the previous section, here there are eight intersecting points with the hyper elliptic curve. At the intersecting points the y-coordinates are same. Therefore we can write, at the intersection points $l(x) = v(x)$ or $l(x) - v(x) \equiv 0\ mod\ u(x)$ since we have to perform polynomial reduction.

For the intersecting points P_1, P_2, P_3 and P_4, we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$((l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (v_{13}x^3 + v_{12}x^2 + v_{11}x + v_{10}) \equiv 0 \bmod x^4 + u_{13}x^3 + u_{12}x^2 + u_{11}x + u_{10}$$

$$\text{Or, } ((l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) \equiv (v_{13}x^3 + v_{12}x^2 + v_{11}x + v_{10}) \bmod x^4 + u_{13}x^3 + u_{12}x^2 + u_{11}x + u_{10}$$

For the intersecting points Q_1 and Q_2, we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$((l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (v_{21}x + v_{20}) \equiv 0 \bmod x^2 + u_{21}x + u_{20}$$

$$\text{Or, } (l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) \equiv (v_{21}x + v_{20}) \bmod x^2 + u_{21}x + u_{20}$$

By reducing the L.H.S with the polynomial expression $x^4 + u_{13}x^3 + u_{12}x^2 + u_{11}x + u_{10}$ and $x^2 + u_{21}x + u_{20}$ comparing with the R.H.S, we get six simultaneous equations.

$$l_5(u_{13}^2 - u_{12}) - l_4u_{13} + l_3 = v_{13} \qquad \text{EQN 1}$$

$$l_5(u_{13}u_{12} - u_{11}) - l_4u_{12} + l_2 = v_{12} \qquad \text{EQN 2}$$

$$l_5(u_{13}u_{11} - u_{10}) - l_4u_{11} + l_{11} = v_{11} \qquad \text{EQN 3}$$

$$l_5(u_{13}u_{11}) - l_4u_{10} + l_0 = v_{10} \qquad \text{EQN 4}$$

$$l_5(u_{21}^4 + u_{20}^2 - 3u_{21}^2u_{20}) + l_4(2u_{21}u_{20} - u_{21}^3) + l_3(u_{21}^2 - u_{20}) - l_2u_{21} + l_1 = v_{21} \qquad \text{EQN 5}$$

$$l_5(u_{21}^3u_{20} - 2u_{21}u_{20}^2) + l_4(u_{20}^2 - u_{21}^2u_{20}) + l_3(u_{21}u_{20}) - l_2u_{20} + l_0 = v_{20} \qquad \text{EQN 6}$$

As discussed before, we can introduce new variables to make the equation less tedious. The variable we choose is shown below to ease the computation:

$$\Delta_0 = u_{21}^4 + u_{20}^2 - 3u_{21}^2u_{20} \qquad \Delta_1 = 2u_{21}u_{20} - u_{21}^3 \qquad \Delta_2 = u_{21} - u_{20}$$

$$\Delta_3 = u_{21}^3u_{20} - 2u_{21}u_{20}^2 \qquad \Delta_4 = u_{20}^2 - u_{21}^2u_{20} \qquad \Delta_5 = u_{21}u_{20}$$

$$\epsilon_o = u_{13}^2 - u_{12} \qquad \epsilon_1 = u_{13}u_{12} - u_{11} \qquad \epsilon_2 = u_{13}u_{11} - u_{10}$$

$$\epsilon_4 = u_{13}u_{10}$$

We can re – write the EQNS to make it easier to solve:

$$l_5\epsilon_o - l_4u_{13} + l_3 = v_{13} \quad \text{EQN 1}$$

$$l_5\epsilon_1 - l_4u_{12} + l_2 = v_{12} \quad \text{EQN 2}$$

$$l_5\epsilon_2 - l_4u_{11} + l_{11} = v_{11} \quad \text{EQN 3}$$

$$l_5\epsilon_3 - l_4u_{10} + l_0 = v_{10} \quad \text{EQN 4}$$

$$\Delta_0l_5 + \Delta_1l_4 + \Delta_2l_3 - l_2u_{21} + l_1 = v_{21} \quad \text{EQN 5}$$

$$l_5\Delta_3 + l_4\Delta_4 + l_3\Delta_5 - l_2u_{20} + l_0 = v_{20} \quad \text{EQN 6}$$

Subtracting the EQN 6 from EQN 5:

$$(u_{20}\Delta_0 - u_{21}\Delta_3)l_5 + (u_{20}\Delta_1 - u_{21}\Delta_4)l_4 + (u_{20}\Delta_2 - u_{21}\Delta_3)l_3 + u_{20}l_1 - u_{21}l_0 = u_{20}v_{21} - u_{21}v_{20} \quad \text{EQN 7}$$

Adding the EQN 7 and EQN 4:

$$(u_{20}\Delta_0 - u_{21}\Delta_3 + u_{21}\epsilon_3)l_5 + (u_{20}\Delta_1 - u_{21}\Delta_4 + u_{21}u_{20})l_4 + (u_{20}\Delta_2 - u_{21}\Delta_5)l_3 + u_{20}l_1 = u_{20}v_{21} - u_{21}v_{20} + u_{21}v_{10} \quad \text{EQN 8}$$

Subtracting the EQN 3 from EQN 8:

$$(u_{20}\Delta_0 - u_{21}\Delta_3 + u_{21}\epsilon_3 - u_{20}\epsilon_2)l_5 + (u_{20}\Delta_1 - u_{21}\Delta_4 - u_{21}u_{10} + u_{11}u_{20})l_4 + (u_{20}\Delta_2 - u_{21}\Delta_5)l_3 = u_{21}(-v_{20} - v_{10}) \quad \text{EQN 9}$$

As discussed before, we can introduce new variables to make the equation less tedious. The variable we choose is shown below to ease the computation:

$$B_0 = u_{20}\Delta_0 - u_{21}\Delta_3 + u_{21}\epsilon_3 - u_{20}\epsilon_2$$

$$B_1 = u_{20}\Delta_1 - u_{21}\Delta_4 - u_{21}u_{10} + u_{11}u_{20}$$

$$B_2 = u_{20}\Delta_2 - u_{21}\Delta_5$$

Subtracting the EQN 1 from EQN 9:

$$(B_0 - B_2\epsilon_0)l_5 + (B_1 + u_{13}B_2)l_4 = u_{21}(-v_{20} - v_{10}) - B_2v_{13} \quad \text{EQN 10}$$

We can simplify the EQN 10 by introducing new variables, as shown below:

$$C_3 = B_0 - B_2\epsilon_0$$
$$C_4 = B_1 + u_3B_2$$
$$C_5 = u_{21}(-v_{20} - v_{10}) - B_2v_{13}$$

$$C_3l_5 + C_4l_4 = C_5 \quad \text{EQN 10}$$

Similarly, we can represent the EQN 7 in a similar manner as shown below:

$$B_3 = u_{20}\Delta_0 - u_{21}\Delta_3$$
$$B_4 = u_{20}\Delta_1 - u_{21}\Delta_4$$
$$B_5 = u_{20}\Delta_2 - u_{21}\Delta_3$$

$$B_3l_5 + B_4l_4 + B_5l_3 + u_{20}l_1 - u_{21}l_0 = u_{20}v_{21} - u_{21}v_{20} \quad \text{EQN 7}$$

The table below shows how the consecutive equations are formed by addition, subtraction and elimination:

EQN 7 + EQN 4	$(u_{21}\epsilon_3 + B_3)l_5 + (-u_{21}u_{10} + B_4)l_4 + B_3l_3 + u_{20}l_1 = u_{20}v_{21} - u_{21}v_{20} + u_{21}v_{10}$.	EQN 11
EQN 11 - EQN 3	$(u_{21}\epsilon_3 + B_3 - u_{20}\epsilon_2)l_5 + (u_{21}u_{10} - u_{21}u_{10} + B_4)l_4 + B_5l_3 = u_{20}v_{21} - u_{21}v_{20} + u_{21}v_{10} - u_{20}v_{11}$.	EQN 12
EQN 12 - EQN 1	$(u_{21}\epsilon_3 + B_3 - u_{20}\epsilon_2 - \epsilon_0B_5)l_5 + (u_{20}u_{11} - u_{21}u_{10} + B_4 + u_{12}B_5)l_4 = u_{20}v_{21} - u_{21}v_{20} + u_{21}v_{10} - u_{20}v_{11} - v_{13}B_5$.	EQN 13

Similarly, we can represent the EQN 13 in a similar manner as shown below:

$$C_0 = u_{21}\epsilon_3 + B_3 - u_{20}\epsilon_2 - \epsilon_0B_5$$

$$C_1 = u_{20}u_{11} - u_{21}u_{10} + B_4 + u_{12}B_5$$
$$C_2 = u_{20}v_{21} - u_{21}v_{20} + u_{21}v_{10} - u_{20}v_{11} - v_{13}B_5$$

$$C_0 l_5 + C_1 l_4 = C_2 \qquad \text{EQN 13}$$
$$C_3 l_5 + C_4 l_4 = C_5 \qquad \text{EQN 10}$$

Solving the equations simultaneously, we will get the expression for l_0, l_1, l_2, l_3, l_4 and l_5, as shown below:

$$inv = (C_4C_2 - C_3C_1)^{-1} \qquad l_5 = inv \cdot (C_4C_2 - C_5C_1) \qquad l_0 = v_{10} + u_{10}l_4 - \epsilon_3 l_5$$

$$l_4 = inv \cdot (C_5C_0 - C_3C_3) \qquad l_3 = v_{13} + u_{13}l_4 - \epsilon_0 l_5$$

$$l_2 = v_{12} + u_{12}l_4 - \epsilon_1 l_5 \qquad l_1 = v_{11} + u_{11}l_4 - \epsilon_1 l_5$$

By using the same methodology used in the earlier sections, we find:

$$u_{33} = L_{(5,4)} - N_6 - 1$$

$$u_{32} = L_{(5,3)} + l_4^2 - u_{33}N_5 - N_4$$

$$u_{31} = L_{(5,2)} + L_{(5,3)} + L_{(4,3)} - f_4 - u_{32}N_5 - u_{33}N_4 - N_3$$

$$u_{30} = L_{(5,1)} + L_{(4,2)} + l_3^2 - f_6 - u_{31}N_5 - u_{32}N_4 - u_{33}N_3 - N_2$$

where the variables are detailed below:

$$N_6 = 1 \qquad N_5 = u_{21} + u_{13} \qquad N_4 = u_{20} + u_{21}u_{13} + u_{12}$$
$$N_3 = u_{20}u_{13} + u_{21}u_{12} + u_{11} \qquad N_2 = u_{20}u_{12} + u_{21}u_{11} + u_{10} \qquad L_{(5,3)} = 2l_5l_3$$
$$L_{(5,4)} = 2l_5l_4 \qquad L_{(5,2)} = 2l_5l_2 \qquad L_{(5,1)} = 2l_5l_1$$
$$L_{(4,2)} = 2l_4l_2 \qquad L_{(4,3)} = l_4l_3$$

Similarly, we can get the result for v_{32}, v_{31} and v_{30} by solving the equation:

$$l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0 \equiv$$

$$\equiv (v_{33}x^3 + v_{32}x^2 + v_{31}x + v_{30}) \bmod x^4 + u_{33}x^3 + u_{32}x^2 + u_{31}x + u_{30}$$

After expanding the equation and comparing the coefficients L.H.S ≡ R.H.S we get:

$$v_{33} = l_5(u_{33}^2 - u_{32}) + l_4(-u_{33}) + l_3$$

$$v_{32} = l_5(u_{33}u_{32} - u_{31}) + l_4(-u_{32}) + l_2$$

$$v_{31} = l_5(u_{33}u_{31} - u_{30}) + l_4(-u_{31}) + l_1$$

$$v_{32} = l_5(u_{33}u_{30}) + l_4(-u_{30}) + l_0$$

5.3.2 Computational Complexity of General Addition Explicit Formula for HEC of $g = 4$

In the section above, we have proposed and derived a General Explicit Formula for Addition on a HEC of genus = 4. For practical purpose one can eliminate the function of $h(x)$ and the co-efficient of f_4 from the function $f(x)$. The computational complexity for the General Addition Explicit Formulae is defined as (no. of Inverses, no. of Multiplication, no. of Squaring).

