

Elliptic Curves

Archived study guides

These archived materials are no longer updated. This archive was exported from a saved copy of these guides.

Export date: 2026-10-05T19:26:30.244Z

Contents

Unit 1 – Foundations of elliptic curves - 6 guides

Unit 2 – Elliptic Curves in Finite Fields - 6 guides

Unit 3 – Elliptic curve cryptography - 6 guides

Unit 4 – Elliptic curves and number theory - 6 guides

Unit 5 – Elliptic Curves: Complex Number Analysis - 6 guides

Unit 6 – Elliptic Curves: Algebraic Geometry Foundations - 6 guides

Unit 7 – Elliptic curves and modular forms - 6 guides

Unit 8 – Algorithms for elliptic curves - 6 guides

Unit 9 – Elliptic Curves in Coding Theory - 6 guides

Unit 10 – Elliptic Curves in Primality Testing & Factoring - 6 guides

Unit 1 – Foundations of elliptic curves

1.1 Weierstrass equations

Weierstrass equations are the backbone of elliptic curve theory. They provide a standardized way to represent these curves algebraically, allowing us to study their properties and behavior. Understanding these equations is crucial for grasping the fundamentals of elliptic curves.

From the general form to simplified versions, Weierstrass equations offer a flexible framework for analyzing elliptic curves. They connect to key concepts like the discriminant, isomorphism classes, and torsion points, forming the foundation for deeper exploration of elliptic curve arithmetic and applications.

Definition of Weierstrass equations

- Weierstrass equations are a specific form of algebraic equations used to define elliptic curves
- They are named after the German mathematician Karl Weierstrass who studied them extensively in the 19th century
- Weierstrass equations provide a standardized way to represent and study the properties of elliptic curves

General form

- The general form of a Weierstrass equation is $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$
- a_1, a_2, a_3, a_4, a_6 are constants that determine the specific shape of the elliptic curve
- This form allows for the presence of various terms and coefficients, making it suitable for representing a wide range of elliptic curves
- The general form can be simplified by applying certain transformations to the equation

Simplified forms

- Weierstrass equations can be simplified into various forms depending on the characteristics of the underlying field
- Over fields of characteristic not equal to 2 or 3, the Weierstrass equation can be reduced to the short Weierstrass form: $y^2 = x^3 + ax + b$
- a and b are constants that define the specific curve
- Over fields of characteristic 2, the simplified form is $y^2 + xy = x^3 + ax^2 + b$
- Over fields of characteristic 3, the simplified form is $y^2 = x^3 + ax^2 + bx + c$
- These simplified forms make it easier to study and classify elliptic curves in different contexts

Relationship to elliptic curves

- Weierstrass equations are intrinsically linked to the study of elliptic curves
- They provide an algebraic representation of elliptic curves, allowing for their geometric and arithmetic properties to be studied

- Understanding Weierstrass equations is crucial for gaining a deeper understanding of elliptic curves and their applications

Cubic equations

- Elliptic curves are defined by cubic equations, which are polynomial equations of degree 3
- The Weierstrass equation is a specific type of cubic equation that satisfies certain conditions
- The cubic nature of elliptic curves gives rise to their unique geometric properties, such as symmetry and the presence of an identity element (the point at infinity)
- Studying cubic equations in the context of Weierstrass forms allows for a systematic analysis of elliptic curves

Projective space

- Elliptic curves are often studied in the context of projective space, which extends the concept of affine space
- Projective space allows for the inclusion of points at infinity, which play a crucial role in the group law of elliptic curves
- The Weierstrass equation can be homogenized to define an elliptic curve in projective space
- The homogenized Weierstrass equation is $Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$
- Studying elliptic curves in projective space provides a more complete and unified framework for understanding their properties

Discriminant of Weierstrass equations

- The discriminant is an important quantity associated with Weierstrass equations that determines the nature of the elliptic curve
- It provides information about the singularities and the structure of the curve
- The discriminant plays a crucial role in classifying elliptic curves and understanding their properties

Formula for discriminant

- The discriminant of a Weierstrass equation is given by a specific formula involving the coefficients of the equation
- For the simplified Weierstrass equation $y^2 = x^3 + ax + b$, the discriminant is $\Delta = -16(4a^3 + 27b^2)$
- The formula for the discriminant varies depending on the specific form of the Weierstrass equation being used
- Calculating the discriminant allows for the determination of important characteristics of the elliptic curve

Nonsingularity condition

- For an elliptic curve to be nonsingular, the discriminant must be nonzero
- A nonzero discriminant indicates that the curve has no cusps or self-intersections

- Nonsingular elliptic curves have a well-defined group law and exhibit desirable properties for cryptographic and mathematical applications
- If the discriminant is zero, the curve is considered singular and does not form a group under the usual addition law
- Checking the nonsingularity condition is essential when working with elliptic curves in practice

Isomorphism classes

- Isomorphism classes group together elliptic curves that are essentially the same up to certain transformations
- Two elliptic curves are considered isomorphic if there exists a change of variables that transforms one curve into the other
- Studying isomorphism classes allows for the classification and simplification of elliptic curves

Changes of variables

- Isomorphisms between elliptic curves can be realized through changes of variables
- Linear transformations of the form $x \mapsto u^2x + r$ and $y \mapsto u^3y + su^2x + t$ can be applied to the Weierstrass equation
- u, r, s, t are constants that define the specific transformation
- These changes of variables preserve the underlying structure and properties of the elliptic curve
- Applying suitable changes of variables can simplify the Weierstrass equation and reveal isomorphisms between curves

Invariants

- Invariants are quantities associated with elliptic curves that remain unchanged under isomorphisms
- The j -invariant is a key invariant that characterizes isomorphism classes of elliptic curves
- It is defined as $j = \frac{c_4^3}{\Delta}$, where c_4 is a certain polynomial in the coefficients of the Weierstrass equation
- Elliptic curves with the same j -invariant are isomorphic, while curves with different j -invariants belong to distinct isomorphism classes
- Invariants provide a way to classify and distinguish between different types of elliptic curves

Reduction of Weierstrass equations

- Reduction refers to the process of simplifying and studying elliptic curves over finite fields or residue class rings
- It involves analyzing the behavior of the Weierstrass equation modulo a prime or prime power
- Reduction techniques are important for understanding the arithmetic and structure of elliptic curves in different contexts

Good vs bad reduction

- Elliptic curves can have either good or bad reduction at a prime p

- Good reduction occurs when the reduced curve modulo P is nonsingular and retains the essential properties of the original curve
- Bad reduction happens when the reduced curve becomes singular or degenerates in some way
- The type of reduction provides insights into the local behavior and structure of the elliptic curve at specific primes

Minimal Weierstrass equations

- A minimal Weierstrass equation is a particular form of the Weierstrass equation that is optimally suited for studying reduction
- It is characterized by having coefficients with minimal valuations at a given prime
- Minimal Weierstrass equations are useful for analyzing the reduction types and computing important quantities associated with elliptic curves
- Finding a minimal Weierstrass equation often involves applying suitable transformations to the original equation

Torsion points

- Torsion points are points on an elliptic curve that have finite order under the group law
- They are points P such that $nP = O$ for some positive integer n , where O is the identity element (point at infinity)
- Torsion points form a subgroup of the elliptic curve group and play a significant role in the study of elliptic curves

Definition and properties

- The order of a torsion point P is the smallest positive integer n such that $nP = O$
- Torsion points are closed under the group law, meaning that the sum of two torsion points is also a torsion point
- The set of torsion points on an elliptic curve is finite and forms a group structure
- The structure and size of the torsion subgroup provide information about the arithmetic properties of the elliptic curve

Torsion subgroups

- The torsion subgroup of an elliptic curve is the set of all torsion points on the curve
- It is a finite abelian group and can be classified up to isomorphism
- The possible structures of torsion subgroups over different fields have been extensively studied
- Over the rational numbers, the torsion subgroup is isomorphic to a product of cyclic groups $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$ with certain restrictions on n and m
- Understanding the torsion subgroup is important for studying the rational points and arithmetic of elliptic curves

Mordell-Weil theorem

- The Mordell-Weil theorem is a fundamental result in the theory of elliptic curves
- It states that the group of rational points on an elliptic curve over a number field is finitely generated
- The theorem has significant implications for understanding the structure and properties of elliptic curves

Statement of theorem

- Let E be an elliptic curve defined over a number field K . Then the group $E(K)$ of K -rational points on E is a finitely generated abelian group
- The group $E(K)$ can be written as a direct sum of a finite torsion subgroup and a free abelian group of rank r
- $E(K) \cong E(K)_{\text{tors}} \oplus \mathbb{Z}^r$
- The rank r is called the Mordell-Weil rank of the elliptic curve E over K
- The theorem guarantees that there are only finitely many rational points on an elliptic curve up to a certain height

Rank of elliptic curves

- The Mordell-Weil rank is a measure of the "size" of the group of rational points on an elliptic curve
- It determines the number of independent points of infinite order on the curve
- Computing the exact rank of an elliptic curve is a challenging problem and is the subject of ongoing research
- There are various methods and techniques for estimating and bounding the rank, such as the use of L -functions and descent algorithms
- The rank of an elliptic curve has implications for its arithmetic properties and the distribution of rational points

Integral points

- Integral points are points on an elliptic curve whose coordinates are integers
- They are a subset of the rational points and have special arithmetic properties
- Studying integral points on elliptic curves is important for understanding their Diophantine aspects and solving certain equations

Definition

- A point $P = (x, y)$ on an elliptic curve E is called an integral point if both x and y are integers
- The set of integral points on E is denoted by $E(\mathbb{Z})$
- Integral points form a subgroup of the group of rational points $E(\mathbb{Q})$
- Finding and characterizing integral points is a fundamental problem in the arithmetic of elliptic curves

Effective bounds

- Effective bounds are explicit estimates or formulas that limit the size or number of integral points on an elliptic curve
- These bounds provide a way to search for and determine all the integral points on a given curve
- Some notable effective bounds include:
 - Siegel's theorem, which gives an upper bound on the number of integral points in terms of the coefficients of the elliptic curve
 - Baker's method, which uses linear forms in logarithms to bound the size of integral points
- Effective bounds are valuable tools for solving Diophantine equations and studying the arithmetic properties of elliptic curves

Rational points of finite order

- Rational points of finite order are torsion points on an elliptic curve whose coordinates are rational numbers
- They have a finite order under the group law and form a subgroup of the torsion subgroup
- Understanding the structure and properties of rational points of finite order is important for studying the arithmetic of elliptic curves

Nagell-Lutz theorem

- The Nagell-Lutz theorem provides a characterization of rational points of finite order on an elliptic curve
- It states that if $P = (x, y)$ is a rational point of finite order on an elliptic curve $y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$, then:
 - $x, y \in \mathbb{Z}$
 - Either $y = 0$ (in which case P has order 2), or y^2 divides the discriminant $\Delta = -16(4a^3 + 27b^2)$
- The theorem gives a practical way to determine the possible torsion points on an elliptic curve and their orders
- It is a useful tool for studying the torsion subgroup and solving Diophantine equations related to elliptic curves

Mazur's theorem

- Mazur's theorem is a fundamental result that classifies the possible torsion subgroups of elliptic curves over the rational numbers
- It states that the torsion subgroup of an elliptic curve over $\mathbb{Q}$ is isomorphic to one of the following 15 groups:
 - $\mathbb{Z}/n\mathbb{Z}$ with $1 \leq n \leq 10$ or $n = 12$
 - $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2m\mathbb{Z}$ with $1 \leq m \leq 4$
- The theorem provides a complete list of the possible structures of rational torsion subgroups

- It has significant implications for understanding the arithmetic and Diophantine properties of elliptic curves over $\mathbb{Q}$
- Mazur's theorem is a deep and influential result in the theory of elliptic curves and has inspired further research in the field

1.2 Group law on elliptic curves

The group law on elliptic curves defines how to "add" points, forming a mathematical structure crucial for cryptography and number theory. It combines geometric intuition with algebraic formulas, allowing us to perform calculations on curve points.

This foundational concept enables the creation of cryptographic systems and factorization algorithms. Understanding the group law is key to grasping how elliptic curves work and why they're so useful in modern applications.

Definition of group law

- Defines a binary operation on the set of points on an elliptic curve that satisfies certain properties
- Provides a way to "add" two points on the curve to obtain a third point also on the curve
- Fundamental concept in the study of elliptic curves and their applications

Geometric interpretation

- The group law has a geometric interpretation based on the chord-and-tangent method
- To add two distinct points P and Q , draw a line through them and find the third point of intersection with the curve, then reflect that point across the x-axis to obtain $P + Q$
- If $P = Q$, draw the tangent line to the curve at P and find the second point of intersection, then reflect that point to obtain $2P$

Algebraic formulas

- The group law can also be expressed using algebraic formulas derived from the Weierstrass equation of the elliptic curve
- For distinct points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, the sum $P + Q = (x_3, y_3)$ is given by:
 - $x_3 = \lambda^2 - x_1 - x_2$
 - $y_3 = \lambda(x_1 - x_3) - y_1$
 - where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$
- For doubling a point $P = (x_1, y_1)$, the result $2P = (x_3, y_3)$ is given by:
 - $x_3 = \lambda^2 - 2x_1$
 - $y_3 = \lambda(x_1 - x_3) - y_1$
 - where $\lambda = \frac{3x_1^2 + a}{2y_1}$ and a is a parameter of the curve

Identity element

- The group law has an identity element, denoted as the point at infinity or O

- Adding any point P on the curve to O results in P itself: $P + O = P$
- O serves as the "zero" element in the group structure of the elliptic curve

Point at infinity

- The point at infinity is a special point that does not have finite coordinates
- It is the neutral element of the group law and acts as the identity
- Geometrically, it can be thought of as a point "infinitely far up" the y -axis
- In projective coordinates, it is often represented as $(0 : 1 : 0)$ or $(1 : 0 : 0)$

Inverse elements

- For every point P on the elliptic curve, there exists an inverse point $-P$ such that $P + (-P) = O$
- The inverse of a point $P = (x, y)$ is the point $-P = (x, -y)$, obtained by reflecting P across the x -axis
- Adding a point to its inverse results in the identity element, the point at infinity

Negatives of points

- The negative of a point $P = (x, y)$ is the point $-P = (x, -y)$
- Geometrically, the negative of a point is obtained by reflecting it across the x -axis
- The negative of the point at infinity is itself: $-O = O$
- The negative of a point satisfies the property $P + (-P) = O$

Associativity

- The group law on elliptic curves is associative, meaning that $(P + Q) + R = P + (Q + R)$ for any points P , Q , and R on the curve
- Associativity ensures that the order of addition does not affect the result
- This property is essential for the group structure of elliptic curves

Geometric proof

- The associativity of the group law can be proved geometrically using the chord-and-tangent method
- The proof involves considering various cases and showing that the resulting points are the same regardless of the order of addition
- The geometric proof relies on the properties of lines and their intersections with the elliptic curve

Algebraic proof

- Associativity can also be proved algebraically using the formulas for point addition and doubling
- The proof involves expanding the expressions for $(P + Q) + R$ and $P + (Q + R)$ and showing that they are equal
- The algebraic proof requires manipulating the equations and using the properties of the underlying field

Commutativity

- The group law on elliptic curves is commutative, meaning that $P + Q = Q + P$ for any points P and Q on the curve
- Commutativity implies that the order of addition does not matter
- This property distinguishes elliptic curves from some other algebraic structures (non-abelian groups)

Abelian group structure

- The commutativity of the group law makes the set of points on an elliptic curve an abelian group
- An abelian group is a group in which the binary operation is commutative
- The abelian group structure of elliptic curves is important for their applications in cryptography and number theory
- Other examples of abelian groups include the integers under addition and the non-zero real numbers under multiplication

Closure

- The group law on elliptic curves satisfies the closure property
- Closure means that the sum of any two points on the curve is also a point on the curve
- This property ensures that the set of points on an elliptic curve is closed under the group operation

Rational points

- If the coefficients of the elliptic curve equation are rational numbers, the curve may have rational points
- A rational point is a point whose coordinates are both rational numbers
- The set of rational points on an elliptic curve forms a subgroup under the group law
- The study of rational points on elliptic curves is a central topic in number theory (Diophantine equations)

Irrational points

- Elliptic curves also contain points with irrational coordinates
- An irrational point is a point where at least one of its coordinates is an irrational number
- The sum of a rational point and an irrational point is always an irrational point
- The closure property ensures that the sum of two irrational points is either a rational point or an irrational point on the curve

Computation of multiples

- Computing multiples of a point on an elliptic curve is a fundamental operation in elliptic curve cryptography

- The scalar multiple of a point P is denoted as kP , where k is an integer and represents adding P to itself k times
- Efficient algorithms exist for computing multiples of points on elliptic curves

Repeated addition

- A simple method for computing kP is through repeated addition
- Start with the point P and repeatedly add P to itself $k-1$ times to obtain kP
- Repeated addition is straightforward but inefficient for large values of k (linear time complexity)
- Example: To compute $5P$, calculate $P + P + P + P + P$

Double-and-add algorithm

- The double-and-add algorithm is a more efficient method for computing multiples of points
- It uses the binary representation of the scalar k and performs point doubling and addition based on the bits
- The algorithm has a logarithmic time complexity, making it suitable for practical applications
- Example: To compute $13P$ (binary: 1101), calculate $2(2(2P + P) + P) + P$

Graphical examples

- Visualizing the group law on elliptic curves can aid in understanding its geometric properties
- Graphical examples illustrate the addition of points and the structure of the group

Over real numbers

- When the elliptic curve is defined over the real numbers, it can be plotted in the Cartesian plane
- The curve has a smooth, symmetric shape that resembles a cubic equation
- The group law can be demonstrated by selecting points on the curve and graphically performing the chord-and-tangent method
- Real-valued elliptic curves provide a visual representation of the group law and its geometric interpretation

Over finite fields

- Elliptic curves can also be defined over finite fields, such as the integers modulo a prime number
- In finite fields, the elliptic curve consists of a finite set of points
- The group law still applies, but the graphical representation is discrete and lacks the smooth appearance of real-valued curves
- Finite field elliptic curves are important in cryptographic applications due to their discrete nature and the difficulty of solving certain problems (discrete logarithm)

Applications

- Elliptic curves and their group law have significant applications in various fields, particularly in cryptography and number theory
- The unique properties of elliptic curves make them suitable for constructing secure and efficient cryptographic schemes

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography approach based on the algebraic structure of elliptic curves
- ECC uses the group law and the difficulty of the elliptic curve discrete logarithm problem (ECDLP) to achieve secure communication
- Key exchange protocols (ECDH) and digital signature algorithms (ECDSA) are built upon the properties of elliptic curves
- ECC provides similar security levels to other cryptographic systems (RSA) with smaller key sizes, leading to improved efficiency

Elliptic curve factorization

- Elliptic curves can be used as a tool for integer factorization, which is the problem of decomposing a composite number into its prime factors
- The elliptic curve factorization method (ECM) is a probabilistic algorithm that attempts to find factors of a number using elliptic curves
- ECM relies on the group law and the structure of elliptic curves over finite fields
- While not the most efficient factorization method for general numbers, ECM is particularly effective for finding small factors and is used in conjunction with other algorithms (quadratic sieve, general number field sieve)

1.3 Elliptic curve point addition

Elliptic curve point addition is a key operation that combines two points on a curve to get a third. It's the foundation for the group structure of elliptic curves, crucial for cryptography and other applications.

The geometric interpretation involves drawing lines through points, while the algebraic formula provides efficient computation. Understanding point addition rules and special cases is essential for working with elliptic curves in various fields.

Definition of point addition

- Point addition is a fundamental operation on elliptic curves that allows us to combine two points on the curve to obtain a third point, also on the curve
- The set of points on an elliptic curve, together with the point at infinity, form an abelian group under the point addition operation
- Point addition is essential for many applications of elliptic curves, such as cryptography and factorization algorithms

Geometric interpretation

- Geometrically, point addition can be visualized by drawing a line through two points on the elliptic curve and finding the third point of intersection
- The third point of intersection is then reflected across the x-axis to obtain the sum of the two original points
- If the line is tangent to the curve at a point, the tangent point is counted twice as an intersection point

Algebraic formula

- The point addition operation can be expressed algebraically using the coordinates of the points involved
- For two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on an elliptic curve $y^2 = x^3 + ax + b$, their sum $R = (x_3, y_3)$ is given by:
 - $x_3 = \lambda^2 - x_1 - x_2$
 - $y_3 = \lambda(x_1 - x_3) - y_1$
 - where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ if $P \neq Q$, and $\lambda = \frac{3x_1^2 + a}{2y_1}$ if $P = Q$
- The algebraic formula allows for efficient computation of point addition in various implementations

Point addition rules

- Point addition on elliptic curves satisfies several important rules that make it a well-defined group operation
- These rules ensure that the set of points on an elliptic curve, together with the point at infinity, form an abelian group under point addition
- Understanding these rules is crucial for working with elliptic curves and their applications

Identity element

- The point at infinity, denoted as $\mathcal{O}$, serves as the identity element for point addition on elliptic curves
- For any point P on the curve, $P + \mathcal{O} = P$ and $\mathcal{O} + P = P$
- The point at infinity is a special point that does not have finite coordinates and is considered to be on every vertical line

Inverse elements

- For every point $P = (x, y)$ on an elliptic curve, there exists an inverse point $-P = (x, -y)$
- The sum of a point and its inverse is the point at infinity: $P + (-P) = \mathcal{O}$
- The inverse of the point at infinity is itself: $-\mathcal{O} = \mathcal{O}$

Commutativity

- Point addition on elliptic curves is commutative, meaning that for any two points P and Q on the curve, $P + Q = Q + P$

- Commutativity is an important property that simplifies many calculations and proofs involving elliptic curves
- The commutativity of point addition is a consequence of the geometric interpretation of the operation

Associativity

- Point addition on elliptic curves is associative, meaning that for any three points P , Q , and R on the curve, $(P + Q) + R = P + (Q + R)$
- Associativity allows for the unambiguous definition of point multiplication, which is the repeated addition of a point to itself
- The associativity of point addition is crucial for the group structure of elliptic curves and their use in cryptographic protocols

Point addition on real numbers

- When working with elliptic curves over the real numbers, point addition can be performed using the geometric and algebraic methods described earlier
- However, there are some special cases to consider when the x-coordinates of the points are identical or when the line connecting the points is vertical
- Understanding these cases is important for the correct implementation of point addition algorithms

Distinct x-coordinates

- When adding two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ with distinct x-coordinates ($x_1 \neq x_2$), the sum $R = (x_3, y_3)$ is computed using the standard algebraic formulas
- The slope of the line connecting P and Q is given by $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$
- The x-coordinate of the sum is $x_3 = \lambda^2 - x_1 - x_2$, and the y-coordinate is $y_3 = \lambda(x_1 - x_3) - y_1$

Identical x-coordinates

- When adding a point $P = (x, y)$ to itself (point doubling), or when adding two points with the same x-coordinate but different y-coordinates, the slope of the tangent line or vertical line is used in the computation
- For point doubling, the slope of the tangent line is given by $\lambda = \frac{3x^2 + a}{2y}$, where a is a parameter of the elliptic curve equation
- The x-coordinate of the sum is $x_3 = \lambda^2 - 2x$, and the y-coordinate is $y_3 = \lambda(x - x_3) - y$

Vertical line exception

- When adding two points with the same x-coordinate and opposite y-coordinates, the line connecting them is vertical
- In this case, the sum of the points is defined as the point at infinity, $\mathcal{O}$
- This exception is consistent with the geometric interpretation of point addition and ensures that the group law holds for all points on the curve

Point addition on finite fields

- Elliptic curves can also be defined over finite fields, such as the integers modulo a prime number ($\mathbb{F}_p$) or binary fields ($\mathbb{F}_{2^m}$)
- Point addition on elliptic curves over finite fields follows the same general principles as over real numbers, but with some additional considerations
- Finite field arithmetic is used to perform the necessary computations, and the field characteristic plays a role in the formulas and algorithms

Modular arithmetic

- When working with elliptic curves over the finite field $\mathbb{F}_p$, all computations are performed modulo the prime number p
- The elliptic curve equation is modified to $y^2 \equiv x^3 + ax + b \pmod{p}$, where a and b are constants in $\mathbb{F}_p$
- Point addition formulas are adapted to use modular arithmetic, ensuring that all intermediate results and the final sum are elements of the finite field

Field characteristic

- The characteristic of a finite field is the smallest positive integer k such that $\underbrace{1 + 1 + \dots + 1}_{k \text{ times}} = 0$ in the field
- For prime fields $\mathbb{F}_p$, the characteristic is the prime number p
- The field characteristic affects the elliptic curve equation and the point addition formulas, particularly in the case of binary fields ($\text{char} = 2$) and fields of characteristic 3 ($\text{char} = 3$)

Distinct x-coordinates

- When adding points with distinct x-coordinates on an elliptic curve over a finite field, the same formulas as in the real number case are used, but with modular arithmetic
- The slope λ is computed as $\lambda \equiv (y_2 - y_1)(x_2 - x_1)^{-1} \pmod{p}$, where $(x_2 - x_1)^{-1}$ is the modular multiplicative inverse of $x_2 - x_1$
- The x-coordinate of the sum is $x_3 \equiv \lambda^2 - x_1 - x_2 \pmod{p}$, and the y-coordinate is $y_3 \equiv \lambda(x_1 - x_3) - y_1 \pmod{p}$

Identical x-coordinates

- When adding a point to itself (point doubling) or adding points with the same x-coordinate but different y-coordinates on an elliptic curve over a finite field, the formulas are adapted using modular arithmetic
- For point doubling, the slope of the tangent line is $\lambda \equiv (3x^2 + a)(2y)^{-1} \pmod{p}$, where $(2y)^{-1}$ is the modular multiplicative inverse of $2y$
- The x-coordinate of the sum is $x_3 \equiv \lambda^2 - 2x \pmod{p}$, and the y-coordinate is $y_3 \equiv \lambda(x - x_3) - y \pmod{p}$

Computation of point addition

- Computing point addition on elliptic curves involves several steps, each of which can be optimized for efficiency and security
- The main steps include calculating the slope of the line connecting the points, finding the intersection of this line with the curve, and reflecting the intersection point across the x-axis
- Various algorithms and techniques can be used to perform these computations, depending on the specific requirements of the application

Slope calculation

- The first step in point addition is to calculate the slope of the line connecting the two points (or the tangent line in the case of point doubling)
- For points with distinct x-coordinates, the slope is computed as $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ (or using modular arithmetic in finite fields)
- For point doubling, the slope of the tangent line is computed as $\lambda = \frac{3x^2 + a}{2y}$ (or using modular arithmetic in finite fields)
- Efficient algorithms for modular multiplication, inversion, and division are used to compute the slope in finite field implementations

Intersection with curve

- Once the slope is calculated, the next step is to find the intersection point of the line with the elliptic curve
- The x-coordinate of the intersection point is computed as $x_3 = \lambda^2 - x_1 - x_2$ (or using modular arithmetic in finite fields)
- The y-coordinate of the intersection point is then computed using the elliptic curve equation: $y_3 = \lambda(x_1 - x_3) - y_1$ (or using modular arithmetic in finite fields)
- These computations involve modular exponentiation and multiplication, which can be optimized using techniques such as Montgomery multiplication or fast modular reduction

Reflection across x-axis

- The final step in point addition is to reflect the intersection point across the x-axis to obtain the sum of the two original points
- This is done by negating the y-coordinate of the intersection point: $(x_3, -y_3)$
- In finite field implementations, negation is performed using modular subtraction: $-y_3 \equiv p - y_3 \pmod{p}$ for prime fields, or using a specific negation algorithm for binary fields
- The reflection step ensures that the sum of the points is another point on the elliptic curve, satisfying the group law

Applications of point addition

- Point addition on elliptic curves has several important applications in cryptography and number theory

- These applications rely on the security and mathematical properties of elliptic curves, such as the difficulty of solving the discrete logarithm problem
- Some of the most notable applications include elliptic curve cryptography, elliptic curve factorization, and elliptic curve primality testing

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography system that uses the algebraic structure of elliptic curves over finite fields
- In ECC, point addition is used to perform scalar multiplication, which is the main operation in key generation, encryption, and digital signature algorithms
- ECC provides the same level of security as other public-key systems (e.g., RSA) with smaller key sizes, making it suitable for resource-constrained environments such as smartphones and IoT devices

Elliptic curve factorization

- Elliptic curve factorization (ECF) is a method for integer factorization that uses elliptic curves to find factors of large composite numbers
- The basic idea behind ECF is to construct an elliptic curve modulo the composite number and perform point additions to find points with certain properties that reveal factors of the original number
- ECF algorithms, such as Lenstra's elliptic curve factorization method, can be more efficient than other factorization methods for certain types of numbers

Elliptic curve primality testing

- Elliptic curve primality testing (ECPT) is a probabilistic primality test that uses elliptic curves to determine whether a given integer is prime or composite
- ECPT algorithms, such as Goldwasser-Kilian and Atkin-Morain tests, rely on the properties of point addition and the group structure of elliptic curves over finite fields
- These tests are based on the fact that the number of points on an elliptic curve modulo a prime number satisfies certain conditions, which can be checked by performing point additions and comparing the results with the expected values

Geometric properties

- The geometric properties of elliptic curves and point addition provide valuable insights into the structure and behavior of these mathematical objects
- Some of the most important geometric properties include the role of tangent lines, collinear points, and inflection points
- Understanding these properties is crucial for the analysis and design of elliptic curve-based algorithms and protocols

Tangent lines

- Tangent lines play a central role in the geometric interpretation of point addition on elliptic curves
- When adding a point to itself (point doubling), the tangent line at that point is used to find the intersection point with the curve, which is then reflected across the x-axis to obtain the sum

- The slope of the tangent line is given by $\lambda = \frac{3x^2 + a}{2y}$, where (x, y) is the point being doubled and a is a parameter of the elliptic curve equation
- Tangent lines are also used in the computation of the group law and in the analysis of the geometric properties of elliptic curves

Collinear points

- Three points on an elliptic curve are said to be collinear if they lie on the same straight line
- Collinearity is closely related to point addition, as the sum of two points on an elliptic curve is defined as the reflection of the third point of intersection of the line connecting the two points with the curve
- If three points are collinear, their sum (using the group law) is equal to the point at infinity, $\mathcal{O}$
- Collinearity properties are used in various elliptic curve-based algorithms, such as the computation of torsion points and the analysis of the group structure

Inflection points

- An inflection point on an elliptic curve is a point where the curve changes from being concave to convex, or vice versa
- Mathematically, an inflection point is a point where the second derivative of the elliptic curve equation equals zero
- For elliptic curves in short Weierstrass form, $y^2 = x^3 + ax + b$, the inflection points are the points where $y = 0$ (i.e., the points where the curve intersects the x-axis)
- Inflection points play a role in the analysis of the geometric and algebraic properties of elliptic curves, and they are used in some cryptographic protocols and attacks

Group law for elliptic curves

- The group law for elliptic curves defines the algebraic structure of the set of points on an elliptic curve, together with the point at infinity
- Under the group law, point addition satisfies the axioms of an abelian group, which include closure, associativity, the existence of an identity element, and the existence of inverse elements
- The group law is essential for understanding the mathematical properties of elliptic curves and their applications in cryptography and other fields

Closure under addition

- The set of points on an elliptic curve, together with the point at infinity, is closed under the point addition operation
- This means that the sum of any two points on the curve (or the point at infinity) is always another point on the curve (or the point at infinity)
- Closure under addition is one of the fundamental properties that make elliptic curves suitable for use in cryptography and other applications
- It ensures that point addition is a well-defined operation and that the set of points forms a mathematical group

Abelian group structure

- The set of points on an elliptic curve, together with the point at infinity, forms an abelian group under the point addition operation
- An abelian group is a group in which the group operation (in this case, point addition) is commutative, meaning that $P + Q = Q + P$ for any two points P and Q on the curve
- The abelian group structure of elliptic curves is characterized by the following axioms:
- Closure: $\forall P, Q \in E, P + Q \in E$

1.4 Elliptic curve point doubling

Elliptic curve point doubling is a key operation in elliptic curve arithmetic. It involves taking a point on the curve and computing its double, which is crucial for many cryptographic algorithms and mathematical applications.

Understanding point doubling is essential for grasping the group law of elliptic curves. It forms the basis for generating cyclic subgroups and plays a vital role in elliptic curve cryptography, factorization methods, and primality proving.

Definition of point doubling

- Point doubling is a fundamental operation in elliptic curve arithmetic that takes a point P on an elliptic curve and computes $2P$, which is the result of adding P to itself
- This operation is crucial for many elliptic curve based algorithms, including key generation, encryption, and digital signature schemes

Algebraic definition

- Algebraically, point doubling is defined as follows: given a point $P = (x_1, y_1)$ on an elliptic curve E , the double of P , denoted as $2P = (x_2, y_2)$, is the point obtained by drawing the tangent line to E at P and finding the other point of intersection of this line with E
- The resulting point is then reflected across the X -axis to obtain $2P$
- The algebraic formula for point doubling depends on the specific form of the elliptic curve equation (short Weierstrass, Montgomery, twisted Edwards)

Geometric interpretation

- Geometrically, point doubling can be visualized as follows: consider an elliptic curve E and a point P on E
- To find $2P$, draw the tangent line to E at P , which intersects E at another point, say Q
- Reflect Q across the X -axis to obtain $2P$
- This geometric interpretation provides an intuitive understanding of the point doubling operation and its relationship to the shape of the elliptic curve

Point doubling formula

- The point doubling formula is a set of equations used to compute the coordinates of $2P$ given the coordinates of P
- The specific formula depends on the form of the elliptic curve equation being used (short Weierstrass, Montgomery, twisted Edwards)
- These formulas are derived using the algebraic definition of point doubling and the properties of the elliptic curve

Derivation of formula

- The derivation of the point doubling formula involves finding the equation of the tangent line at P , substituting it into the elliptic curve equation, and solving for the coordinates of the other point of intersection
- This process involves algebraic manipulation and the use of the elliptic curve's group law properties
- The resulting equations are then simplified to obtain the final point doubling formula

Formula for short Weierstrass form

- For an elliptic curve in short Weierstrass form, $y^2 = x^3 + ax + b$, the point doubling formula is given by:
- $x_2 = \lambda^2 - 2x_1$
- $y_2 = \lambda(x_1 - x_2) - y_1$
- where $\lambda = \frac{3x_1^2 + a}{2y_1}$

Formula for Montgomery form

- For an elliptic curve in Montgomery form, $By^2 = x^3 + Ax^2 + x$, the point doubling formula is given by:
- $x_2 = \frac{(x_1^2 - 1)^2}{4x_1(x_1^2 + Ax_1 + 1)}$
- $y_2 = \frac{(2x_1 + x_2 + A)(x_1 - x_2) - 2y_1}{B}$

Formula for twisted Edwards form

- For an elliptic curve in twisted Edwards form, $ax^2 + y^2 = 1 + dx^2y^2$, the point doubling formula is given by:
- $x_2 = \frac{2xy}{ax^2 + y^2}$
- $y_2 = \frac{y^2 - ax^2}{2 - ax^2 - y^2}$

Point doubling in group law

- Point doubling plays a crucial role in the group law of elliptic curves, which defines the rules for point addition and multiplication

- The group law states that for any two points P and Q on an elliptic curve, there exists a unique point R such that $P + Q = R$
- Point doubling is a special case of point addition where $P = Q$

Role in generating cyclic subgroup

- Point doubling is essential for generating cyclic subgroups of elliptic curve groups
- Given a point P on an elliptic curve, the cyclic subgroup generated by P is the set of all points that can be obtained by repeatedly applying the point doubling operation to P
- The cyclic subgroup generated by P is denoted as $\langle P \rangle = \{P, 2P, 4P, 8P, \dots\}$

Relationship to point addition

- Point doubling is closely related to point addition, as both operations are part of the elliptic curve group law
- Point addition is the operation of adding two distinct points on an elliptic curve, while point doubling is the operation of adding a point to itself
- The formulas for point addition and point doubling share some similarities, but they differ in the specific equations used and the handling of special cases (point at infinity, singularities)

Efficient computation of point doubling

- Efficient computation of point doubling is crucial for the performance of elliptic curve based algorithms, as point doubling is a frequently used operation
- Several techniques can be employed to optimize the computation of point doubling, including the use of different coordinate systems and the application of explicit formulas

Affine coordinates vs projective coordinates

- Elliptic curve points can be represented using different coordinate systems, such as affine coordinates and projective coordinates
- Affine coordinates represent points using two values (x, y) , while projective coordinates use three values (X, Y, Z) to represent points
- Projective coordinates can help avoid the need for expensive field inversions during point doubling, which can improve performance

Explicit formulas for point doubling

- Explicit formulas for point doubling are optimized equations that minimize the number of field operations (additions, multiplications, inversions) required to compute $2P$
- These formulas are derived by applying algebraic manipulations and substitutions to the standard point doubling formulas
- Explicit formulas can be tailored to specific elliptic curve forms (short Weierstrass, Montgomery, twisted Edwards) and coordinate systems (affine, projective)

Cost analysis of point doubling

- Cost analysis involves evaluating the computational cost of point doubling in terms of the number and type of field operations required
- The cost of point doubling depends on the elliptic curve form, coordinate system, and the specific explicit formula used
- By analyzing the cost of different point doubling methods, one can choose the most efficient approach for a given elliptic curve based application

Applications of point doubling

- Point doubling is a fundamental operation in elliptic curve arithmetic and has several important applications in cryptography and computational number theory
- These applications rely on the security and efficiency of elliptic curve based algorithms, which in turn depend on the performance of point doubling

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography approach that uses the algebraic structure of elliptic curves over finite fields
- Point doubling is a crucial operation in ECC, as it is used in key generation, encryption, decryption, and digital signature algorithms
- The security of ECC relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), which involves finding the scalar multiple of a given point

Elliptic curve factorization method

- The elliptic curve factorization method (ECM) is an integer factorization algorithm that uses elliptic curves to find small factors of large composite numbers
- ECM relies on the fact that the order of a random elliptic curve modulo a composite number is likely to have small prime factors
- Point doubling is used in ECM to compute scalar multiples of points on elliptic curves, which is a key step in the factorization process

Elliptic curve primality proving

- Elliptic curve primality proving (ECP) is a method for proving the primality of large integers using elliptic curves
- ECP is based on the idea that if a certain condition holds for an elliptic curve modulo a given integer, then that integer must be prime
- Point doubling is used in ECP to compute the order of points on elliptic curves, which is necessary for verifying the primality condition

Challenges in point doubling

- While point doubling is a well-defined operation in elliptic curve arithmetic, there are some challenges that need to be addressed when implementing point doubling in practice

- These challenges include handling special cases, such as the point at infinity and singularities, and ensuring the security of point doubling against side-channel attacks

Handling point at infinity

- The point at infinity, denoted as $\mathcal{O}$, is a special point on an elliptic curve that serves as the identity element for point addition
- When doubling the point at infinity, the result is the point at infinity itself: $2\mathcal{O} = \mathcal{O}$
- Implementations of point doubling must handle this special case correctly to ensure the consistency and correctness of elliptic curve arithmetic

Dealing with singularities

- Singularities are points on an elliptic curve where the tangent line is not well-defined, which can occur when the denominator of the point doubling formula becomes zero
- In the short Weierstrass form, singularities can happen when the Y -coordinate of the point being doubled is zero
- To handle singularities, implementations of point doubling must check for these special cases and apply appropriate logic to compute the correct result

Resistance to side-channel attacks

- Side-channel attacks are a type of cryptanalytic attack that exploits information leaked by the physical implementation of a cryptographic algorithm, such as timing, power consumption, or electromagnetic radiation
- Point doubling, being a fundamental operation in elliptic curve cryptography, can be a target for side-channel attacks
- To resist these attacks, implementations of point doubling must employ countermeasures, such as constant-time execution, randomization, and masking, to minimize the leakage of sensitive information

1.5 Elliptic curve discriminant and j-invariant

Elliptic curve discriminant and j-invariant are key concepts in the study of elliptic curves. These mathematical tools help characterize the shape, singularities, and isomorphism classes of curves, providing crucial insights into their properties and behavior.

Understanding these invariants is essential for applications in cryptography, number theory, and algebraic geometry. The discriminant determines curve smoothness, while the j-invariant classifies curves into isomorphism classes, both playing vital roles in elliptic curve theory and its practical implementations.

Definition of elliptic curve discriminant

- The elliptic curve discriminant is a fundamental invariant that characterizes the singularity of an elliptic curve
- It is a polynomial expression in the coefficients of the elliptic curve equation
- The discriminant determines whether an elliptic curve is singular or non-singular, which has important implications for its geometric and algebraic properties

Discriminant formula for Weierstrass form

- For an elliptic curve in Weierstrass form $y^2 = x^3 + ax + b$, the discriminant is given by the formula $\Delta = -16(4a^3 + 27b^2)$
- The coefficients a and b in the Weierstrass equation directly determine the value of the discriminant
- The discriminant formula can be derived using the properties of the elliptic curve and its singularities

Discriminant as polynomial in coefficients

- The discriminant is a polynomial expression in terms of the coefficients of the elliptic curve equation
- For curves in Weierstrass form, the discriminant is a sixth-degree polynomial in a and b
- The coefficients of the discriminant polynomial encode information about the singularities and geometric properties of the curve

Non-zero discriminant for non-singular curves

- An elliptic curve is non-singular if and only if its discriminant is non-zero
- A non-zero discriminant indicates that the curve has no cusps, self-intersections, or other singularities
- Non-singular elliptic curves have well-defined group law and arithmetic operations, making them suitable for cryptographic applications

Geometric interpretation of discriminant

- The discriminant provides insight into the geometric properties and singularities of an elliptic curve
- It determines whether the curve is smooth or has singular points, such as cusps or self-intersections
- The geometric interpretation of the discriminant helps in understanding the shape and behavior of elliptic curves

Discriminant and curve smoothness

- A non-zero discriminant indicates that the elliptic curve is smooth and has no singular points
- Smooth curves have a well-defined tangent line at every point and do not intersect themselves
- The smoothness of an elliptic curve is crucial for its algebraic and geometric properties, such as the group law and rational points

Singular vs non-singular curves

- Elliptic curves can be classified as singular or non-singular based on the value of their discriminant
- Singular curves have a discriminant equal to zero and possess singular points (cusps or self-intersections)
- Non-singular curves have a non-zero discriminant and are smooth, without any singularities

Cusps and self-intersections

- Singular elliptic curves can have cusps, which are points where the curve has a sharp corner or a point of self-tangency

- Self-intersections occur when the curve crosses itself at a point, creating a node or a multiple point
- The presence of cusps or self-intersections affects the geometric and algebraic properties of the curve, such as the group law and rational points

Computing the discriminant

- The discriminant can be computed using the coefficients of the elliptic curve equation
- Different forms of the elliptic curve equation lead to different formulas for the discriminant
- Efficient algorithms exist for calculating the discriminant, which is important for various applications

Discriminant for curves in Weierstrass form

- For curves in Weierstrass form $y^2 = x^3 + ax + b$, the discriminant is computed as $\Delta = -16(4a^3 + 27b^2)$
- The coefficients a and b are substituted into the discriminant formula to obtain the value of Δ
- Example: For the curve $y^2 = x^3 - 3x + 2$, we have $a = -3$ and $b = 2$, so $\Delta = -16(4(-3)^3 + 27(2)^2) = -16(-108 + 108) = 0$, indicating a singular curve

Discriminant for curves in other forms

- Elliptic curves can be represented in various forms, such as the generalized Weierstrass form or the Legendre form
- Each form has its own discriminant formula, which can be derived using the properties of the curve and its singularities
- Example: For the Legendre form $y^2 = x(x - 1)(x - \lambda)$, the discriminant is given by $\Delta = 16\lambda^2(\lambda - 1)^2$

Algorithms for discriminant calculation

- Efficient algorithms have been developed to compute the discriminant of elliptic curves
- These algorithms take advantage of the structure of the discriminant polynomial and use techniques such as modular arithmetic and polynomial factorization
- Optimized discriminant calculation is important for applications like elliptic curve cryptography, where the discriminant needs to be computed quickly and securely

Definition of j-invariant

- The j-invariant is another important invariant associated with elliptic curves
- It is a rational function of the coefficients of the elliptic curve equation
- The j-invariant characterizes the isomorphism class of an elliptic curve and provides information about its symmetries and complex multiplication

j-invariant formula

- The j-invariant is defined in terms of the coefficients of the elliptic curve equation

- For curves in Weierstrass form $y^2 = x^3 + ax + b$, the j-invariant is given by the formula
$$j = 1728 \frac{4a^3}{4a^3 + 27b^2}$$

- The formula involves the coefficients a and b and a scaling factor of 1728

j-invariant as function of coefficients

- The j-invariant is a rational function of the coefficients of the elliptic curve equation
- It depends on the specific form of the equation and the values of the coefficients
- Changes in the coefficients can lead to different values of the j-invariant, indicating different isomorphism classes of elliptic curves

j-invariant and isomorphism classes

- The j-invariant characterizes the isomorphism class of an elliptic curve
- Elliptic curves with the same j-invariant are isomorphic, meaning they have the same geometric and algebraic properties up to a change of coordinates
- The j-invariant provides a way to classify elliptic curves into equivalence classes based on their isomorphism type

Properties of j-invariant

- The j-invariant possesses several important properties that make it a valuable tool in the study of elliptic curves
- It is invariant under isomorphisms, captures information about the automorphism group, and is related to complex multiplication
- Understanding the properties of the j-invariant is crucial for various applications in cryptography, number theory, and algebraic geometry

j-invariant and curve isomorphisms

- The j-invariant is an isomorphism invariant, meaning it remains unchanged under isomorphisms of elliptic curves
- If two elliptic curves have the same j-invariant, they are isomorphic and share the same geometric and algebraic properties
- Isomorphic curves can be transformed into each other through a change of coordinates, preserving the essential features of the curve

j-invariant and automorphism group

- The j-invariant is related to the automorphism group of an elliptic curve
- The automorphism group consists of the set of isomorphisms from the curve to itself, which preserve the group law and the point at infinity
- The size and structure of the automorphism group can be determined from the value of the j-invariant

j-invariant and complex multiplication

- The j-invariant plays a crucial role in the theory of complex multiplication of elliptic curves
- Complex multiplication occurs when the endomorphism ring of an elliptic curve is larger than the integers, leading to additional symmetries and algebraic properties
- The j-invariant of an elliptic curve with complex multiplication takes on special values, known as singular moduli, which have important arithmetic and geometric significance

Relationship between discriminant and j-invariant

- The discriminant and j-invariant are closely related invariants of elliptic curves
- They provide complementary information about the singularities, isomorphism classes, and arithmetic properties of the curve
- Understanding the relationship between the discriminant and j-invariant is important for various applications and theoretical investigations

Discriminant in terms of j-invariant

- The discriminant can be expressed in terms of the j-invariant using a specific formula
- For curves in Weierstrass form, the discriminant is given by $\Delta = -\frac{1}{1728}j(j - 1728)$
- This formula highlights the direct relationship between the discriminant and j-invariant, showing how they are interconnected

j-invariant in terms of discriminant

- Conversely, the j-invariant can be expressed in terms of the discriminant using the inverse of the previous formula
- For curves in Weierstrass form, the j-invariant is given by $j = 1728 \frac{4a^3}{4a^3 + 27b^2} = 1728 \frac{4a^3}{\Delta}$
- This formula demonstrates how the j-invariant can be computed directly from the discriminant and the coefficients of the curve

Singular curves and j-invariant

- The j-invariant provides information about the singularities of an elliptic curve
- For singular curves with a discriminant equal to zero, the j-invariant takes on a specific value
- The value of the j-invariant for singular curves depends on the type of singularity (cusp or node) and can be used to classify singular curves

Applications of discriminant and j-invariant

- The discriminant and j-invariant have numerous applications in various fields of mathematics and computer science
- They play a crucial role in elliptic curve cryptography, number theory, and the study of moduli spaces of elliptic curves

- Understanding the applications of these invariants is essential for both theoretical research and practical implementations

Discriminant and j-invariant in cryptography

- Elliptic curve cryptography relies on the properties of the discriminant and j-invariant for secure communication and digital signatures
- The discriminant is used to ensure that the chosen elliptic curve is non-singular and suitable for cryptographic purposes
- The j-invariant is used to classify elliptic curves and select curves with desirable properties, such as resistance to certain attacks or efficient arithmetic operations

Discriminant and j-invariant in number theory

- The discriminant and j-invariant have important connections to various concepts in number theory, such as modular forms and class field theory
- They are used to study the arithmetic properties of elliptic curves over different number fields and to investigate the relationships between elliptic curves and other mathematical objects
- The values of the discriminant and j-invariant can provide insights into the structure and properties of number fields and their associated elliptic curves

Discriminant and j-invariant in moduli spaces

- The discriminant and j-invariant play a central role in the study of moduli spaces of elliptic curves
- Moduli spaces are geometric objects that parametrize families of elliptic curves with certain properties or isomorphism classes
- The discriminant and j-invariant are used to define and study the structure of these moduli spaces, providing a way to classify and understand the relationships between different elliptic curves

1.6 Elliptic curves over various fields

Elliptic curves are fascinating mathematical objects studied across various fields. They have a rich structure that connects complex analysis, number theory, and cryptography. Understanding their behavior over different fields is key to grasping their full potential.

This topic explores elliptic curves defined over real numbers, complex numbers, finite fields, local fields, and global fields. We'll examine their unique properties, group structures, and applications in each setting, building a comprehensive view of these powerful curves.

Definition of elliptic curves

- Elliptic curves are a fundamental object of study in algebraic geometry and number theory
- They have a rich structure and connections to various areas of mathematics, including complex analysis, arithmetic geometry, and cryptography
- Understanding the basic definitions and properties of elliptic curves is essential for studying their behavior over different fields

Weierstrass equation

- An elliptic curve can be defined by a Weierstrass equation of the form $y^2 = x^3 + ax + b$, where a and b are constants
- The Weierstrass equation provides a canonical way to represent elliptic curves and study their properties
- The coefficients a and b determine the shape and features of the elliptic curve
- The Weierstrass equation can be generalized to other fields, such as finite fields or p-adic numbers

Projective closure

- The projective closure of an elliptic curve is obtained by adding a point at infinity, denoted by $\mathcal{O}$
- The point at infinity serves as the identity element for the group law on the elliptic curve
- The projective closure allows for a more uniform treatment of the curve and its points
- In projective coordinates, the Weierstrass equation becomes $Y^2Z = X^3 + aXZ^2 + bZ^3$

Discriminant and singularities

- The discriminant of an elliptic curve, denoted by Δ , is a quantity that determines whether the curve is singular or nonsingular
- For the Weierstrass equation $y^2 = x^3 + ax + b$, the discriminant is given by $\Delta = -16(4a^3 + 27b^2)$
- If $\Delta \neq 0$, the elliptic curve is nonsingular and has distinct roots
- If $\Delta = 0$, the elliptic curve is singular and has a node or a cusp singularity
- Singular elliptic curves have different properties and are not suitable for many applications

Elliptic curves over real numbers

- Elliptic curves over the real numbers have a rich geometric structure and can be visualized as graphs in the Cartesian plane
- Studying elliptic curves over the reals provides intuition and insights into their behavior and properties
- Many concepts and results for elliptic curves over the reals can be generalized to other fields

Graph and basic properties

- The graph of an elliptic curve over the reals is a smooth curve that is symmetric about the x-axis
- The curve intersects the x-axis at three points (counting multiplicity) or a single point if the curve is singular
- The shape of the curve depends on the coefficients a and b in the Weierstrass equation
- The curve has two components: an infinite component and a finite component (if the curve is nonsingular)

Group law and geometric interpretation

- Elliptic curves over the reals have a group law that allows for the addition of points on the curve
- The group law has a geometric interpretation based on the chord-and-tangent method
- To add two points P and Q , draw a line through P and Q (or a tangent line if $P = Q$) and find the third point of intersection with the curve, denoted by R
- The sum $P + Q$ is defined as the reflection of R about the x-axis
- The point at infinity $\mathcal{O}$ serves as the identity element for the group law
- The group law satisfies the associative, inverse, and identity properties

Torsion points and generators

- Torsion points on an elliptic curve over the reals are points of finite order under the group law
- The order of a torsion point P is the smallest positive integer n such that $nP = \mathcal{O}$
- The set of torsion points forms a subgroup of the elliptic curve group
- The structure of the torsion subgroup is described by Mazur's theorem, which states that it can be isomorphic to $\mathbb{Z}/n\mathbb{Z}$ for $n = 1, 2, \dots, 10, 12$ or $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2n\mathbb{Z}$ for $n = 1, 2, 3, 4$
- Generators of the elliptic curve group are points that generate the entire group under the group law
- The rank of an elliptic curve over the reals is the number of independent generators (modulo torsion)

Elliptic curves over complex numbers

- Elliptic curves over the complex numbers have a rich structure and connections to complex analysis and algebraic geometry
- The study of elliptic curves over the complex numbers reveals deep properties and leads to important results in number theory
- Many concepts and techniques from complex analysis, such as lattices and complex tori, play a crucial role in understanding elliptic curves over $\mathbb{C}$

Lattices and complex tori

- A lattice in the complex plane is a discrete subgroup of $\mathbb{C}$ of the form $\Lambda = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$, where ω_1 and ω_2 are linearly independent complex numbers
- The quotient space $\mathbb{C}/\Lambda$ is called a complex torus and has the structure of a compact Riemann surface
- Every complex torus is isomorphic to an elliptic curve over $\mathbb{C}$
- The periods ω_1 and ω_2 determine the shape and properties of the corresponding elliptic curve

Isomorphism with complex tori

- There is a canonical isomorphism between elliptic curves over $\mathbb{C}$ and complex tori
- The isomorphism is given by the Weierstrass $\wp$ -function and its derivative

- The Weierstrass $\wp$ -function is a doubly periodic meromorphic function that satisfies the differential equation $(\wp')^2 = 4\wp^3 - g_2\wp - g_3$, where g_2 and g_3 are constants depending on the lattice
- The isomorphism allows for the study of elliptic curves using the tools and techniques of complex analysis

Endomorphism ring and complex multiplication

- An endomorphism of an elliptic curve over $\mathbb{C}$ is a complex-analytic map from the curve to itself that preserves the group law
- The set of endomorphisms of an elliptic curve forms a ring, called the endomorphism ring
- The endomorphism ring is either $\mathbb{Z}$ or an order in an imaginary quadratic field
- Elliptic curves with complex multiplication are those whose endomorphism ring is larger than $\mathbb{Z}$
- Complex multiplication plays a crucial role in the study of elliptic curves and their arithmetic properties
- Elliptic curves with complex multiplication have special properties and are important in cryptography and the construction of abelian extensions of number fields

Elliptic curves over finite fields

- Elliptic curves over finite fields have important applications in cryptography and coding theory
- The study of elliptic curves over finite fields reveals interesting arithmetic and algebraic properties
- Many concepts and results from the theory of elliptic curves over $\mathbb{C}$ and $\mathbb{R}$ can be adapted to the finite field setting

Definition and basic properties

- An elliptic curve over a finite field $\mathbb{F}_q$ is defined by a Weierstrass equation $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{F}_q$ and the discriminant $\Delta \neq 0$
- The set of points on an elliptic curve over $\mathbb{F}_q$, denoted by $E(\mathbb{F}_q)$, consists of the solutions to the Weierstrass equation together with the point at infinity $\mathcal{O}$
- The set $E(\mathbb{F}_q)$ forms a finite abelian group under the chord-and-tangent group law
- The group law on $E(\mathbb{F}_q)$ can be computed efficiently using arithmetic in the finite field

Group structure and order of points

- The group structure of $E(\mathbb{F}_q)$ is determined by the number of points on the curve, denoted by $\#E(\mathbb{F}_q)$
- The order of a point $P \in E(\mathbb{F}_q)$ is the smallest positive integer n such that $nP = \mathcal{O}$
- The group $E(\mathbb{F}_q)$ is either cyclic or isomorphic to a product of two cyclic groups
- The possible group structures of $E(\mathbb{F}_q)$ are classified by the Rück-Voloch theorem
- The number of points on an elliptic curve over $\mathbb{F}_q$ satisfies the Hasse-Weil bound: $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$

Hasse's theorem and bounds

- Hasse's theorem provides a tight bound on the number of points on an elliptic curve over a finite field

- The theorem states that $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- The quantity $a_q = q+1 - \#E(\mathbb{F}_q)$ is called the trace of Frobenius and satisfies $|a_q| \leq 2\sqrt{q}$
- The Hasse bound is a consequence of the Riemann hypothesis for the zeta function of the elliptic curve
- The Hasse bound is used in the construction of elliptic curve cryptographic schemes and in the analysis of their security

Supersingular vs ordinary curves

- Elliptic curves over finite fields can be classified as either supersingular or ordinary
- An elliptic curve E over $\mathbb{F}_q$ is supersingular if the trace of Frobenius a_q satisfies $a_q \equiv 0 \pmod{p}$, where $q = p^n$ and p is the characteristic of the field
- Equivalently, E is supersingular if the endomorphism ring of E over the algebraic closure of $\mathbb{F}_q$ is an order in a quaternion algebra
- Supersingular curves have special properties and are important in cryptography and the construction of abelian varieties
- Ordinary curves are those that are not supersingular
- Most elliptic curves over finite fields are ordinary, and they have different properties compared to supersingular curves

Elliptic curves over local fields

- Local fields are an important tool in number theory and arithmetic geometry
- The study of elliptic curves over local fields provides insights into their local behavior and leads to global results
- Many concepts and techniques from the theory of elliptic curves over $\mathbb{C}$ and $\mathbb{R}$ can be adapted to the local field setting

p-adic numbers and local fields

- The p-adic numbers $\mathbb{Q}_p$ are a completion of the rational numbers with respect to the p-adic absolute value
- A local field is a field that is complete with respect to a discrete valuation and has a finite residue field
- Examples of local fields include the p-adic numbers $\mathbb{Q}_p$, the field of formal Laurent series $\mathbb{F}_q((t))$, and the completion of a number field at a prime ideal
- Local fields have a rich arithmetic structure and are used in the study of Diophantine equations and arithmetic geometry

Reduction modulo p and Néron models

- The reduction of an elliptic curve E over a local field K modulo the maximal ideal of the valuation ring of K is an important tool in the study of elliptic curves
- The reduction of E can be either good, multiplicative, or additive, depending on the valuation of the discriminant of E

- The Néron model of an elliptic curve E over a local field K is a smooth group scheme over the valuation ring of K that extends E
- The Néron model captures the local behavior of the elliptic curve and its points
- The special fiber of the Néron model is related to the reduction type of the elliptic curve

Kodaira-Néron classification of special fibers

- The Kodaira-Néron classification describes the possible special fibers of the Néron model of an elliptic curve over a local field
- The classification is based on the reduction type of the elliptic curve and the structure of the special fiber
- The possible reduction types are: I_n (good reduction), I_n^* (multiplicative reduction), II, III, IV, II^* , III^* , IV^* (additive reduction)
- The special fiber can have several components and can be described using the concept of intersection graphs
- The Kodaira-Néron classification is used in the study of the arithmetic and geometric properties of elliptic curves over local fields

Tate's algorithm and local zeta functions

- Tate's algorithm is a method for computing the reduction type and the local zeta function of an elliptic curve over a local field
- The algorithm is based on the properties of the Weierstrass equation of the elliptic curve and the valuation of its coefficients
- The local zeta function of an elliptic curve E over a local field K is a generating function that encodes information about the number of points on the reductions of E modulo powers of the maximal ideal of K
- The local zeta function is a rational function and satisfies a functional equation
- Tate's algorithm and local zeta functions are important tools in the study of the arithmetic of elliptic curves and their L-functions

Elliptic curves over global fields

- Global fields are an important class of fields in number theory and arithmetic geometry
- The study of elliptic curves over global fields combines local and global techniques and leads to deep results in Diophantine geometry
- Many concepts and results from the theory of elliptic curves over local fields and finite fields can be combined to study elliptic curves over global fields

Mordell-Weil theorem and rank

- The Mordell-Weil theorem states that the group of rational points $E(K)$ on an elliptic curve E over a number field K is finitely generated
- The group $E(K)$ is isomorphic to $E(K)_{\text{tors}} \oplus \mathbb{Z}^r$, where $E(K)_{\text{tors}}$ is the torsion subgroup and r is the rank of E over K

- The rank of an elliptic curve measures the size of the free part of the group of rational points
- Computing the rank of an elliptic curve is a difficult problem, and there are no known algorithms for determining the rank in general
- The Birch and Swinnerton-Dyer conjecture relates the rank of an elliptic curve to the behavior of its L-function at $s = 1$

Torsion subgroup and Nagell-Lutz theorem

- The torsion subgroup $E(K)_{\text{tors}}$ of an elliptic curve E over a number field K consists of the points of finite order
- The structure of the torsion subgroup is described by the Mazur-Kenku theorem, which states that $E(K)_{\text{tors}}$ is isomorphic to one of 15 possible groups
- The Nagell-Lutz theorem provides a criterion for determining whether a point on an elliptic curve over $\mathbb{Q}$ is a torsion point
- The theorem states that if $P = (x, y)$ is a torsion point on $E : y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$, then $x, y \in \mathbb{Z}$ and either $y = 0$ or y^2 divides the discriminant of E
- The Nagell-Lutz theorem is a useful tool for computing the torsion subgroup of an elliptic curve over $\mathbb{Q}$

Height functions and canonical height

- Height functions are an important tool in Diophantine geometry and the study of rational points on algebraic varieties
- The height of a point measures its arithmetic complexity and provides a way to quantify the size of rational points
- The naive height of a point $P = (x, y)$ on

Unit 2 – Elliptic Curves in Finite Fields

2.1 Finite field arithmetic

Finite field arithmetic is the backbone of elliptic curve cryptography. It provides the mathematical foundation for defining and operating on elliptic curves, enabling secure and efficient cryptographic schemes.

Understanding prime fields and binary fields is crucial. These finite structures support addition, subtraction, multiplication, and division operations, forming the basis for elliptic curve operations like point addition and scalar multiplication.

Finite field fundamentals

- Finite fields, also known as Galois fields, are algebraic structures with a finite number of elements that support addition, subtraction, multiplication, and division operations
- Understanding the properties and characteristics of finite fields is crucial for elliptic curve cryptography as they form the foundation for defining elliptic curves and performing arithmetic operations on them

Prime fields and binary fields

- Finite fields can be classified into two main categories: prime fields ($\mathbb{F}_p$) and binary fields ($\mathbb{F}_{2^m}$)
- Prime fields are defined over a prime number p and have elements ranging from 0 to $p - 1$
- Example: $\mathbb{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
- Binary fields are defined over a power of 2, denoted as 2^m , and have elements represented as polynomials with coefficients in $\mathbb{F}_2$
- Example: $\mathbb{F}_{2^3} = \{0, 1, x, x + 1, x^2, x^2 + 1, x^2 + x, x^2 + x + 1\}$

Field order and characteristic

- The order of a finite field is the number of elements it contains, denoted as q
- For prime fields, the order is equal to the prime number p , while for binary fields, the order is 2^m
- The characteristic of a finite field is the smallest positive integer n such that $n \cdot 1 = 0$, where 1 is the multiplicative identity of the field
- For prime fields, the characteristic is equal to p
- For binary fields, the characteristic is always 2

Primitive and minimal polynomials

- In binary fields, elements are represented as polynomials, and the field arithmetic is performed modulo an irreducible polynomial
- A primitive polynomial is an irreducible polynomial that generates all non-zero elements of the field through successive powers of a primitive element
- Example: $x^3 + x + 1$ is a primitive polynomial over $\mathbb{F}_2$

- A minimal polynomial is the lowest degree monic polynomial that has a given element as a root
- Used to define field extensions and perform field arithmetic

Arithmetic operations in finite fields

- Finite field arithmetic forms the basis for elliptic curve operations, such as point addition, point doubling, and scalar multiplication
- Efficient implementation of finite field arithmetic is crucial for the performance and security of elliptic curve cryptographic schemes

Addition and subtraction

- Addition and subtraction in finite fields are performed component-wise and follow the rules of the underlying field
- In prime fields, addition and subtraction are performed modulo the prime p
- Example: In $\mathbb{F}_7$, $5 + 3 = 1$ (since $8 \equiv 1 \pmod{7}$)
- In binary fields, addition and subtraction are equivalent to bitwise XOR operation
- Example: In $\mathbb{F}_{2^3}$, $(x^2 + 1) + (x + 1) = x^2 + x$

Multiplication and squaring

- Multiplication in finite fields is performed by multiplying the elements as polynomials and then reducing the result modulo the field's irreducible polynomial
- In prime fields, multiplication is performed modulo the prime p
- Example: In $\mathbb{F}_7$, $3 \times 5 = 1$ (since $15 \equiv 1 \pmod{7}$)
- In binary fields, multiplication is performed using polynomial multiplication and then reduced modulo the irreducible polynomial
- Example: In $\mathbb{F}_{2^3}$ with irreducible polynomial $x^3 + x + 1$,
 $(x^2 + 1) \times (x + 1) = x^3 + x^2 + x + 1 \equiv x^2 + x \pmod{x^3 + x + 1}$
- Squaring is a special case of multiplication where an element is multiplied by itself, often optimized for efficiency

Division and inversion

- Division in finite fields is performed by multiplying the dividend by the multiplicative inverse of the divisor
- Multiplicative inverse of an element a is an element a^{-1} such that $a \times a^{-1} = 1$
- In prime fields, the multiplicative inverse can be computed using the extended Euclidean algorithm or Fermat's little theorem
- Example: In $\mathbb{F}_7$, the multiplicative inverse of 3 is 5 (since $3 \times 5 \equiv 1 \pmod{7}$)
- In binary fields, the multiplicative inverse can be computed using the extended Euclidean algorithm or by solving a system of linear equations

Exponentiation and logarithms

- Exponentiation in finite fields is the process of raising an element to a power, often used in cryptographic operations like key exchange and digital signatures
- Logarithms in finite fields are the inverse operation of exponentiation, used to solve equations of the form $a^x = b$ for x
- Efficient algorithms for exponentiation include binary exponentiation and sliding window methods
- Logarithms in finite fields can be computed using algorithms like the Pohlig-Hellman algorithm or the index calculus method

Efficient implementation techniques

- Implementing finite field arithmetic efficiently is crucial for the performance of elliptic curve cryptographic schemes
- Various techniques and optimizations can be applied to improve the speed and memory usage of finite field operations

Polynomial basis vs normal basis

- Elements in binary fields can be represented using different bases, such as polynomial basis or normal basis
- Polynomial basis represents elements as polynomials with coefficients in $\mathbb{F}_2$, and arithmetic is performed modulo an irreducible polynomial
- Example: In $\mathbb{F}_{2^3}$ with irreducible polynomial $x^3 + x + 1$, the element $x^2 + 1$ is represented as the bitstring 101
- Normal basis represents elements using a special type of basis where the basis elements are obtained by applying the Frobenius automorphism to a single element
- Allows for efficient computation of squaring and exponentiation operations
- The choice of basis depends on the specific requirements of the implementation, such as performance, memory constraints, and available hardware instructions

Look-up tables and precomputation

- Look-up tables can be used to store precomputed values for frequently used operations, such as multiplication or exponentiation
- Precomputing and storing these values in tables can significantly reduce the computation time at the cost of increased memory usage
- Example: Precomputing a multiplication table for all pairs of elements in a small field can replace runtime multiplications with simple table lookups

Karatsuba and Toom-Cook multiplication

- Karatsuba and Toom-Cook are algorithms for fast polynomial multiplication, which can be adapted for finite field multiplication

- These algorithms reduce the number of component multiplications by splitting the polynomials into smaller parts and combining the results
- Karatsuba multiplication has a complexity of $O(n^{\log_2 3})$ compared to the standard $O(n^2)$ algorithm
- Toom-Cook multiplication generalizes Karatsuba and achieves even lower complexity for larger polynomials

Montgomery reduction for modular arithmetic

- Montgomery reduction is a technique for efficient modular arithmetic, particularly useful for prime fields
- It allows for fast modular reduction without the need for expensive division operations
- Montgomery reduction transforms the operands into a special representation called Montgomery form, performs the arithmetic operations, and then converts the result back
- This technique is widely used in implementations of elliptic curve cryptography over prime fields to improve the performance of modular multiplication and exponentiation

Applications in elliptic curve cryptography

- Finite fields are the foundation for elliptic curve cryptography (ECC), which is widely used in various security protocols and applications
- ECC relies on the arithmetic operations and properties of elliptic curves defined over finite fields to provide secure and efficient cryptographic primitives

Scalar multiplication on elliptic curves

- Scalar multiplication is the fundamental operation in ECC, which involves computing the multiple of a point on an elliptic curve by a scalar value
- Given a point P on an elliptic curve and a scalar k , scalar multiplication calculates the point $Q = kP$
- Efficient scalar multiplication is crucial for the performance of ECC schemes and is achieved through various algorithms like double-and-add, sliding window, and Montgomery ladder

Point compression and decompression

- Point compression is a technique used to reduce the size of elliptic curve points for storage or transmission
- It involves representing a point using a single coordinate (usually the x-coordinate) along with a single bit to indicate the sign of the other coordinate
- Point decompression is the process of recovering the original point from its compressed representation
- Efficient point compression and decompression algorithms are important for reducing bandwidth and storage requirements in ECC implementations

Finite field isomorphisms for curve representations

- Elliptic curves can be represented using different coordinate systems, such as affine coordinates, projective coordinates, or Jacobian coordinates

- Finite field isomorphisms allow for converting between different curve representations while preserving the underlying field structure
- These isomorphisms are used to optimize elliptic curve arithmetic by selecting the most efficient representation for a given operation
- Example: Converting between affine and projective coordinates can eliminate the need for expensive inversion operations in point addition and doubling

Optimized arithmetic for specific curve parameters

- Certain elliptic curves have special parameters that enable optimized arithmetic operations
- For example, curves with a special form like Montgomery curves or Edwards curves allow for faster point addition and doubling formulas
- Curve25519 is a popular Montgomery curve that provides efficient and secure scalar multiplication
- Other curves like FourQ and Curve448 are designed to optimize various aspects of elliptic curve arithmetic, such as reducing the number of field operations or minimizing the impact of side-channel attacks

Security considerations

- Implementing elliptic curve cryptography securely requires careful consideration of various security aspects, such as side-channel attacks, constant-time execution, and fault injection countermeasures
- Proper implementation techniques and best practices are essential to ensure the robustness and security of ECC-based systems

Resistance to side-channel attacks

- Side-channel attacks exploit information leakage from the physical implementation of cryptographic algorithms, such as timing, power consumption, or electromagnetic emanations
- Protecting against side-channel attacks is crucial to prevent unauthorized access to sensitive data or private keys
- Techniques like constant-time execution, regular scalar multiplication algorithms (e.g., Montgomery ladder), and randomization of intermediate values help mitigate the risk of side-channel attacks

Constant-time implementations

- Constant-time implementations ensure that the execution time of cryptographic operations is independent of the input data and private keys
- This prevents timing-based side-channel attacks, where an attacker could deduce sensitive information by measuring the execution time of specific operations
- Constant-time techniques include using conditional move instructions, avoiding data-dependent branches, and ensuring that all operations take the same amount of time regardless of the input

Randomized projective coordinates

- Randomized projective coordinates are a countermeasure against certain types of side-channel attacks, such as differential power analysis (DPA)

- By randomizing the projective representation of elliptic curve points during scalar multiplication, an attacker's ability to correlate power traces with specific point operations is significantly reduced
- This technique involves multiplying the projective coordinates by a random non-zero field element before performing the scalar multiplication and then dividing the result by the same random element to obtain the final point

Fault injection countermeasures

- Fault injection attacks involve physically manipulating the device or its environment to induce errors during the execution of cryptographic operations
- These errors can be exploited to recover secret information or bypass security checks
- Countermeasures against fault injection attacks include using error detection and correction codes, implementing redundant computations, and performing consistency checks on intermediate and final results
- Example: Verifying the correctness of scalar multiplication by checking that the resulting point still lies on the elliptic curve can detect certain types of fault injection attacks

2.2 Elliptic curves over prime fields

Elliptic curves over prime fields form the foundation of modern cryptography. These smooth algebraic curves, defined by cubic equations, create finite abelian groups that enable secure and efficient computations. Their unique properties make them ideal for encryption, digital signatures, and key exchange protocols.

The Weierstrass equation $y^2 = x^3 + ax + b$ defines elliptic curves over prime fields. Points on these curves form a group under addition, with the point at infinity as the identity element. This group structure, combined with the hardness of the discrete logarithm problem, underpins elliptic curve cryptography's strength.

Definition of elliptic curves

- Elliptic curves are smooth, projective, algebraic curves of genus one, which means they have no cusps or self-intersections
- They are defined by a cubic equation in two variables, often referred to as the Weierstrass equation
- Elliptic curves have a rich algebraic structure, forming an abelian group under a geometric addition operation

Weierstrass equation

- The Weierstrass equation for an elliptic curve over a field K is given by $y^2 = x^3 + ax + b$, where $a, b \in K$ and the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$
- The coefficients a and b determine the shape and properties of the elliptic curve
- The Weierstrass equation can be transformed into other forms, such as the Montgomery or Edwards form, which may have computational advantages

Elliptic curves vs elliptic functions

- Elliptic curves are not the same as elliptic functions, despite the similar name

- Elliptic functions are doubly periodic meromorphic functions on the complex plane, while elliptic curves are algebraic curves defined by a cubic equation
- The name "elliptic" comes from the historical connection between elliptic functions and the arc length of ellipses

Discriminant and singularities

- The discriminant $\Delta = 4a^3 + 27b^2$ of an elliptic curve $y^2 = x^3 + ax + b$ determines whether the curve is smooth or singular
- If $\Delta \neq 0$, the curve is smooth and has no cusps or self-intersections
- If $\Delta = 0$, the curve is singular and has a node (self-intersection) or a cusp, depending on the multiplicity of the roots of the cubic equation

Affine vs projective coordinates

- Elliptic curves can be represented using affine coordinates (x, y) or projective coordinates $(X : Y : Z)$
- Affine coordinates are the standard Cartesian coordinates on the plane, while projective coordinates introduce a third variable Z to represent points at infinity
- Projective coordinates are often used in elliptic curve arithmetic to avoid special cases and improve efficiency by eliminating the need for field inversions

Elliptic curves over prime fields

- Elliptic curves over prime fields are a fundamental building block for elliptic curve cryptography and other applications
- Prime fields provide a finite set of elements with a well-defined arithmetic structure, which is essential for secure and efficient computations

Prime fields and finite fields

- A prime field $\mathbb{F}_p$ is a finite field with p elements, where p is a prime number
- Elements of a prime field are represented by integers modulo p , with addition and multiplication performed modulo p
- Finite fields, also known as Galois fields, are fields with a finite number of elements and include prime fields as a special case

Elliptic curve equations over prime fields

- An elliptic curve over a prime field $\mathbb{F}_p$ is defined by the Weierstrass equation $y^2 \equiv x^3 + ax + b \pmod{p}$, where $a, b \in \mathbb{F}_p$ and $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$
- The coefficients a and b are chosen to ensure that the curve is smooth and has no singularities over the prime field
- Elliptic curves over prime fields have a finite number of points, which form a group under the addition operation

Point representation and uniqueness

- Points on an elliptic curve over a prime field are represented by pairs (x, y) , where $x, y \in \mathbb{F}_p$ satisfy the curve equation
- Each x -coordinate corresponds to at most two y -coordinates, which are negatives of each other modulo p
- The point at infinity, denoted by $\mathcal{O}$, is a special point that serves as the identity element for the group of points on the curve

Point at infinity

- The point at infinity $\mathcal{O}$ is a unique point that does not have a finite x or y coordinate
- It is the neutral element for the group of points on the elliptic curve, meaning that $P + \mathcal{O} = P$ for any point P on the curve
- In projective coordinates, the point at infinity is represented as $(0 : 1 : 0)$, while in affine coordinates, it is often treated as a special case

Group law for elliptic curves

- The set of points on an elliptic curve, together with the point at infinity, forms an abelian group under a geometric addition operation
- The group law defines how to add two points on the curve to obtain a third point, also on the curve
- The group structure of elliptic curves is the foundation for their use in cryptography and other applications

Geometric definition of addition

- The addition of two points P and Q on an elliptic curve is defined geometrically by the following steps:
 1. Draw a line through P and Q (if $P = Q$, draw the tangent line at P)
 2. The line intersects the curve at a third point R'
 3. Reflect R' across the x -axis to obtain the sum $R = P + Q$
- If the line through P and Q is vertical, the sum is defined as the point at infinity $\mathcal{O}$

Algebraic formulas for addition

- The geometric definition of addition can be translated into algebraic formulas for the coordinates of the sum point $R = (x_r, y_r)$
- For two distinct points $P = (x_p, y_p)$ and $Q = (x_q, y_q)$, the sum $R = P + Q$ is given by:
 - $x_r = \lambda^2 - x_p - x_q$
 - $y_r = \lambda(x_p - x_r) - y_p$
- where $\lambda = \frac{y_q - y_p}{x_q - x_p}$
- For doubling a point $P = (x_p, y_p)$, the sum $R = 2P$ is given by:
 - $x_r = \lambda^2 - 2x_p$
 - $y_r = \lambda(x_p - x_r) - y_p$

- where $\lambda = \frac{3x_p^2 + a}{2y_p}$

Associativity and commutativity

- The addition operation on elliptic curves is associative, meaning that $(P + Q) + R = P + (Q + R)$ for any points P , Q , and R on the curve
- Addition is also commutative, so $P + Q = Q + P$ for any points P and Q
- These properties are essential for the group structure of elliptic curves and their use in cryptographic protocols

Identity element and inverses

- The point at infinity $\mathcal{O}$ serves as the identity element for the group of points on an elliptic curve, so $P + \mathcal{O} = P$ for any point P
- Every point $P = (x, y)$ on the curve has an inverse $-P = (x, -y)$, which satisfies $P + (-P) = \mathcal{O}$
- The inverse of the point at infinity is itself, i.e., $-\mathcal{O} = \mathcal{O}$

Graphical examples of addition

- Consider the elliptic curve $y^2 = x^3 - x$ over the real numbers. To add the points $P = (0, 0)$ and $Q = (1, 0)$: 1. Draw a line through P and Q , which is the X -axis 2. The line intersects the curve at the point $R' = (-1, 0)$ 3. Reflect R' across the X -axis to obtain the sum $R = P + Q = (-1, 0)$
- To double the point $P = (0, 0)$: 1. Draw the tangent line to the curve at P , which is the Y -axis 2. The tangent line intersects the curve at the point at infinity $\mathcal{O}$ 3. Therefore, $2P = P + P = \mathcal{O}$

Elliptic curve arithmetic

- Elliptic curve arithmetic involves computing scalar multiples of points on the curve, which is the foundation for elliptic curve cryptography
- Efficient computation techniques are essential for practical implementations of elliptic curve algorithms

Scalar multiplication and repeated addition

- Scalar multiplication is the operation of adding a point P on an elliptic curve to itself k times, denoted as $kP = P + P + \dots + P$ (k times)
- Scalar multiplication can be computed using repeated addition, but this is inefficient for large values of k
- More efficient methods, such as the double-and-add algorithm or window-based methods, are used in practice

Efficient computation techniques

- The double-and-add algorithm for scalar multiplication works by expressing k in binary and iteratively doubling and adding points:
- For each bit of k (from most to least significant):
- Double the current point

- If the current bit is 1, add P to the current point
- Window-based methods, such as the sliding window or w-NAF method, precompute multiples of P and use them to reduce the number of additions required
- Other techniques, such as projective coordinates, Jacobian coordinates, or Montgomery ladders, can be used to improve efficiency by reducing the number of field inversions or providing resistance against side-channel attacks

Doubling vs addition formulas

- Point doubling (computing $2P$) and point addition (computing $P + Q$) have different algebraic formulas and costs
- Doubling formulas are typically more efficient than addition formulas, as they require fewer field operations
- For example, in affine coordinates, point doubling requires 1 field inversion and 4 field multiplications, while point addition requires 1 field inversion and 3 field multiplications
- Using projective or Jacobian coordinates can eliminate the need for field inversions, which are often the most expensive operations

Finite cyclic subgroups

- The group of points on an elliptic curve over a finite field is a finite abelian group, which means it can be decomposed into a product of cyclic subgroups
- A cyclic subgroup is a subgroup generated by a single element, i.e., all elements in the subgroup are scalar multiples of a generator point
- The order of a cyclic subgroup is the number of elements in the subgroup, which is also the order of its generator point
- Lagrange's theorem states that the order of a subgroup divides the order of the group, so the order of any point on the curve divides the total number of points on the curve

Torsion points and orders

- A torsion point on an elliptic curve is a point of finite order, i.e., a point P for which there exists an integer k such that $kP = \mathcal{O}$
- The order of a torsion point is the smallest positive integer k satisfying $kP = \mathcal{O}$
- The set of all torsion points on an elliptic curve forms a subgroup called the torsion subgroup
- For elliptic curves over prime fields, the torsion subgroup is either cyclic or the product of two cyclic subgroups
- The possible torsion subgroup structures for elliptic curves over prime fields are: $\mathbb{Z}_n$ for $n = 1, 2, \dots, 10, 12$, or $\mathbb{Z}_2 \times \mathbb{Z}_{2n}$ for $n = 1, 2, 3, 4$

Elliptic curve discrete logarithm problem

- The security of elliptic curve cryptography relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP)

- The ECDLP is the problem of finding an integer k given points P and Q on an elliptic curve such that $Q = kP$
- The hardness of the ECDLP is essential for the security of elliptic curve-based cryptographic protocols

Hardness assumption and security

- The hardness of the ECDLP is assumed to be exponential in the size of the underlying field, meaning that the best known algorithms for solving the ECDLP have a running time that is exponential in the number of bits required to represent field elements
- This is in contrast to the integer factorization problem and the discrete logarithm problem in finite fields, which have subexponential-time algorithms (e.g., the number field sieve)
- As a result, elliptic curve cryptography can achieve the same level of security as RSA or finite field-based systems with much smaller key sizes

Comparison to integer factorization

- The security of RSA, the most widely used public-key cryptosystem, relies on the difficulty of factoring large integers
- While the best known algorithms for integer factorization, such as the number field sieve, have a subexponential running time, they are still much faster than the best known algorithms for solving the ECDLP
- This means that elliptic curve cryptography can use significantly smaller key sizes than RSA while maintaining the same level of security
- For example, a 256-bit elliptic curve key provides a similar level of security as a 3072-bit RSA key

Pollard's rho algorithm for ECDLP

- Pollard's rho algorithm is a probabilistic algorithm for solving the ECDLP that has an expected running time of $O(\sqrt{n})$, where n is the order of the cyclic subgroup generated by the base point
- The algorithm works by generating a pseudorandom sequence of points on the elliptic curve and looking for a collision in the sequence
- When a collision is found, the ECDLP can be solved with high probability by solving a linear equation in the exponents
- Pollard's rho algorithm is the most efficient general-purpose algorithm for solving the ECDLP, but it is still exponential in the size of the underlying field

Attacks and security considerations

- The security of elliptic curve cryptography can be compromised if the elliptic curve parameters are not chosen carefully
- Some classes of elliptic curves, such as supersingular curves or curves with small embedding degrees, are vulnerable to specific attacks and should be avoided
- Side-channel attacks, which exploit information leaked during the execution of elliptic curve algorithms (e.g., timing or power consumption), can also pose a threat to the security of elliptic curve cryptosystems

- To mitigate these risks, it is important to use standardized, well-studied elliptic curves and to implement side-channel countermeasures, such as constant-time algorithms or randomized projective coordinates

Applications of elliptic curves over prime fields

- Elliptic curves over prime fields have numerous applications in cryptography and secure communication protocols
- The security and efficiency of elliptic curve-based systems make them an attractive choice for resource-constrained environments, such as mobile devices or embedded systems

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography approach that uses the algebraic structure of elliptic curves over finite fields
- ECC can be used for encryption, digital signatures, and key exchange protocols
- Some common elliptic curve-based cryptographic algorithms include the Elliptic Curve Digital Signature Algorithm (ECDSA), the Elliptic Curve Integrated Encryption Scheme (ECIES), and the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol

Digital signature algorithms

- The Elliptic Curve Digital Signature Algorithm (ECDSA) is a widely used digital signature scheme based on elliptic curves
- ECDSA is the elliptic curve analogue of the Digital Signature Algorithm (DSA) and is used in numerous cryptographic protocols, such as Transport Layer Security (TLS) and Bitcoin
- The security of ECDSA relies on the difficulty of solving the ECDLP, and it offers smaller signature sizes compared to RSA or DSA for the same level of security

Diffie-Hellman key exchange

- The Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol is an adaptation of the classic Diffie-Hellman protocol that uses elliptic curve arithmetic
- In ECDH, two parties agree on a common elliptic curve and a base point P . Each party generates a private key k and comp

2.3 Elliptic curves over binary fields

Elliptic curves over binary fields are a fascinating area of study in cryptography. These curves, defined over finite fields of characteristic 2, offer unique properties that make them efficient and secure for various cryptographic applications.