Finite Field	Curve Properties		Cost	
		Inverses (I)	Multiplication (M)	Squaring (s)
$\mathbb{F}_q$	$h(x), f(x)$	1	74	9
$\mathbb{F}_q$	$h(x) = 0$	1	71	9

Table 5.6: Complexity comparison between the explicit formulae for HEC of genus 4 for different curve property.

5.3.3 Comparison of proposed explicit formulae and existing (Addition) for HEC $g = 4$.

The proposed work is compared with the existing addition for the HEC for genus 4 has been compared. The table below shows the list of complexity comparison.

Previous Work	Finite Field	Cost			Improvement Percentage (%)
		Inverse (I)	Multiplication (M)	Squaring (S)	
Cantor [23]	$\mathbb{F}_q$	6	386 M/S		100
Nagao [23]	$\mathbb{F}_q$	2	289 M/S		135
C. Paar [33]	$\mathbb{F}_q$	2	160	4	160
This work	$\mathbb{F}_q$	1	71	9	181

Table 5.7: Comparison between the explicit formulas for (genus = 4) curves over $\mathbb{F}_q$ of previous work and the present work.

ALGORITHM V

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 4, HEC: $y^2 + (h_2x^2 + h_1x + h_0)y = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 10

Input	Genus 4 HEC: $y^2 + h(x)y = f(x)$; $h(x) = h_2x^2 + h_1x + h_0$; $f(x) = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 +$ $f_1x + f_0$; Divisor $D_1 = [u_1(x), v_1(x)]$, $D_2 = [u_2(x), v_2(x)]$; $u_1(x) = x^4 + u_{13}x^3 + u_{12}x^2 + u_{11}x + u_{10}$, $v_1(x) = v_{13}x^3 + v_{12}x^2 + v_{11}x + v_{10}$; $u_2(x) = x^2 + u_{21}x + u_{20}$, $v_1(x) = v_{21}x + v_{20}$;	
Output	$D_3 = [u_3(x), v_3(x)] = D_1 + D_2$, $u_3(x) = x^4 + u_{33}x^3 + u_{32}x^2 + u_{31}x + u_{30}$, $v_3(x) = v_{33}x^3 + v_{32}x^2 + v_{31}x + v_{30}$;	Cost (I, M, S)

Step	Expressions	Cost
1	$U_{21} = u_{21}^2$, $U_{20} = u_{20}^2$, $U_{2120} = u_{21}u_{20}$;	(0,1,2)
2	$\Delta_0 = U_{21}^2 + U_{20} - 3U_{21}u_{20}$, $\Delta_1 = 2U_{2120} - U_{2120}u_{21}$, $\Delta_2 = u_{21} - u_{20}$, $\Delta_3 = U_{21}U_{2120} - 2U_{20}u_{21}$, $\Delta_4 = U_{20} - U_{21}u_{20}$, $\Delta_5 = U_{2120}$;	(0,5,1)
3	$\epsilon_o = u_{13}^2 - u_{12}$, $\epsilon_1 = u_{13}u_{12} - u_{11}$, $\epsilon_2 = u_{13}u_{11} - u_{10}$, $\epsilon_4 = u_{13}u_{10}$;	(0,3,1)
4	$M_0 = u_{20}\Delta_0$, $M_1 = u_{20}\Delta_1$, $M_2 = u_{20}\Delta_2$, $M_3 = u_{21}\Delta_3$, $M_4 = u_{21}\Delta_4$, $M_5 = u_{21}\Delta_5$;	(0,6,0)
5	$D_0 = u_{21}\epsilon_3 - u_{20}\epsilon_2$, $D_1 = -u_{21}u_{10} - u_{11}u_{20}$, $D_2 = u_{21}v_{20}$, $D_3 = u_{21}v_{10}$;	(0,4,0)
6	$B_0 = M_0 - M_3 + D_0$, $B_1 = M_1 - M_4 + D_1$, $B_2 = M_2 - M_5$, $B_3 = M_0 -$ M_3, $B_4 = M_1 - M_4$, $B_5 = M_2 - M_3$;	(0,0,0)
7	$C_0 = D_0 + B_3 - \epsilon_0B_5$, $C_1 = D_1 + B_4 - u_2B_5$, $C_2 = u_{20}v_{21} - D_2 +$ $D_3 - u_{20}v_{11} - v_{13}B_5$, $C_3 = B_0 - \epsilon_0B_2$, $C_4 = B_1 + u_{13}B_2$, $C_5 = -D_2 - D_3 - v_{13}B_2$;	(0,8,0)
8	$inv = (C_4C_0 - C_3C_1)^{-1}$;	(1,2,0)
9	$l_5 = inv \cdot (C_4C_2 - C_5C_1)$, $l_4 = inv \cdot (C_5C_0 - C_3C_2)$, $l_3 = v_3 + u_3l_4 - \epsilon_0l_5$, $l_2 = v_2 + u_2l_4 - \epsilon_1l_5$, $l_1 = v_1 + u_1l_4 - \epsilon_2l_5$, $l_0 = v_0 + u_0l_4 - \epsilon_3l_5$;	(0,14,0)
10	$N_6 = 1$, $N_5 = u_{21} + u_{13}$, $N_4 = u_{20} + u_{21}u_{13} + u_{12}$, $N_3 = u_{20}u_{13} + u_{21}u_{12} + u_{11}$, $N_2 = u_{20}u_{12} + u_{21}u_{11} + u_{10}$;	(0,5,0)

11	$L_{(5,3)} = 2l_5l_3$, $L_{(5,4)} = 2l_5l_4$, $L_{(5,2)} = 2l_5l_2$, $L_{(5,1)} = 2l_5l_1$, $L_{(4,2)} = 2l_4l_2$, $L_{(4,3)} = l_4l_3$;	(0,6,2)
12	**Compute $\boldsymbol{u_3(x) = x^4 + u_{33}x^3 + u_{32}x^2 + u_{31}x + u_{30}}$:** $u_{33} = L_{(5,4)} - N_6 - 1$, $u_{32} = L_{(5,3)} + l_4^2 - f_8 - u_{33}N_5 - N_4$, $u_{31} = L_{(5,2)} + L_{(5,3)} + L_{(4,3)} + h_2l_5 - f_4 - u_{32}N_5 - u_{33}N_4 - N_3$, $u_{30} = L_{(5,1)} + L_{(4,2)} + l_3^2 + h_2l_4 + h_1l_5 - f_6 - u_{31}N_5 - u_{32}N_4 - u_{33}N_3 - N_2$;	(0,9,2)
13	**Compute $\boldsymbol{v_3(x) = v_{33}x^3 + v_{32}x^2 + v_{31}x + v_{30}}$:** $v_{33} = l_5(u_{33}^2 - u_{32}) + l_4(-u_{33}) + l_3$, $v_{32} = l_5(u_{33}u_{32} - u_{31}) + l_4(-u_{32}) + l_2$, $v_{31} = l_5(u_{33}u_{31} - u_{30}) + l_4(-u_{31}) + l_1$, $v_{32} = l_5(u_{33}u_{30}) + l_4(-u_{30}) + l_0$;	(0,11,1)
Sum		(1,74, 9)

ALGORITHM VI

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 4, HEC: $y^2 = x^9 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 10

Input	Genus 4 HEC: $y^2 + h(x)y = f(x)$; $f(x) = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D_1 = [u_1(x), v_1(x)]$, $D_2 = [u_2(x), v_2(x)]$; $u_1(x) = x^4 + u_{13}x^3 + u_{12}x^2 + u_{11}x + u_{10}$, $v_1(x) = v_{13}x^3 + v_{12}x^2 + v_{11}x + v_{10}$; $u_2(x) = x^2 + u_{21}x + u_{20}$, $v_1(x) = v_{21}x + v_{20}$;	
Output	$D_3 = [u_3(x), v_3(x)] = D_1 + D_2$, $u_3(x) = x^4 + u_{33}x^3 + u_{32}x^2 + u_{31}x + u_{30}$, $v_3(x) = v_{33}x^3 + v_{32}x^2 + v_{31}x + v_{30}$;	Cost (I, M, S)

Step	Expressions	Cost
1	$U_{21} = u_{21}^2$, $U_{20} = u_{20}^2$, $U_{2120} = u_{21}u_{20}$;	(0,1,2)
2	$\Delta_0 = U_{21}^2 + U_{20} - 3U_{21}u_{20}$, $\Delta_1 = 2U_{2120} - U_{2120}u_{21}$, $\Delta_2 = u_{21} - u_{20}$, $\Delta_3 = U_{21}U_{2120} - 2U_{20}u_{21}$, $\Delta_4 = U_{20} - U_{21}u_{20}$, $\Delta_5 = U_{2120}$;	(0,5,1)
3	$\epsilon_o = u_{13}^2 - u_{12}$, $\epsilon_1 = u_{13}u_{12} - u_{11}$, $\epsilon_2 = u_{13}u_{11} - u_{10}$, $\epsilon_4 = u_{13}u_{10}$;	(0,3,1)
4	$M_0 = u_{20}\Delta_0$, $M_1 = u_{20}\Delta_1$, $M_2 = u_{20}\Delta_2$, $M_3 = u_{21}\Delta_3$, $M_4 = u_{21}\Delta_4$, $M_5 = u_{21}\Delta_5$;	(0,6,0)
5	$D_0 = u_{21}\epsilon_3 - u_{20}\epsilon_2$, $D_1 = -u_{21}u_{10} - u_{11}u_{20}$, $D_2 = u_{21}v_{20}$, $D_3 = u_{21}v_{10}$;	(0,4,0)
6	$B_0 = M_0 - M_3 + D_0$, $B_1 = M_1 - M_4 + D_1$, $B_2 = M_2 - M_5$, $B_3 = M_0 - M_3$, $B_4 = M_1 - M_4$, $B_5 = M_2 - M_3$;	(0,0,0)
7	$C_0 = D_0 + B_3 - \epsilon_0B_5$, $C_1 = D_1 + B_4 - u_2B_5$, $C_2 = u_{20}v_{21} - D_2 + D_3 - u_{20}v_{11} - v_{13}B_5$, $C_3 = B_0 - \epsilon_0B_2$, $C_4 = B_1 + u_{13}B_2$, $C_5 = -D_2 - D_3 - v_{13}B_2$;	(0,8,0)
8	$inv = (C_4C_0 - C_3C_1)^{-1}$;	(1,2,0)
9	$l_5 = inv \cdot (C_4C_2 - C_5C_1)$, $l_4 = inv \cdot (C_5C_0 - C_3C_2)$, $l_3 = v_3 + u_3l_4 - \epsilon_0l_5$, $l_2 = v_2 + u_2l_4 - \epsilon_1l_5$, $l_1 = v_1 + u_1l_4 - \epsilon_2l_5$, $l_0 = v_0 + u_0l_4 - \epsilon_3l_5$;	(0,14,0)
10	$N_6 = 1$, $N_5 = u_{21} + u_{13}$, $N_4 = u_{20} + u_{21}u_{13} + u_{12}$, $N_3 = u_{20}u_{13} + u_{21}u_{12} + u_{11}$, $N_2 = u_{20}u_{12} + u_{21}u_{11} + u_{10}$;	(0,5,0)