The arithmetic of these curves involves point addition, doubling, and scalar multiplication. Understanding these operations and their efficient implementation is crucial for developing robust elliptic curve cryptosystems in binary fields.

Definitions of elliptic curves over binary fields

- Elliptic curves over binary fields are a type of algebraic curve defined over a finite field of characteristic 2

- Binary fields, denoted as $\mathbb{F}_2^m$, consist of elements that can be represented using m bits
- Elliptic curves over binary fields have unique properties and are widely used in cryptography due to their efficiency and security

Weierstrass equation for binary fields

- The Weierstrass equation for an elliptic curve over a binary field $\mathbb{F}_2^m$ is given by $y^2 + xy = x^3 + ax^2 + b$, where $a, b \in \mathbb{F}_2^m$
- The coefficients a and b must satisfy the condition $b \neq 0$ to ensure the curve is non-singular
- Example: $y^2 + xy = x^3 + x + 1$ over $\mathbb{F}_2^3$

Group law for elliptic curves over binary fields

- The set of points on an elliptic curve over a binary field, together with a special point called the "point at infinity" denoted as $\mathcal{O}$, form an abelian group under the group law
- The group law defines the rules for point addition and point doubling on the curve
- The group law for binary fields differs from that of prime fields due to the characteristic of the field
- Example: Given points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, the sum $P + Q = (x_3, y_3)$ is computed using specific formulas involving the coordinates of P and Q

Arithmetic of elliptic curves over binary fields

- Arithmetic operations on elliptic curves over binary fields involve point addition, point doubling, and scalar multiplication
- These operations form the basis for various cryptographic protocols and algorithms based on elliptic curves
- Efficient implementation of these operations is crucial for the performance and security of elliptic curve cryptosystems

Point addition in binary fields

- Point addition is the operation of adding two distinct points on an elliptic curve over a binary field to obtain a third point on the curve
- The formula for point addition in binary fields is derived from the group law and involves the coordinates of the two points being added
- Example: Given points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, the sum $P + Q = (x_3, y_3)$ is computed as:
- $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a$
- $y_3 = \lambda(x_1 + x_3) + x_3 + y_1$
- where $\lambda = \frac{y_1 + y_2}{x_1 + x_2}$

Point doubling in binary fields

- Point doubling is the operation of adding a point to itself on an elliptic curve over a binary field

- The formula for point doubling in binary fields is a special case of the point addition formula when the two points being added are the same
- Example: Given a point $P = (x_1, y_1)$, the doubled point $2P = (x_3, y_3)$ is computed as:
- $x_3 = \lambda^2 + \lambda + a$
- $y_3 = x_1^2 + (\lambda + 1)x_3$
- where $\lambda = x_1 + \frac{y_1}{x_1}$

Scalar multiplication in binary fields

- Scalar multiplication is the operation of multiplying a point on an elliptic curve over a binary field by a scalar (an integer)
- Scalar multiplication is computed using a combination of point addition and point doubling operations
- Efficient algorithms for scalar multiplication, such as the double-and-add method or the Montgomery ladder, are used to optimize the computation
- Example: Given a point P and a scalar k , the scalar multiple kP is computed by repeatedly applying point addition and point doubling operations based on the binary representation of k

Isomorphism classes of elliptic curves over binary fields

- Isomorphism classes of elliptic curves over binary fields classify curves that have the same structure and properties
- Two elliptic curves over a binary field are isomorphic if there exists an isomorphism (a bijective rational map) between them that preserves the group structure
- Understanding isomorphism classes is important for identifying equivalent curves and optimizing implementations

j-invariant for binary fields

- The j-invariant is a numerical invariant associated with an elliptic curve that characterizes its isomorphism class
- For an elliptic curve over a binary field with Weierstrass equation $y^2 + xy = x^3 + ax^2 + b$, the j-invariant is given by $j = \frac{a^{12}}{b^2}$
- Curves with the same j-invariant are isomorphic, while curves with different j-invariants belong to different isomorphism classes
- Example: The curves $y^2 + xy = x^3 + x + 1$ and $y^2 + xy = x^3 + x$ over $\mathbb{F}_{2^3}$ have the same j-invariant and are isomorphic

Isomorphisms of elliptic curves over binary fields

- An isomorphism between two elliptic curves over a binary field is a bijective rational map that preserves the group structure
- Isomorphisms can be used to map points from one curve to another, enabling the transfer of cryptographic protocols between isomorphic curves

- Example: The map $(x, y) \mapsto (u^2x + r, u^3y + u^2sx + t)$, where $u \neq 0, r, s, t$ are constants in the binary field, defines an isomorphism between elliptic curves over binary fields

Discrete logarithm problem in binary fields

- The discrete logarithm problem (DLP) in binary fields is the problem of finding an integer k such that $Q = kP$, given points P and Q on an elliptic curve over a binary field
- The DLP forms the basis for the security of many elliptic curve cryptosystems, as it is believed to be computationally infeasible for properly chosen curves and field sizes
- Algorithms for solving the DLP in binary fields, such as Pollard's rho and Baby-step giant-step, have a time complexity that is exponential in the field size

Pollard's rho algorithm for binary fields

- Pollard's rho algorithm is a probabilistic algorithm for solving the DLP on elliptic curves over binary fields
- The algorithm uses a pseudo-random walk on the elliptic curve to find a collision, which leads to the solution of the DLP
- Pollard's rho has an expected running time of $O(\sqrt{\pi n/2})$, where n is the order of the base point P
- Example: Given points P and Q on an elliptic curve over a binary field, Pollard's rho algorithm finds an integer k such that $Q = kP$ by iteratively computing pseudo-random points until a collision occurs

Baby-step giant-step algorithm for binary fields

- The Baby-step giant-step (BSGS) algorithm is a deterministic algorithm for solving the DLP on elliptic curves over binary fields
- BSGS is based on the idea of precomputing a table of baby steps (jP) and then taking giant steps ($Q - iP$) to find a collision
- The algorithm has a time complexity of $O(\sqrt{n})$ and a space complexity of $O(\sqrt{n})$, where n is the order of the base point P
- Example: Given points P and Q on an elliptic curve over a binary field, BSGS computes a table of baby steps (jP) for $j = 0, 1, \dots, \lfloor \sqrt{n} \rfloor$ and then searches for a collision with giant steps ($Q - iP$) for $i = 0, 1, \dots, \lfloor \sqrt{n} \rfloor$

Supersingular vs ordinary curves

- Elliptic curves over binary fields can be classified as either supersingular or ordinary based on their group structure and security properties
- The distinction between supersingular and ordinary curves is important for selecting appropriate curves for cryptographic applications
- Supersingular curves have additional structure that can be exploited for certain cryptographic protocols, but they also have potential vulnerabilities

Definitions of supersingular and ordinary curves

- An elliptic curve over a binary field $\mathbb{F}_2^m$ is called supersingular if the trace of the Frobenius endomorphism is divisible by the characteristic of the field (2)
- Equivalently, an elliptic curve over $\mathbb{F}_2^m$ is supersingular if the number of points on the curve is equal to $2^m + 1$
- Curves that are not supersingular are called ordinary curves
- Example: The curve $y^2 + y = x^3 + x$ over $\mathbb{F}_2^m$ is supersingular for all m , while the curve $y^2 + xy = x^3 + x + 1$ over $\mathbb{F}_2^m$ is ordinary for all m

Differences in group structure and security

- Supersingular curves have a unique group structure that allows for efficient pairings, such as the Weil pairing and Tate pairing
- The additional structure of supersingular curves can be exploited for certain cryptographic protocols, such as identity-based encryption and short signatures
- However, supersingular curves are more susceptible to certain attacks, such as the MOV attack, which reduces the DLP on the curve to the DLP in an extension field
- Ordinary curves, on the other hand, do not have the same additional structure and are generally considered more secure for traditional elliptic curve cryptography
- Example: The MOV attack can reduce the DLP on a supersingular curve over $\mathbb{F}_2^m$ to the DLP in $\mathbb{F}_2^{6m}$, while ordinary curves are not vulnerable to this attack

Pairings on elliptic curves over binary fields

- Pairings are bilinear maps that take two points on an elliptic curve over a binary field and map them to an element in an extension field
- Pairings, such as the Weil pairing and Tate pairing, have found numerous applications in cryptography, including identity-based encryption, short signatures, and other advanced protocols
- Efficient computation of pairings is crucial for the practicality of pairing-based cryptographic schemes

Weil pairing for binary fields

- The Weil pairing is a bilinear map that takes two points of order r on an elliptic curve over a binary field and maps them to an element in the r -th roots of unity in an extension field
- The Weil pairing is defined using the divisor class group of the elliptic curve and has various properties, such as bilinearity, non-degeneracy, and antisymmetry
- Example: Given points P and Q of order r on an elliptic curve over $\mathbb{F}_2^m$, the Weil pairing $e_r(P, Q)$ maps to an element in the r -th roots of unity in $\mathbb{F}_2^{km}$, where k is the embedding degree

Tate pairing for binary fields

- The Tate pairing is another bilinear map that takes a point of order r and a divisor class on an elliptic curve over a binary field and maps them to an element in an extension field

- The Tate pairing is defined using the evaluation of functions on divisors and has similar properties to the Weil pairing, such as bilinearity and non-degeneracy
- The Tate pairing is generally more efficient to compute than the Weil pairing and is often preferred in practical implementations
- Example: Given a point P of order r and a divisor class D on an elliptic curve over $\mathbb{F}_{2^m}$, the Tate pairing $e_r(P, D)$ maps to an element in $\mathbb{F}_{2^{km}}^* / (\mathbb{F}_{2^{km}}^*)^r$, where k is the embedding degree

Applications of pairings in cryptography

- Pairings on elliptic curves over binary fields have enabled the construction of various advanced cryptographic protocols
- Identity-based encryption (IBE) schemes, such as the Boneh-Franklin IBE, use pairings to allow for public keys to be derived from identities, simplifying key management
- Short signature schemes, like the Boneh-Lynn-Shacham (BLS) signature, use pairings to produce compact signatures that are verifiable using the signer's public key
- Other applications include attribute-based encryption, searchable encryption, and anonymous credentials
- Example: The Boneh-Franklin IBE scheme uses the Weil pairing on supersingular curves over binary fields to construct an efficient and secure identity-based encryption system

Efficient implementations of elliptic curves over binary fields

- Implementing elliptic curve cryptography over binary fields requires careful consideration of various factors, such as field representation, curve parameters, and optimization techniques
- Efficient implementations are crucial for the performance and practicality of elliptic curve cryptosystems in real-world applications
- Hardware and software optimizations can significantly improve the speed and scalability of elliptic curve operations over binary fields

Polynomial basis vs normal basis representations

- Binary fields can be represented using either a polynomial basis or a normal basis
- In a polynomial basis representation, field elements are represented as polynomials over $\mathbb{F}_2$ modulo an irreducible polynomial of degree m
- In a normal basis representation, field elements are represented as linear combinations of a normal element and its conjugates
- The choice of basis representation can impact the efficiency of field arithmetic and elliptic curve operations
- Example: The polynomial basis representation of $\mathbb{F}_{2^3}$ using the irreducible polynomial $x^3 + x + 1$ represents field elements as polynomials $a_0 + a_1x + a_2x^2$, while a normal basis representation uses a normal element β and its conjugates β^2 and β^4

Optimizations for point arithmetic in binary fields

- Point arithmetic on elliptic curves over binary fields can be optimized using various techniques to improve performance
- Projective coordinates, such as Lopez-Dahab coordinates, can be used to avoid expensive field inversions by representing points using additional coordinates
- Efficient formulas for point addition and point doubling can be derived by exploiting the properties of binary fields and the curve equation
- Precomputation techniques, like window methods or comb methods, can be used to speed up scalar multiplication by precomputing and storing multiples of the base point
- Example: Using Lopez-Dahab projective coordinates, point addition on an elliptic curve over a binary field can be performed using only field multiplications and squarings, avoiding the need for field inversions

Hardware implementations for binary fields

- Hardware implementations of elliptic curve cryptography over binary fields can provide significant performance improvements compared to software implementations
- Field arithmetic operations, such as multiplication and squaring, can be efficiently implemented using dedicated hardware circuits
- Specialized hardware architectures, like systolic arrays or finite field processors, can be used to parallelize and accelerate elliptic curve operations
- Hardware implementations can also benefit from side-channel countermeasures to protect against attacks like power analysis or timing attacks
- Example: A hardware implementation of elliptic curve scalar multiplication over $\mathbb{F}_{2^{163}}$ using a systolic array architecture can achieve a throughput of several million point multiplications per second while occupying a small area on an FPGA or ASIC

2.4 Elliptic curve point counting

Elliptic curve point counting is a crucial aspect of elliptic curve theory. It involves determining the number of points on a curve over a finite field, which is essential for applications in cryptography and primality testing.

Efficient algorithms like Schoof's, SEA, and Satoh's have been developed to tackle this problem. These methods use various mathematical techniques to compute the number of points, improving on the naive exhaustive search approach.

Importance of point counting

- Point counting is a fundamental problem in elliptic curve theory that involves determining the number of points on an elliptic curve over a finite field
- Efficient point counting algorithms are crucial for various applications of elliptic curves, such as cryptography and primality testing
- The security of elliptic curve cryptosystems relies on the difficulty of solving the discrete logarithm problem on the curve, which is closely related to the number of points on the curve

Hasse's theorem

Hasse interval

- Hasse's theorem states that the number of points on an elliptic curve over a finite field $\mathbb{F}_q$ lies within the interval $[q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$
- This interval, known as the Hasse interval, provides a useful bound on the number of points on the curve
- The Hasse interval narrows down the search space for point counting algorithms, making them more efficient

Hasse bound proof

- The proof of Hasse's theorem relies on the Riemann hypothesis for function fields, which is a deep result in algebraic geometry
- The proof involves studying the zeta function of the elliptic curve and its relation to the number of points on the curve
- The Hasse bound can be derived from the Weil conjectures, which generalize the Riemann hypothesis to varieties over finite fields

Naive point counting

Exhaustive search method

- The naive approach to point counting involves iterating through all the elements of the finite field $\mathbb{F}_q$ and testing whether each point satisfies the elliptic curve equation
- For each element $x \in \mathbb{F}_q$, compute $y^2 = x^3 + ax + b$ and check if y is a square in $\mathbb{F}_q$
- If y is a square, then $(x, \sqrt{y})$ and $(x, -\sqrt{y})$ are points on the curve; otherwise, x does not correspond to a point on the curve

Limitations of naive approach

- The exhaustive search method has a time complexity of $O(q)$, which becomes impractical for large finite fields
- As the size of the finite field grows, the naive approach becomes computationally infeasible
- More sophisticated point counting algorithms are needed to efficiently determine the number of points on elliptic curves over large finite fields

Schoof's algorithm

Characteristic equation

- Schoof's algorithm relies on the characteristic equation of the Frobenius endomorphism ϕ of the elliptic curve
- The characteristic equation is given by $\phi^2 - t\phi + q = 0$, where t is the trace of Frobenius and q is the order of the finite field

- The number of points on the elliptic curve is related to the trace of Frobenius by the equation $\#E(F_q) = q + 1 - t$

Division polynomials

- Division polynomials are a sequence of polynomials ψ_n associated with an elliptic curve that describe the n -torsion points on the curve
- Schoof's algorithm uses division polynomials to compute the trace of Frobenius modulo various small primes ℓ
- By computing $t \bmod \ell$ for sufficiently many primes ℓ , the trace of Frobenius can be recovered using the Chinese Remainder Theorem

Computation of trace

- For each prime ℓ , Schoof's algorithm computes the action of the Frobenius endomorphism on the ℓ -torsion points of the elliptic curve
- This action is described by a polynomial equation involving the division polynomials ψ_ℓ and $\phi(\psi_\ell)$
- By solving this equation, the algorithm determines the value of $t \bmod \ell$, which is a piece of information about the trace of Frobenius

Polynomial time complexity

- Schoof's algorithm has a polynomial time complexity of $O(\log^8 q)$, which is a significant improvement over the naive approach
- The algorithm's efficiency comes from the fact that it works with polynomials of degree $O(\ell^2)$ and performs computations modulo small primes ℓ
- While polynomial time, the exponent of 8 in the complexity makes Schoof's algorithm still relatively slow for large finite fields

Schoof-Elkies-Atkin (SEA) algorithm

Improvements over Schoof's algorithm

- The SEA algorithm builds upon Schoof's algorithm by incorporating ideas from Elkies and Atkin to improve its efficiency
- SEA uses a combination of Elkies primes and Atkin primes to reduce the degree of the polynomials involved in the computations
- By working with lower-degree polynomials, SEA achieves a better time complexity compared to Schoof's original algorithm

Elkies primes

- Elkies primes are primes ℓ for which the characteristic equation $\phi^2 - t\phi + q = 0$ splits over the ℓ -torsion points
- For Elkies primes, the action of Frobenius on the ℓ -torsion points can be described by a lower-degree polynomial (degree $O(\ell)$)

- SEA uses modular polynomials to efficiently identify Elkies primes and compute the corresponding eigenvalues of Frobenius

Atkin primes

- Atkin primes are primes ℓ for which the characteristic equation $\phi^2 - t\phi + q = 0$ does not split over the ℓ -torsion points
- For Atkin primes, SEA uses a different approach based on modular equations to extract information about the trace of Frobenius
- While Atkin primes do not provide as much information as Elkies primes, they still contribute to the overall efficiency of the SEA algorithm

SEA vs Schoof comparison

- SEA improves upon Schoof's algorithm by reducing the degree of the polynomials involved in the computations
- The use of Elkies and Atkin primes in SEA leads to a time complexity of $O(\log^6 q)$, which is better than Schoof's $O(\log^8 q)$
- In practice, SEA is significantly faster than Schoof's algorithm and has been widely implemented for point counting on elliptic curves

Satoh's algorithm

Canonical lift

- Satoh's algorithm is based on the idea of lifting the elliptic curve and its Frobenius endomorphism to the $\mathcal{P}$ -adic numbers
- The canonical lift of an elliptic curve is a unique lift of the curve to the $\mathcal{P}$ -adic numbers that preserves certain properties
- By working with the canonical lift, Satoh's algorithm can efficiently compute the trace of Frobenius using $\mathcal{P}$ -adic methods

p-adic numbers

- The $\mathcal{P}$ -adic numbers are a completion of the rational numbers with respect to the $\mathcal{P}$ -adic metric, which measures the divisibility of integers by powers of $\mathcal{P}$
- In Satoh's algorithm, the elliptic curve and its Frobenius endomorphism are lifted to the $\mathcal{P}$ -adic numbers, allowing for efficient computations
- The $\mathcal{P}$ -adic arithmetic used in Satoh's algorithm is well-suited for computer implementations and can be performed with high precision

Advantages of Satoh's method

- Satoh's algorithm has a time complexity of $O(\log^3 q)$, which is a significant improvement over Schoof's algorithm and SEA
- The use of $\mathcal{P}$ -adic numbers in Satoh's algorithm allows for efficient computations and eliminates the need for working with high-degree polynomials

- Satoh's method is particularly well-suited for point counting on elliptic curves over fields of small characteristic, where other algorithms may be less efficient

Kedlaya's algorithm

Monsky-Washnitzer cohomology

- Kedlaya's algorithm is based on the theory of Monsky-Washnitzer cohomology, which is a p -adic cohomology theory for algebraic varieties
- The Monsky-Washnitzer cohomology of an elliptic curve is closely related to its zeta function, which encodes information about the number of points on the curve
- By computing the action of Frobenius on the Monsky-Washnitzer cohomology, Kedlaya's algorithm can efficiently determine the zeta function and the number of points on the curve

Zeta function computation

- The zeta function of an elliptic curve over a finite field is a generating function that encodes information about the number of points on the curve over various field extensions
- Kedlaya's algorithm computes the zeta function of the elliptic curve by working with the Monsky-Washnitzer cohomology and the Frobenius endomorphism
- The coefficients of the zeta function can be efficiently computed using p -adic methods, leading to a fast point counting algorithm

Efficiency of Kedlaya's approach

- Kedlaya's algorithm has a time complexity of $O(\log^3 q)$, which is the same as Satoh's algorithm and better than Schoof's algorithm and SEA
- The use of Monsky-Washnitzer cohomology and p -adic methods in Kedlaya's algorithm allows for efficient computations and avoids the need for working with high-degree polynomials
- Kedlaya's approach is applicable to elliptic curves over fields of any characteristic and has been widely implemented in practice

Hyperelliptic curve point counting

Challenges in higher genus curves

- Hyperelliptic curves are a generalization of elliptic curves to higher genus, with a more complex geometry and arithmetic
- Point counting on hyperelliptic curves is more challenging than on elliptic curves due to the increased complexity of the curve's structure
- The algorithms used for elliptic curve point counting, such as Schoof's algorithm and SEA, do not directly generalize to hyperelliptic curves

Adapted algorithms for hyperelliptic curves

- Several algorithms have been developed to address the challenges of point counting on hyperelliptic curves

- The Schoof-Pila algorithm is an adaptation of Schoof's algorithm to hyperelliptic curves, using the action of Frobenius on the ℓ -torsion points of the Jacobian variety
- The p -adic algorithms, such as Kedlaya's algorithm and its variants, have been extended to work with hyperelliptic curves by considering the Monsky-Washnitzer cohomology of the curve
- These adapted algorithms have time complexities that depend on both the genus of the curve and the size of the finite field, making them less efficient than their elliptic curve counterparts

Applications of point counting

Elliptic curve cryptography

- Point counting is a crucial component in the implementation of elliptic curve cryptography (ECC)
- ECC relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP), which is closely related to the number of points on the curve
- Efficient point counting algorithms are used to determine the security level of an elliptic curve and to generate cryptographically secure curves for use in ECC protocols

Pairing-based cryptography

- Pairing-based cryptography is a subfield of cryptography that uses bilinear pairings on elliptic curves to construct advanced cryptographic primitives
- The efficiency of pairing computations depends on the number of points on the elliptic curve and the embedding degree of the curve
- Point counting algorithms are used to find elliptic curves with suitable properties for pairing-based cryptography, such as curves with a prime or near-prime number of points and a high embedding degree

Elliptic curve primality proving

- Elliptic curve primality proving (ECPP) is a method for proving the primality of large integers using elliptic curves
- ECPP relies on the properties of elliptic curves over finite fields and the number of points on these curves
- Point counting algorithms, such as Schoof's algorithm and SEA, are used in ECPP to determine the number of points on elliptic curves over various finite fields and to construct the primality proof
- ECPP is an important tool in computational number theory and has applications in cryptography and secure communication protocols

2.5 Hasse's theorem and the Hasse interval

Hasse's theorem is a key result in elliptic curve theory, bounding the number of points on curves over finite fields. It introduces the Hasse interval, which provides a range for possible point counts and has important implications for cryptography and number theory.

The theorem connects to broader topics like the Riemann hypothesis for function fields and generalizations to higher-dimensional varieties. It's crucial for understanding elliptic curve structure and forms a foundation for advanced results in algebraic geometry and number theory.

Hasse's theorem

- Fundamental result in the theory of elliptic curves over finite fields
- Provides bounds on the number of points on an elliptic curve over a finite field
- Has important implications for the structure and properties of elliptic curves

Statement of theorem

- Let E be an elliptic curve over a finite field $\mathbb{F}_q$ with q elements
- Denote the number of $\mathbb{F}_q$ -rational points on E by $\#E(\mathbb{F}_q)$
- Hasse's theorem states that $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- In other words, the number of points on E over $\mathbb{F}_q$ is close to $q + 1$

Proof of theorem

- Original proof by Helmut Hasse in 1934 used complex analysis and zeta functions
- Modern proofs rely on the Riemann hypothesis for function fields (proven by André Weil)
- Key steps include relating the number of points to the trace of the Frobenius endomorphism
- The Riemann hypothesis for function fields bounds the magnitude of the trace

The Hasse interval

- Range of possible values for the number of points on an elliptic curve over a finite field
- Follows directly from Hasse's theorem

Definition of interval

- For an elliptic curve E over $\mathbb{F}_q$, the Hasse interval is $[q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$
- Represents the range of values that $\#E(\mathbb{F}_q)$ can take

Bounds of interval

- Lower bound: $q + 1 - 2\sqrt{q}$
- Upper bound: $q + 1 + 2\sqrt{q}$
- The bounds are symmetric around $q + 1$, the expected number of points

Significance of interval

- Shows that the number of points on an elliptic curve is close to the expected value
- Provides a way to estimate the number of points without explicitly counting them
- Has implications for the security of elliptic curve cryptography (curves with extreme point counts may be less secure)

Elliptic curve point counts

- The number of points on an elliptic curve over a finite field
- Closely related to the Hasse interval

Relationship to Hasse interval

- The point count $\#E(\mathbb{F}_q)$ always lies within the Hasse interval $[q + 1 - 2\sqrt{q}, q + 1 + 2\sqrt{q}]$
- Curves with point counts near the bounds of the interval are relatively rare

Computing point counts

- Naïve approach: enumerate all points on the curve and count them (inefficient for large fields)
- More efficient methods include Schoof's algorithm and its improvements (Schoof-Elkies-Atkin algorithm)
- Point counts can also be computed using the zeta function of the elliptic curve

Examples of point counts

- Over $\mathbb{F}_5$, the curve $y^2 = x^3 + x + 1$ has 9 points (including the point at infinity)
- Over $\mathbb{F}_{11}$, the curve $y^2 = x^3 + 2x + 1$ has 12 points
- The Hasse interval for $\mathbb{F}_5$ is $[1, 11]$, and for $\mathbb{F}_{11}$ it is $[5, 19]$

Applications of Hasse's theorem

- Hasse's theorem has significant applications in various areas of mathematics and computer science
- Provides a foundation for understanding the structure and properties of elliptic curves

In cryptography

- Elliptic curve cryptography (ECC) relies on the difficulty of the elliptic curve discrete logarithm problem
- Hasse's theorem ensures that the number of points on the curve is not too small (which would make the system insecure)
- Also used to estimate the computational complexity of attacks on ECC systems

In computational number theory

- Hasse's theorem is used in the study of elliptic curves over finite fields and their zeta functions
- Plays a role in the computation of point counts and the construction of elliptic curve factoring algorithms (Lenstra's ECM)

Other areas of mathematics

- Appears in the study of modular forms and L-functions (via the modularity theorem for elliptic curves)
- Used in the proof of the Weil conjectures for elliptic curves over finite fields
- Has connections to the theory of complex multiplication and the classification of elliptic curves

Generalizations of Hasse's theorem

- Hasse's theorem has been generalized to various other settings beyond elliptic curves over finite fields
- These generalizations extend the ideas and techniques used in the original theorem

To higher genus curves

- Weil's theorem for curves of higher genus over finite fields
- Bounds the number of points in terms of the genus and the field size
- Involves the Riemann hypothesis for function fields and the Weil conjectures

To abelian varieties

- Abelian varieties are higher-dimensional analogues of elliptic curves
- Tate's theorem bounds the number of points on an abelian variety over a finite field
- Relies on the Weil conjectures and the theory of endomorphisms of abelian varieties

Other generalizations

- Hasse-Weil bound for algebraic varieties over finite fields
- Deligne's proof of the Weil conjectures for general algebraic varieties
- Generalizations to curves and varieties over local fields and number fields

Related theorems and results

- Several other theorems and conjectures are closely related to Hasse's theorem
- These results provide additional insights into the structure and properties of elliptic curves

Deuring's theorem

- Characterizes the endomorphism rings of elliptic curves over finite fields
- Relates the endomorphism ring to the trace of the Frobenius endomorphism
- Has applications in the theory of complex multiplication and the construction of elliptic curves with prescribed properties

Waterhouse's theorem

- Classifies the possible group structures of elliptic curves over finite fields
- Shows that every group of order within the Hasse interval occurs as the group of points on some elliptic curve
- Provides a way to construct elliptic curves with desired group structures

Sato-Tate conjecture

- Describes the distribution of the traces of Frobenius for an elliptic curve over varying finite fields

- Predicts that the traces follow a specific probability distribution related to random matrices
- Has been proven for certain classes of elliptic curves (e.g., those with complex multiplication)
- Connects the arithmetic of elliptic curves to deep questions in analytic number theory and harmonic analysis

2.6 Supersingular and ordinary elliptic curves

Supersingular and ordinary elliptic curves are two distinct classes of curves with unique properties. These curves play crucial roles in elliptic curve cryptography, offering different advantages and challenges in various applications.

Understanding the differences between supersingular and ordinary curves is essential for cryptographers and mathematicians. From endomorphism rings to isogeny structures, these curves exhibit distinct characteristics that impact their use in cryptographic protocols and mathematical research.

Supersingular elliptic curves

- Supersingular elliptic curves are a special class of elliptic curves with unique properties that distinguish them from ordinary elliptic curves
- These curves have been extensively studied in the context of elliptic curve cryptography due to their efficient computation and potential for constructing secure cryptographic protocols
- Understanding the characteristics and behavior of supersingular curves is crucial for their effective application in cryptography and other areas of mathematics

Definition of supersingularity

- An elliptic curve E defined over a field k of characteristic $p > 0$ is called supersingular if its p -torsion subgroup $E[p]$ is trivial
- In other words, the group of points on E annihilated by multiplication by p is the identity element $\mathcal{O}$
- Equivalently, E is supersingular if and only if the trace of the Frobenius endomorphism of E is divisible by p
- Supersingularity is a geometric property of the curve that remains invariant under isomorphisms and extensions of the base field
- Examples of supersingular curves include the curve $y^2 = x^3 - x$ over $\mathbb{F}_p$ for $p \equiv 3 \pmod{4}$ and the curve $y^2 = x^3 + 1$ over $\mathbb{F}_p$ for $p \equiv 2 \pmod{3}$

Endomorphism rings of supersingular curves

- The endomorphism ring of a supersingular elliptic curve E over a field k is a maximal order in a quaternion algebra over $\mathbb{Q}$
- A quaternion algebra is a 4-dimensional central simple algebra over a field
- The endomorphism ring of a supersingular curve is always non-commutative, unlike the case for ordinary curves
- The structure of the endomorphism ring provides insights into the isogeny graph of the curve and its potential for use in cryptography

- For example, the supersingular curve $y^2 = x^3 + x$ over $\mathbb{F}_p$ for $p \equiv 3 \pmod{4}$ has endomorphism ring isomorphic to the maximal order in the quaternion algebra ramified at p and ∞

Isogenies of supersingular curves

- Isogenies are rational maps between elliptic curves that preserve the group structure
- For supersingular curves, the isogeny graph is a regular graph with a high degree of connectivity
- This property is exploited in the construction of cryptographic protocols based on supersingular isogeny problems
- The isogeny graph of supersingular curves over a finite field $\mathbb{F}_{p^2}$ is known as the supersingular isogeny graph
- Isogenies of supersingular curves can be efficiently computed using Vélú's formulas, which provide explicit expressions for the coefficients of the codomain curve
- Examples of isogenies include the multiplication-by- m map $[m] : E \rightarrow E$ and the Frobenius endomorphism $\pi : E \rightarrow E$ defined by $(x, y) \mapsto (x^p, y^p)$

Supersingular curves over finite fields

- Supersingular curves over finite fields have a rich structure and are of particular interest in cryptography
- Over a finite field $\mathbb{F}_q$ of characteristic p , there are approximately $\lfloor q/12 \rfloor$ isomorphism classes of supersingular curves
- This is significantly fewer than the number of isomorphism classes of ordinary curves over the same field
- The supersingular isogeny graph over $\mathbb{F}_{p^2}$ is a $(p+1)$ -regular graph, meaning that each vertex (curve) has exactly $p+1$ outgoing edges (isogenies)
- Supersingular curves over finite fields have been used to construct cryptographic protocols such as the supersingular isogeny Diffie-Hellman key exchange (SIDH) and the supersingular isogeny hash function (SIH)

Cryptographic applications of supersingular curves

- Supersingular curves have found numerous applications in elliptic curve cryptography due to their unique properties and efficient computation
- The supersingular isogeny Diffie-Hellman key exchange (SIDH) is a post-quantum key exchange protocol based on the difficulty of finding isogenies between supersingular curves
- SIDH offers small key sizes and efficient computation compared to other post-quantum candidates
- The supersingular isogeny hash function (SIH) is a cryptographic hash function that maps binary strings to supersingular elliptic curves
- SIH is collision-resistant and can be used for various cryptographic purposes, such as digital signatures and password hashing
- Supersingular curves have also been used in the construction of zero-knowledge proofs, identity-based encryption schemes, and other cryptographic primitives

- The security of cryptographic protocols based on supersingular curves relies on the hardness of problems such as the supersingular isogeny problem and the supersingular computational Diffie-Hellman problem

Ordinary elliptic curves

- Ordinary elliptic curves are the most common type of elliptic curves and have been extensively studied in mathematics and cryptography
- Unlike supersingular curves, ordinary curves have a commutative endomorphism ring and a different structure of the isogeny graph
- Understanding the properties and behavior of ordinary curves is essential for their effective use in cryptographic applications and theoretical investigations

Definition of ordinary curves

- An elliptic curve E defined over a field k of characteristic $p > 0$ is called ordinary if it is not supersingular
- In other words, the p -torsion subgroup $E[p]$ is not trivial, and the trace of the Frobenius endomorphism is not divisible by p
- Ordinary curves constitute the majority of elliptic curves over a given field, with supersingular curves being a special case
- The property of being ordinary is preserved under isomorphisms and extensions of the base field
- Examples of ordinary curves include the curve $y^2 = x^3 - x$ over $\mathbb{F}_p$ for $p \equiv 1 \pmod{4}$ and the curve $y^2 = x^3 + ax + b$ over $\mathbb{Q}$ for most choices of a and b

Endomorphism rings of ordinary curves

- The endomorphism ring of an ordinary elliptic curve E over a field k is an order in an imaginary quadratic field
- An imaginary quadratic field is a number field of the form $\mathbb{Q}(\sqrt{-d})$, where d is a positive square-free integer
- The endomorphism ring of an ordinary curve is always commutative, unlike the case for supersingular curves
- The structure of the endomorphism ring provides information about the isogenies and complex multiplication properties of the curve
- For example, the ordinary curve $y^2 = x^3 - x$ over $\mathbb{F}_p$ for $p \equiv 1 \pmod{4}$ has endomorphism ring isomorphic to the ring of integers of the imaginary quadratic field $\mathbb{Q}(\sqrt{-1})$

Isogenies of ordinary curves

- Isogenies between ordinary elliptic curves have a different structure compared to those of supersingular curves
- The isogeny graph of ordinary curves is not as highly connected as the supersingular isogeny graph, and the degrees of isogenies are more restricted

- Computing isogenies between ordinary curves is generally more challenging than in the supersingular case, as the formulas for isogenies are more complex
- Isogenies of ordinary curves have been used in the construction of cryptographic protocols, such as the CRS (Charles-Goren-Lauter) hash function and the CSIDH (Commutative Supersingular Isogeny Diffie-Hellman) key exchange

Ordinary curves over finite fields

- Ordinary curves over finite fields have been widely studied and applied in elliptic curve cryptography
- Over a finite field $\mathbb{F}_q$ of characteristic p , the number of isomorphism classes of ordinary curves is approximately q , which is significantly larger than the number of supersingular curves
- Ordinary curves over finite fields have a rich theory of complex multiplication, which relates the endomorphism ring of the curve to the ring of integers of an imaginary quadratic field
- The complex multiplication method can be used to construct ordinary curves with desired properties, such as a specific group order or endomorphism ring

Cryptographic applications of ordinary curves

- Ordinary elliptic curves have been extensively used in the design and implementation of various cryptographic protocols
- The most common use of ordinary curves is in the Elliptic Curve Diffie-Hellman (ECDH) key exchange and the Elliptic Curve Digital Signature Algorithm (ECDSA)
- These protocols rely on the hardness of the elliptic curve discrete logarithm problem (ECDLP) for their security
- Ordinary curves have also been used in the construction of pairing-based cryptography, where bilinear pairings on elliptic curves are used to build advanced cryptographic primitives
- Examples include the Boneh-Franklin identity-based encryption scheme and the Boneh-Lynn-Shacham (BLS) signature scheme
- The choice of ordinary curves for cryptographic applications is guided by security considerations, efficiency requirements, and standards set by organizations such as NIST (National Institute of Standards and Technology)
- Some widely used ordinary curves in cryptography include secp256k1 (used in Bitcoin), NIST P-256, and Curve25519

Supersingular vs ordinary curves

- Supersingular and ordinary elliptic curves have distinct properties that make them suitable for different cryptographic applications
- Understanding the differences between these two classes of curves is crucial for selecting the appropriate type of curve for a given cryptographic protocol or mathematical investigation
- The comparison of supersingular and ordinary curves involves aspects such as their endomorphism rings, isogeny structures, and performance in cryptographic implementations

Differences in endomorphism rings

- The endomorphism rings of supersingular and ordinary curves have fundamentally different structures
- Supersingular curves have non-commutative endomorphism rings that are maximal orders in quaternion algebras over $\mathbb{Q}$
- This non-commutativity leads to a rich and complex structure of the endomorphism ring
- Ordinary curves, on the other hand, have commutative endomorphism rings that are orders in imaginary quadratic fields
- The commutativity of the endomorphism ring allows for a simpler and more well-understood structure
- The differences in endomorphism rings have implications for the isogeny graphs and the potential for constructing cryptographic protocols based on isogeny problems

Differences in isogeny structures

- The isogeny graphs of supersingular and ordinary curves exhibit distinct characteristics that impact their use in cryptography
- Supersingular curves have a highly connected isogeny graph, with each curve having a large number of outgoing isogenies (approximately $p + 1$ over $\mathbb{F}_{p^2}$)
- This dense connectivity is exploited in the construction of cryptographic protocols like SIDH and SIH
- Ordinary curves have a sparser isogeny graph, with fewer isogenies between curves and more restricted degrees of isogenies
- The structure of the isogeny graph of ordinary curves is more complex and less well-understood compared to the supersingular case
- The differences in isogeny structures affect the design and security analysis of isogeny-based cryptographic protocols

Comparative performance in cryptography

- The performance of supersingular and ordinary curves in cryptographic implementations depends on various factors, such as the specific protocol, the chosen parameters, and the implementation details
- Supersingular curves often allow for more efficient computation of certain operations, such as point multiplication and isogeny evaluation
- This efficiency is due to the special structure of supersingular curves and the availability of optimized algorithms for these operations
- Ordinary curves, while generally slower in terms of computation, benefit from a wider range of available curves and more mature implementation techniques
- The use of ordinary curves in cryptography has a longer history, and there are well-established standards and best practices for their implementation
- The comparative performance of supersingular and ordinary curves in cryptography is an active area of research, and the choice between the two depends on the specific requirements and constraints of the application

Comparative security in cryptography

- The security of cryptographic protocols based on supersingular and ordinary curves relies on different hardness assumptions and is subject to different types of attacks
- Supersingular curves are often used in the construction of post-quantum cryptographic protocols, as they are believed to be resistant to attacks by quantum computers
- The security of supersingular isogeny-based protocols, such as SIDH, relies on the hardness of the supersingular isogeny problem and its variants
- Ordinary curves, when used in classical elliptic curve cryptography, rely on the hardness of the elliptic curve discrete logarithm problem (ECDLP)
- The security of ordinary curve-based protocols, such as ECDH and ECDSA, is well-studied, and there are established guidelines for selecting secure curve parameters
- The comparative security of supersingular and ordinary curves is an ongoing research topic, and the choice between the two depends on the specific security requirements and the assumed capabilities of potential adversaries
- Factors such as key sizes, available attacks, and the impact of future cryptanalytic advances should be considered when comparing the security of supersingular and ordinary curve-based cryptography

Unit 3 – Elliptic curve cryptography

3.1 Elliptic curve Diffie-Hellman (ECDH) key exchange

Elliptic curve Diffie-Hellman (ECDH) key exchange is a crucial cryptographic protocol for secure communication. It allows two parties to establish a shared secret key over an insecure channel using elliptic curve math. ECDH offers advantages like smaller key sizes and improved efficiency compared to traditional Diffie-Hellman.