11	$L_{(5,3)} = 2l_5l_3, L_{(5,4)} = 2l_5l_4, L_{(5,2)} = 2l_5l_2, L_{(5,1)} = 2l_5l_1,$ $L_{(4,2)} = 2l_4l_2, L_{(4,3)} = l_4l_3;$	(0,6,2)
12	**Compute $\boldsymbol{u_3(x) = x^4 + u_{33}x^3 + u_{32}x^2 + u_{31}x + u_{30}}$:** $u_{33} = L_{(5,4)} - N_6 - 1, u_{32} = L_{(5,3)} + l_4^2 - u_{33}N_5 - N_4,$ $u_{31} = L_{(5,2)} + L_{(5,3)} + L_{(4,3)} - f_4 - u_{32}N_5 - u_{33}N_4 - N_3,$ $u_{30} = L_{(5,1)} + L_{(4,2)} + l_3^2 - f_6 - u_{31}N_5 - u_{32}N_4 - u_{33}N_3 - N_2;$	(0,6,2)
13	**Compute $\boldsymbol{v_3(x) = v_{33}x^3 + v_{32}x^2 + v_{31}x + v_{30}}$:** $v_{33} = l_5(u_{33}^2 - u_{32}) + l_4(-u_{33}) + l_3,$ $v_{32} = l_5(u_{33}u_{32} - u_{31}) + l_4(-u_{32}) + l_2,$ $v_{31} = l_5(u_{33}u_{31} - u_{30}) + l_4(-u_{31}) + l_1,$ $v_{32} = l_5(u_{33}u_{30}) + l_4(-u_{30}) + l_0;$	(0,11,1)
Sum		(1,71, 9)

5.4 Explicit Formulae (Doubling) for HEC of genus 2

As discussed in the previous sections that we can apply the group law operations in the Jacobian and we can generate explicit formulae for the hyper elliptic curve of genus 2, 3 and 4 from the intersecting points. Similar to the elliptic curve tangent and chord method, we can also generate doubling explicit formulae for HECs. The figure below is the graphical representation of the Jacobian variety curve and the hyper elliptic curve of genus 2.

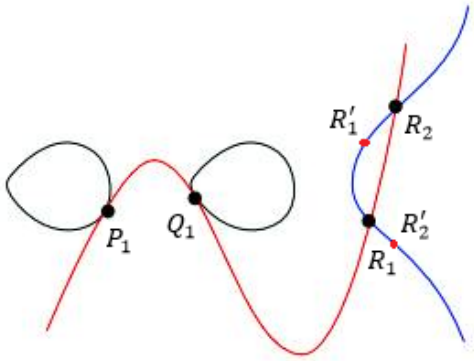

Figure 5.4: Hyper Elliptic Curve of genus 2 as it touch the Jacobian variety curve.

Since the points of touch between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$[P_1] + [P_1] + [Q_1] + [Q_1] + [R_1] + [R_2] = 0.$$

$$2[P_1] + 2[Q_1] = -[R_1] - [R_2] = [R_1'] + [R_2']$$

The Cartesian points or the affine space points P_1 and Q_1 could be transformed to individual divisor class group based on Mumford Representation, which is to be discussed in a separate section.

5.4.1 Generating Doubling Explicit Formulae for HEC of genus 2

Let's consider a general Hyper Elliptic Curve C of genus $g = 2$ over the finite field $\mathbb{F}_q$:

HEC: $y^2 = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

Since, we will be working with this particular expression $[P_1] + [P_1] + [Q_1] + [Q_1] + [R_1] + [R_2] = 0$. We can convert the coordinate $[P_1]$ and $[Q_1]$ to polynomial expression using Mumford.

The divisor class group, D for the point $[P_1]$, $[Q_1]$ and for the point $[R_1]$, $[R_1]$ is shown below:

$$D = \prod_{r=1}^{2}(x - x_{P1})(x - x_{Q2}) - 2P_\infty = [x^2 + u_1x + u_0, v_1x + v_0]$$

$$D' = \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2}) - 2P_\infty = [x^2 + u_1'x + u_0', v_1'x + v_0']$$

From the figure 4 above we can assert the polynomial expression of $l(x)$ to be $l(x) = l_3x^3 + l_2x^2 + l_1x + l_0$. The Jacobian curve $y = l(x)$ is a cubic function since we can see in the graph that the function has two extreme points and intersecting with the hyper elliptic curve with four Cartesian points.

At the intersecting points the y-coordinates are same. Therefore we can write, at the intersection points $l(x) = v(x)$ or $l(x) - v(x) \equiv 0\ mod\ u(x)$ since we have to perform polynomial reduction.

For the intersecting points $[P_1]$ and $[Q_1]$, we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$(l_3x^3 + l_2x^2 + l_1x + l_0) - (v_1x + v_0) \equiv 0 \bmod x^2 + u_1x + u_0$$

Or, $(l_3x^3 + l_2x^2 + l_1x + l_0) \equiv (v_1x + v_0) \bmod x^2 + u_1x + u_0$

By reducing the L.H.S with the polynomial expression $x^2 + u_1x + u_0$ and comparing with the R.H.S, we get two equations.

$$(u_1^2 - u_0)l_3 + (-u_1)l_2 + l_1 \equiv v_1 \qquad \text{EQN 1}$$

$$(u_1u_0)l_3 + (-u_0)l_2 + l_0 \equiv v_0 \qquad \text{EQN 2}$$

Since, the Jacobian variety curve touches at two points on the hyper elliptic curve. We can conclude that at that point the gradient is same. So we find the derivative of the Jacobian and the hyper elliptic curves:

Derivative of Jacobian Curve:

$$\frac{dy}{dx} = 3l_3x^2 + 2l_2x + l_1$$

Derivative of the hyper elliptic curve:

$$2y \cdot \frac{dy}{dx} = 5x^4 + 4f_4x^4 + 3f_3x^3 + 2f_2x + f_1$$

By substituting $v(x) = v_1x + v_0$ of the divisor $D = [u(x), v(x)]$ and the derivative of the Jacobian curve, we get the expression as shown below:

$$2(3l_3x^2 + 2l_2x + l_1) \cdot (v_1x + v_0) = 5x^4 + 4f_4x^4 + 3f_3x^3 + 2f_2x + f_1 \bmod (x^2 + u_1x + u_0)$$

By reducing the L.H.S with the polynomial expression $x^2 + u_1x + u_0$ and comparing with the R.H.S, we get another pair of equations.

$$\Delta_0 l_3 + \Delta_1 l_2 + \Delta_2 l_1 \equiv \epsilon_0 \qquad \text{EQN 3}$$

$$\Delta_3 l_3 + \Delta_4 l_2 + \Delta_5 l_1 \equiv \epsilon_1 \qquad \text{EQN 4}$$

Where the variables Δ_0, Δ_1, Δ_2, Δ_3, Δ_4, Δ_5, ϵ_0 and ϵ_1 are shown below:

$$\Delta_0 = 6v_1(u_1^2 - u_0) - 6u_1v_0 \qquad \Delta_1 = -4u_1v_1 + 4v_0$$

$$\Delta_2 = 2v_1 \qquad \Delta_3 = 6u_1u_0v_1 - 6u_0v_0$$

$$\Delta_4 = -4u_0v_1 \qquad \Delta_2 = 2v_0$$

$$\epsilon_0 = 5[-u_1(u_1^2 - u_0) + u_1u_0] + 3f_3(-u_1) + 2f_2 \qquad \epsilon_1 = 5[-u_0(u_1^2 - u_0)] + 3f_3(-u_0) + f_1$$

By solving the equations EQN 1, 2, 3 and 4 we can find the co-efficient of the Jacobian variety curve $l(x) = l_3x^3 + l_2x^2 + l_1x + l_0$. The results are shown below:

$$l_3 = \frac{M_4\theta_1 - M_1\theta_2}{M_0M_4 - M_1M_3} \qquad l_2 = \frac{M_0\theta_2 - M_3\theta_1}{M_0M_4 - M_1M_3}$$

$$l_1 = v_1 + u_1l_2 - (u_1^2 - u_0)l_3 \qquad l_0 = v_0 + u_0l_2 - (u_1u_0)l_3$$

Where the variable are used to simplify the calculation and to make the result appear less tedious. The variable used here is shown below:

$$M_0 = \Delta_3 - \Delta_5(u_1^2 - u_0) \qquad M_1 = \Delta_4 - \Delta_5(-u_1) \qquad M_3 = \Delta_0 - \Delta_2(u_1^2 - u_0)$$

$$M_4 = \Delta_1 - \Delta_2(-u_1) \qquad \theta_1 = \epsilon_1 - \Delta_5v_1 \qquad \theta_2 = \epsilon_0 - \Delta_2v_1$$

At the point of touch, in this case it is the Jacobian Variety and the Hyper Elliptic Curve of genus 2, the value of the y on both curve is the same. So we can replace the y expression of the hyper elliptic curve by the Jacobian variety curve $y = l_3x^3 + l_2x^2 + l_1x + l_0$.