ECDH relies on the elliptic curve discrete logarithm problem for security. The protocol involves key generation, exchange of public keys, and derivation of a shared secret. ECDH is widely used in secure protocols like TLS and applications like messaging encryption and IoT security.

Overview of ECDH

- Elliptic Curve Diffie-Hellman (ECDH) is a key agreement protocol that allows two parties to establish a shared secret key over an insecure channel
- ECDH is based on the mathematical properties of elliptic curves and provides a secure and efficient method for key exchange
- ECDH plays a crucial role in modern cryptography, enabling secure communication and protecting sensitive data in various applications

Definition of ECDH

- ECDH is a variant of the Diffie-Hellman key exchange protocol that uses elliptic curve cryptography (ECC) instead of traditional finite field arithmetic
- It involves two parties, Alice and Bob, who agree on a common elliptic curve and a base point on that curve
- Alice and Bob each generate a private key and compute a corresponding public key using scalar multiplication on the base point
- They exchange their public keys and combine them with their own private keys to derive a shared secret key

Comparison vs Diffie-Hellman

- ECDH offers several advantages over the original Diffie-Hellman protocol, which uses modular exponentiation in a finite field
- ECDH requires smaller key sizes to achieve the same level of security as Diffie-Hellman, making it more efficient in terms of computation and bandwidth
- The security of ECDH relies on the elliptic curve discrete logarithm problem (ECDLP), which is considered harder to solve than the discrete logarithm problem used in Diffie-Hellman

Role in cryptography

- ECDH is widely used in various cryptographic protocols and applications to establish secure communication channels

- It is employed in secure communication protocols such as Transport Layer Security (TLS) and Internet Protocol Security (IPsec) to negotiate session keys
- ECDH is also used in secure messaging applications, email encryption, and secure storage systems to protect sensitive data
- The use of ECDH helps ensure confidentiality, integrity, and authentication in digital communications

ECDH process

- The ECDH process involves several steps to establish a shared secret key between two parties
- It consists of a setup phase, key generation, key exchange, and shared secret derivation

Setup phase

- In the setup phase, the two parties agree on a common elliptic curve and a base point (generator point) on that curve
- The elliptic curve is defined by its parameters, such as the curve equation, field size, and cofactor
- The chosen curve should be standardized and have well-understood security properties (NIST curves, Curve25519)
- The base point is a fixed point on the curve that serves as a starting point for scalar multiplication

Key generation

- Each party generates a private key, which is a randomly selected integer within the range of the curve's order
- The private key is kept secret and known only to the respective party
- Each party computes their public key by performing scalar multiplication of the base point with their private key
- The public key is a point on the elliptic curve derived from the private key and the base point

Key exchange

- The two parties exchange their public keys over an insecure channel
- Alice sends her public key to Bob, and Bob sends his public key to Alice
- The public keys are not sensitive and can be transmitted openly without compromising the security of the shared secret

Shared secret derivation

- Each party combines the received public key with their own private key to derive the shared secret
- Alice computes the shared secret by performing scalar multiplication of Bob's public key with her private key
- Bob computes the shared secret by performing scalar multiplication of Alice's public key with his private key
- The resulting shared secret is a point on the elliptic curve that is known only to Alice and Bob

- The shared secret can be further processed (hash function) to obtain a symmetric key for subsequent encryption and decryption

Elliptic curve fundamentals

- Elliptic curves are the mathematical foundation of ECDH and play a crucial role in its security and efficiency
- Understanding the properties and operations of elliptic curves is essential for implementing and analyzing ECDH

Elliptic curve equation

- An elliptic curve over a finite field is defined by an equation of the form $y^2 = x^3 + ax + b$, where a and b are constants and the discriminant $4a^3 + 27b^2 \neq 0$
- The equation defines a set of points (x, y) that satisfy the curve equation, along with a special point called the point at infinity
- Elliptic curves used in cryptography are typically defined over prime fields ($\mathbb{F}_p$) or binary fields ($\mathbb{F}_{2^m}$)

Point addition

- Elliptic curves have a group structure under the operation of point addition
- Given two points P and Q on the curve, their sum $R = P + Q$ is defined as the point obtained by drawing a line through P and Q and finding the third point of intersection with the curve
- The point addition operation satisfies the group axioms (closure, associativity, identity, inverses) and forms an abelian group
- Point addition is used in ECDH to compute public keys and derive the shared secret

Scalar multiplication

- Scalar multiplication is the operation of adding a point P to itself k times, where k is a scalar (integer)
- The result of scalar multiplication is denoted as kP , which is a point on the elliptic curve
- Scalar multiplication is the core operation in ECDH, used for generating public keys and deriving the shared secret
- Efficient algorithms (double-and-add, window methods) are used to optimize scalar multiplication computations

Generator point

- The generator point, also known as the base point, is a fixed point on the elliptic curve that is agreed upon by the communicating parties
- It serves as the starting point for scalar multiplication and is used to generate public keys
- The generator point should have a large prime order to ensure the security of the cryptographic scheme
- Standardized curves (NIST curves, Curve25519) have well-defined generator points with known properties

Security of ECDH

- The security of ECDH relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP)
- ECDLP is the problem of finding the scalar k given a point P and its scalar multiple kP on the elliptic curve
- The security of ECDH is based on the computational infeasibility of solving ECDLP for well-chosen elliptic curves and key sizes

Elliptic curve discrete logarithm problem

- ECDLP is the foundation of the security of elliptic curve cryptography, including ECDH
- Given a point P and its scalar multiple kP , the goal of ECDLP is to find the value of k
- Solving ECDLP is considered computationally infeasible for properly selected elliptic curves and sufficiently large key sizes
- The best known algorithms for solving ECDLP have a subexponential running time, making it impractical to solve for well-chosen parameters

Computational infeasibility

- The security of ECDH relies on the computational infeasibility of solving ECDLP
- Computational infeasibility means that the time and resources required to solve ECDLP exceed the capabilities of an adversary
- The choice of elliptic curve parameters (curve equation, field size) and key sizes directly affects the computational infeasibility of ECDLP
- Larger key sizes and carefully selected curves provide higher levels of security against known attacks

Comparison vs integer factorization

- ECDLP is considered harder to solve than the integer factorization problem (IFP) used in RSA cryptography
- IFP involves factoring a large composite number into its prime factors, which becomes increasingly difficult as the key size grows
- ECDLP, on the other hand, relies on the difficulty of finding discrete logarithms on elliptic curves, which is believed to be a harder problem
- As a result, ECDH can achieve the same level of security as RSA with smaller key sizes, leading to improved efficiency and performance

Key size vs security level

- The security level of ECDH depends on the size of the elliptic curve key (private key) used in the protocol
- Larger key sizes provide higher levels of security against known attacks, such as the Pollard's rho algorithm and the baby-step giant-step algorithm

- NIST recommends using elliptic curve key sizes of at least 256 bits to achieve a security level equivalent to 128-bit symmetric encryption
- Key sizes of 384 bits and 521 bits are recommended for higher security levels, providing security comparable to 192-bit and 256-bit symmetric encryption, respectively
- It is important to choose appropriate key sizes based on the desired security level and the longevity of the protected data

Advantages of ECDH

- ECDH offers several advantages over other key exchange protocols, making it an attractive choice for various applications
- These advantages include smaller key sizes, improved efficiency, reduced bandwidth, and suitability for constrained devices

Smaller key sizes

- ECDH requires significantly smaller key sizes compared to other public-key cryptographic schemes like RSA and Diffie-Hellman
- For the same level of security, ECDH keys are typically 2-3 times smaller than RSA keys
- Smaller key sizes result in reduced storage requirements, faster computations, and lower bandwidth consumption
- This makes ECDH particularly suitable for resource-constrained environments and devices with limited storage and processing capabilities

Improved efficiency

- ECDH computations involve scalar multiplication on elliptic curves, which can be efficiently implemented using specialized algorithms
- The smaller key sizes of ECDH lead to faster key generation, key exchange, and shared secret derivation compared to other schemes
- Efficient implementations of ECDH can leverage the parallelism and hardware acceleration available in modern processors
- The improved efficiency of ECDH enables faster establishment of secure communication channels and reduces the overall latency in cryptographic protocols

Reduced bandwidth

- The smaller key sizes of ECDH result in reduced bandwidth requirements for key exchange and data transmission
- Transmitting smaller public keys and encrypted data saves network bandwidth and reduces the overall communication overhead
- This is particularly beneficial in bandwidth-constrained environments, such as mobile networks and wireless sensor networks
- Reduced bandwidth consumption also leads to faster transmission times and improved responsiveness in communication protocols

Suitability for constrained devices

- ECDH is well-suited for resource-constrained devices, such as embedded systems, smart cards, and Internet of Things (IoT) devices
- The smaller key sizes and improved efficiency of ECDH make it feasible to implement secure key exchange on devices with limited processing power, memory, and battery life
- ECDH can be optimized for specific hardware architectures and can take advantage of hardware acceleration for elliptic curve operations
- The use of ECDH in constrained devices enables secure communication and authentication in various applications, such as wireless sensor networks and smart home systems

Applications of ECDH

- ECDH is widely used in various applications that require secure key exchange and establishment of shared secrets
- It is employed in secure communication protocols, virtual private networks (VPNs), messaging and email encryption, and Internet of Things (IoT) security

Secure communication protocols

- ECDH is used in popular secure communication protocols, such as Transport Layer Security (TLS) and Secure Shell (SSH)
- TLS uses ECDH for key exchange during the handshake process to establish a secure session between a client and a server
- SSH employs ECDH for secure key exchange and authentication, ensuring confidentiality and integrity of remote communication
- ECDH provides strong security and efficiency in these protocols, protecting sensitive data transmitted over untrusted networks

VPN and TLS

- Virtual Private Networks (VPNs) use ECDH to establish secure tunnels for remote access and encrypted communication
- ECDH is used in VPN protocols like Internet Key Exchange (IKE) and Internet Protocol Security (IPsec) to negotiate session keys
- TLS, the protocol that secures web communication (HTTPS), uses ECDH for key exchange in its handshake process
- ECDH ensures the confidentiality and integrity of data transmitted over VPNs and TLS-secured connections, protecting against eavesdropping and tampering

Messaging and email

- ECDH is employed in secure messaging and email encryption applications to protect the privacy and confidentiality of user communication
- End-to-end encrypted messaging apps (Signal, WhatsApp) use ECDH to derive shared secrets for message encryption and decryption

- Email encryption protocols, such as Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extensions (S/MIME), can utilize ECDH for key exchange
- ECDH ensures that only the intended recipients can access and read the encrypted messages, preventing unauthorized access and interception

Internet of Things (IoT)

- ECDH plays a crucial role in securing communication and authentication in the Internet of Things (IoT) ecosystem
- IoT devices, such as smart home appliances, wearables, and industrial sensors, often have limited resources and require efficient security mechanisms
- ECDH is used to establish secure communication channels between IoT devices and cloud platforms or gateways
- It enables secure firmware updates, device provisioning, and data exchange in IoT networks, protecting against unauthorized access and tampering
- ECDH's small key sizes and efficiency make it suitable for resource-constrained IoT devices, ensuring strong security while minimizing overhead

Implementations of ECDH

- ECDH is implemented using standardized elliptic curves and is supported by various libraries and frameworks
- Proper implementation of ECDH is crucial to ensure its security and interoperability across different systems

Standardized curves

- Standardized elliptic curves are recommended for use in ECDH implementations to ensure security and compatibility
- NIST (National Institute of Standards and Technology) has defined a set of elliptic curves for cryptographic purposes (NIST P-256, P-384, P-521)
- Other standardized curves include Curve25519 and Curve448, which are designed to provide strong security and resistance against side-channel attacks
- Using standardized curves ensures that the implementation follows best practices and has been thoroughly analyzed by the cryptographic community

Curve25519 and Curve448

- Curve25519 and Curve448 are popular elliptic curves used in ECDH implementations
- Curve25519 is a 255-bit elliptic curve that offers high security and efficiency, making it suitable for a wide range of applications
- Curve448 is a 448-bit elliptic curve that provides even higher security levels and is recommended for future-proofing cryptographic systems
- These curves have been designed to resist known attacks and provide strong security guarantees

- They have been widely adopted in various cryptographic libraries and protocols due to their performance and security properties

Libraries and frameworks

- ECDH is supported by numerous cryptographic libraries and frameworks across different programming languages
- OpenSSL is a widely used open-source library that provides ECDH functionality in C and C++
- Java Cryptography Architecture (JCA) and Bouncy Castle library offer ECDH support for Java applications
- Microsoft .NET Framework includes ECDH classes in the System.Security.Cryptography namespace
- Various other libraries (libsodium, TweetNaCl) provide high-level APIs for ECDH in different programming languages, simplifying the implementation process

Best practices for implementation

- Implementing ECDH securely requires following best practices and guidelines to mitigate potential vulnerabilities
- Use standardized and well-established elliptic curves (NIST curves, Curve25519) that have been thoroughly analyzed and tested
- Generate cryptographically secure random numbers for private keys using reliable random number generators
- Validate public keys received from other parties to ensure they are valid points on the elliptic curve and not the point at infinity
- Use secure key derivation functions (HKDF, PBKDF2) to derive session keys from the shared secret
- Implement proper error handling and avoid leaking sensitive information through error messages or side channels
- Regularly update and patch cryptographic libraries and frameworks to address any discovered vulnerabilities or weaknesses

Attacks on ECDH

- ECDH, like any cryptographic protocol, is subject to various attacks that aim to compromise its security
- Understanding the potential attacks and implementing appropriate countermeasures is crucial to ensure the robustness of ECDH implementations

Invalid curve attacks

- Invalid curve attacks exploit the lack of proper validation of received public keys in ECDH implementations
- An attacker can manipulate the public key by sending a point that lies on a different elliptic curve or a malformed point

- If the implementation does not validate the received point, it may perform computations on the invalid curve, leading to potential vulnerabilities
- Countermeasures include validating the received public key to ensure it is a valid point on the agreed-upon elliptic curve

Small subgroup attacks

- Small subgroup attacks target ECDH implementations that use elliptic curves with small subgroups
- If the order of the elliptic curve is not prime or has small factors, an attacker can exploit the structure of the subgroups to obtain information about the private key
- Countermeasures involve using elliptic curves with prime order or performing additional checks to ensure the points lie in the correct subgroup
- Validating the order of the received public key and using cofactor multiplication can mitigate small subgroup attacks

Timing and side-channel attacks

- Timing attacks

3.2 Elliptic curve digital signature algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) is a powerful cryptographic tool that leverages elliptic curves for secure digital signatures. It offers enhanced security with shorter key sizes compared to traditional algorithms, making it ideal for resource-constrained environments.

ECDSA involves key generation, signature creation, and verification processes. Understanding its implementation, security considerations, and applications is crucial for ensuring robust digital authentication and integrity in various systems, from cryptocurrencies to secure communication protocols.

Basics of ECDSA

- Elliptic Curve Digital Signature Algorithm (ECDSA) is a cryptographic signature scheme based on the mathematical properties of elliptic curves
- ECDSA provides secure digital signatures for authentication, data integrity, and non-repudiation in various applications
- Understanding the foundations of elliptic curve cryptography and digital signature algorithms is essential for effectively using ECDSA

Elliptic curve cryptography foundations

- Elliptic curves are mathematical curves defined by the equation $y^2 = x^3 + ax + b$ over a finite field
- Elliptic curve cryptography (ECC) utilizes the algebraic structure of elliptic curves to construct cryptographic primitives
- ECC offers equivalent security to other public-key cryptosystems (RSA) with shorter key sizes, leading to improved efficiency

Digital signature algorithms overview

- Digital signature algorithms enable the creation and verification of digital signatures
- Signatures provide authentication, integrity, and non-repudiation for digital messages or documents
- Common digital signature algorithms include RSA, DSA, and ECDSA, each with its own mathematical foundations and properties

ECDSA vs other signature schemes

- ECDSA is based on elliptic curve cryptography, while RSA and DSA use integer factorization and discrete logarithm problems, respectively
- ECDSA offers shorter key sizes compared to RSA and DSA for equivalent security levels, making it more suitable for resource-constrained environments
- ECDSA has gained popularity due to its efficiency and wide adoption in various cryptographic protocols and applications (Bitcoin, TLS)

ECDSA key generation

- Key generation is the process of creating a pair of cryptographic keys: a private key and a corresponding public key
- Proper key generation is crucial for the security of ECDSA, as the private key must remain secret and the public key is used for signature verification
- ECDSA key generation involves selecting appropriate elliptic curve domain parameters and generating a random private key

Elliptic curve domain parameters

- Elliptic curve domain parameters define the specific elliptic curve and finite field used in ECDSA
- Domain parameters include the curve equation, field size, base point, order of the base point, and cofactor
- Standardized domain parameters (NIST P-256, secp256k1) are commonly used to ensure interoperability and security

Private key selection

- The private key is a randomly selected integer from the range $[1, n - 1]$, where n is the order of the base point
- Cryptographically secure random number generators (CSPRNGs) are used to generate the private key to ensure unpredictability
- The private key must be kept secret and protected from unauthorized access or disclosure

Public key calculation

- The public key is derived from the private key by performing scalar multiplication of the private key with the base point on the elliptic curve
- Given a private key d and base point G , the public key Q is calculated as $Q = dG$

- The public key is a point on the elliptic curve and can be shared freely without compromising the security of the private key

ECDSA signature generation

- Signature generation is the process of creating a digital signature for a message using the signer's private key
- ECDSA signature generation involves several steps, including message hashing, ephemeral key generation, and signature calculation
- The resulting signature consists of two components (r, s) that are used for verification

Message hashing

- The message to be signed is first hashed using a cryptographic hash function (SHA-256) to obtain a fixed-size digest
- Hashing ensures the integrity of the message and prevents the signing of arbitrarily large messages
- The hash digest is then used as input to the signature calculation process

Ephemeral key generation

- An ephemeral key pair is generated for each signature to ensure randomness and prevent certain attacks
- The ephemeral private key k is randomly selected from the range $[1, n - 1]$, where n is the order of the base point
- The corresponding ephemeral public key R is calculated as $R = kG$, where G is the base point

Signature calculation

- The signature components (r, s) are calculated using the ephemeral key, private key, and hash digest
- The r component is derived from the x-coordinate of the ephemeral public key R
- The s component is calculated as $s = k^{-1}(z + rd) \bmod n$, where z is the hash digest, d is the private key, and n is the order of the base point

Signature encoding

- The signature components (r, s) are encoded into a standardized format for transmission and storage
- Common encoding formats include ASN.1 DER (Distinguished Encoding Rules) and concatenated r and s values
- Proper encoding ensures interoperability and compatibility across different ECDSA implementations

ECDSA signature verification

- Signature verification is the process of validating the authenticity and integrity of a digital signature
- ECDSA signature verification involves several steps, including public key validation, signature decoding, and the verification process itself

- Successful verification confirms that the signature was generated by the owner of the corresponding private key and that the message has not been tampered with

Public key validation

- The public key used for verification must be validated to ensure it is a valid point on the elliptic curve
- Validation checks include verifying that the public key satisfies the curve equation and that it is not the point at infinity
- Public key validation prevents attacks that attempt to use invalid or malformed public keys

Signature decoding

- The received signature is decoded from its encoded format back into the (r, s) components
- Decoding reverses the encoding process and extracts the individual signature components for verification
- Proper decoding is necessary to ensure the correct values are used in the verification process

Signature verification process

- The verification process uses the public key, message hash, and signature components to determine the validity of the signature
- The verifier calculates two elliptic curve points u_1 and u_2 using the signature components, public key, and message hash
- The verifier then computes a point R' as $R' = u_1G + u_2Q$, where G is the base point and Q is the public key
- The signature is considered valid if the x-coordinate of R' matches the r component of the signature

Verification outcomes

- If the signature verification process is successful, it confirms that the signature was generated by the owner of the corresponding private key and that the message has not been modified
- A failed verification indicates that the signature is invalid, which may be due to an incorrect signature, a tampered message, or an invalid public key
- Proper handling of verification outcomes is crucial for maintaining the security and integrity of the system

ECDSA security considerations

- ECDSA provides strong security guarantees when implemented and used correctly
- However, several security considerations must be taken into account to prevent potential vulnerabilities and attacks
- Proper implementation, key management, and protection against side-channel attacks are essential for maintaining the security of ECDSA

Randomness in key generation

- The security of ECDSA heavily relies on the randomness of the private key and ephemeral keys used in signature generation
- Inadequate randomness can lead to predictable keys and vulnerabilities such as key recovery attacks
- Using cryptographically secure random number generators (CSPRNGs) and properly seeding them is crucial for ensuring randomness

Signature malleability

- ECDSA signatures are malleable, meaning that given a valid signature (r, s) , an attacker can create a different valid signature $(r, -s \bmod n)$ without knowledge of the private key
- Signature malleability can lead to issues in certain protocols and applications that rely on unique signatures
- Proper signature encoding and verification procedures should be followed to mitigate the impact of signature malleability

Side-channel attacks

- Side-channel attacks exploit information leakage from the physical implementation of ECDSA to recover sensitive data, such as the private key
- Examples of side-channel attacks include timing attacks, power analysis attacks, and electromagnetic emanation attacks
- Implementing countermeasures such as constant-time operations, randomization techniques, and physical security measures can help mitigate side-channel attacks

Key management best practices

- Proper key management is essential for maintaining the security of ECDSA keys throughout their lifecycle
- Best practices include secure key generation, safe key storage (hardware security modules), and regular key rotation and renewal
- Access control, backup and recovery procedures, and secure key destruction should also be implemented to prevent unauthorized access or loss of keys

ECDSA implementation aspects

- Implementing ECDSA requires careful consideration of various aspects to ensure efficiency, interoperability, and security
- Efficient algorithms, adherence to standards, hardware acceleration, and thorough testing and validation are important for successful ECDSA implementations
- Proper implementation choices and practices contribute to the overall performance and reliability of ECDSA in real-world applications

Efficient algorithms for ECDSA

- Efficient algorithms for elliptic curve operations (point addition, scalar multiplication) are crucial for optimizing ECDSA performance
- Techniques such as pre-computation, windowing methods, and projective coordinates can significantly speed up ECDSA operations
- Implementing efficient algorithms helps reduce computational overhead and improve the scalability of ECDSA-based systems

Interoperability and standards

- Adhering to established standards and guidelines ensures interoperability among different ECDSA implementations
- Standards such as NIST FIPS 186-4, ANSI X9.62, and RFC 6979 provide specifications for ECDSA key generation, signature generation, and encoding
- Following these standards promotes compatibility and allows seamless integration of ECDSA in various protocols and applications

Hardware acceleration for ECDSA

- Hardware acceleration can significantly improve the performance of ECDSA operations, especially in resource-constrained environments
- Dedicated hardware modules (cryptographic co-processors) or instruction set extensions (Intel SHA extensions) can offload ECDSA computations from the main processor
- Utilizing hardware acceleration reduces the computational burden on the system and enables faster signature generation and verification

Testing and validation of implementations

- Comprehensive testing and validation of ECDSA implementations are essential to ensure correctness, security, and compliance with standards
- Test vectors and reference implementations provide a basis for verifying the correctness of ECDSA operations
- Security audits, code reviews, and formal verification techniques help identify and mitigate potential vulnerabilities or implementation flaws
- Regular testing and validation processes contribute to the overall reliability and trustworthiness of ECDSA implementations

ECDSA applications and use cases

- ECDSA has found widespread adoption in various domains due to its security, efficiency, and versatility
- From cryptocurrency systems to secure communication protocols and digital identity solutions, ECDSA plays a crucial role in ensuring the integrity and authenticity of digital transactions and communications

- As technology evolves, ECDSA continues to be a key component in shaping the future of secure digital interactions

ECDSA in cryptocurrency systems

- ECDSA is extensively used in cryptocurrency systems, such as Bitcoin and Ethereum, for securing transactions and user identities
- In these systems, ECDSA is used to generate public-private key pairs for user addresses and to sign transactions to prove ownership and authorize transfers
- The security and efficiency of ECDSA contribute to the overall integrity and scalability of cryptocurrency networks

ECDSA for secure communication protocols

- ECDSA is employed in various secure communication protocols to establish authenticated and confidential channels
- Protocols such as Transport Layer Security (TLS) and Secure Shell (SSH) use ECDSA for key exchange and digital signatures
- ECDSA helps protect the privacy and integrity of communication channels, ensuring that data is securely transmitted between parties

ECDSA in digital identity systems

- ECDSA is used in digital identity systems to provide strong authentication and non-repudiation
- Digital certificates, which bind public keys to identities, often rely on ECDSA for signature generation and verification
- ECDSA-based digital signatures enable secure authentication, ensuring that only authorized entities can access sensitive resources or perform critical actions

Future directions for ECDSA

- As quantum computing advances, the security of traditional public-key cryptography, including ECDSA, may be threatened
- Research is ongoing to develop quantum-resistant cryptographic algorithms that can withstand attacks from quantum computers
- Post-quantum cryptography standards, such as those being developed by NIST, aim to provide alternatives to ECDSA for long-term security
- Continued development and adoption of ECDSA in new domains, such as Internet of Things (IoT) and blockchain applications, will drive further innovation and security enhancements

3.3 Elliptic curve integrated encryption scheme (ECIES)

ECIES combines elliptic curve cryptography with symmetric encryption for secure messaging. It uses a Diffie-Hellman key exchange on elliptic curves to create a shared secret key, which then encrypts and authenticates messages.

The scheme involves key generation, encryption, and decryption steps. It provides strong security based on the Elliptic Curve Discrete Logarithm Problem, making it suitable for various applications requiring

high-level protection and efficiency.

Overview of ECIES

- Elliptic Curve Integrated Encryption Scheme (ECIES) is a hybrid encryption scheme that combines the security of elliptic curve cryptography with the efficiency of symmetric encryption
- ECIES provides a standardized way to securely encrypt and decrypt messages using elliptic curve cryptography, ensuring confidentiality, integrity, and authentication
- The scheme leverages the Diffie-Hellman key exchange protocol on elliptic curves to establish a shared secret key, which is then used for symmetric encryption and authentication

Definition of ECIES

- ECIES is a public-key encryption scheme that utilizes elliptic curve cryptography to encrypt and decrypt messages
- It combines the Elliptic Curve Diffie-Hellman (ECDH) key exchange with symmetric encryption and message authentication
- ECIES provides semantic security, which means that an adversary cannot learn any information about the plaintext from the ciphertext without knowing the private key

Components of ECIES

- ECIES consists of three main components: key generation, encryption, and decryption
- Key generation involves generating a pair of public and private keys based on an elliptic curve
- Encryption combines the recipient's public key with a randomly generated ephemeral key to derive a shared secret, which is then used to encrypt the message and compute an authentication tag
- Decryption uses the recipient's private key and the ephemeral key to derive the shared secret, decrypt the message, and verify the authentication tag

ECIES encryption process

- The ECIES encryption process involves several steps to securely encrypt a message using the recipient's public key and a randomly generated ephemeral key

Key generation in ECIES

- The sender generates an ephemeral key pair (k, R) , where k is a random integer and $R = kG$ is a point on the elliptic curve (G is the base point)
- The sender also obtains the recipient's public key Q , which is a point on the elliptic curve

Encryption steps in ECIES

- The sender computes a shared secret point $S = kQ$ using the ephemeral private key k and the recipient's public key Q
- The shared secret point S is then passed through a key derivation function (KDF) to derive a symmetric encryption key K_{enc} and a MAC key K_{mac}

- The sender encrypts the plaintext message m using the symmetric encryption key K_{enc} and a chosen symmetric encryption algorithm (e.g., AES) to obtain the ciphertext C
- The sender computes an authentication tag t over the ciphertext C using the MAC key K_{mac} and a chosen MAC algorithm (e.g., HMAC)

Ciphertext structure in ECIES

- The final ciphertext in ECIES consists of three components: the ephemeral public key R , the encrypted message C , and the authentication tag t
- The ciphertext is represented as (R, C, t) , where R is a point on the elliptic curve, C is the encrypted message, and t is the authentication tag
- The ciphertext can be transmitted to the recipient, who can use their private key to decrypt the message and verify its authenticity

ECIES decryption process

- The ECIES decryption process involves recovering the plaintext message from the ciphertext using the recipient's private key and the ephemeral public key included in the ciphertext

Decryption steps in ECIES

- The recipient receives the ciphertext (R, C, t) , where R is the ephemeral public key, C is the encrypted message, and t is the authentication tag
- The recipient computes the shared secret point $S = dR$ using their private key d and the ephemeral public key R
- The shared secret point S is passed through the same key derivation function (KDF) used in the encryption process to derive the symmetric encryption key K_{enc} and the MAC key K_{mac}
- The recipient verifies the authenticity of the ciphertext by computing the authentication tag t' over the ciphertext C using the MAC key K_{mac} and comparing it with the received tag t . If the tags match, the ciphertext is considered authentic

Plaintext recovery in ECIES

- If the authentication tag is verified successfully, the recipient decrypts the ciphertext C using the symmetric encryption key K_{enc} and the chosen symmetric encryption algorithm (e.g., AES) to obtain the plaintext message m
- The recipient has now successfully recovered the original plaintext message m from the ciphertext (R, C, t)
- If the authentication tag verification fails, the recipient should discard the ciphertext as it may have been tampered with or corrupted during transmission

Security of ECIES

- ECIES provides strong security guarantees based on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the security of the underlying symmetric encryption and MAC algorithms

Assumptions in ECIES security

- The security of ECIES relies on the assumption that the ECDLP is computationally infeasible to solve
- This means that given a point Q on the elliptic curve, it is extremely difficult to find the scalar k such that $Q = kG$, where G is the base point
- ECIES also assumes that the underlying symmetric encryption algorithm (e.g., AES) and MAC algorithm (e.g., HMAC) are secure and resistant to known attacks

Security proofs for ECIES

- ECIES has been proven to provide indistinguishability under chosen plaintext attack (IND-CPA) and indistinguishability under chosen ciphertext attack (IND-CCA) security
- IND-CPA security ensures that an adversary cannot distinguish between the encryptions of two different messages, even if they can obtain encryptions of chosen plaintexts
- IND-CCA security provides an additional layer of protection, ensuring that an adversary cannot learn any information about the plaintext even if they can obtain decryptions of chosen ciphertexts (except for the challenge ciphertext)
- The security proofs of ECIES rely on the assumptions mentioned earlier and the security of the underlying primitives

Comparison of ECIES vs other schemes

- ECIES offers several advantages over other public-key encryption schemes, such as RSA and ElGamal
- Elliptic curve cryptography provides higher security with smaller key sizes compared to RSA, reducing storage and transmission overhead
- ECIES is more efficient in terms of computation compared to schemes like ElGamal, which requires multiple exponentiations
- ECIES also provides built-in message authentication, eliminating the need for a separate MAC algorithm

Applications of ECIES

- ECIES has found widespread adoption in various domains that require secure communication and data protection

Use cases for ECIES

- ECIES is commonly used in secure messaging applications to encrypt and decrypt messages between users
- It is also employed in secure email communication, ensuring the confidentiality and integrity of email messages
- ECIES is used in blockchain and cryptocurrency systems to secure transactions and protect user privacy
- Internet of Things (IoT) devices and smart contracts often rely on ECIES for secure communication and data exchange

Advantages of ECIES in applications

- ECIES provides strong security guarantees based on the hardness of the ECDLP, making it suitable for applications that require high levels of security
- The small key sizes in elliptic curve cryptography reduce storage and transmission overhead, making ECIES efficient for resource-constrained devices
- The built-in message authentication in ECIES eliminates the need for a separate MAC algorithm, simplifying the implementation and reducing the attack surface
- ECIES is flexible and can be used with various elliptic curves and symmetric encryption algorithms, allowing developers to choose the most suitable parameters for their specific application

Limitations of ECIES in applications

- ECIES requires the secure distribution and management of public keys, which can be challenging in large-scale systems
- The performance of ECIES may be slower compared to symmetric encryption algorithms, especially for large messages
- ECIES relies on the security of the underlying elliptic curve and symmetric encryption algorithms, so any weaknesses discovered in these primitives can impact the security of ECIES
- Implementing ECIES correctly and securely requires careful attention to details, such as proper key generation, secure random number generation, and protection against side-channel attacks

Variants and extensions of ECIES

- Several variants and extensions of ECIES have been proposed to address specific security requirements and improve efficiency

Modified ECIES schemes

- Ephemeral-Static ECIES (ES-ECIES) is a variant that uses a static key pair for the recipient instead of generating a new ephemeral key pair for each encryption, reducing computational overhead
- Elliptic Curve Integrated Encryption Scheme with Key Encapsulation Mechanism (ECIES-KEM) separates the key encapsulation and data encapsulation steps, allowing for more flexibility in key management
- Schnorr ECIES (SECIES) replaces the ECDH key exchange with Schnorr signatures, providing additional authentication and non-repudiation properties

Integration of ECIES with other primitives

- ECIES can be combined with other cryptographic primitives to enhance security or provide additional functionalities
- ECIES can be integrated with digital signatures (e.g., ECDSA) to provide authentication and non-repudiation of encrypted messages
- ECIES can be used in conjunction with key derivation functions (e.g., HKDF) to derive multiple keys for different purposes (encryption, authentication, etc.) from the shared secret

- ECIES can be combined with padding schemes (e.g., OAEP) to provide additional security against chosen-ciphertext attacks

Standardization of ECIES

- ECIES has been standardized by various organizations to ensure interoperability and promote secure implementations
- The ANSI X9.63 standard defines ECIES and provides guidelines for its implementation
- The IEEE 1363a standard includes ECIES as one of the recommended public-key encryption schemes
- The SECG (Standards for Efficient Cryptography Group) has published SEC 1, which specifies ECIES and provides test vectors for implementation validation
- The NIST (National Institute of Standards and Technology) has included ECIES in its recommendations for key management and public-key cryptography

3.4 Pairing-based cryptography

Pairing-based cryptography uses bilinear maps on elliptic curves to create advanced cryptographic schemes. These pairings, like the Weil and Tate pairings, enable new security features by leveraging the algebraic structure of elliptic curves.

This approach opens up possibilities for identity-based encryption, short signatures, and other novel protocols. However, it also brings challenges in curve selection, efficiency optimization, and balancing security with performance in real-world implementations.