HEC: $y^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$
The Jacobian Variety Curve: $y_{JC} = l_3x^3 + l_2x^2 + l_1x + l_0$

Substituting y in HEC with y_{JC}:

$$y_{JC}{}^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\text{Or, } (l_3x^3 + l_2x^2 + l_1x + l_0)^2 = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\text{Or, } (l_3x^3 + l_2x^2 + l_1x + l_0)^2 - (x^5 + f_3x^3 + f_2x^2 + f_1x + f_0) = 0 \qquad \text{[EXP 1]}$$

Solving the expression [EXP 1] would give the coordinates of the intersecting points and the point of touch. However, we intended to get the results in Mumford form. The expression below is explicitly expressed in the Mumford form if we solve and compare the L.H.S and R.H.S.

$$\prod_{r=1}^{2}(x - x_{P1})(x - x_{Q1}) \cdot \prod_{r=1}^{2}(x - x_{P1})(x - x_{Q1}) \cdot \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2})$$

$$\equiv$$

$$(l_3x^3 + l_2x^2 + l_1x + l_0)^2 - (x^5 + f_3x^3 + f_2x^2 + f_1x + f_0)$$

Comparing the coefficients of the L.H.S with the R.H.S we get:

$$u_1' = 2l_3l_2 - 2u_1 - 1$$

$$u_0' = 2l_3l_1 + l_2^2 - 2u_0 - u_1^2 - 2u_1'u_1$$

Similarly, we can get the result for v_1' and v_0' by solving the equation:

$$l(x) \bmod (x^2 + u_1'x + u_0') \equiv v_1'x + v_0'$$

Or, $l_3x^3 + l_2x^2 + l_1x + l_0 \bmod (x^2 + u_1'x + u_0') \equiv v_1'x + v_0'$

After expanding the equation and comparing the coefficients L.H.S ≡ R.H.S we get:

$$v_1' = (u_1'^2 - u_0')l_3 - u_1'l_2 + l_1$$

$$v_0' = (u_1'u_0')l_3 - u_0'l_2 + l_0$$

5.4.2 Comparison of proposed and existing Explicit Formulae (Doubling) for HEC $g = 2$.

The proposed works has been compared with the Explicit Formulae for doubling for the HEC for genus 2 has been compared. The table below presents list the complexity comparison table.

Previous Work	Finite Field	Curve Properties	Cost			Improvement Percentage (%)
			Inverses (I)	Multiplication (M)	Squaring (S)	
Harley [11,25]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	2	30	-	100
Lange [26]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	2	24	6	103.4
Matsuo [13]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	2	27	-	104.3
Takahashi [15]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	29	-	130
Miyamoto [14]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	27	-	132.9
Lange [27]	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	22	5	135.7
This work	$\mathbb{F}_q$	$h(x) = 0$, $f_4 = 0$	1	23	3	136.7

Table 5.8: Comparison between the explicit formulas (doubling) for (genus = 2) curves over $\mathbb{F}_q$ of previous work and the present work.

TABLE VII

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 2, HEC: $y^2 = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 4

Input	Genus 2 HEC: $y^2 = f(x)$; $h(x) = 0$; $f(x) = x^5 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D = [u(x), v(x)]$; $u(x) = x^2 + u_1x + u_0$, $v(x) = v_1x + v_0$,	
Initial input variables	$U_1 = u_1^2$; $U_0 = u_1u_0$; $UV_{01} = u_0v_1$; $UV_{10} = u_1v_0$; $UV_{11} = u_1v_1$; $UV_{00} = u_0v_0$;	
Output	$D' = [u', v'] = 2D$, $u'(x) = x^2 + u_1'x + u_0'$, $v'(x) = x^2 + v_1'x + v_0'$;	Cost (I, M, S)

Step	Expressions	Cost
1	$\Delta_0 = 6v_1U_1 - UV_{10}$; $\Delta_1 = -4UV_{11} + 4v_0$; $\Delta_2 = 2v_1$; $\Delta_3 = 6v_1U_0 - 6UV_{00}$; $\Delta_4 = -4UV_{01}$; $\Delta_5 = 2v_0$;	(0,2,0)
2	$\epsilon_0 = 5(-u_1U_1 + 2U_0) - 3f_3u_1 + 2f_2$; $\epsilon_1 = 5[-u_0U_0 + u_0^2] - 3f_3u_0 + f_1$;	(0,3,2)
3	$M_0 = \Delta_3 - \Delta_5(U_1 - u_0)$; $M_1 = \Delta_4 - \Delta_5(-u_1)$; $M_3 = \Delta_0 - \Delta_2(U_1 - u_0)$; $M_4 = \Delta_1 - \Delta_2(-u_1)$;	(0,3,0)
4	$\theta_1 = \epsilon_1 - \Delta_5v_1$; $\theta_2 = \epsilon_0 - \Delta_2v_1$;	
5	$inv = (M_0M_4 - M_1M_3)^{-1}$;	(1,2,0)
6	$l_3 = inv \cdot (M_4\theta_1 - M_1\theta_2)$; $l_2 = inv \cdot (M_0\theta_2 - M_3\theta_1)$; $l_1 = v_1 + u_1l_2 - (U_1 - u_0)l_3$; $l_0 = v_0 + u_0l_2 - (U_0)l_3$;	(0,8,0)
7	**Compute $\boldsymbol{u'(x) = x^2 + u_1'x + u_0'}$:** $u_1' = 2l_3l_2 - 2u_1 - 1$; $u_0' = 2l_3l_1 + l_2^2 - 2u_0 - U_1 - 2u_1'u_1$;	(0,3,1)
Initial output variables	$U_1' = u_1'^2$; $U_0' = u_1'u_0'$;	
8	**Compute $\boldsymbol{v'(x) = v_1'x + v_0'}$:** $v_1' = (U_1 - u_0')l_3 - u_1'l_2 + l_1$; $v_0' = U_0'l_3 - u_0'l_2 + l_0$;	(0,2,0)
Sum		(0,23,3)

5.5 Explicit Formulae (Doubling) for HEC of genus 3

As discussed in the section 5.4, we apply this concept to the HEC of genus 3. In this section we apply the same concept to develop a general addition explicit formulae algorithm for genus 3.

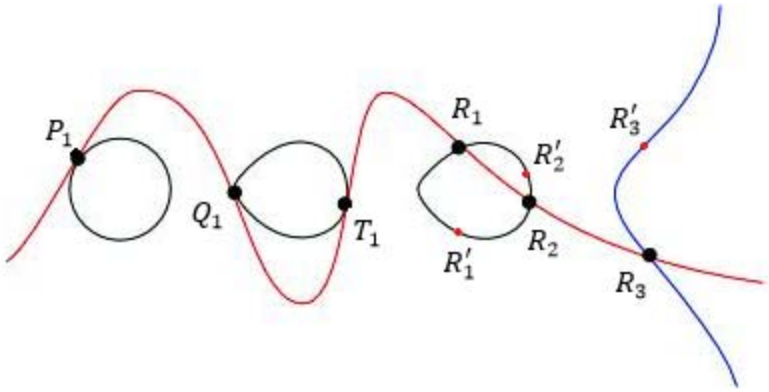

Figure 5.5: Hyper Elliptic Curve of genus 3 as it touch the Jacobian variety curve.

Since the points of touch between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$2[P_1] + 2[Q_1] + 2[S_1] = -[R_1] - [R_2] - [R_3] = [R_1'] + [R_2'] + [R_3'].$$

The Cartesian points or the affine space points P_1, Q_1 and S_1 could be transformed to individual divisor class group based on Mumford Representation, which is to be discussed in a separate section.

5.5.1 Generating Doubling Explicit Formulae for HEC of genus 3

Let's consider a general Hyper Elliptic Curve C of genus $g = 3$ over the finite field $\mathbb{F}_q$:

HEC: $y^2 = x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

Since, we will be working with this particular expression $[P_1] + [P_1] + [Q_1] + [Q_1] + [S_1] + [S_1] + [R_1] + [R_2] + [R_3] = 0$. We can convert the coordinate $[P_1]$ and $[Q_1]$ to polynomial expression using Mumford.

The divisor class group, D for the point $[P_1]$, $[Q_1]$, $[S_1]$ and for the point $[R_1]$, $[R_2]$, $[R_3]$ is shown below:

$D = \prod_{r=1}^{2}(x - x_P)(x - x_Q)(x - x_S) - 3P_\infty =$	$D = [u(x), v(x)]$ $u(x) = x^3 + u_2x^2 + u_1x + u_0$ $v(x) = v_2x^2 + v_1x + v_0$
$D' = \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2})(x - x_{R3}) - 3P_\infty$	$D' = [u'(x), v'(x)]$ $u'(x) = x^3 + u_2'x^2 + u_1'x + u_0'$ $v'(x) = v_2'x^2 + v_1'x + v_0'$

Table 5.9: Corresponding conversion of the Cartesian points to Mumford form for genus 3.

We can assert the polynomial expression of $l(x)$ to be $l(x) = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. The Jacobian curve $y = l(x)$ is a quantic function. At the intersecting points the y-coordinates are same. Therefore we can write, at the intersection points $l(x) = v(x)$ or $l(x) - v(x) \equiv 0\ mod\ u(x)$ since we have to perform polynomial reduction.
For the points $[P_1]$, $[Q_1]$, $[S_1]$ we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$(l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (v_2x^2 + v_1x + v_0) \equiv 0 \bmod x^3 + u_2x^2 + u_1x + u_0$$

$$l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0 \equiv (v_2x^2 + v_1x + v_0) \bmod x^3 + u_2x^2 + u_1x + u_0$$

By reducing the L.H.S with the polynomial expression $x^2 + u_1x + u_0$ and comparing with the R.H.S, we get three equations.

$$A_0l_4 - u_2l_3 + l_2 = v_2 \qquad \text{EQN 1}$$

$$A_1l_4 - u_1l_3 + l_1 = v_1 \qquad \text{EQN 2}$$

$$A_2l_4 - u_0l_3 + l_0 = v_0 \qquad \text{EQN 3}$$

Where the variables A_0, A_1 and A_2:

$$A_0 = (u_2^2 - u_1) \qquad A_1 = (u_2 u_1 - u_0) \qquad A_3 = (u_2 u_0)$$

Since, the Jacobian variety curve touches at two points on the hyper elliptic curve. We can conclude that at that point the gradient is same. So we find the derivative of the Jacobian and the hyper elliptic curves:

Derivative of Jacobian Curve:

$$\frac{dy}{dx} = 4l_4x^3 + 3l_3x^2 + 2l_2x + l_1$$

Derivative of the hyper elliptic curve:

$$2y \cdot \frac{dy}{dx} = 7x^6 + 5f_5x^4 + 4f_4x^3 + 3f_3x^2 + 2f_2x + f_1$$

By substituting $v(x) = v_2x^2 + v_1x + v_0$ of the divisor $D = [u(x), v(x)]$ and the derivative of the Jacobian curve, we get the expression as shown below:

$$2(4l_4x^3 + 3l_3x^2 + 2l_2x + l_1) \cdot (v_2x^2 + v_1x + v_0)$$
$$\equiv$$
$$5x^4 + 4f_4x^4 + 3f_3x^3 + 2f_2x + f_1 \bmod (x^2 + u_1x + u_0)$$

By reducing the L.H.S with the polynomial expression $x^2 + u_1x + u_0$ and comparing with the R.H.S, we get another three equations.

$$\Delta_0 l_4 + \Delta_1 l_3 + \Delta_2 l_2 + \Delta_3 l_1 = \theta_0 \qquad \text{EQN 4}$$

$$\Delta_4 l_4 + \Delta_5 l_3 + \Delta_6 l_2 + \Delta_7 l_1 = \theta_1 \qquad \text{EQN 5}$$

$$\Delta_8 l_4 + \Delta_9 l_3 + \Delta_{10} l_2 + \Delta_{11} l_1 = \theta_2 \qquad \text{EQN 6}$$

Where the variables Δ_0, Δ_1, Δ_2, Δ_3, Δ_4, Δ_5, Δ_6, Δ_7, Δ_8, Δ_9, Δ_{10}, Δ_{11}, θ_0, θ_1 and θ_2 shown below:

$$\Delta_0 = 8B_0v_2 + 8u_2u_2v_1 - 8u_1v_1 - u_2v_0 \qquad \Delta_1 = 6u_2u_2v_2 - 6u_1v_2 - 6u_2v_1 - 6v_0$$

$$\Delta_2 = -4u_2v_2 - 4v_1 \qquad \Delta_3 = -2v_2$$

$$\Delta_4 = 8B_1v_2 + 8u_2u_1v_1 - 8u_0v_1 - 8u_1u_2v_0 \qquad \Delta_5 = 6u_2u_1v_2 - 6u_0v_2 - 6u_1v_1$$

$$\Delta_6 = -4u_1v_2 + 4v_0 \qquad \Delta_7 = 2v_1$$

$$\Delta_8 = 8B_2v_2 + 8u_0u_2v_1 - 8u_0v_0 \qquad \Delta_9 = 6u_2u_0v_2 - 4u_0v_1$$

$$\Delta_{10} = -4u_0v_2 \qquad \Delta_{11} = 2v_1$$

$$\theta_0 = 7C_0 + 5f_5A_0 - 4f_4u_2 + 3f_3 \qquad \theta_1 = 7C_1 + 5f_5A_1 - 4f_4u_1 + 2f_2$$

$$\theta_2 = 7C_2 + 5f_5A_2 - 4f_4u_0 + f_1 \qquad C_0 = U_2^2 + 3u_2^2u_1 + u_2u_0 + u_1^2$$

$$C_1 = u_2^3u_1 - 2u_2u_1^2 - u_2^2u_0 - 2u_1u_0 \qquad C_2 = u_2^3u_0 - u_1u_2u_0 - u_0^2$$

By solving the equations EQN 1, 2, 3, 4, 5 and 6 we can find the co-efficient of the Jacobian variety curve $l(x) = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. The results are shown below:

$$l_4 = \frac{D_3S_0 - D_1S_1}{D_0D_3 - D_1D_2}$$

$$l_3 = \frac{D_0S_1 - D_2S_0}{D_0D_3 - D_1D_2}$$

$$l_2 = v_2 + u_2l_3 - A_0l_4$$

$$l_1 = v_1 + u_1l_3 - A_1l_4$$

$$l_0 = v_0 + u_0l_3 - A_2l_4$$

Where the variable are used to simplify the calculation and to make the result appear less tedious. The variable used here is shown below:

$$M_0 = \Delta_7\Delta_0 - \Delta_4\Delta_3 \qquad M_1 = \Delta_7\Delta_1 - \Delta_5\Delta_3 \qquad M_2 = \Delta_7\Delta_2 - \Delta_6\Delta_3$$

$$M_3 = \Delta_{11}\Delta_0 - \Delta_8\Delta_3 \qquad M_4 = \Delta_{11}\Delta_1 - \Delta_9\Delta_3 \qquad M_5 = \Delta_{11}\Delta_2 - \Delta_{10}\Delta_3$$

$$D_0 = M_3 - A_0 M_5 \qquad D_1 = M_4 - u_2 M_5 \qquad D_2 = M_0 - A_0 M_2$$

$$D_3 = M_1 - u_2 M_2 \qquad S_0 = \epsilon_2 - v_2 M_5 \qquad S_1 = \epsilon_1 - v_2 M_2$$

At the point of touch, in this case it is the Jacobian Variety and the Hyper Elliptic Curve of genus 3, the value of the y on both curve is the same. So we can replace the y expression of the hyper elliptic curve by the Jacobian variety curve $y = l_4 x^4 + l_3 x^3 + l_2 x^2 + l_1 x + l_0$.

HEC: $y^2 = x^7 + f_5 x^5 + f_4 x^4 + f_3 x^3 + f_2 x^2 + f_1 x + f_0$
The Jacobian Variety Curve: $y_{JC} = l_4 x^4 + l_3 x^3 + l_2 x^2 + l_1 x + l_0$

Substituting y in HEC with y_{JC}:

$$y_{JC}{}^2 = x^7 + f_5 x^5 + f_4 x^4 + f_3 x^3 + f_2 x^2 + f_1 x + f_0$$

$$\text{Or, } (l_4 x^4 + l_3 x^3 + l_2 x^2 + l_1 x + l_0)^2 = x^7 + f_5 x^5 + f_4 x^4 + f_3 x^3 + f_2 x^2 + f_1 x + f_0$$

$$\text{Or, } (l_4 x^4 + l_3 x^3 + l_2 x^2 + l_1 x + l_0)^2 - (x^7 + f_5 x^5 + f_4 x^4 + f_3 x^3 + f_2 x^2 + f_1 x + f_0) = 0$$

[EXP 2]

Solving the expression [EXP 2] would give the coordinates of the intersecting points and the point of touch. However, we intended to get the results in Mumford form. The expression below is explicitly expressed in the Mumford form if we solve and compare the L.H.S and R.H.S.

$$\prod_{r=1}^{3}(x - x_P)(x - x_Q)(x - x_S) \prod_{r=1}^{3}(x - x_P)(x - x_Q)(x - x_S) \prod_{r=1}^{3}(x - x_{R1})(x - x_{R2})(x - x_{R3})$$

$$=$$

$$(x^3 + u_2 x^2 + u_1 x + u_0)(x^3 + u_2 x^2 + u_1 x + u_0)(x^3 + u_2' x^2 + u_1' x + u_0')$$

$$\equiv$$

$$(l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2 - (x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0)$$

Comparing the coefficients of the L.H.S with the R.H.S we get:

$$u_2' = 2u_2 + l_4^2$$

$$u_1' = 2l_4l_2 - (2u_1 + u_2^2) - 2u_2u_1'$$

$$u_0' = 2l_4l_2 - 2u_2u_1' - u_1'(2u_1 + u_2^2) - (2u_0 + u_2u_1)$$

Similarly, we can get the result for v_2', v_1' and v_0' by solving the equation:

$$l(x) \bmod (x^3 + u_2x^2 + u_1x + u_0) \equiv v_2x^2 + v_1x + v_0$$

Or, $l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0 \bmod (x^3 + u_2x^2 + u_1x + u_0) \equiv v_2x^2 + v_1x + v_0$

After expanding the equation and comparing the coefficients L.H.S ≡ R.H.S we get:

$$v_2' = l_2 + u_2'l_3 - (u_2'^2 - u_1')l_4$$

$$v_1' = l_1 + u_1'l_3 - (u_2'u_1' - u_0')l_4$$

$$v_0' = l_0 + u_0'l_3 - (u_2'u_0')l_4$$

5.5.2 Comparison of proposed and existing Explicit Formulae (Doubling) for HEC $g = 3$.

The proposed works has been compared with the Explicit Formulae (Doubling) for the HEC for genus 3 has been compared. The table below presents list the complexity comparison table.

Previous Work	Finite Field	Curve Properties	Cost			Improvement Percentage (%)
			Inverse (I)	Multiplication (M)	Squaring (S)	
Kuroki et al [29]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	74 M/S		100
Gonda et al [30]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	71 M/S		103.3
Guyot et al. [31]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	61	9	108.1
Myukai et al. [32]	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	68 M/S		106.4
This work	$\mathbb{F}_q$	$h(x) = 0$, $f_6 = 0$	1	63	3	109.8

Table 5.10: Comparison between the explicit formulas for (genus = 3) curves over $\mathbb{F}_q$ of previous work and the present work.

TABLE VIII

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 3, HEC: $y^2 = x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 6.

Input	Genus 3 HEC: $y^2 + h(x)y = f(x)$; $f(x) = x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D = [u(x), v(x)]$; $u(x) = x^3 + u_2x^2 + u_1x + u_0$, $v(x) = v_2x^2 + v_1x + v_0$,	
Initial input variables	$U_{20} = u_2u_0$; $U_{10} = u_1u_0$; $U_0 = u_0^2$; $U_1 = u_1^2$; $U_2 = u_2^2$; $U_4 = U_2^2$; $U_{21} = u_2u_1$; $U_{221} = U_2u_1$; $U_{220} = U_2u_0$; $UV_{11} = u_1v_1$; $UV_{11} = u_1v_1$; $UV_{01} = u_0v_1$; $UV_{12} = u_1v_2$; $UV_{01} = u_0v_1$; $UV_{20} = u_2v_0$; $UV_{22} = u_2v_2$; $UV_{21} = u_2v_1$; $UV_{00} = u_0v_0$;	
Output	$D' = [u', v'] = 2D$, $u'(x) = x^3 + u_2'x^2 + u_1'x + u_0'$, $v'(x) = v_2'x^2 + v_1'x + v_0'$;	Cost (I, M, S)

Step	Expressions	Cost
1	$A_0 = U_2 - u_1$; $A_1 = U_{21} - u_0$; $A_2 = U_{20}$;	(0,0,0)
2	$B_0 = -u_2U_2 + 2U_{21} - u_0$; $B_1 = U_{20} + U_{221} - U_1$; $B_2 = -U_{220} + U_{10}$;	(0,1,0)
3	$C_0 = U_4 + 3U_{221} + U_{20} + U_1$; $C_1 = u_2U_{221} - 2u_2U_1 - U_{220} - 2U_{10}$; $C_2 = U_2U_{20} - u_1U_{20} - U_0$;	(0,4,0)
4	$\Delta_0 = 8B_0v_2 + 8u_2UV_{21} - 8UV_{11} - UV_{20}$; $\Delta_1 = 6u_2UV_{22} - 6UV_{12} - 6UV_{21} - 6v_0$; $\Delta_2 = -4UV_{22} - 4v_1$; $\Delta_3 = -2v_2$; $\Delta_4 = 8B_1v_2 + 8u_2UV_{11} - 8UV_{01} - 8u_1UV_{20}$; $\Delta_5 = 6u_2UV_{12} - 6UV_{02} - 6UV_{11}$; $\Delta_6 = -4UV_{12} + 4v_0$; $\Delta_7 = 2v_1$; $\Delta_8 = 8B_2v_2 + 8u_0UV_{21} - 8UV_{00}$; $\Delta_9 = 6u_2UV_{02} - 4UV_{01}$; $\Delta_{10} = -4UV_{02}$; $\Delta_{11} = 2v_1$;	(0,11,0)
5	$M_0 = \Delta_7\Delta_0 - \Delta_4\Delta_3$; $M_1 = \Delta_7\Delta_1 - \Delta_5\Delta_3$; $M_2 = \Delta_7\Delta_2 - \Delta_6\Delta_3$; $M_3 = \Delta_{11}\Delta_0 - \Delta_8\Delta_3$; $M_4 = \Delta_{11}\Delta_1 - \Delta_9\Delta_3$; $M_5 = \Delta_{11}\Delta_2 - \Delta_{10}\Delta_3$;	(0,12,0)
6	$\theta_0 = 7C_0 + 5f_5A_0 - 4f_4u_2 + 3f_3$; $\theta_1 = 7C_1 + 5f_5A_1 - 4f_4u_1 + 2f_2$; $\theta_2 = 7C_2 + 5f_5A_2 - 4f_4u_0 + f_1$;	(0,6,0)
7	$\epsilon_1 = \Delta_7\theta_0 - \Delta_3\theta_1$; $\epsilon_2 = \Delta_{11}\theta_0 - \Delta_3\theta_2$;	(0,4,0)
8	$D_0 = M_3 - A_0M_5$; $D_1 = M_4 - u_2M_5$; $D_2 = M_0 - A_0M_2$; $D_3 = M_1 - u_2M_2$;	(0,4,0)
9	$S_0 = \epsilon_2 - v_2M_5$; $S_1 = \epsilon_1 - v_2M_2$;	(0,2,0)
10	$inv = (D_0D_3 - D_1D_2)^{-1}$;	(1,2,0)