Bilinear pairings on elliptic curves

- Bilinear pairings are mathematical functions that map pairs of points on an elliptic curve to elements in a finite field
- Pairings enable the construction of advanced cryptographic schemes by leveraging the algebraic structure of elliptic curves
- Two main types of pairings used in cryptography are the Weil pairing and the Tate pairing

Weil pairing

Definition of Weil pairing

- The Weil pairing is a bilinear map that takes two points on an elliptic curve and outputs an element in a finite field
- Denoted as $e_n : E[n] \times E[n] \rightarrow \mu_n$, where $E[n]$ is the n -torsion subgroup and μ_n is the group of n -th roots of unity
- Satisfies bilinearity, non-degeneracy, and alternating properties

Properties of Weil pairing

- Bilinearity: $e_n(P_1 + P_2, Q) = e_n(P_1, Q) \cdot e_n(P_2, Q)$ and $e_n(P, Q_1 + Q_2) = e_n(P, Q_1) \cdot e_n(P, Q_2)$
- Non-degeneracy: If $e_n(P, Q) = 1$ for all $Q \in E[n]$, then $P = \mathcal{O}$ (the point at infinity)

- Alternating: $e_n(P, P) = 1$ for all $P \in E[n]$
- Compatibility with isogenies: If $\phi : E_1 \rightarrow E_2$ is an isogeny, then $e_n(\phi(P), \phi(Q)) = e_n(P, Q)^{\deg(\phi)}$

Computation of Weil pairing

- The Weil pairing can be computed using Miller's algorithm, which evaluates rational functions at divisors
- Involves computing line functions and evaluating them at points on the elliptic curve
- Requires careful selection of divisors and efficient arithmetic on the elliptic curve and finite field

Tate pairing

Definition of Tate pairing

- The Tate pairing is another bilinear map that takes points on an elliptic curve and outputs an element in a finite field
- Denoted as $e_T : E(\mathbb{F}_q)[n] \times E(\mathbb{F}_{q^k})/nE(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n$
- Maps a point in the n -torsion subgroup and a coset of $E(\mathbb{F}_{q^k})$ to an element in the quotient group $\mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^n$

Properties of Tate pairing

- Bilinearity: Similar to the Weil pairing, the Tate pairing satisfies bilinearity properties
- Non-degeneracy: The Tate pairing is non-degenerate in the sense that for any non-zero point $P \in E(\mathbb{F}_q)[n]$, there exists a point $Q \in E(\mathbb{F}_{q^k})$ such that $e_T(P, Q) \neq 1$
- Compatibility with isogenies: The Tate pairing is also compatible with isogenies between elliptic curves

Computation of Tate pairing

- The Tate pairing can be computed using Miller's algorithm, similar to the Weil pairing
- Involves evaluating rational functions at divisors and performing operations on the elliptic curve and finite field
- Requires careful selection of parameters and efficient implementations to achieve practical performance

Tate pairing vs Weil pairing

- The Tate pairing has some advantages over the Weil pairing in terms of computation and flexibility
- The Tate pairing can be computed more efficiently than the Weil pairing in certain cases
- The Tate pairing allows for a wider range of elliptic curves and subgroups to be used compared to the Weil pairing
- However, the Weil pairing has a simpler definition and can be easier to work with in some theoretical settings

Pairing-friendly elliptic curves

Supersingular curves

- Supersingular elliptic curves have a special structure that makes them suitable for pairing-based cryptography
- They have a large embedding degree, which determines the size of the finite field used in the pairing computation
- Examples of supersingular curves include curves over binary fields and curves with j -invariant 0 or 1728

Ordinary curves

- Ordinary elliptic curves can also be used for pairing-based cryptography, but they require more careful selection of parameters
- The embedding degree of ordinary curves is typically smaller than that of supersingular curves
- Suitable ordinary curves include MNT curves, BN curves, and KSS curves

Construction of pairing-friendly curves

- Constructing pairing-friendly elliptic curves involves finding curves with desirable properties for efficient pairing computation and security
- Methods for constructing pairing-friendly curves include the Cocks-Pinch method, the Dupont-Engge-Morain method, and the Brezing-Weng method
- The choice of construction method depends on the desired embedding degree, security level, and efficiency requirements

Security of pairing-based cryptography

Bilinear Diffie-Hellman problem

- The bilinear Diffie-Hellman (BDH) problem is a computational problem that underlies the security of many pairing-based cryptographic schemes
- Given P, aP, bP, cP for some points P on an elliptic curve and unknown scalars a, b, c , the BDH problem is to compute $e(P, P)^{abc}$
- The hardness of the BDH problem is assumed in the security proofs of various pairing-based protocols

Decisional bilinear Diffie-Hellman problem

- The decisional bilinear Diffie-Hellman (DBDH) problem is a decisional variant of the BDH problem
- Given P, aP, bP, cP, Z , the DBDH problem is to determine whether $Z = e(P, P)^{abc}$ or a random element in the target group
- The DBDH assumption states that solving the DBDH problem is computationally infeasible for any polynomial-time algorithm

Security reductions and assumptions

- The security of pairing-based cryptographic schemes is often proved by reducing the security to the hardness of the BDH or DBDH problem
- Other assumptions used in the security analysis of pairing-based cryptography include the q -strong Diffie-Hellman assumption and the external Diffie-Hellman assumption
- Reductions and assumptions provide a formal framework for analyzing the security of pairing-based protocols and relating them to well-studied computational problems

Applications of pairing-based cryptography

Identity-based encryption

- Identity-based encryption (IBE) is a type of public-key encryption where the public key is derived from a user's identity (email address, phone number)
- Pairings enable the construction of efficient IBE schemes, such as the Boneh-Franklin IBE scheme and the Cocks IBE scheme
- IBE simplifies key management and eliminates the need for public key certificates

Attribute-based encryption

- Attribute-based encryption (ABE) is a generalization of IBE where users' private keys and ciphertexts are associated with attributes or policies
- Pairings are used to construct expressive ABE schemes, such as ciphertext-policy ABE (CP-ABE) and key-policy ABE (KP-ABE)
- ABE enables fine-grained access control and supports complex access policies based on attributes

Short signatures

- Pairings can be used to construct short signature schemes, such as the Boneh-Lynn-Shacham (BLS) signature scheme
- BLS signatures are much shorter than traditional digital signatures (RSA, ECDSA) while providing comparable security
- Short signatures are useful in scenarios with limited bandwidth or storage, such as blockchain applications and IoT devices

Tripartite Diffie-Hellman key exchange

- Pairings enable the construction of three-party Diffie-Hellman key exchange protocols, such as the Joux protocol
- In a tripartite Diffie-Hellman key exchange, three parties can simultaneously compute a shared secret key without the need for multiple rounds of communication
- Tripartite key exchange has applications in secure multi-party computation and collaborative key establishment

Efficiency of pairing computations

Miller's algorithm

- Miller's algorithm is the fundamental algorithm for computing pairings on elliptic curves
- It involves evaluating rational functions at divisors and performing operations on the elliptic curve and finite field
- Miller's algorithm has a complexity of $O(\log n)$ for a pairing of order n , making it relatively efficient

Optimizations and improvements

- Various optimizations and improvements have been proposed to enhance the efficiency of pairing computations
- Techniques such as the use of affine coordinates, the employment of efficient arithmetic in the underlying field, and the exploitation of the twisted curve can significantly speed up pairing computations
- Preprocessing and precomputation techniques can also be used to reduce the online computation time of pairings

Comparison of pairing types

- The efficiency of pairing computations depends on the choice of pairing type (Weil pairing, Tate pairing, optimal pairings)
- Tate pairings are generally more efficient than Weil pairings due to their simpler definition and the ability to use distortion maps
- Optimal pairings, such as the Ate pairing and its variants (R-ate pairing, optimal Ate pairing), further improve the efficiency by reducing the loop length in Miller's algorithm
- The choice of pairing type depends on the specific requirements of the cryptographic protocol and the trade-offs between security, efficiency, and implementation complexity

Challenges in pairing-based cryptography

Selecting suitable curves and parameters

- Selecting suitable elliptic curves and parameters is crucial for the security and efficiency of pairing-based cryptography
- Factors to consider include the embedding degree, the size of the underlying field, the security level, and the existence of efficient algorithms for pairing computation
- Standardized curves, such as the BN curves and the BLS curves, provide well-studied options for pairing-friendly curves

Balancing security and efficiency

- Pairing-based cryptography often involves a trade-off between security and efficiency
- Higher security levels require larger key sizes and more complex computations, which can impact the performance of pairing-based protocols

- Techniques such as using asymmetric pairings and optimizing pairing computations can help balance security and efficiency
- It is important to carefully assess the security requirements and performance constraints of the specific application when designing pairing-based cryptographic schemes

Implementation considerations

- Implementing pairing-based cryptography requires careful consideration of various factors to ensure security and efficiency
- Side-channel attacks, such as timing attacks and power analysis attacks, can potentially leak sensitive information during pairing computations
- Countermeasures, such as constant-time implementations and randomization techniques, should be employed to mitigate side-channel vulnerabilities
- Efficient algorithms and libraries for elliptic curve arithmetic and finite field operations are essential for practical implementations of pairing-based cryptography
- Proper parameter validation, key management, and secure key storage are also important aspects of implementing pairing-based cryptographic protocols

3.5 Security of elliptic curve cryptosystems

Elliptic curve cryptosystems are a cornerstone of modern digital security. Their strength lies in the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally difficult to solve, providing robust protection for sensitive data and communications.

While ECC offers strong security, it's not invulnerable. Attacks like side-channel, invalid curve, and small subgroup attacks can exploit weaknesses in implementation. Proper security practices, standardized protocols, and awareness of quantum computing threats are crucial for maintaining ECC's effectiveness.

Hardness of ECDLP

- The security of elliptic curve cryptography relies on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP)
- ECDLP involves finding the discrete logarithm of a point on an elliptic curve with respect to a given base point
- The hardness of ECDLP is the foundation for the security guarantees provided by ECC-based cryptographic schemes

Discrete logarithm problem in ECC

- Given an elliptic curve E over a finite field, a base point P on E , and another point Q on E , the ECDLP is to find an integer k such that $Q = kP$
- The integer k is called the discrete logarithm of Q with respect to the base point P
- The difficulty of solving the ECDLP is what makes ECC suitable for cryptographic applications

Complexity of solving ECDLP

- The best-known algorithms for solving ECDLP have a complexity of $O(\sqrt{n})$, where n is the order of the base point P

- This subexponential complexity is significantly harder than the complexity of integer factorization, which is the basis for RSA security
- The high complexity of ECDLP allows ECC to achieve the same level of security as RSA with much smaller key sizes

Comparison vs integer factorization

- Integer factorization, used in RSA, has a subexponential complexity of $O(e^{((1.9 * (\ln n)^{1/3}) * (\ln \ln n)^{2/3}))})$
- In contrast, ECDLP has a complexity of $O(\sqrt{n})$, making it harder to solve than integer factorization for the same key size
- This difference in complexity allows ECC to use smaller key sizes compared to RSA while maintaining the same level of security (160-bit ECC key $\approx$ 1024-bit RSA key)

Attacks on ECC

- While ECC is considered secure, it is not immune to various types of attacks that exploit vulnerabilities in implementations or specific curve parameters
- These attacks highlight the importance of proper implementation and parameter selection to maintain the security of ECC-based systems
- Understanding and mitigating these attacks is crucial for developers and security professionals working with ECC

Side-channel attacks

- Side-channel attacks exploit information leakage from the physical implementation of ECC, such as timing, power consumption, or electromagnetic emanations
- An example is the timing attack, where an attacker analyzes the time taken for different operations to deduce information about the private key
- Countermeasures include constant-time implementations, blinding techniques, and physical shielding

Invalid curve attacks

- Invalid curve attacks occur when an attacker forces a victim to perform ECC operations on a maliciously crafted curve that is not part of the specified domain parameters
- By using invalid curves, an attacker can exploit weaknesses to recover the victim's private key
- Proper validation of input points and adhering to standardized domain parameters help prevent invalid curve attacks

Small subgroup attacks

- Small subgroup attacks target ECC implementations that do not properly validate the order of points on the curve
- An attacker can exploit the existence of small subgroups to learn information about the private key
- Countermeasures include validating the order of input points and using curves with prime order or cofactor = 1

Fault attacks

- Fault attacks involve inducing errors in the computation of ECC operations, such as point multiplication, to reveal information about the private key
- An example is the sign change fault attack, where an attacker manipulates the sign of a point coordinate to observe changes in the output
- Countermeasures include error detection and correction, randomizing intermediate values, and physical protection against fault injection

Secure implementation practices

- Implementing ECC securely requires following best practices and guidelines to minimize the risk of vulnerabilities and attacks
- Proper implementation practices ensure that the theoretical security of ECC is maintained in real-world applications
- Developers should adhere to these practices when implementing ECC in cryptographic libraries, protocols, and systems

Proper domain parameters

- Use standardized and well-vetted domain parameters for ECC, such as those recommended by NIST, Brainpool, or SafeCurves
- Avoid using custom or untested curve parameters that may have weaknesses or backdoors
- Ensure that the chosen curve is appropriate for the security level required by the application

Cryptographically secure RNGs

- Use cryptographically secure random number generators (CSPRNGs) for generating ECC private keys and other random values
- Avoid using weak or predictable sources of randomness, such as system time or user input
- Regularly seed and update the CSPRNG with sufficient entropy to maintain its security properties

Constant-time implementations

- Implement ECC operations in constant time to prevent timing attacks
- Avoid conditional branches, variable-time arithmetic operations, and other timing dependencies on secret data
- Use constant-time comparison functions and masking techniques to ensure uniform execution time

Validating input points

- Always validate input points received from untrusted sources to prevent invalid curve, small subgroup, and other attacks
- Check that the input points lie on the correct elliptic curve and have the expected order
- Reject or throw an exception for invalid input points to prevent further processing

Standardized ECC protocols

- Several standardized protocols have been developed to facilitate the use of ECC in various cryptographic applications
- These protocols provide a consistent and interoperable way to perform common ECC operations, such as digital signatures and key exchange
- Using standardized protocols helps ensure the security and compatibility of ECC implementations across different systems and platforms

ECDSA for digital signatures

- The Elliptic Curve Digital Signature Algorithm (ECDSA) is a widely used protocol for creating and verifying digital signatures using ECC
- ECDSA signatures provide integrity, authentication, and non-repudiation for messages or data
- The protocol involves key generation, signature creation, and signature verification steps, all based on ECC operations

ECDH for key exchange

- Elliptic Curve Diffie-Hellman (ECDH) is a key agreement protocol that allows two parties to establish a shared secret key over an insecure channel
- ECDH is based on the Diffie-Hellman key exchange protocol but uses ECC operations instead of modular exponentiation
- The shared secret key can be used for symmetric encryption, message authentication, or other cryptographic purposes

EdDSA vs ECDSA

- Edwards-curve Digital Signature Algorithm (EdDSA) is an alternative to ECDSA that uses twisted Edwards curves and provides better performance and security properties
- EdDSA has simpler and more secure implementations compared to ECDSA, reducing the risk of side-channel attacks and implementation errors
- Examples of EdDSA include Ed25519 and Ed448, which are gaining popularity in various applications and protocols (Signal, Tor)

ECC in real-world applications

- ECC has been widely adopted in various real-world applications due to its strong security, efficient performance, and small key sizes
- The use of ECC in these applications ensures the confidentiality, integrity, and authenticity of sensitive data and communications
- As the need for strong encryption grows, ECC continues to play a crucial role in securing digital systems and infrastructure

Use in TLS/SSL

- ECC is supported in the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols for secure web communications
- ECC cipher suites in TLS/SSL provide forward secrecy and efficient key exchange, enhancing the security of HTTPS websites
- Examples of ECC cipher suites include ECDHE-ECDSA-AES128-GCM-SHA256 and ECDHE-RSA-AES256-GCM-SHA384

Cryptocurrency wallet security

- ECC is the foundation for securing private keys in many cryptocurrency wallets, such as Bitcoin and Ethereum
- Users' private keys are used to sign transactions and prove ownership of funds, making ECC essential for preventing unauthorized access and theft
- Hardware wallets often use ECC to generate and store private keys securely, providing an additional layer of protection

Government and military usage

- Government and military organizations use ECC for secure communications, data protection, and authentication
- The US National Security Agency (NSA) has recommended the use of ECC for both unclassified and classified information
- ECC is used in various government-issued smart cards, identity documents, and secure communication devices

Quantum computing impact

- The advent of quantum computing poses a significant threat to the security of many cryptographic systems, including ECC
- Quantum computers, with their ability to perform certain computations exponentially faster than classical computers, could break ECC and other public-key cryptosystems
- Understanding the impact of quantum computing on ECC is crucial for planning and transitioning to post-quantum secure alternatives

Shor's algorithm threat

- Shor's algorithm is a quantum algorithm that can efficiently solve the discrete logarithm problem and integer factorization
- If a sufficiently powerful quantum computer is built, Shor's algorithm could break ECC and RSA in polynomial time
- The existence of Shor's algorithm necessitates the development and adoption of quantum-resistant cryptographic schemes

Doubling of key sizes

- One short-term mitigation strategy against quantum attacks is to double the key sizes used in ECC
- Doubling the key size increases the complexity of the ECDLP, making it harder for quantum computers to solve
- For example, using a 512-bit ECC key instead of a 256-bit key provides a higher level of quantum resistance

Post-quantum alternatives to ECC

- Post-quantum cryptography (PQC) refers to cryptographic algorithms that are believed to be secure against quantum computer attacks
- Examples of PQC algorithms include lattice-based (NTRU, LWE), code-based (McEliece), and multivariate (Rainbow) cryptosystems
- Researchers and standardization bodies are working on evaluating and standardizing PQC algorithms to replace ECC and other vulnerable schemes in the future

3.6 Quantum-resistant elliptic curve cryptography

Quantum-resistant elliptic curve cryptography is a cutting-edge field addressing the looming threat of quantum computers to current encryption methods. It explores new mathematical structures and algorithms that can withstand attacks from both classical and quantum computers.

This topic delves into specific quantum-resistant curves, like supersingular isogeny-based cryptography, and their implementations. It covers security assumptions, standardization efforts, and the challenges of integrating these new methods into existing systems.

Quantum computing threats

- Quantum computers leverage principles of quantum mechanics to perform certain computations exponentially faster than classical computers
- Rapid development in quantum computing poses significant threats to the security of widely used cryptographic systems, including those based on elliptic curve cryptography (ECC)

Shor's algorithm

- Quantum algorithm developed by Peter Shor in 1994 that can efficiently solve the integer factorization and discrete logarithm problems
- Ability to break the security of cryptographic schemes that rely on the hardness of these problems, such as RSA and classical ECC
- Requires a sufficiently large and stable quantum computer to be practically implemented, but the threat is considered serious enough to warrant the development of quantum-resistant alternatives

Impact on ECC security

- Classical ECC relies on the elliptic curve discrete logarithm problem (ECDLP), which is believed to be hard for classical computers

- Shor's algorithm can solve the ECDLP in polynomial time on a quantum computer, rendering classical ECC insecure
- Key sizes used in classical ECC (256-384 bits) would be vulnerable to quantum attacks, necessitating a transition to quantum-resistant alternatives

Quantum-resistant curves

- Specific classes of elliptic curves and related mathematical structures that are believed to be resistant to attacks by quantum computers
- Rely on different hard problems that are not known to be efficiently solvable by quantum algorithms like Shor's algorithm
- Examples include supersingular isogeny-based cryptography and curves over non-commutative algebraic structures

Supersingular isogeny key exchange

- Key exchange protocol based on the difficulty of finding isogenies between supersingular elliptic curves
- Utilizes the commutativity of the endomorphism ring of supersingular elliptic curves to establish a shared secret between two parties
- Proposed by Jao and De Feo in 2011 as a quantum-resistant alternative to classical Diffie-Hellman key exchange

Supersingular isogeny Diffie-Hellman

- Variant of the supersingular isogeny key exchange that more closely resembles the classical Diffie-Hellman protocol
- Parties exchange public keys derived from their respective secret isogenies and compute the shared secret using the other party's public key and their own secret isogeny
- Offers forward secrecy and can be used as a drop-in replacement for classical Diffie-Hellman in various protocols

Commutative supersingular isogeny Diffie-Hellman

- Extension of the supersingular isogeny Diffie-Hellman protocol that leverages the commutativity of the endomorphism ring
- Enables the construction of more complex key exchange and encryption schemes, such as tripartite Diffie-Hellman and public-key encryption
- Provides additional flexibility and functionality compared to the basic supersingular isogeny Diffie-Hellman protocol

Security of quantum-resistant ECC

- Security of quantum-resistant ECC schemes relies on the hardness of specific mathematical problems that are not known to be efficiently solvable by quantum computers
- Extensive research is being conducted to assess the security of these schemes and identify potential vulnerabilities

Difficulty of isogeny problems

- Supersingular isogeny-based cryptography relies on the hardness of finding isogenies between supersingular elliptic curves
- Two main problems: the supersingular isogeny problem (SIP) and the computational supersingular isogeny Diffie-Hellman (CSIDH) problem
- Best known quantum algorithms for these problems have exponential complexity, providing a strong basis for the security of isogeny-based schemes

Quantum security assumptions

- Security proofs for quantum-resistant ECC schemes often rely on specific assumptions about the capabilities of quantum adversaries
- Examples include the quantum random oracle model (QROM) and the quantum generic group model (QGGM)
- These assumptions help to analyze the security of schemes in the presence of quantum adversaries and guide the selection of appropriate parameters

Side-channel attack resistance

- Side-channel attacks exploit information leakage from the physical implementation of cryptographic algorithms to recover secret keys or sensitive data
- Quantum-resistant ECC schemes must be carefully implemented to minimize the risk of side-channel attacks
- Techniques such as constant-time implementations, randomization, and masking can help to mitigate side-channel vulnerabilities

Implementations of quantum-resistant ECC

- Practical implementations of quantum-resistant ECC schemes are essential for their adoption and deployment in real-world systems
- Efficient algorithms and optimized implementations are being developed to ensure the feasibility of these schemes

Efficient algorithms

- Researchers are developing efficient algorithms for key generation, key exchange, encryption, and decryption in quantum-resistant ECC schemes
- Optimizations include improved isogeny computation techniques, faster arithmetic in finite fields, and efficient representations of elliptic curve points
- These algorithms aim to reduce the computational overhead and improve the performance of quantum-resistant ECC implementations

Key sizes vs classical ECC

- Quantum-resistant ECC schemes typically require larger key sizes compared to classical ECC to achieve equivalent security levels

- For example, supersingular isogeny-based schemes may require key sizes of 2048-4096 bits to match the security of 256-bit classical ECC
- Larger key sizes impact storage requirements and communication bandwidth, but the increased security against quantum attacks is considered a worthwhile trade-off

Performance comparisons

- Performance of quantum-resistant ECC schemes is actively being evaluated and compared to classical ECC and other post-quantum cryptographic alternatives
- Factors considered include computational efficiency, memory usage, and communication overhead
- Results indicate that quantum-resistant ECC schemes can achieve practical performance levels, although they may be slower than classical ECC in some scenarios

Standardization efforts

- Standardization of quantum-resistant ECC schemes is crucial for ensuring interoperability, security, and wide-scale adoption
- Several organizations and initiatives are working towards the standardization of quantum-resistant cryptography, including ECC-based schemes

NIST post-quantum cryptography competition

- The National Institute of Standards and Technology (NIST) is conducting a multi-round competition to select and standardize post-quantum cryptographic algorithms
- Several quantum-resistant ECC schemes, such as SIKE (Supersingular Isogeny Key Encapsulation), have been submitted and are being evaluated
- The competition aims to identify secure and efficient algorithms suitable for standardization and wide-scale deployment

IEEE standards

- The Institute of Electrical and Electronics Engineers (IEEE) is developing standards for quantum-resistant cryptography, including ECC-based schemes
- IEEE P1363.3 standard focuses on identity-based public-key cryptography, including supersingular isogeny-based protocols
- These standards provide guidelines for the implementation and use of quantum-resistant ECC schemes in various applications

Integration in protocols

- Quantum-resistant ECC schemes are being integrated into existing and new cryptographic protocols to ensure their security against quantum attacks
- Examples include post-quantum versions of TLS (Transport Layer Security), IKE (Internet Key Exchange), and VPN (Virtual Private Network) protocols
- Integration efforts aim to facilitate the smooth transition from classical to quantum-resistant cryptography while maintaining compatibility with existing infrastructure

Unit 4 – Elliptic curves and number theory

4.1 Elliptic curves over the rational numbers

Elliptic curves over rational numbers are a fascinating area of study in number theory. They combine algebraic geometry with arithmetic, offering insights into fundamental mathematical problems.

These curves have a rich structure, forming groups under addition. Understanding their rational points, integral points, and isogenies is crucial for applications in cryptography and solving Diophantine equations.

Definition of elliptic curves

- Elliptic curves are a fundamental object of study in number theory and algebraic geometry
- They have a rich structure and connections to various branches of mathematics
- Understanding their definition is crucial for studying their properties and applications

Weierstrass equation

- An elliptic curve over a field K can be defined by a Weierstrass equation of the form $y^2 = x^3 + Ax + B$, where $A, B \in K$
- The coefficients A and B must satisfy the condition that the discriminant $\Delta = -16(4A^3 + 27B^2)$ is nonzero to ensure the curve is smooth
- The Weierstrass equation can be transformed into a shorter form $y^2 = x^3 - 27c_4x - 54c_6$ using a change of variables

Smooth projective curves

- Elliptic curves are smooth projective curves of genus one with a specified base point
- They can be viewed as the set of solutions to a cubic equation in the projective plane $\mathbb{P}^2$
- The smoothness condition means that the curve has no cusps or self-intersections
- The genus one condition implies that the curve has a unique point at infinity, which serves as the identity element for the group law

Discriminant and j-invariant

- The discriminant Δ of an elliptic curve is a quantity that measures how singular the curve is
- If $\Delta \neq 0$, the curve is smooth and has no singular points
- The j-invariant $j(E) = c_4^3/\Delta$ is an important invariant of an elliptic curve that characterizes its isomorphism class
- Two elliptic curves over a field K are isomorphic if and only if they have the same j-invariant

Rational points on elliptic curves

- Rational points on an elliptic curve are points whose coordinates are rational numbers
- They form a group under a natural group law, which gives elliptic curves a rich algebraic structure

- Studying rational points is a central problem in arithmetic geometry and has applications in cryptography

Definition and examples

- A rational point on an elliptic curve E defined over $\mathbb{Q}$ is a point $(x, y) \in E$ such that $x, y \in \mathbb{Q}$
- The set of rational points on E is denoted by $E(\mathbb{Q})$
- Examples of rational points on the elliptic curve $y^2 = x^3 - x$ include $(0, 0)$, $(1, 0)$, and $(-1, 0)$

Group law

- The set of rational points $E(\mathbb{Q})$ forms an abelian group under a natural group law
- The group law is defined geometrically by the chord-and-tangent process
- Given two points P and Q on E , the sum $P + Q$ is defined as the reflection of the third intersection point of the line through P and Q with E
- The identity element is the point at infinity, denoted by $\mathcal{O}$

Geometric interpretation

- The group law on an elliptic curve has a beautiful geometric interpretation
- Adding two points P and Q can be visualized by drawing a line through P and Q , finding the third intersection point with the curve, and reflecting it across the x-axis
- If $P = Q$, the line is taken to be the tangent line at P , and the reflection of the double intersection point is the result
- This geometric description allows for a visual understanding of the group structure

Algebraic formulas

- The group law on an elliptic curve can also be described algebraically using explicit formulas
- Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ be two points on an elliptic curve E given by $y^2 = x^3 + Ax + B$
- If $P \neq Q$, then the sum $P + Q = (x_3, y_3)$ is given by:
 - $x_3 = \lambda^2 - x_1 - x_2$
 - $y_3 = \lambda(x_1 - x_3) - y_1$
 - where $\lambda = (y_2 - y_1)/(x_2 - x_1)$
- If $P = Q$, then the double $2P = (x_3, y_3)$ is given by:
 - $x_3 = \lambda^2 - 2x_1$
 - $y_3 = \lambda(x_1 - x_3) - y_1$
 - where $\lambda = (3x_1^2 + A)/(2y_1)$

Mordell-Weil theorem

- The Mordell-Weil theorem is a fundamental result in the theory of elliptic curves

- It describes the structure of the group of rational points on an elliptic curve over a number field
- The theorem has important consequences for understanding the arithmetic of elliptic curves

Statement and consequences

- The Mordell-Weil theorem states that for an elliptic curve E over a number field K , the group $E(K)$ of K -rational points is finitely generated
- This means that $E(K)$ is isomorphic to $\mathbb{Z}^r \oplus E(K)_{\text{tors}}$, where r is a non-negative integer called the rank of E over K , and $E(K)_{\text{tors}}$ is the torsion subgroup of $E(K)$
- The theorem implies that there are only finitely many torsion points and that the rank is a measure of the "size" of the group of rational points
- It also provides a basis for studying the arithmetic of elliptic curves and their rational points

Rank and torsion

- The rank r of an elliptic curve E over a number field K is the number of independent points of infinite order in $E(K)$
- Determining the rank is a difficult problem, and there is no general algorithm known for computing it
- The torsion subgroup $E(K)_{\text{tors}}$ consists of the points of finite order in $E(K)$
- The possible torsion subgroups over $\mathbb{Q}$ are known and classified by Mazur's theorem
- Understanding the rank and torsion of an elliptic curve provides insights into its structure and properties

Examples and computations

- Consider the elliptic curve $E : y^2 = x^3 - x$ over $\mathbb{Q}$
- The torsion subgroup $E(\mathbb{Q})_{\text{tors}}$ consists of the points $\mathcal{O}$, $(0, 0)$, $(1, 0)$, and $(-1, 0)$, forming a group isomorphic to $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$
- The rank of E over $\mathbb{Q}$ is 0, so $E(\mathbb{Q}) \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$
- Another example is the elliptic curve $E : y^2 = x^3 - 4x$, which has rank 1 over $\mathbb{Q}$ and torsion subgroup isomorphic to $\mathbb{Z}/2\mathbb{Z}$
- Computing the rank and torsion of an elliptic curve often involves a combination of algebraic and analytic techniques, such as descent methods and L-functions

Integral points on elliptic curves

- Integral points on an elliptic curve are points whose coordinates are integers
- Studying integral points is a natural question in Diophantine geometry and has connections to other problems in number theory
- Several important theorems and results are known about the structure and finiteness of integral points

Nagell-Lutz theorem

- The Nagell-Lutz theorem provides a criterion for determining the torsion points on an elliptic curve over $\mathbb{Q}$ with integral coefficients
- It states that if $E : y^2 = x^3 + Ax + B$ with $A, B \in \mathbb{Z}$, then any torsion point $(x, y) \in E(\mathbb{Q})_{\text{tors}}$ satisfies either $y = 0$ or y^2 divides the discriminant Δ
- This theorem gives a practical way to find all the torsion points on an elliptic curve and helps in understanding the torsion subgroup

Siegel's theorem

- Siegel's theorem is a fundamental result about the finiteness of integral points on curves of genus at least one
- For an elliptic curve E over $\mathbb{Q}$, Siegel's theorem implies that the set of integral points $E(\mathbb{Z})$ is finite
- The proof of Siegel's theorem is non-effective, meaning it does not provide an explicit bound on the size of the integral points
- Effective versions of Siegel's theorem have been proved for specific classes of elliptic curves, such as those with complex multiplication

Elliptic logarithms

- Elliptic logarithms are a tool for studying integral points on elliptic curves
- The elliptic logarithm is a function that maps points on an elliptic curve to a complex number, analogous to the natural logarithm for real numbers
- It satisfies a group homomorphism property and can be used to derive bounds on the size of integral points
- Elliptic logarithms play a role in the proof of Siegel's theorem and in the study of linear forms in elliptic logarithms, which has applications to Diophantine equations

Elliptic curves over finite fields

- Elliptic curves can also be studied over finite fields, where they exhibit interesting properties and have important applications
- The theory of elliptic curves over finite fields is a rich area of research with connections to number theory, algebraic geometry, and cryptography
- Several key results and algorithms are known for elliptic curves over finite fields

Hasse's theorem

- Hasse's theorem, also known as the Hasse-Weil bound, gives an estimate for the number of points on an elliptic curve over a finite field
- It states that for an elliptic curve E over a finite field $\mathbb{F}_q$ of characteristic p , the number of $\mathbb{F}_q$ -rational points $\#E(\mathbb{F}_q)$ satisfies the inequality $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- This theorem provides a tight bound on the number of points and has implications for the structure and properties of elliptic curves over finite fields

Supersingular vs ordinary curves

- Elliptic curves over finite fields can be classified into two types: supersingular and ordinary curves
- An elliptic curve E over a finite field of characteristic p is called supersingular if p divides the trace of Frobenius $a_p = p+1-\#E(\mathbb{F}_p)$, and ordinary otherwise
- Supersingular curves have special properties and are of interest in cryptography and the theory of modular forms
- Ordinary curves are more common and have a simpler structure, making them suitable for cryptographic applications

Schoof's algorithm

- Schoof's algorithm is a polynomial-time algorithm for counting the number of points on an elliptic curve over a finite field
- It uses the action of the Frobenius endomorphism on the ℓ -torsion points of the curve for various small primes ℓ to determine the trace of Frobenius modulo ℓ
- By combining the information modulo several primes, Schoof's algorithm computes the exact number of points on the curve
- The algorithm has a running time of $O(\log^8 q)$ for a curve over $\mathbb{F}_q$ and has been further improved by various optimizations and variants (SEA, AGM, etc.)

Rational isogenies

- Isogenies are a fundamental concept in the study of elliptic curves and their relationships
- A rational isogeny between two elliptic curves is a non-constant morphism that preserves the group structure
- Isogenies provide a way to relate different elliptic curves and have applications in cryptography and the theory of modular curves

Definition and examples

- An isogeny between two elliptic curves E_1 and E_2 over a field K is a non-constant rational map $\phi : E_1 \rightarrow E_2$ that is also a group homomorphism
- The degree of an isogeny is the degree of the corresponding rational map
- Examples of isogenies include multiplication-by- n maps, which are isogenies from an elliptic curve to itself, and the Frobenius endomorphism over finite fields
- Isogenies can be classified into separable and inseparable isogenies based on the separability of the corresponding function field extension

Vélu's formulas

- Vélu's formulas provide explicit equations for computing isogenies between elliptic curves
- Given an elliptic curve E and a finite subgroup G of E , Vélu's formulas describe the equation of the quotient curve E/G and the isogeny $\phi : E \rightarrow E/G$

- The formulas involve the coordinates of the points in G and the coefficients of the curve E
- Vélu's formulas are used in the computation of isogenies and the construction of isogeny graphs

Isogeny graphs

- Isogeny graphs are a way to visualize the relationships between elliptic curves through isogenies
- The vertices of an isogeny graph represent elliptic curves (up to isomorphism), and the edges represent isogenies between them
- The degree of an isogeny is often attached as a label to the corresponding edge
- Isogeny graphs have a rich structure and are studied in the context of modular curves and the moduli space of elliptic curves
- They also have applications in cryptography, such as in the construction of hash functions and the analysis of isogeny-based cryptographic protocols

Isogeny-based cryptography

- Isogeny-based cryptography is a relatively new area that uses isogenies between elliptic curves for constructing cryptographic protocols
- The security of these protocols relies on the difficulty of computing isogenies between elliptic curves and the hardness of the isogeny problem
- Examples of isogeny-based cryptographic protocols include the supersingular isogeny Diffie-Hellman (SIDH) key exchange and the supersingular isogeny hash function
- Isogeny-based cryptography is believed to be resistant to attacks by quantum computers and is a candidate for post-quantum cryptography

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a modern public-key cryptography approach that uses the algebraic structure of elliptic curves over finite fields
- ECC provides similar security levels to traditional cryptosystems (like RSA) with smaller key sizes, making it efficient for use in constrained environments
- The security of ECC relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)

Diffie-Hellman key exchange

- The Diffie-Hellman key exchange protocol can be adapted to use elliptic curves, resulting in the elliptic curve Diffie-Hellman (ECDH) key exchange
- In ECDH, two parties agree on an elliptic curve E over a finite field and a base point $P \in E$ of large order
- Each party chooses a secret integer a and b , respectively, and computes the public points aP and bP
- The shared secret is then computed as abP , which can be used to derive a symmetric encryption key
- The security of ECDH relies on the difficulty of computing the secret integers a and b given the public points aP and bP (the ECDLP)

Elliptic curve digital signature algorithm (ECDSA)

- The elliptic curve digital signature algorithm (ECDSA) is a variant of the digital signature algorithm (DSA) that uses elliptic curves
- In ECDSA, the signer has a private key d and a corresponding public key $Q = dP$, where P is a base point on an agreed-upon elliptic curve
- To sign a message m , the signer chooses a random integer k and computes the point $kP = (x, y)$
- The signature consists of two components: (r, s)

4.2 Mordell-Weil theorem

The Mordell-Weil theorem is a cornerstone of elliptic curve theory. It states that the group of rational points on an elliptic curve over a number field is finitely generated. This powerful result connects algebra, geometry, and number theory.

Understanding the Mordell-Weil theorem is crucial for studying elliptic curves' structure and applications. It impacts areas like Diophantine equations, cryptography, and number theory, providing a framework for exploring rational points on these fascinating mathematical objects.

Statement of Mordell-Weil theorem

- Fundamental theorem in the study of elliptic curves over global fields such as the rational numbers $\mathbb{Q}$ or number fields
- Asserts that the group of rational points on an elliptic curve, denoted $E(\mathbb{Q})$ or more generally $E(K)$ for a number field K , is finitely generated
- Has important implications for understanding the structure and properties of elliptic curves and their rational points

For elliptic curves over $\mathbb{Q}$

- When E is defined over the rational numbers $\mathbb{Q}$, the Mordell-Weil theorem states that $E(\mathbb{Q})$ is a finitely generated abelian group
- This means $E(\mathbb{Q})$ is isomorphic to $\mathbb{Z}^r \oplus E(\mathbb{Q})_{\text{tors}}$, where r is a non-negative integer called the rank and $E(\mathbb{Q})_{\text{tors}}$ is the torsion subgroup
- The torsion subgroup $E(\mathbb{Q})_{\text{tors}}$ is finite and consists of all points of finite order, while the rank r measures the "size" of the infinite part of $E(\mathbb{Q})$

For elliptic curves over number fields

- The Mordell-Weil theorem extends to elliptic curves defined over arbitrary number fields K
- In this case, $E(K)$ is also a finitely generated abelian group isomorphic to $\mathbb{Z}^r \oplus E(K)_{\text{tors}}$
- The rank r and the structure of the torsion subgroup $E(K)_{\text{tors}}$ depend on the number field K and can be more complex than in the case of $\mathbb{Q}$

Finitely generated abelian groups

- Algebraic structures that play a central role in the statement and proof of the Mordell-Weil theorem

- Can be decomposed into a direct sum of a free abelian group of finite rank and a finite torsion subgroup
- Understanding their properties is crucial for studying the structure of elliptic curve groups

Definition and properties

- An abelian group G is finitely generated if there exists a finite set $\{g_1, \dots, g_n\} \subseteq G$ such that every element of G can be expressed as a linear combination of the g_i with integer coefficients
- The rank of a finitely generated abelian group is the size of a maximal linearly independent subset, or equivalently, the dimension of the tensor product $G \otimes_{\mathbb{Z}} \mathbb{Q}$ as a $\mathbb{Q}$ -vector space
- Every finitely generated abelian group G is isomorphic to a direct sum of the form $\mathbb{Z}^r \oplus G_{\text{tors}}$, where r is the rank and G_{tors} is the torsion subgroup

Torsion subgroup

- The torsion subgroup of a finitely generated abelian group G is the subgroup consisting of all elements of finite order
- It is a finite abelian group and can be decomposed into a direct sum of cyclic groups of prime power order
- For elliptic curves over $\mathbb{Q}$, the possible structures of the torsion subgroup are classified by Mazur's theorem, which lists 15 possible isomorphism classes

Rank of group

- The rank of a finitely generated abelian group G measures the size of the free part of G
- It is a non-negative integer and can be defined as the dimension of the $\mathbb{Q}$ -vector space $G \otimes_{\mathbb{Z}} \mathbb{Q}$
- For elliptic curves, the rank is a key invariant that is often difficult to compute and is related to important conjectures like the Birch and Swinnerton-Dyer conjecture

Proof of Mordell-Weil theorem

- The proof of the Mordell-Weil theorem is a deep and technical result that combines ideas from algebraic geometry, number theory, and group theory
- It proceeds in two main steps: first establishing the weak Mordell-Weil theorem, and then using height functions to prove the full statement
- The proof is a cornerstone of the theory of elliptic curves and has led to the development of important techniques and concepts

Weak Mordell-Weil theorem

- States that for an elliptic curve E over a number field K , the quotient group $E(K)/mE(K)$ is finite for every positive integer m
- Proved using techniques from Galois cohomology and the Kummer exact sequence
- Provides a first step towards the full Mordell-Weil theorem by showing that $E(K)$ is "almost" finitely generated

Height functions on elliptic curves

- A key tool in the proof of the Mordell-Weil theorem, used to measure the "size" or "complexity" of rational points on elliptic curves
- Can be defined using the canonical height, which is a quadratic form on $E(K)$ that behaves well under multiplication by integers
- Satisfy important properties like the Northcott property, which states that there are only finitely many points of bounded height

Descent via height functions

- A technique used to prove the full Mordell-Weil theorem by combining the weak Mordell-Weil theorem with height functions
- Involves constructing a suitable height function and using it to show that the quotient group $E(K)/mE(K)$ is not only finite but also of bounded height
- Leads to the conclusion that $E(K)$ is finitely generated by applying the Northcott property and the ascending chain condition for finitely generated abelian groups

Computing Mordell-Weil groups

- The Mordell-Weil theorem guarantees that the group of rational points on an elliptic curve is finitely generated, but computing the rank and generators can be a challenging problem
- Various methods and algorithms have been developed to compute Mordell-Weil groups, both in theory and in practice
- Computing Mordell-Weil groups is an active area of research with connections to computational number theory and cryptography

Computation of torsion subgroup

- The torsion subgroup of an elliptic curve over $\mathbb{Q}$ can be effectively computed using division polynomials and Mazur's theorem
- Over number fields, the computation of the torsion subgroup is more complex but can be achieved using techniques from algebraic number theory (Lutz-Nagell theorem)
- Efficient algorithms for computing torsion subgroups have been implemented in computer algebra systems like Sage and Magma

Computation of rank

- Computing the rank of an elliptic curve is a more difficult problem than computing the torsion subgroup
- Techniques for bounding the rank include descent methods (2-descent, 3-descent), the use of L-functions and the Birch and Swinnerton-Dyer conjecture, and the method of Selmer groups
- In practice, a combination of these methods is often used to obtain upper and lower bounds on the rank, but determining the exact rank can be challenging

Algorithms for finding generators

- Once the rank of an elliptic curve is known, finding a set of generators for the Mordell-Weil group is the next step
- Methods for finding generators include the use of height functions and the Mordell-Weil sieve, which systematically searches for points of bounded height
- The Heegner point construction can be used to find generators in some cases, particularly when the rank is equal to the analytic rank predicted by the Birch and Swinnerton-Dyer conjecture

Applications of Mordell-Weil theorem

- The Mordell-Weil theorem has numerous applications in various areas of mathematics, including Diophantine equations, cryptography, and number theory
- It provides a powerful tool for studying the structure and properties of elliptic curves and their rational points
- Many important problems and conjectures in number theory are related to the Mordell-Weil theorem and the behavior of elliptic curve groups

In Diophantine equations

- The Mordell-Weil theorem can be used to study the solvability of certain Diophantine equations, particularly those that can be related to elliptic curves
- For example, the theorem can be applied to prove the non-existence of rational solutions to certain cubic equations or to find all rational points on specific elliptic curves
- The Mordell-Weil theorem is also connected to the study of integral points on elliptic curves and the Siegel-Mahler theorem

In cryptography

- Elliptic curve cryptography (ECC) is a widely used form of public-key cryptography that relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)
- The Mordell-Weil theorem guarantees that the group of rational points on an elliptic curve used in ECC is finitely generated, which is important for the security and efficiency of the cryptosystem
- Understanding the structure and properties of Mordell-Weil groups can help in the design and analysis of ECC protocols and in the study of potential attacks

In number theory

- The Mordell-Weil theorem is closely related to many important problems and conjectures in number theory, such as the Birch and Swinnerton-Dyer conjecture and the Tate-Shafarevich group
- It provides a framework for studying the arithmetic of elliptic curves and their L-functions, which encode deep information about the distribution of rational points
- The theorem also has connections to other areas of number theory, such as modular forms, Galois representations, and the Langlands program

Examples and exercises

- To gain a deeper understanding of the Mordell-Weil theorem and its applications, it is helpful to work through concrete examples and exercises
- These can range from simple computations of torsion subgroups and ranks to more advanced problems involving Diophantine equations and cryptographic protocols
- Examples and exercises also serve to illustrate the key concepts and techniques used in the study of elliptic curves and Mordell-Weil groups

Elliptic curves with various ranks

- Consider the elliptic curve $E : y^2 = x^3 - x$ over $\mathbb{Q}$. Compute its torsion subgroup and rank, and find a set of generators for the Mordell-Weil group $E(\mathbb{Q})$
- Investigate the elliptic curve $E : y^2 = x^3 - 4x$ over $\mathbb{Q}$, which has rank 0. Prove that $E(\mathbb{Q})$ consists only of the point at infinity and the two torsion points $(0, 0)$ and $(2, 0)$
- Study the elliptic curve $E : y^2 + y = x^3 - 7x + 6$ over $\mathbb{Q}$, which has rank 3. Find a set of generators for $E(\mathbb{Q})$ using the method of descent

Computing Mordell-Weil groups step-by-step

- For the elliptic curve $E : y^2 = x^3 + 2x - 1$ over $\mathbb{Q}$, compute the torsion subgroup using division polynomials and Mazur's theorem
- Apply the method of 2-descent to find an upper bound on the rank of the elliptic curve $E : y^2 = x^3 - 5x + 4$ over $\mathbb{Q}$
- Use the Heegner point construction to find a non-torsion point on the elliptic curve $E : y^2 = x^3 - 4x$ over $\mathbb{Q}(\sqrt{-7})$, and use this point to generate a subgroup of the Mordell-Weil group

Challenge problems and open questions

- Investigate the Birch and Swinnerton-Dyer conjecture for specific elliptic curves, and compute the analytic rank using L-functions and modular symbols
- Study the Tate-Shafarevich group of an elliptic curve and its relationship to the Mordell-Weil group and the Cassels-Tate pairing
- Explore unsolved problems related to the Mordell-Weil theorem, such as the existence of elliptic curves with arbitrarily large rank or the distribution of ranks in families of elliptic curves

4.3 Elliptic curve L-functions

Elliptic curve L-functions are complex analytic functions that encode crucial arithmetic and geometric information about elliptic curves. They play a vital role in number theory and cryptography, providing insights into the structure and properties of these curves.

The study of L-functions involves analytic continuation, functional equations, and special values. The modularity theorem connects elliptic curves to modular forms, enabling powerful analytical techniques. Understanding L-functions is key to solving important problems in number theory.

Definition of elliptic curve L-functions

- Elliptic curve L-functions are complex analytic functions associated to elliptic curves defined over number fields
- They encode important arithmetic and geometric information about the elliptic curve
- Understanding the properties of L-functions is crucial for solving many problems in number theory and cryptography

Hasse-Weil L-function

- The Hasse-Weil L-function is defined as an Euler product over the primes of the number field
- For each prime p , the local factor is determined by the number of points on the elliptic curve modulo p
- It is initially defined as a Dirichlet series convergent in a half-plane
- The coefficients of the L-function are related to the trace of Frobenius on the elliptic curve

Analytic continuation

- One of the key properties of L-functions is that they admit an analytic continuation to the entire complex plane
- This means the L-function can be extended beyond its initial domain of convergence
- The analytic continuation is achieved using the functional equation and the Phragmén-Lindelöf principle
- Analytic continuation allows studying the behavior of L-functions at critical points, such as $s = 1$

Functional equation

- Elliptic curve L-functions satisfy a functional equation relating values at s and $2 - s$
- The functional equation involves a gamma factor and a root number (sign of the functional equation)
- The functional equation is a key tool in proving the analytic continuation of L-functions
- It also provides a symmetry in the distribution of zeros and special values of the L-function

Modularity theorem for elliptic curves

- The modularity theorem states that every elliptic curve over $\mathbb{Q}$ is modular
- This means the L-function of the elliptic curve coincides with the L-function of a modular form
- Modularity establishes a deep connection between elliptic curves and modular forms
- It has significant consequences for the study of L-functions and the arithmetic of elliptic curves

Taniyama-Shimura conjecture

- The Taniyama-Shimura conjecture, now a theorem, asserts the modularity of all elliptic curves over $\mathbb{Q}$
- It was first posed in the 1950s and became a central problem in number theory
- The conjecture relates two seemingly different objects: elliptic curves and modular forms

- Its proof has far-reaching consequences, including the proof of Fermat's Last Theorem

Wiles' proof

- In 1995, Andrew Wiles proved the Taniyama-Shimura conjecture for semistable elliptic curves
- The proof was completed by Taylor-Wiles for all elliptic curves over $\mathbb{Q}$
- Wiles' proof uses the Langlands-Tunnell theorem and Galois representations associated to modular forms
- The key idea is to establish a "Galois-theoretic" modularity lifting theorem using deformation theory

Consequences for L-functions

- The modularity theorem implies that the L-function of an elliptic curve over $\mathbb{Q}$ is entire
- This means it extends to an analytic function on the whole complex plane
- Modularity also provides a way to compute the coefficients of the L-function using modular symbols
- It enables the study of special values and zeros of L-functions using tools from the theory of modular forms

Special values of L-functions

- Special values of L-functions, particularly at $s = 1$, have significant arithmetic meaning
- They are related to important invariants of the elliptic curve, such as the rank and the Tate-Shafarevich group
- Conjectures like the Birch and Swinnerton-Dyer conjecture aim to understand these special values
- Computing and studying special values is a central problem in the theory of elliptic curves

Birch and Swinnerton-Dyer conjecture

- The BSD conjecture relates the rank of an elliptic curve to the order of vanishing of its L-function at $s = 1$
- It predicts that the rank equals the analytic rank (the order of the zero at $s = 1$)
- The conjecture also expresses the leading coefficient of the Taylor expansion at $s = 1$ in terms of arithmetic invariants
- These invariants include the Tate-Shafarevich group, the regulator, and the real period
- BSD is one of the most important open problems in number theory, with significant progress made in special cases

L-function at $s=1$

- The value of the L-function at $s = 1$ is of particular interest, as it encodes arithmetic information
- When the rank is zero, the L-function is non-zero at $s = 1$, and its value is related to the Tate-Shafarevich group
- In the rank zero case, the BSD conjecture predicts the precise value of the L-function at $s = 1$

- This has been proved for elliptic curves with complex multiplication and in some other special cases
- Computing the value of the L-function at $s = 1$ is a challenging problem, often relying on analytic techniques

Relation to elliptic curve rank

- The rank of an elliptic curve is the rank of its Mordell-Weil group (the group of rational points)
- It measures the number of independent infinite order points on the curve
- The BSD conjecture predicts a direct relation between the rank and the order of vanishing of the L-function at $s = 1$
- A zero of order r at $s = 1$ should correspond to a rank r elliptic curve
- Proving the BSD conjecture would provide a way to determine the rank of an elliptic curve analytically

Zeros of L-functions

- The distribution of zeros of L-functions encodes important information about the elliptic curve
- The location of zeros is related to questions about the distribution of prime numbers and the size of the Tate-Shafarevich group
- Studying the zeros of L-functions is a central problem in analytic number theory
- Conjectures like the Riemann hypothesis for elliptic curves aim to understand the distribution of zeros

Distribution of zeros

- The Generalized Riemann Hypothesis (GRH) for elliptic curve L-functions states that all non-trivial zeros lie on the line $\Re(s) = 1$
- This is an analog of the classical Riemann hypothesis for the Riemann zeta function
- GRH has significant consequences for the distribution of prime numbers and the growth of arithmetic invariants
- It implies optimal bounds for the error term in the prime number theorem for the elliptic curve
- Unconditionally, it is known that a positive proportion of zeros lie on the critical line $\Re(s) = 1/2$

Riemann hypothesis for elliptic curves

- The Riemann hypothesis for elliptic curve L-functions is a special case of the GRH
- It asserts that all non-trivial zeros of the L-function have real part equal to $1/2$
- The Riemann hypothesis has been verified numerically for many elliptic curves
- However, it remains unproved in general, even for a single elliptic curve
- Proving the Riemann hypothesis would have significant implications for the distribution of prime numbers and the size of the Tate-Shafarevich group

Consequences for elliptic curve arithmetic

- The distribution of zeros of L-functions has important consequences for the arithmetic of elliptic curves

- It controls the distribution of prime numbers p for which the elliptic curve has a given number of points modulo p
- The Riemann hypothesis implies optimal bounds for the error term in the prime number theorem for the elliptic curve
- This means the number of points on the elliptic curve modulo p is well-approximated by the prime number theorem
- Zeros of L-functions also appear in the Birch and Swinnerton-Dyer conjecture, relating the rank to the order of vanishing at $s = 1$

Twists of L-functions

- Twists of L-functions are obtained by modifying the Euler factors at a finite set of primes
- They correspond to twists of the elliptic curve by characters of the Galois group
- Studying twists of L-functions provides insight into the arithmetic of the original elliptic curve
- Different types of twists, such as quadratic twists and higher order twists, have specific arithmetic interpretations

Quadratic twists

- Quadratic twists are the most well-studied type of twist, corresponding to quadratic characters
- The quadratic twist of an elliptic curve E by a squarefree integer d is the curve E^d defined by the equation $dy^2 = x^3 + ax + b$
- The L-function of E^d is related to the L-function of E by a simple factor
- The sign of the functional equation of E^d is related to the sign of d and the sign of E
- Quadratic twists are used to study the rank and the Tate-Shafarevich group of elliptic curves

Higher order twists

- Higher order twists correspond to characters of higher order, such as cubic or quartic characters
- They are defined similarly to quadratic twists, by modifying the defining equation of the elliptic curve
- The L-functions of higher order twists are related to the L-function of the original curve by a twist factor
- The functional equation and analytic properties of the twisted L-function can be deduced from those of the original L-function
- Higher order twists are used to study the distribution of ranks and the structure of the Tate-Shafarevich group

Relation to elliptic curve isogenies

- Twists of elliptic curves are related to isogenies, which are morphisms between elliptic curves that preserve the group structure
- Quadratic twists correspond to isogenies of degree 2, while higher order twists correspond to isogenies of higher degree
- The L-functions of isogenous elliptic curves are closely related

- They have the same degree and share many analytic properties
- Studying twists and isogenies provides insight into the arithmetic and geometric structure of elliptic curves

L-functions and modular forms

- The modularity theorem establishes a deep connection between L-functions of elliptic curves and L-functions of modular forms
- It states that every elliptic curve over $\mathbb{Q}$ is modular, meaning its L-function coincides with the L-function of a modular form
- This connection allows the use of powerful tools from the theory of modular forms to study elliptic curve L-functions
- Many properties of L-functions, such as analytic continuation and functional equation, can be proved using modularity

Modularity of L-functions

- The L-function of a modular form f is defined as a Dirichlet series with coefficients given by the Fourier coefficients of f
- It shares many properties with the L-function of an elliptic curve, such as analytic continuation and functional equation
- The modularity theorem implies that the L-function of an elliptic curve over $\mathbb{Q}$ is the L-function of a modular form
- This allows the use of techniques from the theory of modular forms to study elliptic curve L-functions
- Modularity also provides a way to compute the coefficients of the L-function using modular symbols

Hecke operators and eigenforms

- Hecke operators are linear operators acting on the space of modular forms
- They are defined using the Fourier coefficients of modular forms and have important arithmetic properties
- Eigenforms are modular forms that are eigenvectors for all Hecke operators
- They play a crucial role in the theory of modular forms and have a simple L-function
- The coefficients of the L-function of an eigenform are given by its Hecke eigenvalues
- This provides a way to compute the L-function of a modular form and, by modularity, the L-function of an elliptic curve

Congruences between modular forms

- Congruences between modular forms are relations between their Fourier coefficients modulo a prime or a prime power
- They are related to congruences between the coefficients of the corresponding L-functions
- Studying congruences between modular forms provides insight into the arithmetic of elliptic curves

- It allows the use of techniques from modular forms to study the Tate-Shafarevich group and the Birch and Swinnerton-Dyer conjecture
- Congruences also play a role in the proof of the modularity theorem, through the use of Galois representations and modularity lifting theorems

Computational aspects of L-functions

- Computing values and properties of L-functions is a challenging problem, requiring a combination of analytic and algebraic techniques
- Efficient algorithms for L-function evaluation are essential for numerical investigations and applications
- Numerical computation of L-functions is used to test conjectures and guide theoretical research
- L-functions also have important applications in cryptography, such as the construction of elliptic curve cryptosystems

Efficient algorithms for L-function evaluation

- Evaluating L-functions requires computing the coefficients of the Dirichlet series and summing the series efficiently
- This can be done using techniques from analytic number theory, such as approximate functional equations and Riemann-Siegel type formulas
- For elliptic curve L-functions, modularity provides a way to compute the coefficients using modular symbols
- This leads to efficient algorithms for computing the L-function and its special values
- Efficient algorithms for L-function evaluation are implemented in computer algebra systems and specialized software packages

Numerical verification of conjectures

- Numerical computation of L-functions is used to test conjectures and provide evidence for theoretical results
- This includes verifying the Riemann hypothesis for specific elliptic curves and testing the Birch and Swinnerton-Dyer conjecture
- Numerical data can guide the search for proofs and counterexamples
- It can also suggest new conjectures and research directions
- High-precision computation of L-functions requires advanced techniques from numerical analysis and rigorous error control

Applications to cryptography

- Elliptic curve cryptography relies on the difficulty of the discrete logarithm problem on elliptic curves
- The security of elliptic curve cryptosystems depends on the distribution of prime numbers and the size of the Tate-Shafarevich group

- L-functions of elliptic curves are used to study the distribution of prime numbers and the growth of arithmetic invariants
- This provides a way to assess the security of elliptic curve cryptosystems and guide the selection of parameters
- Computational techniques for L-functions, such as efficient algorithms for evaluation and numerical verification of conjectures, are used in the analysis and design of elliptic curve cryptosystems

4.4 Birch and Swinnerton-Dyer conjecture

The Birch and Swinnerton-Dyer conjecture connects the arithmetic of elliptic curves to their L-functions. It predicts that the order of vanishing of an elliptic curve's L-function at a specific point equals the curve's rank.

This conjecture has profound implications for number theory and algebraic geometry. It links algebraic properties of elliptic curves to analytic properties of L-functions, offering insights into the structure of rational points on these curves.

Statement of the conjecture

- The Birch and Swinnerton-Dyer conjecture is a central problem in number theory that connects the arithmetic of elliptic curves to their L-functions
- It provides a deep link between the algebraic structure of elliptic curves and analytic properties of their associated L-functions
- The conjecture has far-reaching consequences and has been a driving force in the development of modern algebraic number theory

Rank of elliptic curves

- The rank of an elliptic curve E over a number field K , denoted $rank(E(K))$, is the rank of the Mordell-Weil group $E(K)$ as a finitely generated abelian group
- The rank measures the number of independent rational points of infinite order on the elliptic curve
- Determining the rank is a difficult problem and the conjecture predicts a deep connection between the rank and the order of vanishing of the L-function

L-functions of elliptic curves

- The L-function $L(E, s)$ of an elliptic curve E over a number field K is a complex analytic function that encodes arithmetic information about the curve
- It is defined as an Euler product over the primes of K , with local factors determined by the reduction type of E at each prime
- The L-function is expected to have an analytic continuation to the entire complex plane and satisfy a functional equation

Order of vanishing

- The Birch and Swinnerton-Dyer conjecture predicts that the order of vanishing of the L-function $L(E, s)$ at $s = 1$ is equal to the rank of the elliptic curve E
- In other words, the conjecture asserts that $ord_{s=1} L(E, s) = rank(E(K))$

- This connection between the analytic behavior of the L-function and the algebraic rank is a central feature of the conjecture

Tate-Shafarevich group

- The Tate-Shafarevich group, denoted $\text{Ш}(E/K)$, is a mysterious group associated to an elliptic curve E over a number field K
- It measures the failure of the local-global principle for principal homogeneous spaces of E
- The Birch and Swinnerton-Dyer conjecture predicts that the Tate-Shafarevich group is finite and its order is related to the leading coefficient of the L-function at $s = 1$

Connections to other problems

- The Birch and Swinnerton-Dyer conjecture has deep connections to several other major problems in number theory
- It provides a unifying framework that relates seemingly disparate areas of mathematics
- Proving the conjecture would have significant implications for our understanding of elliptic curves and their role in modern number theory

Congruent number problem

- The congruent number problem asks which integers can be the area of a right triangle with rational side lengths
- It can be reformulated in terms of the rank of elliptic curves of the form $y^2 = x^3 - n^2x$
- The Birch and Swinnerton-Dyer conjecture implies a criterion for determining whether a given integer is a congruent number

Fermat's Last Theorem

- Fermat's Last Theorem states that the equation $x^n + y^n = z^n$ has no non-zero integer solutions for $n > 2$
- The proof of Fermat's Last Theorem by Andrew Wiles relied on establishing the modularity of certain elliptic curves
- The Birch and Swinnerton-Dyer conjecture played a crucial role in guiding the strategy of the proof and highlighting the importance of elliptic curves

ABC conjecture

- The ABC conjecture is a deep conjecture in number theory that relates the prime factors of three integers satisfying $a + b = c$
- It has connections to the Birch and Swinnerton-Dyer conjecture through the study of elliptic curves and their L-functions
- Some special cases of the ABC conjecture have been proven using techniques inspired by the Birch and Swinnerton-Dyer conjecture

Computational evidence

- Extensive computational evidence has been gathered in support of the Birch and Swinnerton-Dyer conjecture
- Numerical calculations have verified the conjecture for a large number of elliptic curves
- However, the conjecture remains unproven in general and computational methods have limitations

Elliptic curve databases

- Large databases of elliptic curves have been compiled, such as the L-functions and Modular Forms Database (LMFDB)
- These databases contain information about the rank, L-function, and other invariants of millions of elliptic curves
- They provide a rich source of data for testing the Birch and Swinnerton-Dyer conjecture and exploring patterns and relationships

Verification for specific cases

- The Birch and Swinnerton-Dyer conjecture has been verified for many individual elliptic curves using a combination of theoretical and computational techniques
- For example, the conjecture has been proven for all elliptic curves of rank 0 and 1 over the rational numbers
- Computational methods, such as Heegner point constructions and descent calculations, have been used to confirm the conjecture in specific cases

Limitations of computational methods

- Despite the extensive computational evidence, the Birch and Swinnerton-Dyer conjecture remains unproven in general
- Computational methods are limited by the size and complexity of the elliptic curves being considered
- As the rank and the coefficients of the elliptic curves increase, the calculations become more challenging and time-consuming
- Proving the conjecture will require new theoretical insights and techniques beyond computational verification

Partial results

- While the Birch and Swinnerton-Dyer conjecture remains unproven in full generality, significant partial results have been established
- These results provide strong evidence for the conjecture and have deepened our understanding of elliptic curves and their L-functions
- The partial results have been achieved through a combination of sophisticated techniques from algebraic geometry, harmonic analysis, and representation theory

Coates-Wiles theorem

- The Coates-Wiles theorem, proven by John Coates and Andrew Wiles, establishes a special case of the Birch and Swinnerton-Dyer conjecture
- It states that if an elliptic curve E over a number field K has complex multiplication and rank 0, then the L-function $L(E, s)$ does not vanish at $s = 1$
- This result provided the first concrete evidence for the conjecture and highlighted the importance of complex multiplication in the study of elliptic curves

Gross-Zagier theorem

- The Gross-Zagier theorem, proven by Benedict Gross and Don Zagier, relates the height of Heegner points on an elliptic curve to the derivative of its L-function at $s = 1$
- It provides a formula for the canonical height of Heegner points in terms of the derivative of the L-function
- The theorem has important implications for the Birch and Swinnerton-Dyer conjecture and has been used to prove cases of the conjecture for elliptic curves of rank 1

Kolyvagin's work

- Victor Kolyvagin made significant progress on the Birch and Swinnerton-Dyer conjecture using techniques from Iwasawa theory and Euler systems
- He introduced the concept of Euler systems, which are collections of cohomology classes that satisfy certain compatibility relations
- Kolyvagin's work led to the proof of the weak Birch and Swinnerton-Dyer conjecture for elliptic curves of rank 0 and 1 over the rational numbers

Modularity theorem

- The modularity theorem, originally known as the Taniyama-Shimura conjecture and proven by Andrew Wiles and others, establishes a deep connection between elliptic curves and modular forms
- It states that every elliptic curve over the rational numbers is modular, meaning it corresponds to a modular form of a specific level and weight
- The modularity theorem has important consequences for the Birch and Swinnerton-Dyer conjecture, as it allows techniques from the theory of modular forms to be applied to the study of elliptic curves

Generalizations and variants

- The Birch and Swinnerton-Dyer conjecture has been generalized and extended in various directions
- These generalizations and variants aim to capture more intricate arithmetic properties of elliptic curves and their L-functions
- They provide a broader framework for understanding the deep connections between algebra, geometry, and analysis in the study of elliptic curves

Birch and Swinnerton-Dyer conjecture over number fields

- The original Birch and Swinnerton-Dyer conjecture was formulated for elliptic curves over the rational numbers
- It has been generalized to elliptic curves over arbitrary number fields, taking into account the more complex arithmetic and algebraic structure
- The conjecture over number fields involves the rank of the Mordell-Weil group, the order of vanishing of the L-function, and additional arithmetic invariants such as the Tate-Shafarevich group and the regulator

Equivariant Birch and Swinnerton-Dyer conjecture

- The equivariant Birch and Swinnerton-Dyer conjecture is a refinement that takes into account the action of the absolute Galois group on the elliptic curve and its invariants
- It formulates the conjecture in terms of equivariant L-functions and equivariant Selmer groups
- The equivariant version provides a more precise description of the arithmetic properties of elliptic curves and has connections to non-commutative Iwasawa theory

p-adic Birch and Swinnerton-Dyer conjecture

- The p-adic Birch and Swinnerton-Dyer conjecture is an analogue of the classical conjecture that considers p-adic L-functions instead of complex L-functions
- It relates the rank of an elliptic curve to the order of vanishing of its p-adic L-function at certain points
- The p-adic version has important applications in Iwasawa theory and the study of Selmer groups and Galois cohomology

Birch and Swinnerton-Dyer conjecture for abelian varieties

- The Birch and Swinnerton-Dyer conjecture has been generalized to abelian varieties, which are higher-dimensional analogues of elliptic curves
- The conjecture for abelian varieties involves the rank of the Mordell-Weil group, the order of vanishing of the L-function, and additional arithmetic invariants such as the Tate-Shafarevich group and the Néron-Tate height pairing
- The study of the Birch and Swinnerton-Dyer conjecture for abelian varieties has led to important developments in the theory of motives and algebraic cycles

Current status and future directions

- Despite significant progress, the Birch and Swinnerton-Dyer conjecture remains one of the most challenging open problems in mathematics
- The conjecture has far-reaching consequences and its resolution would have a profound impact on our understanding of elliptic curves and number theory
- Current research focuses on developing new techniques and insights to tackle the remaining obstacles and make further progress towards a complete proof

Consequences of the conjecture

- A proof of the Birch and Swinnerton-Dyer conjecture would have numerous consequences and applications
- It would provide a powerful tool for determining the rank of elliptic curves and understanding their arithmetic properties
- The conjecture has implications for other areas of mathematics, such as the study of Diophantine equations, Galois representations, and algebraic cycles
- Its resolution would also have practical applications in cryptography and coding theory, where elliptic curves play a crucial role

Obstacles to a complete proof

- Proving the Birch and Swinnerton-Dyer conjecture in full generality faces significant obstacles and challenges
- One major difficulty lies in understanding the structure and finiteness of the Tate-Shafarevich group
- Another challenge is relating the algebraic and analytic ranks of elliptic curves and controlling the error terms in the asymptotic formulas
- The conjecture is also closely tied to deep conjectures in Iwasawa theory and the equivariant Tamagawa number conjecture, which remain unproven

Recent progress and breakthroughs

- In recent years, there have been significant breakthroughs and progress towards the Birch and Swinnerton-Dyer conjecture
- New techniques, such as the use of Euler systems, p -adic methods, and automorphic representations, have led to important advances
- Notable results include the proof of the conjecture for elliptic curves over function fields, the establishment of the p -part of the conjecture in certain cases, and the development of new methods for bounding Selmer groups and Tate-Shafarevich groups
- The modularity theorem and its generalizations have also opened up new avenues for studying elliptic curves and their L -functions

Open questions and research directions

- Many open questions and research directions remain in the study of the Birch and Swinnerton-Dyer conjecture
- One important area of investigation is the development of new Euler systems and their applications to the conjecture
- Another active area of research is the study of p -adic methods and their potential for proving cases of the conjecture
- Generalizations and refinements of the conjecture, such as the equivariant and p -adic versions, provide fertile ground for further exploration

- The connections between the Birch and Swinnerton-Dyer conjecture and other areas of mathematics, such as arithmetic geometry, Iwasawa theory, and automorphic forms, continue to drive new discoveries and insights

4.5 Elliptic curves and Diophantine equations

Elliptic curves are smooth, projective algebraic curves with a rich structure. They're key in number theory and cryptography, defined by equations like $y^2 = x^3 + ax + b$. Their points form an abelian group under a geometric addition operation.

Diophantine equations are polynomial equations with integer coefficients, seeking integer solutions. They connect to elliptic curves, as many Diophantine problems can be reframed using elliptic curves. This link has led to breakthroughs in number theory and cryptography.

Elliptic curves

- Elliptic curves are smooth, projective algebraic curves of genus one with a specified basepoint
- They have a rich algebraic structure and are important objects of study in number theory and cryptography
- Elliptic curves can be defined over any field, including the complex numbers, the rational numbers, and finite fields

Weierstrass form

- The Weierstrass form of an elliptic curve is the equation $y^2 = x^3 + ax + b$, where a and b are constants
- Every elliptic curve can be written in Weierstrass form by a suitable change of variables
- The discriminant of the Weierstrass equation, given by $\Delta = -16(4a^3 + 27b^2)$, must be nonzero for the curve to be smooth

Elliptic curve group law

- The set of points on an elliptic curve forms an abelian group under a geometric addition operation called the group law
- The group law is defined by the following rule: three points on the curve add up to the identity element (the point at infinity) if and only if they are collinear
- The group law can be expressed algebraically using the Weierstrass equation, making it efficient to compute

Points of finite order

- A point P on an elliptic curve is said to have finite order if there exists a positive integer n such that $nP = O$, where O is the identity element (point at infinity)
- The order of a point P is the smallest positive integer n such that $nP = O$
- Points of finite order are also called torsion points

Torsion subgroup

- The set of all points of finite order on an elliptic curve forms a subgroup called the torsion subgroup
- The torsion subgroup is always finite and can be classified completely over the complex numbers (Mazur's theorem) and the rational numbers (Nagell-Lutz theorem)
- The torsion subgroup is an important invariant of an elliptic curve and can provide information about its structure and properties

Rank of elliptic curves

- The rank of an elliptic curve is the number of independent points of infinite order in its group of rational points
- The rank is a measure of the "size" of the group of rational points and is an important invariant of the curve
- Determining the rank of an elliptic curve is a difficult problem, and there is no known algorithm for computing it in general

Diophantine equations

- Diophantine equations are polynomial equations with integer coefficients for which integer solutions are sought
- They are named after the ancient Greek mathematician Diophantus of Alexandria, who studied them extensively
- Diophantine equations have a rich history and have been the subject of much research in number theory

Linear Diophantine equations

- A linear Diophantine equation is an equation of the form $ax + by = c$, where a , b , and c are integers
- The existence of solutions to a linear Diophantine equation can be determined using the Euclidean algorithm
- If solutions exist, they can be parametrized using the extended Euclidean algorithm

Pythagorean triples

- A Pythagorean triple is a set of three positive integers (a, b, c) satisfying the equation $a^2 + b^2 = c^2$
- Pythagorean triples correspond to right triangles with integer side lengths
- All Pythagorean triples can be generated from the primitive ones using scaling and permutation

Fermat's Last Theorem

- Fermat's Last Theorem states that the equation $x^n + y^n = z^n$ has no non-trivial integer solutions for $n > 2$
- The theorem was conjectured by Pierre de Fermat in 1637 but was not proved until 1995 by Andrew Wiles

- The proof of Fermat's Last Theorem relies on deep connections between elliptic curves and modular forms

Elliptic curves as Diophantine equations

- Elliptic curves can be viewed as Diophantine equations since they are defined by polynomial equations with integer coefficients
- The group of rational points on an elliptic curve corresponds to the integer solutions of its defining equation
- Many Diophantine problems, such as finding perfect powers or solving certain types of equations, can be reduced to questions about elliptic curves

Mordell-Weil Theorem

- The Mordell-Weil Theorem states that the group of rational points on an elliptic curve is finitely generated
- It was first proved by Louis Mordell in 1922 for elliptic curves over $\mathbb{Q}$ and later generalized by André Weil to elliptic curves over any number field
- The Mordell-Weil Theorem is a fundamental result in the study of elliptic curves and has many important consequences

Finitely generated abelian groups

- An abelian group is finitely generated if it can be generated by a finite set of elements
- Every finitely generated abelian group is isomorphic to a direct sum of cyclic groups
- The structure theorem for finitely generated abelian groups allows us to classify them up to isomorphism

Proof of Mordell-Weil Theorem

- The proof of the Mordell-Weil Theorem relies on the theory of heights on elliptic curves
- The height of a rational point measures its "complexity" and satisfies certain properties that allow for a descent argument
- The proof proceeds by showing that there are only finitely many rational points of bounded height, and then using the group law to generate the entire group from these points

Consequences of Mordell-Weil Theorem

- The Mordell-Weil Theorem implies that the group of rational points on an elliptic curve has a finite torsion subgroup and a free abelian part of finite rank
- The theorem allows us to study the structure of the group of rational points and to compute it in specific cases
- The Mordell-Weil Theorem has applications to many Diophantine problems, such as the congruent number problem and the study of integral points on elliptic curves

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography approach based on the algebraic structure of elliptic curves over finite fields
- ECC requires smaller key sizes than other public-key cryptography systems (such as RSA) for equivalent security, making it more efficient
- Elliptic curve cryptography is widely used in practice, including in the Bitcoin protocol and in the TLS standard

Elliptic Curve Diffie-Hellman (ECDH)

- Elliptic Curve Diffie-Hellman is a key agreement protocol that allows two parties to establish a shared secret over an insecure channel
- ECDH is based on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)
- In ECDH, the parties agree on an elliptic curve and a base point, then each party generates a private-public key pair and exchanges the public keys to compute the shared secret

Elliptic Curve Digital Signature Algorithm (ECDSA)

- The Elliptic Curve Digital Signature Algorithm is a digital signature scheme based on elliptic curve cryptography
- ECDSA is used to verify the authenticity of a message and the identity of the sender
- In ECDSA, the signer generates a private-public key pair and uses the private key to sign the message, while the verifier uses the public key to check the signature

Security of elliptic curve cryptography

- The security of elliptic curve cryptography relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP)
- The ECDLP is believed to be harder than the discrete logarithm problem in finite fields (for comparable key sizes), which is the basis for the security of other cryptographic systems like DSA
- The most efficient known algorithms for solving the ECDLP have exponential running time, making ECC secure for appropriately chosen parameters

Elliptic curves over finite fields

- Elliptic curves can be defined over finite fields $\mathbb{F}_q$, where q is a prime power
- The group of $\mathbb{F}_q$ -rational points on an elliptic curve, denoted by $E(\mathbb{F}_q)$, is a finite group
- Elliptic curves over finite fields have applications in cryptography and coding theory

Hasse's Theorem

- Hasse's Theorem gives a bound on the number of $\mathbb{F}_q$ -rational points on an elliptic curve E
- The theorem states that $|E(\mathbb{F}_q) - (q + 1)| \leq 2\sqrt{q}$
- Hasse's Theorem allows us to estimate the size of the group $E(\mathbb{F}_q)$ and to study its properties

Supersingular vs ordinary curves

- An elliptic curve over a finite field is called supersingular if its endomorphism ring is an order in a quaternion algebra, and ordinary otherwise
- Supersingular curves have special properties that make them useful in certain cryptographic applications (such as pairing-based cryptography)
- Most elliptic curves over finite fields are ordinary, and they are used in standard elliptic curve cryptography

Pairing-based cryptography

- Pairing-based cryptography is a subfield of elliptic curve cryptography that uses bilinear pairings on elliptic curves
- A bilinear pairing is a map that takes two points on an elliptic curve and returns an element of a finite field, satisfying certain properties
- Pairings can be used to construct advanced cryptographic protocols, such as identity-based encryption and short signatures

Elliptic curves in number theory

- Elliptic curves are central objects in modern number theory and have connections to many other areas of mathematics
- The study of elliptic curves over various fields (such as the rational numbers, number fields, and finite fields) leads to deep and fascinating questions
- Elliptic curves have been used to solve many long-standing problems in number theory, such as Fermat's Last Theorem and the congruent number problem

Congruent number problem

- A congruent number is a positive integer that is the area of a right triangle with rational side lengths
- The congruent number problem asks which integers are congruent numbers
- The problem can be reformulated in terms of the existence of rational points on certain elliptic curves, providing a link between geometry and arithmetic

Birch and Swinnerton-Dyer conjecture

- The Birch and Swinnerton-Dyer conjecture is one of the most important open problems in number theory and is one of the Millennium Prize Problems
- The conjecture relates the rank of an elliptic curve (the number of independent infinite-order rational points) to the behavior of its L-function at $s = 1$
- The conjecture has been proved in some special cases but remains open in general

Elliptic curves and modular forms

- There is a deep connection between elliptic curves and modular forms, which are certain analytic functions on the upper half-plane satisfying transformation properties

- The Modularity Theorem, proved by Wiles and others, states that every elliptic curve over $\mathbb{Q}$ is modular, meaning that it corresponds to a modular form
- This connection between elliptic curves and modular forms was a crucial ingredient in the proof of Fermat's Last Theorem and has led to many other important results in number theory

4.6 Elliptic curves and the ABC conjecture

Elliptic curves are smooth, projective algebraic curves of genus one with a specified base point. They form an abelian group under a defined group law and play a crucial role in number theory, cryptography, and coding theory.

The ABC conjecture connects prime factors of relatively prime integers to their sum. If proven, it would have significant consequences for various areas of mathematics, including elliptic curves and Diophantine equations.