11	**Computing the co-efficient of Jacobian Variety Curve: $y = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$:** $l_4 = inv \cdot (D_3S_0 - D_1S_1)$; $l_3 = inv \cdot (D_0S_1 - D_2S_0)$; $l_2 = v_2 + u_2l_3 - A_0l_4$; $l_1 = v_1 + u_1l_3 - A_1l_4$; $l_0 = v_0 + u_0l_3 - A_2l_4$;	(0,12,0)
12	$L_{(4,2)} = 2l_4l_2 + l_3^2$; $U'_{21} = u_2u'_1$;	(0,2,1)
13	**Compute $u'(x) = x^3 + u'_2x^2 + u'_1x + u'_0$:** $u'_2 = 2u_2 + l_4^2$; $u'_1 = L_{(4,2)} - (2u_1 + U_2) - 2U'_{21}$; $u'_0 = L_{(4,2)} - 2U'_{21} - u'_1(2u_1 + U_2) - (2u_0 + U_{21})$;	(0,1,1)
14	**Compute $v'(x) = v'_2x^2 + v'_1x + v'_0$:** $v'_2 = l_2 + u'_2l_3 - (u'^2_2 - u'_1)l_4$; $v'_1 = l_1 + u'_1l_3 - (u'_2u'_1 - u'_0)l_4$; $v'_0 = l_0 + u'_0l_3 - (u'_2u'_0)l_4$;	(0,6,1)
Sum		(0,63,3)

5.6 Explicit Formulae (Doubling) for HEC of genus 4

As discussed in the section 5.5, we apply this concept to the HEC of genus 4. In this section we apply the same concept to develop a general addition explicit formulae algorithm for genus 4.

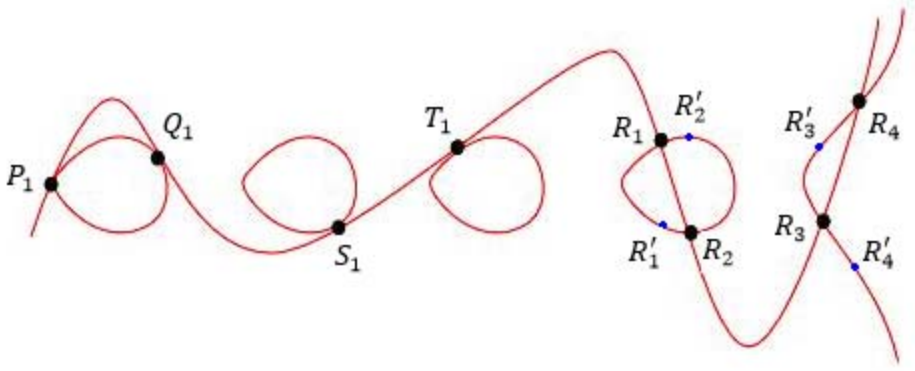

Figure 5.6: Hyper Elliptic Curve of genus 4 as it touch the Jacobian variety curve.

Since the points of touch between the Jacobian variety curve with the hyper elliptic curve sums up to zero. Therefore:

$$2[P_1] + 2[Q_1] + 2[S_1] + 2[T_1] = -[R_1] - [R_2] - [R_3] - [R_4] = [R_1'] + [R_2'] + [R_3'] + [R_4'].$$

$$2[P_1] + 2[Q_1] + 2[S_1] + 2[T_1] = -[R_1] - [R_2] - [R_3] = [R_1'] + [R_2'] + [R_3'] + [R_4'].$$

The Cartesian points or the affine space points P_1, Q_1, T_1 and S_1 could be transformed to individual divisor class group based on Mumford Representation, which is to be discussed in a separate section.

5.6.1 Generating Doubling Explicit Formulae for HEC of genus 4

Let's consider a general Hyper Elliptic Curve C of genus $g = 4$ over the finite field $\mathbb{F}_q$:

HEC: $y^2 = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$;

Since, we will be working with this particular expression $2[P_1] + 2[Q_1] + 2[S_1] + 2[T_1] = -[R_1] - [R_2] - [R_3] - [R_4]$. We can convert the coordinate $[P_1]$, $[Q_1]$, $[S_1]$ and $[T_1]$ to polynomial expression using Mumford as shown below:

$D = \prod_{r=1}^{2}(x - x_P)(x - x_Q)(x - x_S) - 3P_\infty =$	$D = [u(x), v(x)]$ $u(x) = x^3 + u_2x^2 + u_1x + u_0$ $v(x) = v_2x^2 + v_1x + v_0$
$D' = \prod_{r=1}^{2}(x - x_{R1})(x - x_{R2})(x - x_{R3}) - 3P_\infty$	$D' = [u'(x), v'(x)]$ $u'(x) = x^3 + u'_2x^2 + u'_1x + u'_0$ $v'(x) = v'_2x^2 + v'_1x + v'_0$

Table 5.11: Corresponding conversion of the Cartesian points to Mumford form for genus 4.

$$D = \prod_{r=1}^{4}(x - x_P)(x - x_Q)(x - x_S)(x - x_T) - 4P_\infty = x^4 + u_3x^3 + u_2x^2 + u_1x + u_0$$

$$D' = \prod_{r=1}^{4}(x - x_{R1})(x - x_{R2})(x - x_{R3})(x - x_{R4}) - 4P_\infty = x^4 + u'_3x^3 + u'_2x^2 + u'_1x + u'_0$$

We can assert the polynomial expression of $l(x)$ to be $l(x) = l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. The Jacobian curve $y = l(x)$ is a pentic function. At the touch points the y-coordinates are same. Therefore we can write, $l(x) = v(x)$ or $l(x) - v(x) \equiv 0\ mod\ u(x)$ since we have to perform polynomial reduction.

For the points $[P_1]$, $[Q_1]$, $[S_1]$, $[T_1]$ we can write it in the form of $l(x) - v(x) \equiv 0\ mod\ u(x)$.

$$(l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0) - (v_3x^3 + v_2x^2 + v_1x + v_0)$$
$$\equiv 0 \bmod x^4 + u_3x^3 + u_2x^2 + u_1x + u_0$$

$$l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$$
$$\equiv$$
$$v_3x^3 + v_2x^2 + v_1x + v_0 \bmod x^4 + u_3x^3 + u_2x^2 + u_1x + u_0$$

By reducing the L.H.S with the polynomial expression $x^4 + u_3x^3 + u_2x^2 + u_1x + u_0$ and comparing with the R.H.S, we get four equations.

$$A_0l_5 - u_3l_4 + l_3 = v_3 \quad \text{EQN 1}$$

$$A_1l_5 - u_2l_4 + l_2 = v_2 \quad \text{EQN 2}$$

$$A_2l_5 - u_1l_4 + l_1 = v_1 \quad \text{EQN 3}$$

$$A_3l_5 - u_0l_4 + l_0 = v_0 \quad \text{EQN 4}$$

Derivative of Jacobian Curve:

$$\frac{dy}{dx} = 5l_5x^4 + 4l_4x^3 + 3l_3x^2 + 2l_2x + l_1$$

Derivative of the hyper elliptic curve:

$$2y \cdot \frac{dy}{dx} = 9x^8 + 7f_7x^6 + 6f_6x^5 + 5f_5x^4 + 4f_4x^3 + 3f_3x^2 + 2f_2x + f_1$$

By substituting $v(x) = v_3x^3 + v_2x^2 + v_1x + v_0$ of the divisor $D = [u(x), v(x)]$ and the derivative of the Jacobian curve, we get the expression as shown below:

$$2(5l_5x^4 + 4l_4x^3 + 3l_3x^2 + 2l_2x + l_1) \cdot (v_3x^3 + v_2x^2 + v_1x + v_0)$$
$$\equiv$$
$$9x^8 + 7f_7x^6 + 6f_6x^5 + 5f_5x^4 + 4f_4x^3 + 3f_3x^2 + 2f_2x + f_1 \bmod u(x)$$

By reducing the L.H.S with the polynomial expression $x^4 + u_3x^3 + u_2x^2 + u_1x + u_0$, and comparing with the R.H.S, we get four equations.

$$M_0l_5 + M_1l_4 + M_2l_3 + M_3l_2 + M_3l_2 = \epsilon_0 \quad \text{EQN 5}$$

$$M_5l_5 + M_6l_4 + M_7l_3 + M_8l_2 + M_9l_2 = \epsilon_1 \quad \text{EQN 6}$$

$$P_0l_5 + P_1l_5 + P_2l_3 + P_3l_2 + P_3l_2 = \epsilon_2 \quad \text{EQN 7}$$

$$P_5l_5 + P_6l_5 + P_7l_3 + P_8l_2 + P_9l_2 = \epsilon_2 \quad \text{EQN 8}$$

By solving the equations EQN 1, 2, 3, 4, 5, 6,7 and 8 we can find the co-efficient of the Jacobian variety curve $l(x) = l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$. The results are shown below:

$$l_5 = \frac{\Delta_3\theta_0 - \Delta_1\theta_1}{\Delta_3\Delta_0 - \Delta_2\Delta_1}$$

$$l_4 = \frac{\Delta_0\theta_1 - \Delta_2\theta_0}{\Delta_3\Delta_0 - \Delta_2\Delta_1}$$

$$l_3 = v_3 + u_3l_4 - A_0l_5$$

$$l_2 = v_2 + u_2l_4 - A_1l_5$$

$$l_1 = v_1 + u_1l_4 - A_2l_5$$

$$l_0 = v_0 + u_0l_4 - A_3l_5$$

At the point of touch, in this case it is the Jacobian Variety and the Hyper Elliptic Curve of genus 4, the value of the y on both curve is the same. So we can replace the y expression of the hyper elliptic curve by the Jacobian variety curve $y = l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$.