Elliptic curves and ABC conjecture

- Elliptic curves are a fundamental object of study in number theory and algebraic geometry with deep connections to the ABC conjecture
- Understanding the properties and structure of elliptic curves is essential for grasping the significance and implications of the ABC conjecture
- The study of elliptic curves over various fields, their group law, and associated theorems provide the mathematical framework for exploring the ABC conjecture

Definition of elliptic curves

- Elliptic curves are smooth, projective algebraic curves of genus one with a specified base point
- Can be defined over any field, including the complex numbers, rational numbers, and finite fields
- Possess a rich algebraic structure, forming an abelian group under a specified group law
- Play a crucial role in various branches of mathematics, such as number theory, cryptography, and coding theory

Weierstrass equation

- Elliptic curves can be described by a Weierstrass equation of the form $y^2 = x^3 + ax + b$, where a and b are constants
- The discriminant $\Delta = -16(4a^3 + 27b^2)$ determines the singularity of the curve
- If $\Delta \neq 0$, the curve is non-singular and defines an elliptic curve
- If $\Delta = 0$, the curve is singular and not an elliptic curve
- The Weierstrass equation provides a standard form for representing and studying elliptic curves

Group law on elliptic curves

- Elliptic curves possess a group structure under a well-defined addition operation
- The group law allows for the addition of two points on the curve to obtain another point on the curve
- The group law is defined geometrically using the chord-and-tangent method

- Given two points P and Q on the curve, draw a line through them
- The line intersects the curve at a third point R
- The reflection of R about the X -axis is the sum $P + Q$
- The group law satisfies the axioms of an abelian group (associativity, identity, inverses, commutativity)

Elliptic curves over finite fields

- Elliptic curves can be defined over finite fields $\mathbb{F}_q$, where q is a prime power
- The number of points on an elliptic curve over $\mathbb{F}_q$, denoted by $\#E(\mathbb{F}_q)$, is finite
- Hasse's theorem provides bounds for $\#E(\mathbb{F}_q)$: $|q + 1 - \#E(\mathbb{F}_q)| \leq 2\sqrt{q}$
- Elliptic curves over finite fields have applications in cryptography and coding theory (error-correcting codes)

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptosystem based on the algebraic structure of elliptic curves over finite fields
- ECC relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)
- Given points P and Q on an elliptic curve, finding an integer k such that $Q = kP$ is computationally infeasible
- ECC offers similar security levels to other cryptosystems (RSA) with smaller key sizes, making it more efficient
- Used in various cryptographic protocols, such as key exchange (ECDH), digital signatures (ECDSA), and encryption

Rank of elliptic curves

- The rank of an elliptic curve E over a field K , denoted by $\text{rank}(E(K))$, is the number of independent points of infinite order in the group $E(K)$
- The rank measures the "size" of the group of rational points on the curve
- Determining the rank of an elliptic curve is a difficult problem, and there is no general algorithm for computing it
- The Birch and Swinnerton-Dyer conjecture relates the rank to the behavior of the L-function associated with the elliptic curve

Torsion points on elliptic curves

- Torsion points on an elliptic curve are points of finite order under the group law
- The set of torsion points on an elliptic curve E over a field K forms a subgroup, denoted by $E(K)_{\text{tors}}$
- The structure of the torsion subgroup is well-understood for elliptic curves over $\mathbb{Q}$
- Mazur's theorem classifies the possible torsion subgroups of elliptic curves over $\mathbb{Q}$
- There are only 15 possible torsion subgroups, with orders ranging from 1 to 12

- Torsion points play a role in the study of rational points on elliptic curves and in the Mordell-Weil theorem

Mordell-Weil theorem

- The Mordell-Weil theorem states that for an elliptic curve E over a number field K , the group of K -rational points $E(K)$ is finitely generated
- The group $E(K)$ can be decomposed as $E(K) \cong \mathbb{Z}^r \oplus E(K)_{\text{tors}}$, where r is the rank and $E(K)_{\text{tors}}$ is the torsion subgroup
- The theorem implies that the rational points on an elliptic curve can be described by a finite set of generators
- The proof of the Mordell-Weil theorem is non-constructive and does not provide an algorithm for finding the generators

Elliptic curves and Diophantine equations

- Elliptic curves are closely related to Diophantine equations, which are polynomial equations with integer coefficients and solutions
- Many Diophantine equations can be transformed into the study of rational points on elliptic curves
- The Mordell-Weil theorem implies that the set of rational solutions to certain Diophantine equations is finite
- Elliptic curves provide a powerful tool for studying and solving certain types of Diophantine equations (Fermat's equation, Catalan's equation)

ABC conjecture

- The ABC conjecture is a profound and far-reaching conjecture in number theory that connects the prime factors of three relatively prime integers to their sum
- If proven true, the ABC conjecture would have significant consequences for various areas of mathematics, including elliptic curves and Diophantine equations
- The conjecture remains unproven, but substantial progress has been made towards its resolution

Statement of ABC conjecture

- For any $\varepsilon > 0$, there exist only finitely many triples of relatively prime positive integers (a, b, c) satisfying $a + b = c$ and $\text{rad}(abc) < c^{1-\varepsilon}$, where $\text{rad}(n)$ is the radical of n
- The conjecture asserts that if the sum of two relatively prime integers a and b is equal to c , then the product of their distinct prime factors (radical) is usually not much smaller than c
- The conjecture is stated in terms of the quality $q(a, b, c) = \frac{\log c}{\log \text{rad}(abc)}$, with the conjecture implying that the quality is bounded for all but finitely many triples

Radical of an integer

- The radical of a positive integer n , denoted by $\text{rad}(n)$, is the product of the distinct prime factors of n
- For example, $\text{rad}(12) = \text{rad}(2^2 \cdot 3) = 2 \cdot 3 = 6$

- The radical measures the "smoothness" of an integer, with smaller radicals indicating a higher degree of smoothness
- The radical plays a central role in the formulation of the ABC conjecture

Quality vs size in ABC conjecture

- The ABC conjecture relates the quality $q(a, b, c) = \frac{\log c}{\log \text{rad}(abc)}$ to the size of the integers a , b , and c
- The conjecture suggests that triples (a, b, c) with high quality (i.e., c is large compared to $\text{rad}(abc)$) are rare
- Triples with quality greater than 1 are of particular interest, as they represent instances where the radical is small relative to the size of c
- The ABC conjecture effectively limits the number of high-quality triples, with only finitely many exceeding any given quality threshold

Consequences of ABC conjecture

- The ABC conjecture has numerous consequences in various areas of mathematics, particularly in number theory and Diophantine equations
- Some notable consequences include:
 - Fermat's Last Theorem for sufficiently large exponents
 - The Mordell conjecture on the finite generation of rational points on curves of genus ≥ 2
 - The Erdős-Woods conjecture on the existence of perfect powers with small gaps
 - The Wieferich prime conjecture on the non-existence of certain prime numbers
- The ABC conjecture provides a unifying framework for understanding and proving various long-standing conjectures in number theory

ABC conjecture and Fermat's last theorem

- The ABC conjecture implies Fermat's Last Theorem for sufficiently large exponents
- Fermat's Last Theorem states that the equation $x^n + y^n = z^n$ has no non-zero integer solutions for $n > 2$
- If the ABC conjecture holds, then there exists a constant N such that Fermat's Last Theorem is true for all exponents $n > N$
- The proof of Fermat's Last Theorem by Andrew Wiles does not rely on the ABC conjecture, but the conjecture provides an alternative approach to the problem

ABC conjecture and Mordell conjecture

- The ABC conjecture implies the Mordell conjecture, which states that a curve of genus ≥ 2 over a number field has only finitely many rational points
- The Mordell conjecture was proven by Gerd Faltings in 1983, but the ABC conjecture provides a more elementary proof

- The ABC conjecture can be used to bound the height of rational points on curves of genus ≥ 2 , which leads to the finite generation of rational points
- The connection between the ABC conjecture and the Mordell conjecture highlights the deep interplay between Diophantine equations and the structure of rational points on curves

Masser-Oesterlé conjecture

- The Masser-Oesterlé conjecture, also known as the *abc* conjecture, is an equivalent formulation of the ABC conjecture
- The conjecture states that for every $\varepsilon > 0$, there exists a constant C_ε such that for any relatively prime integers a , b , and c satisfying $a + b = c$, the inequality $\max(|a|, |b|, |c|) \leq C_\varepsilon \cdot \text{rad}(abc)^{1+\varepsilon}$ holds
- The Masser-Oesterlé conjecture provides a more explicit bound on the size of the integers a , b , and c in terms of the radical of their product
- The conjecture is named after David Masser and Joseph Oesterlé, who independently formulated it in the 1980s

Progress towards proving ABC conjecture

- Despite significant efforts, the ABC conjecture remains unproven
- Several partial results and special cases of the conjecture have been established
- The conjecture has been proven for certain classes of triples (a, b, c) , such as those where a , b , and c have specific divisibility properties
- Lower bounds on the quality of triples have been obtained, providing evidence in favor of the conjecture
- In 2012, Shinichi Mochizuki proposed a proof of the ABC conjecture using his theory of Inter-universal Teichmüller Theory (IUT)
- Mochizuki's proof is highly complex and relies on novel mathematical techniques that are not yet fully understood by the mathematical community
- The validity of Mochizuki's proof remains a topic of ongoing discussion and verification
- The ABC conjecture continues to be a central problem in number theory, with active research aimed at its resolution

Connections between ABC conjecture and elliptic curves

- The ABC conjecture has deep connections to the theory of elliptic curves
- Elliptic curves can be used to construct high-quality ABC triples
- For example, the Frey curve $y^2 = x(x - a)(x + b)$ associated with an ABC triple (a, b, c) has good reduction properties related to the radical of abc
- The study of elliptic curves and their reduction types can provide insights into the structure of ABC triples
- The ABC conjecture has implications for the rank and torsion subgroups of elliptic curves

- If the ABC conjecture is true, it would imply that the rank of elliptic curves over $\mathbb{Q}$ is uniformly bounded
- The conjecture also has consequences for the possible torsion subgroups of elliptic curves over number fields
- The interplay between the ABC conjecture and elliptic curves highlights the deep connections between different areas of number theory and algebraic geometry

Unit 5 – Elliptic Curves: Complex Number Analysis

5.1 Complex tori and lattices

Complex tori and lattices form the foundation for understanding elliptic curves from a geometric perspective. These structures bridge the gap between abstract algebra and complex analysis, providing a visual representation of the underlying mathematical concepts.

By studying lattices in the complex plane and their quotients, we gain insights into the periodicity of elliptic functions and the structure of elliptic curves. This connection allows us to explore isomorphisms, endomorphisms, and complex multiplication, deepening our understanding of these fascinating mathematical objects.

Lattices in the complex plane

- Lattices are discrete subgroups of the complex plane that form a regular grid-like structure
- Play a fundamental role in the study of elliptic curves and their properties as complex tori
- Provide a way to construct elliptic functions and study their periodicity

Definition of lattice

- A lattice Λ in the complex plane is a discrete subgroup of $\mathbb{C}$ generated by two linearly independent complex numbers ω_1 and ω_2
- Can be written as $\Lambda = \{m\omega_1 + n\omega_2 \mid m, n \in \mathbb{Z}\}$
- The complex numbers ω_1 and ω_2 are called the periods of the lattice
- Example: The Gaussian integers $\mathbb{Z}[i] = \{a + bi \mid a, b \in \mathbb{Z}\}$ form a lattice in $\mathbb{C}$

Basis of a lattice

- A basis of a lattice Λ is a pair of complex numbers (ω_1, ω_2) that generate the lattice
- The basis is not unique, but any two bases are related by a matrix in $\text{GL}_2(\mathbb{Z})$
- The ratio $\tau = \omega_2/\omega_1$ is called the modular parameter and determines the shape of the lattice up to scaling
- Example: The lattice $\Lambda = \{m + n\sqrt{3}i \mid m, n \in \mathbb{Z}\}$ has a basis $(1, \sqrt{3}i)$

Fundamental parallelogram

- The fundamental parallelogram of a lattice Λ with basis (ω_1, ω_2) is the set $\{s\omega_1 + t\omega_2 \mid 0 \leq s, t < 1\}$
- Provides a geometric representation of the lattice and its periodicity
- Any point in the complex plane can be uniquely written as a point in the fundamental parallelogram plus a lattice point

- The area of the fundamental parallelogram is given by $|\operatorname{Im}(\omega_1 \bar{\omega}_2)|$

Quotient of $\mathbb{C}$ by a lattice

- The quotient of the complex plane by a lattice is a key construction in the study of elliptic curves
- Identifies points in the complex plane that differ by a lattice point, resulting in a compact Riemann surface

Torus as quotient space

- The quotient space $\mathbb{C}/\Lambda$ obtained by identifying points in the complex plane that differ by a lattice point is a torus
- The torus inherits a complex structure from the complex plane, making it a compact Riemann surface
- Elliptic curves can be viewed as tori equipped with an algebraic structure

Fundamental domain

- A fundamental domain for the quotient space $\mathbb{C}/\Lambda$ is a subset of $\mathbb{C}$ that contains exactly one representative from each equivalence class
- The fundamental parallelogram is an example of a fundamental domain
- Any meromorphic function on $\mathbb{C}/\Lambda$ can be lifted to a doubly periodic meromorphic function on $\mathbb{C}$

Isomorphic lattices

- Two lattices Λ_1 and Λ_2 are isomorphic if there exists a non-zero complex number α such that $\alpha\Lambda_1 = \Lambda_2$
- Isomorphic lattices give rise to isomorphic quotient spaces and elliptic curves
- The modular parameter τ determines the isomorphism class of a lattice up to scaling

Elliptic functions

- Elliptic functions are meromorphic functions on the complex plane that are doubly periodic with respect to a lattice
- They provide a rich source of examples of meromorphic functions on compact Riemann surfaces
- Elliptic functions are closely related to elliptic curves and play a key role in their study

Doubly periodic functions

- A function $f: \mathbb{C} \rightarrow \mathbb{C}$ is doubly periodic with respect to a lattice Λ if $f(z + \omega) = f(z)$ for all $z \in \mathbb{C}$ and $\omega \in \Lambda$
- Doubly periodic functions can be viewed as meromorphic functions on the quotient space $\mathbb{C}/\Lambda$
- The periods of a doubly periodic function are the generators of the lattice Λ

Liouville's theorem

- Liouville's theorem states that any doubly periodic meromorphic function on $\mathbb{C}$ must be constant

- Implies that non-constant elliptic functions must have poles
- The number and order of the poles determine the function up to an additive constant

Weierstrass $\wp$ function

- The Weierstrass $\wp$ function is a fundamental example of an elliptic function
- Defined as $\wp(z; \Lambda) = \frac{1}{z^2} + \sum_{\omega \in \Lambda \setminus \{0\}} \left(\frac{1}{(z - \omega)^2} - \frac{1}{\omega^2} \right)$
- Has a double pole at each lattice point and satisfies the differential equation $(\wp')^2 = 4\wp^3 - g_2\wp - g_3$
- The coefficients g_2 and g_3 are invariants of the lattice and determine the isomorphism class of the associated elliptic curve

Elliptic curves as complex tori

- Elliptic curves can be viewed as complex tori equipped with an algebraic structure
- This perspective provides a geometric interpretation of elliptic curves and their properties
- The complex analytic and algebraic definitions of elliptic curves are equivalent

Embedding into projective space

- An elliptic curve $E(\mathbb{C})$ can be embedded into the complex projective plane $\mathbb{P}^2(\mathbb{C})$ using the Weierstrass $\wp$ function and its derivative
- The embedding is given by $z \mapsto [\wp(z) : \wp'(z) : 1]$ for $z \in \mathbb{C} \setminus \Lambda$
- The point at infinity is mapped to $[0 : 1 : 0]$, completing the curve in projective space

Algebraic vs analytic definition

- Algebraically, an elliptic curve is defined as a smooth projective curve of genus one with a specified base point
- Analytically, an elliptic curve is defined as the quotient space $\mathbb{C}/\Lambda$ for some lattice Λ
- The Weierstrass embedding provides an isomorphism between the algebraic and analytic definitions

Isomorphism classes

- Two elliptic curves E_1 and E_2 are isomorphic as complex tori if and only if their associated lattices Λ_1 and Λ_2 are isomorphic
- The isomorphism class of an elliptic curve is determined by the j -invariant, which is a function of the lattice invariants g_2 and g_3
- Elliptic curves with the same j -invariant are isomorphic over $\mathbb{C}$, but may not be isomorphic over other fields

Homomorphisms of complex tori

- Homomorphisms of complex tori are holomorphic maps that preserve the group structure
- They provide a way to study the relationships between different elliptic curves

- Isogenies are a particularly important class of homomorphisms

Complex analytic maps

- A complex analytic map between complex tori $\mathbb{C}/\Lambda_1$ and $\mathbb{C}/\Lambda_2$ is a holomorphic map that descends from a linear map on the complex plane
- The linear map must satisfy $\alpha\Lambda_1 \subseteq \Lambda_2$ for some complex number α
- Complex analytic maps are necessarily group homomorphisms

Isogenies

- An isogeny is a surjective homomorphism between elliptic curves that is also an algebraic morphism
- Isogenies can be viewed as complex analytic maps between the corresponding complex tori
- The degree of an isogeny is the degree of the corresponding algebraic morphism
- Example: Multiplication by n is an isogeny from an elliptic curve to itself for any integer n

Isogeny kernels and degrees

- The kernel of an isogeny $\phi : E_1 \rightarrow E_2$ is the subgroup $\ker(\phi) = \{P \in E_1 \mid \phi(P) = O\}$, where O is the identity element
- The degree of an isogeny is equal to the order of its kernel
- Isogenies of degree n correspond to index n sublattices of the associated lattice
- Example: The multiplication by n isogeny has kernel isomorphic to $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$ and degree n^2

Endomorphism rings

- The endomorphism ring of an elliptic curve E is the set of isogenies from E to itself, together with the zero map
- Endomorphism rings provide a way to study the symmetries and arithmetic properties of elliptic curves
- The structure of the endomorphism ring is related to the complex multiplication of the associated lattice

Definition and properties

- The endomorphism ring $\text{End}(E)$ is a ring under pointwise addition and composition of isogenies
- Contains the subring $\mathbb{Z}$ corresponding to multiplication by integers
- Can be viewed as a subring of the ring of algebraic integers in a number field
- The endomorphism ring is either $\mathbb{Z}$ or an order in an imaginary quadratic field

Complex multiplication

- An elliptic curve E has complex multiplication if its endomorphism ring is larger than $\mathbb{Z}$
- Equivalent to the associated lattice Λ having additional symmetries beyond scaling

- Elliptic curves with complex multiplication have special arithmetic properties and are important in number theory
- Example: The elliptic curve $y^2 = x^3 - x$ has complex multiplication by the Gaussian integers $\mathbb{Z}[i]$

Orders in imaginary quadratic fields

- If E has complex multiplication, then $\text{End}(E)$ is an order in an imaginary quadratic field $K = \mathbb{Q}(\sqrt{-d})$ for some square-free positive integer d
- The order is determined by the discriminant of the lattice associated to E
- The ring class field of the order is the smallest field over which all elliptic curves with that endomorphism ring are defined
- The Hilbert class field of K is the ring class field of its maximal order and plays a key role in the study of elliptic curves with complex multiplication

5.2 Elliptic functions and Weierstrass $\wp$ -function

Elliptic functions are complex-valued functions with two independent periods. They're crucial in studying elliptic curves, which have applications in cryptography and number theory. The Weierstrass $\wp$ -function is a key example, providing a link between complex analysis and algebraic geometry.

The $\wp$ -function has unique properties that make it fundamental in elliptic curve theory. It satisfies a specific differential equation, has a Laurent series expansion, and exhibits homogeneity. These characteristics allow us to connect abstract mathematical concepts with practical applications in various fields.

Definition of elliptic functions

- Elliptic functions are a crucial class of functions in complex analysis and algebraic geometry
- They are closely related to elliptic curves and have numerous applications in various fields of mathematics and physics
- Understanding the properties and behavior of elliptic functions is essential for studying elliptic curves and their applications

Doubly periodic functions

- Elliptic functions are doubly periodic, meaning they are periodic in two independent directions in the complex plane
- For an elliptic function $f(z)$ and two complex numbers ω_1 and ω_2 with $\omega_1/\omega_2 \notin \mathbb{R}$, $f(z + m\omega_1 + n\omega_2) = f(z)$ for all $m, n \in \mathbb{Z}$
- The periods ω_1 and ω_2 form a lattice in the complex plane

Poles and residues

- Elliptic functions have a finite number of poles in each period parallelogram
- The sum of the residues at the poles in a period parallelogram is always zero
- This property distinguishes elliptic functions from other meromorphic functions

Liouville's theorem

- Liouville's theorem states that a doubly periodic function that is holomorphic (analytic and single-valued) on the entire complex plane must be constant
- This theorem implies that non-constant elliptic functions must have poles

Fundamental parallelogram

- A fundamental parallelogram is a region in the complex plane that represents the periodicity of an elliptic function
- It is formed by the vectors ω_1 and ω_2 , which are the periods of the elliptic function
- The function's behavior within the fundamental parallelogram determines its behavior throughout the entire complex plane

Weierstrass $\wp$ -function

- The Weierstrass $\wp$ -function is a specific elliptic function that plays a central role in the theory of elliptic curves
- It is named after the German mathematician Karl Weierstrass, who introduced and studied this function extensively
- The $\wp$ -function has several important properties that make it a fundamental object in the study of elliptic functions and elliptic curves

Definition and properties

- The Weierstrass $\wp$ -function is defined as a sum over a lattice Λ in the complex plane:
$$\wp(z; \Lambda) = \frac{1}{z^2} + \sum_{\omega \in \Lambda \setminus \{0\}} \left(\frac{1}{(z - \omega)^2} - \frac{1}{\omega^2} \right)$$
- It is an even function, meaning $\wp(-z) = \wp(z)$, and has double poles at each lattice point
- The $\wp$ -function is doubly periodic with respect to the lattice Λ

Differential equation

- The Weierstrass $\wp$ -function satisfies the following differential equation:
$$(\wp'(z))^2 = 4\wp(z)^3 - g_2\wp(z) - g_3$$
- The constants g_2 and g_3 are called the invariants of the lattice Λ and determine the specific $\wp$ -function
- This differential equation is crucial for understanding the connection between $\wp$ -functions and elliptic curves

Laurent series expansion

- The Weierstrass $\wp$ -function has a Laurent series expansion around $z = 0$:

$$\wp(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} (2k+1)G_{2k+2}z^{2k}$$

- The coefficients G_{2k+2} are called Eisenstein series and are related to the invariants g_2 and g_3

- The Laurent series expansion is useful for studying the behavior of the $\wp$ -function near its poles

Homogeneity and invariance

- The Weierstrass $\wp$ -function is homogeneous of degree -2 , meaning $\wp(\lambda z; \lambda \Lambda) = \lambda^{-2} \wp(z; \Lambda)$ for any non-zero complex number λ
- It is also invariant under unimodular transformations of the lattice Λ , i.e., transformations that preserve the area of the fundamental parallelogram

Poles and residues of $\wp$ -function

- The Weierstrass $\wp$ -function has double poles at each lattice point, with a residue of zero
- The behavior of the $\wp$ -function near its poles is essential for understanding its properties and connection to elliptic curves
- The Laurent series expansion of the $\wp$ -function around a pole provides insight into its local behavior

Algebraic properties of $\wp$ -function

- The Weierstrass $\wp$ -function possesses several algebraic properties that are fundamental to its application in the theory of elliptic curves
- These properties include the addition theorem, duplication formula, and division polynomials
- Understanding these algebraic properties is crucial for studying the group structure of elliptic curves and their isogenies

Addition theorem

- The addition theorem for the Weierstrass $\wp$ -function states that for any two complex numbers z_1 and z_2 (excluding poles), $\wp(z_1 + z_2) = -\wp(z_1) - \wp(z_2) + \frac{1}{4} \left(\frac{\wp'(z_1) - \wp'(z_2)}{\wp(z_1) - \wp(z_2)} \right)^2$
- This theorem allows the addition of points on an elliptic curve to be expressed in terms of the $\wp$ -function and its derivative
- The addition theorem is a key ingredient in the group law of elliptic curves

Duplication formula

- The duplication formula for the Weierstrass $\wp$ -function is a special case of the addition theorem when $z_1 = z_2$: $\wp(2z) = -2\wp(z) + \frac{1}{4} \left(\frac{\wp''(z)}{\wp'(z)} \right)^2$
- This formula is useful for studying the doubling of points on an elliptic curve and has applications in elliptic curve cryptography

Division polynomials

- Division polynomials are a sequence of polynomials $\psi_n(x, y)$ that are related to the division of points on an elliptic curve
- They can be defined using the Weierstrass $\wp$ -function and its derivatives
- Division polynomials play a crucial role in the study of torsion points and isogenies of elliptic curves

Relation to elliptic curves

- The Weierstrass $\wp$ -function and its derivative $\wp'$ provide a parameterization of elliptic curves
- An elliptic curve in Weierstrass form can be written as $y^2 = 4x^3 - g_2x - g_3$, where g_2 and g_3 are the invariants of the associated lattice
- The map $(z, \Lambda) \mapsto (\wp(z; \Lambda), \wp'(z; \Lambda))$ establishes an isomorphism between the complex torus $\mathbb{C}/\Lambda$ and the elliptic curve

Eisenstein series and $\wp$ -function

- Eisenstein series are a family of modular forms that are closely related to the Weierstrass $\wp$ -function
- They play a crucial role in the study of elliptic functions and modular forms
- Understanding the connection between Eisenstein series and the $\wp$ -function provides insight into the modularity properties of elliptic curves

Definition of Eisenstein series

- The Eisenstein series $G_{2k}(\Lambda)$ of weight $2k$ for a lattice Λ is defined as $G_{2k}(\Lambda) = \sum_{\omega \in \Lambda \setminus \{0\}} \omega^{-2k}$
- These series converge absolutely for $k \geq 2$ and are modular forms of weight $2k$ for the modular group $SL_2(\mathbb{Z})$
- The Eisenstein series G_4 and G_6 are particularly important, as they are related to the invariants g_2 and g_3 of the Weierstrass $\wp$ -function

Connection to $\wp$ -function

- The invariants g_2 and g_3 of the Weierstrass $\wp$ -function can be expressed in terms of the Eisenstein series: $g_2 = 60G_4$ and $g_3 = 140G_6$
- The Laurent series expansion of the $\wp$ -function involves the Eisenstein series as coefficients:

$$\wp(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} (2k+1)G_{2k+2}z^{2k}$$

- This connection highlights the modular properties of the $\wp$ -function and its relation to the theory of modular forms

Modularity of Eisenstein series

- The Eisenstein series G_{2k} are modular forms of weight $2k$ for the modular group $SL_2(\mathbb{Z})$
- This means that they satisfy certain transformation properties under the action of $SL_2(\mathbb{Z})$ on the upper half-plane
- The modularity of Eisenstein series is a key ingredient in the proof of the modularity theorem for elliptic curves over $\mathbb{Q}$

Lattices and $\wp$ -function

- The Weierstrass $\wp$ -function is closely related to lattices in the complex plane

- The properties of the $\wp$ -function, such as its periods and invariants, are determined by the associated lattice
- Understanding the relationship between lattices and the $\wp$ -function is essential for studying the arithmetic and geometric properties of elliptic curves

Lattice invariants g_2 and g_3

- The invariants g_2 and g_3 of a lattice Λ are defined as $g_2(\Lambda) = 60 \sum_{\omega \in \Lambda \setminus \{0\}} \omega^{-4}$ and $g_3(\Lambda) = 140 \sum_{\omega \in \Lambda \setminus \{0\}} \omega^{-6}$
- These invariants determine the Weierstrass $\wp$ -function associated with the lattice Λ up to a constant factor
- The discriminant of the lattice is given by $\Delta = g_2^3 - 27g_3^2$ and is non-zero for non-degenerate lattices

Relation between lattices and $\wp$ -function

- The Weierstrass $\wp$ -function $\wp(z; \Lambda)$ is uniquely determined by its associated lattice Λ
- The periods of the $\wp$ -function are given by the basis vectors of the lattice
- The invariants g_2 and g_3 of the $\wp$ -function are determined by the lattice invariants $g_2(\Lambda)$ and $g_3(\Lambda)$

Isomorphic lattices and $\wp$ -functions

- Two lattices Λ_1 and Λ_2 are isomorphic if there exists a non-zero complex number λ such that $\Lambda_2 = \lambda\Lambda_1$
- Isomorphic lattices give rise to homothetic $\wp$ -functions, i.e., $\wp(z; \Lambda_2) = \lambda^{-2}\wp(\lambda^{-1}z; \Lambda_1)$
- The j -invariant, defined as $j(\Lambda) = 1728 \frac{g_2^3}{\Delta}$, is an important quantity that characterizes isomorphism classes of lattices and elliptic curves

Complex tori and $\wp$ -function

- Complex tori, which are quotients of the complex plane by a lattice, are closely related to elliptic curves and the Weierstrass $\wp$ -function
- The $\wp$ -function provides a natural parameterization of complex tori and establishes an isomorphism between complex tori and elliptic curves
- Studying complex tori and their relationship to the $\wp$ -function is crucial for understanding the geometry and arithmetic of elliptic curves

Uniformization of complex tori

- Every complex torus $\mathbb{C}/\Lambda$ can be uniformized by the complex plane $\mathbb{C}$ using the Weierstrass $\wp$ -function
- The map $z \mapsto (\wp(z; \Lambda), \wp'(z; \Lambda))$ defines an isomorphism between the complex torus $\mathbb{C}/\Lambda$ and the elliptic curve $y^2 = 4x^3 - g_2(\Lambda)x - g_3(\Lambda)$
- This uniformization provides a powerful tool for studying the geometry and topology of complex tori and elliptic curves

Elliptic curves as complex tori

- Every elliptic curve over $\mathbb{C}$ can be realized as a complex torus $\mathbb{C}/\Lambda$ for some lattice Λ
- The Weierstrass $\wp$ -function associated with the lattice Λ provides a parameterization of the elliptic curve
- This perspective allows the application of complex analytic techniques to the study of elliptic curves

Isogenies and complex multiplication

- An isogeny between two elliptic curves is a surjective holomorphic map that preserves the group structure
- Isogenies can be studied using the Weierstrass $\wp$ -function and the associated lattices
- Elliptic curves with complex multiplication, i.e., those admitting an isogeny to themselves, have lattices with special arithmetic properties and are of particular interest in number theory

Applications of $\wp$ -function

- The Weierstrass $\wp$ -function has numerous applications in various areas of mathematics and physics
- Its connection to elliptic integrals, as well as its appearance in physical models and cryptographic protocols, highlights its importance and versatility
- Exploring the applications of the $\wp$ -function provides insight into the practical significance of elliptic functions and elliptic curves

Elliptic integrals and $\wp$ -function

- Elliptic integrals, such as $\int \frac{dx}{\sqrt{4x^3 - g_2x - g_3}}$, can be evaluated using the Weierstrass $\wp$ -function
- The inverse of the $\wp$ -function, called the Weierstrass elliptic function, is related to elliptic integrals and provides a way to express them in terms of elliptic functions
- This connection is important in the study of various physical problems, such as the motion of a pendulum or the dynamics of a spinning top

Elliptic functions in physics

- Elliptic functions, including the Weierstrass $\wp$ -function, appear naturally in several physical models
- They are used to describe the motion of particles in periodic potentials, such as in the study of crystal lattices or the dynamics of electrons in a solid
- Elliptic functions also play a role in the theory of integrable systems, such as the Korteweg-de Vries equation and the sine-Gordon equation

Elliptic functions in cryptography

- Elliptic curves and elliptic functions have found important applications in cryptography
- The group structure of elliptic curves over finite fields is used in the construction of public-key cryptographic protocols, such as the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Elliptic Curve Diffie-Hellman (ECDH) key exchange

- The Weierstrass $\wp$ -function and its properties are essential for understanding the arithmetic of elliptic curves and their use in cryptographic applications

5.3 Isomorphism between elliptic curves and complex tori

Elliptic curves and complex tori are deeply connected mathematical objects. By viewing elliptic curves as quotients of the complex plane by lattices, we gain insights into their geometry and topology, bridging complex analysis and algebraic geometry.

This perspective reveals the isomorphism between elliptic curves and complex tori, established through the Weierstrass $\wp$ -function. This connection allows us to study elliptic curves using complex analysis tools and explore their group structure, j -invariants, and applications in number theory.

Elliptic curves as complex tori

- Elliptic curves can be viewed as complex tori, which are quotient spaces of the complex plane by a lattice
- This perspective allows for a deeper understanding of the geometry and topology of elliptic curves
- Studying elliptic curves as complex tori reveals connections to other areas of mathematics, such as complex analysis and algebraic geometry

Lattices in the complex plane

- A lattice in the complex plane is a discrete subgroup of $\mathbb{C}$ generated by two linearly independent complex numbers ω_1 and ω_2
- The lattice $\Lambda = \{n\omega_1 + m\omega_2 : n, m \in \mathbb{Z}\}$ forms a regular grid in the complex plane
- The choice of generating vectors ω_1 and ω_2 determines the shape and orientation of the lattice (rectangular, square, hexagonal)
- The fundamental parallelogram of the lattice is the region $\{a\omega_1 + b\omega_2 : 0 \leq a, b < 1\}$, which tiles the complex plane when translated by lattice vectors

Quotient spaces and tori

- The quotient space $\mathbb{C}/\Lambda$ is obtained by identifying points in the complex plane that differ by a lattice vector
- $\mathbb{C}/\Lambda$ has the topology of a torus (donut shape) since opposite sides of the fundamental parallelogram are identified
- The complex structure on $\mathbb{C}$ induces a complex structure on the quotient space $\mathbb{C}/\Lambda$, making it a complex torus
- The periods of the lattice ω_1 and ω_2 determine the shape and size of the resulting torus

Periods of elliptic curves

- Every elliptic curve E has an associated lattice Λ_E , and the quotient space $\mathbb{C}/\Lambda_E$ is isomorphic to E as a complex manifold
- The periods of an elliptic curve are the generating vectors ω_1 and ω_2 of its associated lattice Λ_E

- The ratio $\tau = \omega_2/\omega_1$ (in the upper half-plane) uniquely determines the isomorphism class of the elliptic curve
- The j -invariant of an elliptic curve can be expressed in terms of its periods, linking the complex analytic and algebraic perspectives

Isomorphism theorem

- The isomorphism theorem states that every elliptic curve over $\mathbb{C}$ is isomorphic to a complex torus $\mathbb{C}/\Lambda$ for some lattice Λ
- This isomorphism provides a powerful tool for studying elliptic curves using complex analysis and algebraic geometry
- The isomorphism is established using the Weierstrass $\wp$ -function and its properties

Weierstrass $\wp$ -function

- The Weierstrass $\wp$ -function associated to a lattice Λ is defined as

$$\wp(z) = \frac{1}{z^2} + \sum_{\omega \in \Lambda \setminus \{0\}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right)$$
- $\wp(z)$ is an even, doubly periodic meromorphic function with poles at the lattice points
- The $\wp$ -function satisfies the differential equation $\wp'(z)^2 = 4\wp(z)^3 - g_2\wp(z) - g_3$, where g_2 and g_3 are constants depending on the lattice
- The map $\varphi : \mathbb{C}/\Lambda \rightarrow E, z \mapsto (\wp(z), \wp'(z))$ defines an isomorphism between the complex torus $\mathbb{C}/\Lambda$ and the elliptic curve $E : y^2 = 4x^3 - g_2x - g_3$

Elliptic functions and meromorphic functions

- An elliptic function is a meromorphic function on $\mathbb{C}$ that is doubly periodic with respect to a lattice Λ
- The field of elliptic functions associated to a lattice Λ is generated by the Weierstrass $\wp$ -function and its derivative $\wp'$
- Every elliptic function can be expressed as a rational function in $\wp$ and $\wp'$
- The isomorphism theorem implies that the field of meromorphic functions on an elliptic curve is isomorphic to the field of elliptic functions associated to its lattice

Isomorphism between elliptic curves and tori

- The isomorphism $\varphi : \mathbb{C}/\Lambda \rightarrow E$ establishes a one-to-one correspondence between points on the complex torus and points on the elliptic curve
- The isomorphism preserves the complex analytic structure, making it a biholomorphic map
- Under the isomorphism, the group structure on the elliptic curve (defined by the chord-tangent law) corresponds to the natural group structure on the torus (addition of complex numbers modulo the lattice)
- The isomorphism allows for the study of elliptic curves using the tools of complex analysis and the study of complex tori using algebraic geometry

Properties of the isomorphism

- The isomorphism between elliptic curves and complex tori has several important properties that highlight the deep connections between these objects
- These properties allow for the transfer of results and techniques between the realms of complex analysis and algebraic geometry

Isomorphism preserving group structure

- The isomorphism $\varphi : \mathbb{C}/\Lambda \rightarrow E$ is a group homomorphism, preserving the additive group structure on both sides
- Addition of points on the elliptic curve E (defined by the chord-tangent law) corresponds to addition of complex numbers modulo the lattice Λ
- The identity element of the group E is the point at infinity, which corresponds to the zero element of $\mathbb{C}/\Lambda$
- Torsion points on the elliptic curve (points of finite order) correspond to division points of the lattice (complex numbers that are rational multiples of lattice vectors)

Isomorphism as a biholomorphic map

- The isomorphism $\varphi : \mathbb{C}/\Lambda \rightarrow E$ is a biholomorphic map, meaning it is a bijective holomorphic map with a holomorphic inverse
- The complex analytic structure on the torus $\mathbb{C}/\Lambda$ (inherited from $\mathbb{C}$) is preserved under the isomorphism
- Holomorphic maps between elliptic curves correspond to holomorphic maps between their associated complex tori that respect the lattice structure
- The isomorphism allows for the study of geometric properties of elliptic curves using complex analytic techniques

Isomorphism and j-invariant

- The j-invariant of an elliptic curve E is a complex number that uniquely determines its isomorphism class over $\mathbb{C}$
- Two elliptic curves over $\mathbb{C}$ are isomorphic if and only if they have the same j-invariant
- The j-invariant can be expressed in terms of the periods ω_1 and ω_2 of the associated lattice Λ_E , linking the complex analytic and algebraic perspectives
- The j-invariant provides a way to classify elliptic curves over $\mathbb{C}$ and study their moduli space

Consequences and applications

- The isomorphism between elliptic curves and complex tori has far-reaching consequences and applications in various areas of mathematics
- It allows for the transfer of results and techniques between complex analysis, algebraic geometry, and number theory, leading to deep insights and new discoveries

Complex multiplication of elliptic curves

- An elliptic curve E has complex multiplication if its associated lattice Λ_E is invariant under multiplication by a complex number $\alpha \notin \mathbb{Q}$
- Elliptic curves with complex multiplication have special arithmetic properties and are important in number theory
- The j -invariants of elliptic curves with complex multiplication are algebraic integers, and their minimal polynomials have interesting properties
- The theory of complex multiplication provides a link between elliptic curves and class field theory

Elliptic curves over $\mathbb{C}$ vs over $\mathbb{Q}$

- The isomorphism theorem holds for elliptic curves over $\mathbb{C}$, but the situation is more complicated for elliptic curves over $\mathbb{Q}$ (or other fields)
- Not every elliptic curve over $\mathbb{Q}$ is isomorphic to a complex torus with a lattice defined over $\mathbb{Q}$
- The study of elliptic curves over $\mathbb{Q}$ leads to questions about rational points, torsion subgroups, and the Mordell-Weil theorem
- The interplay between the complex analytic and algebraic aspects of elliptic curves over $\mathbb{Q}$ is a central theme in arithmetic geometry

Modular curves and moduli spaces

- Modular curves are algebraic curves that parametrize isomorphism classes of elliptic curves with additional structure (e.g., torsion points, isogenies)
- The j -invariant provides a natural map from a modular curve to the moduli space of elliptic curves (a coarse moduli space)
- The study of modular curves and their properties leads to important results in number theory, such as the proof of Fermat's Last Theorem
- The isomorphism between elliptic curves and complex tori provides a foundation for the study of moduli spaces and their compactifications

5.4 Elliptic curves and the modular lambda function

Elliptic curves are fascinating mathematical objects with applications in cryptography and number theory. They're defined by cubic equations and form abelian groups under a special addition law. The Weierstrass equation is a standard way to represent them.

The modular lambda function provides an alternative way to parameterize elliptic curves. It's defined on the complex upper half-plane and has important connections to modular forms. This function allows us to study elliptic curves using complex analysis techniques.