HEC: $y^2 = x^9 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
The Jacobian Variety Curve: $y_{JC} = l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$

Substituting y in HEC with y_{JC}:

$$y_{JC}{}^2 = x^9 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\text{Or, } (l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2$$
$$=$$
$$x^9 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$$

$$\text{Or, } (l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2 -$$
$$(x^7 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0) = 0 \qquad \text{[EXP 2]}$$

Solving the expression [EXP 2] would give the coordinates of the intersecting points and the point of touch. However, we intended to get the results in Mumford form. The expression below is explicitly expressed in the Mumford form if we solve and compare the L.H.S and R.H.S.

$$(x^4 + u_3x^3 + u_2x^2 + u_1x + u_0) \cdot (x^4 + u_3x^3 + u_2x^2 + u_1x + u_0) \cdot (x^4 + u_3x^3 + u_2x^2 + u_1x + u_0)$$
$$\equiv$$
$$(l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0)^2 - x^9 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$$

Comparing the coefficients of the L.H.S with the R.H.S we get:

$$u_3' = -2u_3$$

$$u_2' = l_5^2 - u_3'(2u_3) - (2u_2 + U_0)$$

$$u_1' = 2l_5l_4 - 1 - u_2'(2u_3) + u_3'(2u_2 + U_0) - (2u_1 + 2U_{23})$$

$$u_0' = 2l_5l_3 + l_4^2 - u_1'(2u_3) - u_2'(2u_2 + U_0) - u_3'(2u_1 + 2U_{23}) - (2u_0 + U_{13} + U_2)$$

Similarly, we can get the result for v_2', v_1' and v_0' by solving the equation:

$$l(x) \bmod (x^4 + u_3x^3 + u_2x^2 + u_1x + u_0) \equiv v_3x^3 + v_2x^2 + v_1x + v_0$$

$$\text{Or, } l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0 \equiv v_3x^3 + v_2x^2 + v_1x + v_0 \; mod \; (x^4 + u_3x^3 + u_2x^2 + u_1x + u_0)$$

After expanding the equation and comparing the coefficients L.H.S ≡ R.H.S we get:

$$v_3' = A_0'l_5 - u_3'l_4 + l_3$$

$$v_2' = A_1'l_5 - u_2'l_4 + l_2$$

$$v_1' = A_2'l_5 - u_1'l_4 + l_1$$

$$v_0' = A_3'l_5 - u_0'l_4 + l_0$$

5.6.2 Comparison of proposed and existing Explicit Formulae (Doubling) for HEC $g = 4$

The proposed works has been compared with the Explicit Formulae (Doubling) for the HEC for genus 4 has been compared. The table below presents list the complexity comparison table.

Previous Work	Finite Field	Cost			Improvement Percentage (%)
		Inverse (I)	Multiplication (M)	Squaring (S)	
Cantor [23]	$\mathbb{F}_q$	6	359 M/S		100
Nagao [23]	$\mathbb{F}_q$	2	268 M/S		135.7
C. Paar [33]	$\mathbb{F}_q$	2	193	16	146.3
This work	$\mathbb{F}_q$	1	98	3	178.6

Table 5.12: Comparison between the explicit formulae's (doubling) for (genus = 4) curves over $\mathbb{F}_q$ of previous work and the present work.

TABLE IX

EXPLICIT FORMULA FOR ADDITION ON A HYPER ELLIPTIC CURVE OF GENUS 4, HEC: $y^2 = x^9 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$ OVER THE GALOIS FIELD $GF(p)$. NUMBER OF COORDINATES: 8

Input	Genus 4 HEC: $y^2 + h(x)y = f(x)$; $f(x) = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; Divisor $D = [u(x), v(x)]$; $u(x) = x^4 + u_3x^3 + u_2x^2 + u_1x + u_0$, $v(x) = v_3x^3 + v_2x^2 + v_1x + v_0$;	
Initial input variables	$U_0 = u_3^2$; $U_{30} = U_0u_0$; $U_{32} = U_0u_2$; $U_{31} = U_0u_1$; $U_{320} = u_0U_{23}$; $U_{321} = u_1U_{23}$; $U_{13} = u_3u_1$; $U_{03} = u_3u_0$; $U_{02} = u_2u_0$; $U_{01} = u_1u_0$; $U_{23} = u_3u_2$; $U_{12} = u_2u_1$; $U_1 = U_0^2$; $U_2 = u_2^2$; $U_3 = u_1^2$; $U_4 = u_0^2$;	
Output	$D' = [u', v'] = 2D$, $u'(x) = x^4 + u_3'x^3 + u_2'x^2 + u_1'x + u_0'$; $v'(x) = v_3'x^3 + v_2'x^2 + v_1'x + v_0'$;	Cost (I, M, S)

Step	Expressions	Cost
1	$A_0 = U_0 - u_2$; $A_1 = U_{23} - u_1$; $A_2 = U_{13} - u_0$; $A_3 = U_{03}$;	(0,0,0)
2	$B_0 = -u_3U_0 + 2U_{23} - u_1$; $B_1 = -U_{32} + U_2 + U_{13} - u_0$; $B_2 = -U_{31} + U_{12} + U_{03}$; $B_3 = U_{02} - U_{30}$;	(0,1,0)
3	$C_0 = U_1 - 3U_{32} + 2U_{13} + U_2 - u_0$; $C_1 = u_3U_{32} - 2u_3U_2 + 2U_{12} + U_{03}$; $C_2 = u_3U_{31} - 2U_{321} - U_{30} + U_3 + U_{02}$; $C_3 = u_3U_{30} - 2U_{320} - u_0 + U_{01}$;	(0,4,0)
4	$D_0 = -u_3U_1 + 3U_{32} + u_3U_{32} + 2U_{12} - 4U_{31} - 3u_3U_2 + 2U_{03}$; $D_1 = u_3U_{31} - u_0U_{32} - 4U_{321} + U_3 + 2U_{02} + 3U_0U_2 - u_2U_2$; $D_2 = 2U_{01} - U_0U_{31} + 3u_1U_{32} - 2u_3U_3 - u_1U_2 + u_3U_{20} - 2U_{320} - u_0$; $D_3 = -U_0U_{30} + 3u_2U_{30} - 2u_0U_{13} - u_0U_2 + U_4$;	(0,16,0)
5	$M_0 = 10C_0v_3 + 10B_0v_2 + 10A_0v_1 + 10u_3v_0$; $M_1 = 8B_0v_3 + 8A_0v_2 - 8u_3v_1 + 8v_0$; $M_2 = 6A_0v_3 - 6u_3v_2 + 6v_1$; $M_3 = -4u_3v_3 + 4v_2$; $M_4 = 2v_3$; $M_5 = 10C_1v_3 + 10B_1v_2 + 10A_1v_1 + 10u_2v_0$; $M_6 = 8B_1v_3 + 8A_1v_2 - 8u_2v_1$; $M_7 = 6A_1v_3 - 6u_2v_2 + 6v_0$; $M_9 = 2v_2$;	(0,19,0)
6	$\epsilon_0 = 9D_0 + 7f_7B_0 + 6f_6A_0 - 5f_5u_3 + 4f_4$; $\epsilon_1 = 9D_1 + 7f_7B_1 + 6f_6A_1 - 5f_5u_2 + 3f_3$;	(0,6,0)

7	$\Delta_0 = M_0 - M_2A_0 - M_3A_1 - M_4A_2$; $\Delta_1 = M_1 - u_3M_2 + u_2M_3 + u_1M_4$; $\Delta_2 = M_5 - M_7A_0 - M_8A_1 - M_9A_2$; $\Delta_3 = M_6 - u_3M_7 + u_2M_8 + u_1M_9$;	(0,13,0)
8	$\theta_0 = \epsilon_0 - M_2v_3 - M_3v_2 - M_4v_1$; $\theta_1 = \epsilon_1 - M_7v_3 - M_8v_2 - M_9v_1$;	(0,6,0)
9	$inv = (\Delta_3\Delta_0 - \Delta_2\Delta_1)^{-1}$;	(1,2,0)
10	**Computing the co-efficient of Jacobian Variety Curve: $y = l_5x^5 + l_4x^4 + l_3x^3 + l_2x^2 + l_1x + l_0$:** $l_5 = inv \cdot (\Delta_3\theta_0 - \Delta_1\theta_1)$; $l_4 = inv \cdot (\Delta_0\theta_1 - \Delta_2\theta_0)$; $l_3 = v_3 + u_3l_4 - A_0l_5$; $l_2 = v_2 + u_2l_4 - A_1l_5$; $l_1 = v_1 + u_1l_4 - A_2l_5$; $l_0 = v_0 + u_0l_4 - A_3l_5$;	(0,12,0)
11	**Compute $u'(x) = x^3 + u_2'x^2 + u_1'x + u_0'$:** $u_3' = -2u_3$; $u_2' = l_5^2 - u_3'(2u_3) - (2u_2 + U_0)$; $u_1' = 2l_5l_4 - 1 - u_2'(2u_3) + u_3'(2u_2 + U_0) - (2u_1 + 2U_{23})$; $u_0' = 2l_5l_3 + l_4^2 - u_1'(2u_3) - u_2'(2u_2 + U_0) - u_3'(2u_1 + 2U_{23}) - (2u_0 + U_{13} + U_2)$;	(0,8,2)
12	$A_0' = u_3'^2 - u_2'$; $A_1' = u_3'u_2' - u_1'$; $A_2' = u_3'u_1' - u_0'$; $A_2' = u_3'u_0'$;	(0,3,1)
13	**Compute $v'(x) = v_2'x^2 + v_1'x + v_0'$:** $v_3' = A_0'l_5 - u_3'l_4 + l_3$; $v_2' = A_1'l_5 - u_2'l_4 + l_2$; $v_1' = A_2'l_5 - u_1'l_4 + l_1$; $v_0' = A_3'l_5 - u_0'l_4 + l_0$;	(0,8,0)
Sum		(1,98,3)

Chapter 6

Discussions and Future Works

In the field of cyber – security, especially for public key infrastructure. There is a demand for shorter key size and faster computation. Shorter key size is needed since the mobile devices stores limited amount of space and faster computation is necessary because of limited power supply and processing capabilities. Elliptic Curve Cryptosystem has been studied extensively and has already been implemented in our mobile devices such as Blackberry to public key cryptosystem. The key size used by the ECC is much shorter than of RSA and provides the same security strength. However, it is believed theoretically that the Hyper Elliptic Curve Cryptosystem can provide the same security strength with much shorter key size of ECC.

In this thesis, a brief introduction is provided in the first chapter. Then, in chapter 2 the mathematical background of groups, rings, finite field and basic introduction of Hyper Elliptic Curve with examples. Chapter 3, discusses the HECC is details such as group operation and its comparison with ECC. In the chapter 4, we gave an overview of the existing computational methods and subsequently in chapter 5 we have proposed an algorithm for faster computation for Hyper Elliptic Curve Cryptography. We also discussed the process used derive the algorithm for curves of genus 2, 3 and 4. In the complexity comparison table, we have noticed that the as the number of genus of the curve increases the number of total operations to perform group operation decreases compare to the recent existing work. There is significant rise in efficiency in terms of percentage from the recent previous work for the hyper elliptic curves of genus 3 and 4.

In the future study, we can work on the hardware implementation of Hyper Elliptic Curve Cryptosystems, i.e, key exchange and digital signature. Although this thesis is solely based on the computation in Affine Space. Later we can derived an explicit formulae in projective space, which will delete the inversion operation. As discussed in this thesis, inverse operation is computationally intensive. Would it be possible to develop an explicit formulae algorithm of 0 inversion in affine space? The explicit formulae shown in the thesis is takes the coefficient of the Mumford Representation of the Cartesian points as input. There is

another method of transforming the Cartesian points into divisor class by Chow Representation [41], [42]. Is it possible to develop more efficient explicit formulae with Chow Representation instead of Mumford Representation? All these options can be explored in the future.

Later, we can propose explicit formulae over the finite field of $GF(2^m)$.

Appendix

A. MATLAB SCRIPT FOR PSEUDO CODE FOR CALCULATING: $y^2 mod\ p$

```
% Power Mod Calculator

% A = y.^2 mod p

syms A y p

y = input('Enter the range of y: ');
p = input('Enter the value of mod p: ');

A = mod(power(y,p),p);

W = ['y = ', num2str(y)];
Z = ['y.^n mod p = ', num2str(A)];

disp(W)
disp(Z)
```

B. MATLAB SCRIPT FOR PSEUDO CODE FOR CALCULATING: $(x^3 + x) \bmod p$

```
% Power Mod Calculator

% B = (x.^3 + x) mod p

syms B x p

x = input('Enter the range of x: ');
p = input('Enter the value of mod p: ');

B = mod((x.^3 + x),p);

W = ['x = ', num2str(x)];
Z = ['x.^3 + x mod p = ', num2str(B)];

disp(W)
disp(Z)
```

C. MATLAB SCRIPT FOR PSEUDO CODE TO DETERMINE THE POINTS ON EC

```
% EllipticCurvePoints
% E = {(x,y): y^2 = x^3 + x mod p} U {O}

% A = y.^2 mod p
% B = (x.^3 + x) mod p
% A represents the left side of the equation
% B represents the right side of the equation

syms A B x y p

y = input('Enter the range of y: '); % range of y is from 0:p-1
x = input('Enter the range of x: '); % range of x is from 0:p-1
count = 0;

p = input('Enter the value of mod p: ');

disp('The valid co-ordinates or points in the curve')

A = mod(y.^2,p);
B = mod((x.^3 + x),p);

for b = [1:p]
        for  a = [1:p]
            if B(b) == A(a)
                %Z = ['B[', num2str(b) ,']', '===' , 'A[', num2str(a)
,']'];
                %W = ['x = ', num2str(x(b)) , ' === ' , 'y = ',
num2str(y(a))];
                %disp(Z)
                %W = ['(x,y) = (', num2str(x(b)) ,',',
num2str(y(a)),')'];
                count = count + 1;
                %disp(W)
            end
        end
end
count = count + 1;
Z = ['Number of Valid points = ', num2str(count)];
disp(Z)
```

D. MATLAB SCRIPT FOR PSEUDO CODE TO DETERMINE THE POINTS ON HEC

```
% HyperEllipticCurvePoints
% E = {(x,y): y^2 = x^5 + 1184x^3 + 1846x^2 + 956x + 560 mod p} U {O}

% A = (y.^2) mod p
% B = (x.^5 + 1184*x.^3 + 1846*x.^2 + 956*x + 560) mod p
% A represents the left side of the equation
% B represents the right side of the equation

syms A B x y p

y = input('Enter the range of y: '); % range of y is from 0:p-1
x = input('Enter the range of x: '); % range of x is from 0:p-1
count = 0;

p = input('Enter the value of mod p: ');

disp('The valid co-ordinates or points in the curve')

A = mod((y.^2),p);
B = mod((x.^5 + 1184*x.^3 + 1846*x.^2 + 956*x + 560),p);

for b = [1:p]
        for  a = [1:p]
            if B(b) == A(a)
                W = ['(x,y) = (', num2str(x(b)) ,',',
num2str(y(a)),')'];
                count = count + 1;
                disp(W)

            end
        end
end
count = count + 1;
Z = ['Number of Valid points = ', num2str(count)];
disp(Z)
```

Bibliography

[1] Koblitz, Neal. "Elliptic curve cryptosystems." *Mathematics of computation*48.177 (1987): 203-209.

[2] Koblitz, Neal. "Hyperelliptic cryptosystems." *Journal of cryptology* 1.3 (1989): 139-150.

[3] Cantor, David G. "Computing in the Jacobian of a hyperelliptic curve."*Mathematics of computation* 48.177 (1987): 95-101.

[4] CLANCY III, THOMAS CHARLES. *Analysis of FPGA-based hyperelliptic curve cryptosystems*. Diss. University of Illinois at Urbana-Champaign, 2002.

[5] Pelzl, Jan, et al. "Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves." *Cryptographic Hardware and Embedded Systems-CHES 2003*. Springer Berlin Heidelberg, 2003. 351-365.

[6] Diffie, Whitfield, and Martin E. Hellman. "New directions in cryptography."*Information Theory, IEEE Transactions on* 22.6 (1976): 644-654.

[7] Merkle, Ralph C. "Secure communications over insecure channels."*Communications of the ACM* 21.4 (1978): 294-299.

[8] Rivest, Ronald L., Adi Shamir, and Len Adleman. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21.2 (1978): 120-126.

[9] Paar, Christof, and Jan Pelzl. *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[10] ElGamal, Taher. "A public key cryptosystem and a signature scheme based on discrete logarithms." *Advances in cryptology*. Springer Berlin Heidelberg, 1984.

[11] Harley, R. "Fast arithmetic on genus two curves; 2000."

[12] Lange, Tanja. "Efficient arithmetic on hyperelliptic curves." *IACR Cryptology ePrint Archive* 2002 (2002): 107.

[13] Matsuo, Kazuto, Jinhui Chao, and Shigeo Tsujii. "Fast genus two hyperelliptic curve cryptosystems." Technical Report ISEC2001-23, IEICE, 2001. Pages 89-96.

[14] Miyamoto, Yosuke, et al. "A fast addition algorithm of genus two hyperelliptic curve." The 2002 Symposium on Cryptography and Information Security—SCIS 2002, IEICE Japan. 2002.

[15] Takahashi, Masashi. "Improving Harley algorithms for Jacobians of genus 2 hyperelliptic curves." SCIS, IEICE Japan (2002).

[16] Gallian, Joseph. *Contemporary abstract algebra*. Nelson Education, 2009.

[17] Herstein, Israel Nathan, and Israel N. Herstein. *Abstract algebra*. Macmillan, 1990.

[18] Roman, Steven. *Field theory*. Vol. 158. Springer Science & Business Media, 2005.

[19] V. Miller, "Use of elliptic curves in cryptography," in Advance in Cryptology - CRYPTO'85, ser. LNCS 218, H. C. Williams, Ed. Berlin, Germany: Springer-Verlag, pp. 417-426, 1986.

[20] N. Koblitz, "A Family of Jacobian Suitable for Discrete Log Cryptosystems," Advance in Cryptology - CRYPTO'88, ser. LNCS 403, Shafi Goldwasser, Ed. Berlin, Germany: Springer-Verlag, pp. 94-99, 1988.

[21] Montgomery, Peter L. "Speeding the Pollard and elliptic curve methods of factorization." *Mathematics of computation* 48.177 (1987): 243-264.

[22] Chudnovsky, David V., and Gregory V. Chudnovsky. "Sequences of numbers generated by addition in formal groups and new primality and factorization tests." *Advances in Applied Mathematics* 7.4 (1986): 385-434.

[23] Nagao, Koh-ichi. "Improving group law algorithms for Jacobians of hyperelliptic curves." Algorithmic Number Theory. Springer Berlin Heidelberg, 2000. 439-447.

[24] Lange, Tanja. "Efficient Arithmetic on Genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae." *IACR Cryptology ePrint Archive* 2002 (2002): 121.

[25] Gaudry, Pierrick, and Robert Harley. "Counting points on hyperelliptic curves over finite fields." Algorithmic number theory. Springer Berlin Heidelberg, 2000. 313-332.

[26] Essen, Universitiat-Gesamthochschule, and Tanja Lange. "Efficient Arithmetic on Hyperelliptic Curves."

[27] Lange, Tanja. "Formulae for arithmetic on genus 2 hyperelliptic curves." Applicable Algebra in Engineering, Communication and Computing 15.5 (2005): 295-328.

[28] Fan, Xinxin, Thomas Wollinger, and Guang Gong. "Efficient explicit formulae for genus 3 hyperelliptic curve cryptosystems." the IEEE Journal proceedings of (2007).

[29] Kuroki, Junichi, et al. "Fast genus three hyperelliptic curve cryptosystems." The 2002 Symposium on Cryptography and Information Security, Japan—SCIS 2002. 2002

[30] Gonda, Masaki, et al. "Improvements of addition algorithm on genus 3 hyperelliptic curves and their implementation." IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences 88.1 (2005): 89-96.

[31] Guyot, Cyril, Kiumars Kaveh, and Vijay M. Patankar. "Explicit algorithm for the arithmetic on the hyperelliptic Jacobians of genus 3." JOURNAL-RAMANUJAN MATHEMATICAL SOCIETY 19.2 (2004): 75-115.

[32] Nyukai, J., et al. On the resultant computation in the addition Harley algorithms on hyperelliptic curves. IEICE Technical Report, ISEC2006-5, 2006.

[33] Pelzl, Jan, Thomas Wollinger, and Christof Paar. "Low cost security: Explicit formulae for genus-4 hyperelliptic curves." Selected Areas in Cryptography. Springer Berlin Heidelberg, 2003.

[34] Silverman, Joseph H. *Advanced topics in the arithmetic of elliptic curves*. Vol. 151. Springer Science & Business Media, 2013.

[35] Mumford, David, Chidambaran Padmanabhan Ramanujam, and Jurij Ivanovič Manin. *Abelian varieties*. Vol. 108. Oxford: Oxford university press, 1974.

[36] Mumford, David, and C. Musili. *Tata lectures on theta II*. Vol. 43. Birkhäuser, 2007.

[37] Galbraith, Steven D. *Mathematics of public key cryptography*. Cambridge University Press, 2012.

[38] Hartshorne, Robin. *Algebraic geometry*. Vol. 52. Springer Science & Business Media, 1977.

[39] Milne, James S. "Jacobian varieties." *Arithmetic geometry*. Springer New York, 1986. 167-212.

[40] Mumford, David. *Curves and their Jacobians*. University of Michigan Press, c1975, 1975.

[41] Chow, Wei-Liang. "On equivalence classes of cycles in an algebraic variety."*Annals of Mathematics* (1956): 450-479.

[42] Huang, Ming-Deh, and Doug Ierardi. "Efficient algorithms for the Riemann-Roch problem and for addition in the Jacobian of a curve." *Journal of Symbolic Computation* 18.6 (1994): 519-539.

Vita Auctoris

NAME:	Raqib Ahmed ASIF
PLACE OF BIRTH:	Dhaka, P.R. Bangladesh
YEAR OF BIRTH:	1986
EDUCATION:	North South University, B.Sc., Dhaka, P.R. Bangladesh, 2010 University of Windsor, MENG., Windsor, ON, CANADA, 2012 University of Windsor, M.A.Sc., Windsor, ON, CANADA, 2016