Elliptic curves

- Elliptic curves are a type of algebraic curve defined by a cubic equation in two variables
- They have important applications in number theory, cryptography, and complex analysis

- Elliptic curves over finite fields are used in elliptic curve cryptography for secure communication and digital signatures

Weierstrass equation

- The Weierstrass equation is a standard form for representing elliptic curves
- It is given by the equation $y^2 = x^3 + ax + b$, where a and b are constants
- The discriminant $\Delta = -16(4a^3 + 27b^2)$ must be nonzero for the curve to be non-singular

Elliptic curve points

- Points on an elliptic curve are solutions to the Weierstrass equation
- They form an abelian group under the elliptic curve group law
- The set of points on an elliptic curve over a field K is denoted $E(K)$

Point at infinity

- The point at infinity, denoted $\mathcal{O}$, is a special point on an elliptic curve
- It serves as the identity element for the elliptic curve group
- Adding any point P on the curve to $\mathcal{O}$ results in P itself

Elliptic curve group law

- The elliptic curve group law defines addition of points on an elliptic curve
- It satisfies the properties of an abelian group: associativity, identity element, inverse elements, and commutativity
- The group law can be defined geometrically using the chord-and-tangent method

Elliptic curve addition

- Elliptic curve addition is the operation of adding two points on an elliptic curve
- Given points P and Q , their sum $P + Q$ is found by drawing a line through P and Q and finding the third point of intersection with the curve
- The negative of this third point is the sum $P + Q$

Elliptic curve doubling

- Elliptic curve doubling is the operation of adding a point to itself
- Given a point P , its double $2P$ is found by drawing the tangent line to the curve at P and finding the second point of intersection
- The negative of this second point is the double $2P$

Elliptic curve scalar multiplication

- Elliptic curve scalar multiplication is the operation of adding a point to itself multiple times

- Given a point P and a scalar k , the scalar multiple kP is computed by repeated doubling and addition
- Scalar multiplication is the basis for elliptic curve cryptography, as it is computationally infeasible to reverse (discrete logarithm problem)

Elliptic curve discrete logarithm problem

- The elliptic curve discrete logarithm problem (ECDLP) is the problem of finding the scalar k given points P and $Q = kP$ on an elliptic curve
- It is believed to be computationally hard for properly chosen curves and is the foundation of elliptic curve cryptography's security
- The hardness of the ECDLP allows for smaller key sizes compared to other cryptographic schemes (RSA)

Modular lambda function

- The modular lambda function is a special function defined on the complex upper half-plane that parameterizes elliptic curves
- It provides an alternative representation of elliptic curves and has important applications in number theory and cryptography
- The modular lambda function is closely related to the j -invariant of an elliptic curve

Definition of modular lambda function

- The modular lambda function $\lambda(\tau)$ is defined as $\lambda(\tau) = \frac{g_2(\tau)^3}{g_2(\tau)^3 - 27g_3(\tau)^2}$, where g_2 and g_3 are Eisenstein series
- It is a meromorphic function on the complex upper half-plane $\mathbb{H} = \{\tau \in \mathbb{C} : \text{Im}(\tau) > 0\}$
- The modular lambda function is invariant under the action of the modular group $SL(2, \mathbb{Z})$

Computation of modular lambda function

- The modular lambda function can be computed using the q -expansion of the Eisenstein series g_2 and g_3
- It has a Fourier series expansion in terms of the nome $q = e^{2\pi i \tau}$
- Efficient algorithms exist for computing the modular lambda function to high precision

Modular lambda function properties

- The modular lambda function satisfies the transformations $\lambda(-1/\tau) = 1 - \lambda(\tau)$ and $\lambda(\tau + 1) = 1/(1 - \lambda(\tau))$
- It has a simple pole at the cusp $\tau = i\infty$ with residue 1
- The values of λ at the elliptic points $\tau = i$ and $\tau = e^{2\pi i/3}$ are $1/2$ and $-e^{\pi i/3}$, respectively

Modular lambda function vs Weierstrass equation

- The modular lambda function provides an alternative parameterization of elliptic curves compared to the Weierstrass equation

- Every elliptic curve over $\mathbb{C}$ can be written in the form $y^2 = x(x-1)(x-\lambda)$ for some $\lambda \in \mathbb{C} \setminus \{0, 1\}$
- The j-invariant of an elliptic curve can be expressed in terms of the modular lambda function as
$$j(\lambda) = 256 \frac{(\lambda^2 - \lambda + 1)^3}{\lambda^2(\lambda - 1)^2}$$

Applications of modular lambda function

- The modular lambda function has applications in the study of elliptic curves and modular forms
- It provides a way to classify elliptic curves over $\mathbb{C}$ up to isomorphism
- The modular lambda function is used in the computation of elliptic curve isogenies and the study of complex multiplication

Modular lambda function in cryptography

- The modular lambda function can be used to construct elliptic curves with desirable properties for cryptography
- It allows for the efficient generation of elliptic curves with a given j-invariant or endomorphism ring
- The modular lambda function is used in the implementation of elliptic curve cryptography in computer algebra systems (Sage, Magma)

Relationship between elliptic curves and modular lambda function

- The modular lambda function provides a deep connection between elliptic curves and modular forms
- It allows for the study of elliptic curves using the tools of complex analysis and modular forms
- The relationship between elliptic curves and the modular lambda function is central to the proof of the modularity theorem (Taniyama-Shimura conjecture)

Isomorphism between elliptic curves and modular lambda function

- There is an isomorphism between the moduli space of elliptic curves over $\mathbb{C}$ and the quotient space $\mathbb{H}/\mathrm{SL}(2, \mathbb{Z})$
- This isomorphism is given by the modular lambda function, which maps the complex upper half-plane to the complex plane minus $\{0, 1\}$
- The modular lambda function provides a one-to-one correspondence between isomorphism classes of elliptic curves and points in the fundamental domain of $\mathrm{SL}(2, \mathbb{Z})$

Modular lambda function as parameterization of elliptic curves

- The modular lambda function can be used to parameterize elliptic curves over $\mathbb{C}$
- Every elliptic curve can be written in the form $y^2 = x(x-1)(x-\lambda)$ for some $\lambda \in \mathbb{C} \setminus \{0, 1\}$
- The value of λ determines the isomorphism class of the elliptic curve

Advantages of modular lambda function representation

- The modular lambda function provides a more compact representation of elliptic curves compared to the Weierstrass equation
- It simplifies the study of isomorphism classes of elliptic curves and their moduli space

- The modular lambda function allows for the application of complex analysis and modular forms to the study of elliptic curves

Computational aspects

- The modular lambda function has important computational aspects in the study of elliptic curves and their applications
- Efficient algorithms have been developed for computing the modular lambda function and related quantities
- The modular lambda function is used in the implementation of elliptic curve cryptography in various software packages

Efficient algorithms for modular lambda function

- There are efficient algorithms for computing the modular lambda function to high precision
- These algorithms often involve the use of the arithmetic-geometric mean (AGM) iteration and theta functions
- Examples include the Brent-Salamin algorithm and the Borwein algorithm for computing the modular lambda function

Modular lambda function in computer algebra systems

- The modular lambda function is implemented in various computer algebra systems for the study of elliptic curves and modular forms
- Systems like Sage, Magma, and Pari/GP provide functions for computing the modular lambda function and related quantities
- These implementations are used in research and applications involving elliptic curves and modular forms

Modular lambda function in elliptic curve cryptography implementations

- The modular lambda function is used in the implementation of elliptic curve cryptography in various software packages
- It allows for the efficient generation of elliptic curves with desired properties (j-invariant, endomorphism ring)
- The modular lambda function is used in the construction of elliptic curve parameters and the computation of elliptic curve isogenies in cryptographic protocols

Advanced topics

- The modular lambda function is connected to various advanced topics in the study of elliptic curves and modular forms
- These topics involve the interplay between elliptic curves, complex multiplication, and isogenies
- The modular lambda function also appears in the study of higher genus curves and their moduli spaces

Complex multiplication and modular lambda function

- The modular lambda function is closely related to the theory of complex multiplication of elliptic curves
- Elliptic curves with complex multiplication correspond to special values of the modular lambda function
- The modular lambda function can be used to construct elliptic curves with a given endomorphism ring and to study their properties

Modular lambda function and elliptic curve isogenies

- The modular lambda function is used in the study of elliptic curve isogenies
- Isogenies between elliptic curves correspond to certain transformations of the modular lambda function
- The modular lambda function can be used to compute isogenies between elliptic curves and to study their properties

Modular lambda function in higher genus curves

- The modular lambda function has analogues in the study of higher genus curves and their moduli spaces
- These analogues involve Siegel modular forms and the Siegel upper half-space
- The study of higher genus curves and their moduli spaces using modular forms is an active area of research in algebraic geometry and number theory

5.5 Elliptic curve uniformization

Elliptic curve uniformization connects complex analysis, algebraic geometry, and number theory. It reveals how elliptic curves over complex numbers can be viewed as quotients of the complex plane by lattices, providing deep insights into their structure and properties.

This perspective allows us to understand elliptic curves through the lens of complex tori and Weierstrass functions. It forms the foundation for studying isomorphisms, complex multiplication, and connections to modular forms and elliptic functions.

Elliptic curves over complex numbers

- Elliptic curves over the complex numbers provide a rich and beautiful interplay between complex analysis, algebraic geometry, and number theory
- The study of elliptic curves over $\mathbb{C}$ allows for a deep understanding of their geometric and analytic properties, which can then be applied to other fields such as cryptography and mathematical physics

Lattices in the complex plane

- A lattice in the complex plane is a discrete subgroup of $\mathbb{C}$ of the form $L = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$, where $\omega_1, \omega_2 \in \mathbb{C}$ are linearly independent over $\mathbb{R}$
- The generators ω_1 and ω_2 are called the periods of the lattice
- The fundamental parallelogram of a lattice is the set $\{a\omega_1 + b\omega_2 : 0 \leq a, b < 1\}$, which tiles the complex plane under translations by lattice elements

- The shape of the fundamental parallelogram determines the geometry of the lattice (square, rectangular, rhombic)

Quotient of $\mathbb{C}$ by lattice

- The quotient of the complex plane $\mathbb{C}$ by a lattice L , denoted $\mathbb{C}/L$, is the set of equivalence classes of complex numbers under the equivalence relation $z_1 \sim z_2$ if and only if $z_1 - z_2 \in L$
- The quotient $\mathbb{C}/L$ inherits a natural complex analytic structure from $\mathbb{C}$, making it a compact Riemann surface
- The quotient $\mathbb{C}/L$ is topologically equivalent to a torus, obtained by identifying opposite sides of the fundamental parallelogram

Elliptic curves as complex tori

- An elliptic curve over $\mathbb{C}$ can be realized as a complex torus $\mathbb{C}/L$ for some lattice L
- The group law on the elliptic curve corresponds to the natural group structure on the torus, with the point at infinity serving as the identity element
- The Weierstrass $\wp$ -function associated to the lattice L provides a parametrization of the elliptic curve, giving an explicit isomorphism between $\mathbb{C}/L$ and the curve in Weierstrass form

Uniformization theorem

- The uniformization theorem is a fundamental result in the theory of Riemann surfaces, providing a classification of simply connected Riemann surfaces
- In the context of elliptic curves, the uniformization theorem states that every elliptic curve over $\mathbb{C}$ is analytically isomorphic to a complex torus $\mathbb{C}/L$ for some lattice L

Statement of the theorem

- Let E be an elliptic curve over $\mathbb{C}$. Then there exists a lattice L in $\mathbb{C}$ such that E is analytically isomorphic to the complex torus $\mathbb{C}/L$
- The isomorphism is given by the Weierstrass $\wp$ -function associated to the lattice L
- The lattice L is uniquely determined by E up to homothety (scaling by a complex number)

Existence of Weierstrass parametrization

- The Weierstrass $\wp$ -function associated to a lattice L is defined as $\wp(z) = \frac{1}{z^2} + \sum_{\omega \in L \setminus \{0\}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right)$
- The $\wp$ -function is an even, meromorphic function on $\mathbb{C}$ with double poles at the lattice points and satisfies the differential equation $(\wp'(z))^2 = 4\wp(z)^3 - g_2\wp(z) - g_3$, where g_2 and g_3 are constants depending on the lattice L
- The map $\mathbb{C}/L \rightarrow E$ given by $z \mapsto (\wp(z), \wp'(z))$ is an analytic isomorphism between the complex torus and the elliptic curve E in Weierstrass form $y^2 = 4x^3 - g_2x - g_3$

Uniqueness up to isomorphism

- If two elliptic curves E_1 and E_2 over $\mathbb{C}$ are analytically isomorphic, then their corresponding lattices L_1 and L_2 are homothetic, meaning there exists a complex number α such that $L_2 = \alpha L_1$

- Conversely, if two lattices L_1 and L_2 are homothetic, then the corresponding elliptic curves $\mathbb{C}/L_1$ and $\mathbb{C}/L_2$ are analytically isomorphic
- This uniqueness property allows for the classification of elliptic curves over $\mathbb{C}$ up to isomorphism by their associated lattices

Weierstrass $\wp$ function

- The Weierstrass $\wp$ -function is a central object in the study of elliptic curves over the complex numbers, providing a bridge between the analytic and algebraic aspects of the theory
- The $\wp$ -function is a meromorphic function on $\mathbb{C}$ that is doubly periodic with respect to a lattice L

Definition and properties

- The Weierstrass $\wp$ -function associated to a lattice L is defined as $\wp(z) = \frac{1}{z^2} + \sum_{\omega \in L \setminus \{0\}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right)$
- $\wp(z)$ is an even function, satisfying $\wp(-z) = \wp(z)$
- $\wp(z)$ is periodic with respect to the lattice L , meaning $\wp(z + \omega) = \wp(z)$ for all $\omega \in L$
- $\wp(z)$ has double poles at the lattice points and no other poles

Relation to elliptic curves

- The Weierstrass $\wp$ -function parametrizes an elliptic curve in Weierstrass form $y^2 = 4x^3 - g_2x - g_3$, where g_2 and g_3 are constants depending on the lattice L
- The map $\phi: \mathbb{C}/L \rightarrow E$ given by $\phi(z) = (\wp(z), \wp'(z))$ is an analytic isomorphism between the complex torus $\mathbb{C}/L$ and the elliptic curve E
- The group law on the elliptic curve E corresponds to the natural group structure on the torus $\mathbb{C}/L$ under the isomorphism ϕ

Differential equation satisfied by $\wp$

- The Weierstrass $\wp$ -function satisfies the differential equation $(\wp'(z))^2 = 4\wp(z)^3 - g_2\wp(z) - g_3$, where g_2 and g_3 are constants depending on the lattice L
- This differential equation characterizes the $\wp$ -function and plays a crucial role in the study of elliptic curves and their associated functions
- The constants g_2 and g_3 are called the invariants of the Weierstrass equation and determine the isomorphism class of the elliptic curve

Periods and quasi-periods

- The periods and quasi-periods of an elliptic curve are fundamental quantities associated with the lattice uniformization of the curve
- They encode important information about the geometry and arithmetic of the elliptic curve

Periods of elliptic functions

- The periods of an elliptic function f with respect to a lattice L are the complex numbers $\omega \in L$ such that $f(z + \omega) = f(z)$ for all $z \in \mathbb{C}$

- For the Weierstrass $\wp$ -function, the periods are precisely the elements of the lattice L
- The periods form a discrete subgroup of $\mathbb{C}$ and generate the lattice L

Quasi-periods and the period lattice

- The quasi-periods of the Weierstrass $\wp$ -function are the values of the integral $\eta_i = \int_0^{\omega_i} \wp(z) dz$, where ω_1 and ω_2 are the fundamental periods of the lattice L
- The quasi-periods satisfy the relation $\eta_1\omega_2 - \eta_2\omega_1 = \pm 2\pi i$
- The period lattice of an elliptic curve is the lattice generated by the periods ω_1, ω_2 and the quasi-periods η_1, η_2

Legendre relation for periods

- The periods ω_1, ω_2 and quasi-periods η_1, η_2 of an elliptic curve satisfy the Legendre relation $\eta_1\omega_2 - \eta_2\omega_1 = 2\pi i$
- This relation is a consequence of the residue theorem applied to the Weierstrass $\wp$ -function
- The Legendre relation plays a crucial role in the theory of elliptic integrals and the study of the moduli space of elliptic curves

Isomorphisms of elliptic curves

- Isomorphisms of elliptic curves over the complex numbers are an important tool for understanding the geometry and arithmetic of these curves
- They allow for the classification of elliptic curves up to isomorphism and the study of their symmetries

Isomorphisms over $\mathbb{C}$

- Two elliptic curves E_1 and E_2 over $\mathbb{C}$ are isomorphic if there exists a complex analytic isomorphism $\phi : E_1 \rightarrow E_2$ that preserves the group structure
- Isomorphic elliptic curves have the same j -invariant, a complex number that characterizes the isomorphism class of the curve
- Conversely, two elliptic curves with the same j -invariant are isomorphic over $\mathbb{C}$

Isomorphisms preserving the group law

- An isomorphism $\phi : E_1 \rightarrow E_2$ between elliptic curves is said to preserve the group law if $\phi(P + Q) = \phi(P) + \phi(Q)$ for all points $P, Q \in E_1$
- Isomorphisms preserving the group law are precisely the complex analytic isomorphisms between the curves
- The set of isomorphisms preserving the group law forms a group under composition, called the automorphism group of the elliptic curve

Automorphisms and endomorphisms

- An automorphism of an elliptic curve E is an isomorphism $\phi : E \rightarrow E$ from the curve to itself that preserves the group law

- The set of automorphisms of E forms a finite group, denoted $\text{Aut}(E)$, which is either cyclic of order 2, 4, or 6, or the Klein four-group
- An endomorphism of an elliptic curve E is a complex analytic map $\phi : E \rightarrow E$ that preserves the group law, but may not be an isomorphism
- The set of endomorphisms of E forms a ring under pointwise addition and composition, called the endomorphism ring of E

Complex multiplication

- Complex multiplication is a special property that certain elliptic curves over the complex numbers possess, which has far-reaching consequences in number theory and cryptography
- Elliptic curves with complex multiplication have additional symmetries and arithmetic properties that make them particularly interesting to study

Elliptic curves with complex multiplication

- An elliptic curve E over $\mathbb{C}$ is said to have complex multiplication (CM) if its endomorphism ring $\text{End}(E)$ is strictly larger than $\mathbb{Z}$
- Equivalently, E has CM if there exists an imaginary quadratic field K such that $K \subseteq \text{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$
- Elliptic curves with CM are rare; they form a countable subset of the moduli space of elliptic curves

CM by quadratic imaginary fields

- If an elliptic curve E has CM by an imaginary quadratic field K , then K is isomorphic to a subfield of the endomorphism algebra $\text{End}(E) \otimes_{\mathbb{Z}} \mathbb{Q}$
- The conductor of the order $\mathcal{O} = \text{End}(E)$ in K is an important invariant of the curve E , measuring the complexity of its CM structure
- Elliptic curves with CM by the same imaginary quadratic field and conductor are isogenous, meaning there exists a non-constant rational map between them that preserves the group law

Hilbert class field and ray class fields

- The Hilbert class field of an imaginary quadratic field K is the maximal unramified abelian extension of K
- Elliptic curves with CM by K are closely related to the Hilbert class field of K ; their j -invariants generate the Hilbert class field over K
- Ray class fields are generalizations of the Hilbert class field that take into account the conductor of the CM order
- Elliptic curves with CM by K and a given conductor are related to the corresponding ray class field of K

Modular functions and modular forms

- Modular functions and modular forms are central objects in the theory of elliptic curves and number theory
- They provide a powerful tool for understanding the arithmetic and geometric properties of elliptic curves and their moduli spaces

Modular functions and modular curves

- A modular function is a meromorphic function on the upper half-plane $\mathcal{H} = \{z \in \mathbb{C} : \text{Im}(z) > 0\}$ that is invariant under the action of a congruence subgroup Γ of $\text{SL}_2(\mathbb{Z})$
- The quotient space $\Gamma \backslash \mathcal{H}^*$, where $\mathcal{H}^* = \mathcal{H} \cup \mathbb{Q} \cup \{\infty\}$, is a compact Riemann surface called a modular curve
- Modular functions can be thought of as meromorphic functions on modular curves
- The j -invariant of an elliptic curve is an example of a modular function on the full modular group $\text{SL}_2(\mathbb{Z})$

Eisenstein series and cusp forms

- Eisenstein series are special modular forms that are eigenfunctions of the Laplace operator on the upper half-plane
- They play a crucial role in the theory of modular forms and the study of the arithmetic of elliptic curves
- Cusp forms are modular forms that vanish at the cusps (the points at infinity) of a modular curve
- The space of cusp forms of a given weight and level is a finite-dimensional vector space with rich arithmetic properties

Hecke operators and Hecke eigenforms

- Hecke operators are linear operators acting on the space of modular forms of a given weight and level
- They are defined using the action of certain double cosets of the congruence subgroup on the upper half-plane
- Hecke eigenforms are modular forms that are simultaneous eigenfunctions of all Hecke operators
- The Fourier coefficients of Hecke eigenforms encode important arithmetic information, such as the number of points on elliptic curves over finite fields

Elliptic functions and elliptic integrals

- Elliptic functions and elliptic integrals are closely related to the theory of elliptic curves and provide a rich source of examples and applications
- They have a long history in mathematics and have been studied extensively for their analytic and geometric properties

Elliptic integrals of the first kind

- An elliptic integral of the first kind is an integral of the form $\int \frac{dx}{\sqrt{P(x)}}$, where $P(x)$ is a cubic or quartic polynomial with distinct roots
- These integrals arise naturally in the study of the arc length of ellipses and the period lattice of elliptic curves
- The complete elliptic integral of the first kind, denoted $K(m)$, is a special case where the limits of integration are 0 and 1, and $P(x) = (1 - x^2)(1 - mx^2)$

Elliptic integrals of the second kind

- An elliptic integral of the second kind is an integral of the form $\int \sqrt{P(x)} dx$, where $P(x)$ is a cubic or quartic polynomial with distinct roots
- These integrals arise in the study of the area enclosed by ellipses and the quasi-periods of ell

5.6 Elliptic curves and complex multiplication

Elliptic curves over complex numbers provide a fascinating bridge between algebra, geometry, and number theory. Their study reveals deep connections between lattices in the complex plane, elliptic functions, and isomorphism classes of curves.

Complex multiplication adds another layer of richness, linking elliptic curves to imaginary quadratic fields and class field theory. This powerful theory enables the construction of curves with special properties, finding applications in cryptography and computational number theory.

Elliptic curves over complex numbers

- Elliptic curves over the complex numbers $\mathbb{C}$ provide a rich geometric and analytic structure that connects various branches of mathematics
- The study of elliptic curves over $\mathbb{C}$ involves the interplay between algebraic geometry, complex analysis, and number theory
- Understanding elliptic curves over $\mathbb{C}$ lays the foundation for the study of complex multiplication and its applications

Weierstrass equation

- Elliptic curves over $\mathbb{C}$ can be described by the Weierstrass equation $y^2 = x^3 + ax + b$, where $a, b \in \mathbb{C}$ and the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$
- The Weierstrass equation defines a smooth projective curve of genus one with a specified base point $\mathcal{O}$ at infinity
- The coefficients a and b determine the shape and properties of the elliptic curve ($y^2 = x^3 - x$ and $y^2 = x^3 - 1$)

Lattices in complex plane

- A lattice Λ in the complex plane $\mathbb{C}$ is a discrete subgroup of the form $\Lambda = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$, where $\omega_1, \omega_2 \in \mathbb{C}$ are linearly independent over $\mathbb{R}$
- Every lattice Λ defines a complex torus $\mathbb{C}/\Lambda$, which is an elliptic curve when equipped with a suitable group law
- The periods ω_1 and ω_2 determine the shape and size of the lattice ($\Lambda = \mathbb{Z} + i\mathbb{Z}$ and $\Lambda = \mathbb{Z} + \sqrt{2}i\mathbb{Z}$)

Elliptic functions and meromorphic functions

- An elliptic function is a meromorphic function $f: \mathbb{C} \rightarrow \mathbb{C}$ that is periodic with respect to a lattice Λ , i.e., $f(z + \omega) = f(z)$ for all $z \in \mathbb{C}$ and $\omega \in \Lambda$
- Elliptic functions can be expressed in terms of the Weierstrass $\wp$ -function and its derivative $\wp'$, which are examples of meromorphic functions on the complex torus $\mathbb{C}/\Lambda$

- The field of elliptic functions associated with a lattice Λ is isomorphic to the field of rational functions on the corresponding elliptic curve ($\wp(z)$ and $\wp'(z)$)

Isomorphisms between elliptic curves and complex tori

- There is a bijective correspondence between isomorphism classes of elliptic curves over $\mathbb{C}$ and isomorphism classes of complex tori $\mathbb{C}/\Lambda$
- The Weierstrass $\wp$ -function and its derivative $\wp'$ provide an explicit isomorphism between an elliptic curve in Weierstrass form and the corresponding complex torus
- The j -invariant $j(E)$ of an elliptic curve E is a complex number that characterizes the isomorphism class of E and the corresponding lattice Λ ($j(y^2 = x^3 - x) = 1728$ and $j(y^2 = x^3 - 1) = 0$)

Complex multiplication

- Complex multiplication is a special property of certain elliptic curves over $\mathbb{C}$ that have extra endomorphisms beyond the usual multiplication-by- n maps
- The study of complex multiplication connects elliptic curves, imaginary quadratic fields, and class field theory
- Complex multiplication plays a crucial role in constructing elliptic curves with desired properties and has applications in cryptography and computational number theory

Endomorphism rings of elliptic curves

- An endomorphism of an elliptic curve E over $\mathbb{C}$ is a complex-analytic map $\phi : E \rightarrow E$ that is also a group homomorphism
- The set of endomorphisms of E forms a ring $\text{End}(E)$ under pointwise addition and composition
- The endomorphism ring $\text{End}(E)$ is either $\mathbb{Z}$ (generic case) or an order in an imaginary quadratic field (complex multiplication case) ($\text{End}(y^2 = x^3 - x) = \mathbb{Z}[i]$ and $\text{End}(y^2 = x^3 - 1) = \mathbb{Z}[\omega]$, where $\omega = e^{2\pi i/3}$)

Orders in imaginary quadratic fields

- An imaginary quadratic field is a number field $K = \mathbb{Q}(\sqrt{-d})$, where d is a positive square-free integer
- An order $\mathcal{O}$ in K is a subring of K that is a finitely generated $\mathbb{Z}$ -module containing 1 ($\mathcal{O} = \mathbb{Z}[i]$ in $\mathbb{Q}(i)$ and $\mathcal{O} = \mathbb{Z}[\omega]$ in $\mathbb{Q}(\sqrt{-3})$)
- The maximal order in K is the ring of integers $\mathcal{O}_K$, which is the integral closure of $\mathbb{Z}$ in K

Class group of orders

- The class group $\text{Cl}(\mathcal{O})$ of an order $\mathcal{O}$ in an imaginary quadratic field K is the group of fractional ideal classes of $\mathcal{O}$ under ideal multiplication
- The class number $h(\mathcal{O})$ is the order of the class group $\text{Cl}(\mathcal{O})$
- The class group $\text{Cl}(\mathcal{O}_K)$ of the ring of integers $\mathcal{O}_K$ is a finite abelian group that measures the failure of unique factorization in $\mathcal{O}_K$ ($\text{Cl}(\mathbb{Z}[i]) = \{1\}$ and $\text{Cl}(\mathbb{Z}[\omega]) \cong \mathbb{Z}/2\mathbb{Z}$)

Elliptic curves with complex multiplication

- An elliptic curve E over $\mathbb{C}$ has complex multiplication by an order $\mathcal{O}$ in an imaginary quadratic field K if $\text{End}(E) \cong \mathcal{O}$
- The j -invariant $j(E)$ of an elliptic curve E with complex multiplication by $\mathcal{O}$ is an algebraic integer that generates the Hilbert class field of $\mathcal{O}$
- Elliptic curves with complex multiplication have special properties, such as a larger endomorphism ring and a more efficient point counting algorithm ($y^2 = x^3 - x$ has CM by $\mathbb{Z}[i]$ and $y^2 = x^3 - 1$ has CM by $\mathbb{Z}[\omega]$)

Hilbert class field and ray class fields

- The Hilbert class field $H_{\mathcal{O}}$ of an order $\mathcal{O}$ in an imaginary quadratic field K is the maximal unramified abelian extension of K with Galois group isomorphic to $\text{Cl}(\mathcal{O})$
- Ray class fields are generalizations of the Hilbert class field that allow ramification at a finite set of primes
- The theory of complex multiplication provides a way to construct Hilbert class fields and ray class fields using the j -invariants of elliptic curves with complex multiplication ($H_{\mathbb{Z}[i]} = \mathbb{Q}(i)$ and $H_{\mathbb{Z}[\omega]} = \mathbb{Q}(\sqrt{-3}, \sqrt{-1})$)

Elliptic curves over finite fields

- The study of elliptic curves over finite fields $\mathbb{F}_q$, where q is a prime power, has important applications in cryptography and computational number theory
- Elliptic curves over finite fields exhibit different properties compared to elliptic curves over $\mathbb{C}$, such as a finite number of points and a more intricate endomorphism structure
- Understanding the reduction of elliptic curves modulo primes and the Frobenius endomorphism is crucial for the efficient implementation of elliptic curve cryptography

Reduction of elliptic curves modulo primes

- Given an elliptic curve E defined over $\mathbb{Q}$ and a prime p , the reduction of E modulo p is the elliptic curve E_p obtained by reducing the coefficients of the Weierstrass equation of E modulo p
- The reduction E_p is an elliptic curve over the finite field $\mathbb{F}_p$ if p does not divide the discriminant Δ of E (good reduction)
- If p divides Δ , the reduction E_p is either a singular curve (bad reduction) or a curve with a node or a cusp (semistable reduction) ($y^2 = x^3 - x \pmod{5}$ and $y^2 = x^3 - 1 \pmod{7}$)

Supersingular vs ordinary elliptic curves

- An elliptic curve E over a finite field $\mathbb{F}_q$ of characteristic p is called supersingular if the group $E(\mathbb{F}_{q^n})$ has order divisible by p for all $n \geq 1$
- If E is not supersingular, it is called ordinary
- Supersingular elliptic curves have special properties, such as a larger endomorphism ring and a more efficient point counting algorithm ($y^2 = x^3 - x$ over $\mathbb{F}_5$ is supersingular and $y^2 = x^3 - 1$ over $\mathbb{F}_7$ is ordinary)

Frobenius endomorphism

- The Frobenius endomorphism $\pi_q : E \rightarrow E$ of an elliptic curve E over a finite field $\mathbb{F}_q$ is the map $(x, y) \mapsto (x^q, y^q)$
- The Frobenius endomorphism π_q satisfies the characteristic equation $\pi_q^2 - t_q \pi_q + q = 0$, where t_q is the trace of π_q
- The trace t_q determines the number of points on E over $\mathbb{F}_q$ via the formula $\#E(\mathbb{F}_q) = q + 1 - t_q$ ($\pi_5(x, y) = (x^5, y^5)$ on $y^2 = x^3 - x$ over $\mathbb{F}_5$ and $\pi_7(x, y) = (x^7, y^7)$ on $y^2 = x^3 - 1$ over $\mathbb{F}_7$)

Characteristic polynomial of Frobenius

- The characteristic polynomial of the Frobenius endomorphism π_q of an elliptic curve E over $\mathbb{F}_q$ is the polynomial $P_q(T) = T^2 - t_q T + q$
- The roots of $P_q(T)$ are complex numbers of absolute value $\sqrt{q}$ and are conjugate if E is ordinary
- The splitting field of $P_q(T)$ is related to the endomorphism ring of E and plays a role in the complex multiplication method ($P_5(T) = T^2 + 2T + 5$ for $y^2 = x^3 - x$ over $\mathbb{F}_5$ and $P_7(T) = T^2 + 2T + 7$ for $y^2 = x^3 - 1$ over $\mathbb{F}_7$)

Point counting algorithms for elliptic curves

- Determining the number of points on an elliptic curve E over a finite field $\mathbb{F}_q$ is a fundamental problem in elliptic curve cryptography
- The naive algorithm of counting points by exhaustive search has exponential complexity in $\log q$
- More efficient point counting algorithms, such as Schoof's algorithm and its variants (Schoof-Elkies-Atkin algorithm), have polynomial complexity in $\log q$ and rely on the properties of the Frobenius endomorphism and the characteristic polynomial ($\#E(\mathbb{F}_5) = 5$ for $y^2 = x^3 - x$ over $\mathbb{F}_5$ and $\#E(\mathbb{F}_7) = 6$ for $y^2 = x^3 - 1$ over $\mathbb{F}_7$)

Applications of complex multiplication

- The theory of complex multiplication has numerous applications in various areas of mathematics and computer science
- Complex multiplication provides a way to construct elliptic curves with desired properties, such as a prescribed number of points or resistance to certain attacks in cryptography
- The complex multiplication method is also used in primality proving and the computation of class polynomials

Constructing elliptic curves with prescribed number of points

- In elliptic curve cryptography, it is often desirable to construct elliptic curves over finite fields with a prescribed number of points to ensure security and efficiency
- The complex multiplication method allows the construction of such curves by starting with an elliptic curve E with complex multiplication by an order $\mathcal{O}$ and reducing E modulo a suitable prime p
- The resulting curve E_p over $\mathbb{F}_p$ will have a number of points related to the class number of $\mathcal{O}$ and the trace of the Frobenius endomorphism ($y^2 = x^3 - 1$ over $\mathbb{F}_7$ has 6 points)

Primality proving

- Primality proving is the task of determining whether a given integer n is prime or composite
- The complex multiplication method can be used to construct primality proofs for certain classes of integers, such as Mersenne numbers and Fermat numbers
- The idea is to construct an elliptic curve E with complex multiplication by an order $\mathcal{O}$ and show that the reduction E_n has a prime number of points over $\mathbb{F}_n$, implying that n is prime ($y^2 = x^3 - x$ over $\mathbb{F}_5$ has 5 points, proving that 5 is prime)

Generating cryptographically secure elliptic curves

- Elliptic curve cryptography relies on the difficulty of the discrete logarithm problem on elliptic curves over finite fields
- To ensure security, it is important to use elliptic curves that are resistant to known attacks, such as the Pollard rho algorithm and the Pohlig-Hellman algorithm
- The complex multiplication method can be used to generate cryptographically secure elliptic curves with a prescribed number of points and a large embedding degree, which is important for pairing-based cryptography ($y^2 = x^3 - x + 1$ over $\mathbb{F}_{31}$ has 37 points and embedding degree 6)

Complex multiplication method for computing class polynomials

- The complex multiplication method is an algorithm for computing the Hilbert class polynomial $H_D(x)$ associated with an imaginary quadratic discriminant D
- The Hilbert class polynomial $H_D(x)$ is a polynomial whose roots are the j -invariants of elliptic curves with complex multiplication by the order of discriminant D
- The complex multiplication method computes $H_D(x)$ by evaluating the modular j -function at certain points in the upper half-plane related to the class group of the order ($H_{-3}(x) = x$ and $H_{-4}(x) = x^2 + 1728$)

Elliptic curve cryptography and pa

Unit 6 – Elliptic Curves: Algebraic Geometry Foundations

6.1 Elliptic curves as algebraic varieties

Elliptic curves are fascinating algebraic varieties with deep connections to number theory and cryptography. As cubic curves, they possess a unique group structure that enables powerful applications in various fields of mathematics and computer science.

This section explores the defining equations of elliptic curves, their geometric properties as cubic curves, and the group law that governs their points. We'll also examine isomorphisms between curves, their behavior over different fields, and the intriguing properties of torsion points.

Defining equations of elliptic curves

- Elliptic curves are algebraic curves defined by specific types of equations called Weierstrass equations
- Understanding the various forms of equations is crucial for studying the properties and applications of elliptic curves
- Different forms of equations offer advantages in specific contexts, such as simplifying computations or highlighting certain geometric features

Weierstrass equations

- General form: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ where a_1, a_2, a_3, a_4, a_6 are constants
- Simplified Weierstrass equation: $y^2 = x^3 + ax + b$ where a, b are constants
- Most commonly used form due to its generality and simplicity
- Coefficients determine the shape and properties of the elliptic curve

Legendre normal form

- Equation: $y^2 = x(x - 1)(x - \lambda)$ where λ is a constant not equal to 0 or 1
- Obtained from the simplified Weierstrass equation by a change of variables
- Useful for studying elliptic curves over finite fields and in cryptography (e.g., Diffie-Hellman key exchange)

Hesse normal form

- Equation: $x^3 + y^3 + z^3 = 3\mu xyz$ where μ is a constant not equal to 1
- Projective form of an elliptic curve, using homogeneous coordinates $[x : y : z]$
- Symmetric form that highlights the cubic nature of the curve
- Facilitates the study of elliptic curves over various fields (e.g., complex numbers)

Montgomery form

- Equation: $By^2 = x^3 + Ax^2 + x$ where A, B are constants satisfying certain conditions
- Designed to optimize computations in elliptic curve cryptography
- Enables faster point multiplication and resistance to side-channel attacks (e.g., timing attacks)

Edwards form

- Equation: $x^2 + y^2 = 1 + dx^2y^2$ where d is a constant not equal to 0 or 1
- Alternative form that provides efficient arithmetic and symmetry
- Useful for implementing elliptic curve cryptography (e.g., Edwards25519 curve)
- Allows for unified addition formulas and resistance to exceptional points

Elliptic curves as cubic curves

- Elliptic curves are a special class of algebraic curves defined by cubic equations
- Understanding the geometric properties of cubic curves is essential for studying elliptic curves
- Cubic curves have nine points of inflection, which play a crucial role in the group law on elliptic curves

Nonsingularity condition

- For a cubic curve to be an elliptic curve, it must be nonsingular (i.e., have no cusps or self-intersections)
- Nonsingularity ensures that the curve is smooth and has well-defined tangent lines at every point
- The nonsingularity condition is expressed in terms of the discriminant of the cubic equation

Discriminant of cubic polynomials

- The discriminant Δ of a cubic polynomial $f(x) = ax^3 + bx^2 + cx + d$ is given by $\Delta = 18abcd - 4b^3d + b^2c^2 - 4ac^3 - 27a^2d^2$
- For an elliptic curve in simplified Weierstrass form $y^2 = x^3 + ax + b$, the discriminant is $\Delta = -16(4a^3 + 27b^2)$
- The curve is nonsingular if and only if $\Delta \neq 0$
- The sign of the discriminant determines the nature of the roots (e.g., three real roots, one real and two complex roots)

Projective closure

- Elliptic curves can be extended to the projective plane by adding a point at infinity
- The projective closure of an elliptic curve is obtained by homogenizing the equation (e.g., $y^2z = x^3 + axz^2 + bz^3$)
- The point at infinity, denoted by $\mathcal{O}$, is the unique point where the projective curve intersects the line at infinity ($z = 0$)

- The projective closure is essential for defining the group law on elliptic curves

Points at infinity

- In the projective closure of an elliptic curve, there is a single point at infinity $\mathcal{O}$
- The point at infinity serves as the identity element for the group law on the elliptic curve
- Geometrically, the point at infinity can be thought of as the point where all vertical lines intersect the curve
- The existence of the point at infinity is crucial for the completeness and consistency of the group law

Group law on elliptic curves

- Elliptic curves have a natural group structure, which is the foundation for their applications in cryptography and number theory
- The group law defines an addition operation on the points of an elliptic curve, resulting in an abelian group
- The group law has a geometric interpretation that involves drawing lines and finding their intersections with the curve

Chord-tangent construction

- To add two points P and Q on an elliptic curve, draw a line through P and Q (if $P \neq Q$) or the tangent line at P (if $P = Q$)
- The line intersects the curve at a third point R'
- The sum $P + Q$ is defined as the reflection of R' across the x-axis, denoted by $-R'$
- If the line is vertical, the sum is defined as the point at infinity $\mathcal{O}$

Associativity of group law

- The group law on elliptic curves is associative: $(P + Q) + R = P + (Q + R)$ for any points P, Q, R on the curve
- Associativity allows for efficient computation of point multiples (e.g., $nP = P + P + \dots + P$ n times)
- Associativity is a consequence of the geometric properties of the chord-tangent construction

Identity element

- The point at infinity $\mathcal{O}$ serves as the identity element for the group law on elliptic curves
- For any point P on the curve, $P + \mathcal{O} = P$ and $\mathcal{O} + P = P$
- The identity element is the neutral element under the addition operation

Inverse elements

- For any point P on an elliptic curve, there exists an inverse element $-P$ such that $P + (-P) = \mathcal{O}$
- The inverse of a point $P = (x, y)$ is given by $-P = (x, -y)$, the reflection of P across the x-axis

- The existence of inverse elements makes the group law on elliptic curves an abelian group

Geometric interpretation of group law

- The group law on elliptic curves has a beautiful geometric interpretation
- Adding points corresponds to drawing lines and finding their intersections with the curve
- The chord-tangent construction provides a visual way to understand the addition operation
- The geometric interpretation is useful for proving properties of the group law and for visualizing cryptographic protocols

Isomorphisms of elliptic curves

- Isomorphisms are mappings between elliptic curves that preserve the group structure
- Studying isomorphisms is essential for understanding the equivalence classes of elliptic curves and their moduli spaces
- Isomorphisms are used to classify elliptic curves and to simplify their equations

Changes of variables

- Elliptic curves can be transformed into different forms by applying changes of variables
- Linear changes of variables (e.g., $x \mapsto u^2x + r$, $y \mapsto u^3y + su^2x + t$) preserve the Weierstrass form of the equation
- Quadratic changes of variables (e.g., $x \mapsto \frac{x}{z}$, $y \mapsto \frac{y}{z}$) lead to projective forms of the equation (e.g., Hesse normal form)

Admissible changes of coordinates

- Admissible changes of coordinates are changes of variables that preserve the group structure of an elliptic curve
- For Weierstrass equations, admissible changes are linear changes with $u \neq 0$
- Admissible changes form a group called the group of admissible changes of coordinates
- The group of admissible changes is isomorphic to the projective linear group $\text{PGL}(2, K)$ over the base field K

Isomorphism classes

- Two elliptic curves are isomorphic if there exists an admissible change of coordinates that transforms one curve into the other
- Isomorphic curves have the same group structure and share many properties (e.g., the same number of points over finite fields)
- Isomorphism classes partition the set of all elliptic curves into equivalence classes
- The moduli space of elliptic curves parametrizes the isomorphism classes

j-invariant

- The j-invariant is a numerical invariant that characterizes the isomorphism class of an elliptic curve
- For a Weierstrass equation $y^2 = x^3 + ax + b$, the j-invariant is given by $j = 1728 \frac{4a^3}{4a^3 + 27b^2}$
- Two elliptic curves are isomorphic if and only if they have the same j-invariant
- The j-invariant is a crucial tool for classifying elliptic curves and studying their moduli spaces

Automorphisms of elliptic curves

- An automorphism of an elliptic curve is an isomorphism from the curve to itself
- The set of automorphisms of an elliptic curve forms a group under composition
- For elliptic curves over fields of characteristic not equal to 2 or 3, the automorphism group is either cyclic of order 2, 4, or 6
- The order of the automorphism group is related to the j-invariant and the presence of complex multiplication

Elliptic curves over various fields

- Elliptic curves can be defined over any field, and their properties depend on the characteristics of the underlying field
- Studying elliptic curves over different fields reveals rich connections to various branches of mathematics
- The choice of the field affects the group structure, torsion subgroups, and arithmetic properties of elliptic curves

Elliptic curves over real numbers

- Over the field of real numbers $\mathbb{R}$, elliptic curves have a geometric interpretation as curves in the real plane
- The real points of an elliptic curve form either one or two connected components (e.g., a single infinite component or a bounded component and an infinite component)
- The group law on real elliptic curves can be visualized using the chord-tangent construction
- Real elliptic curves are used in some cryptographic protocols and have applications in physics and engineering

Elliptic curves over complex numbers

- Over the field of complex numbers $\mathbb{C}$, elliptic curves have a rich geometric and analytic structure
- Complex elliptic curves are isomorphic to complex tori $\mathbb{C}/\Lambda$, where Λ is a lattice in the complex plane
- The group law on complex elliptic curves corresponds to the addition of complex numbers modulo the lattice
- Complex elliptic curves are central objects in complex analysis, algebraic geometry, and number theory (e.g., the Modularity Theorem)

Elliptic curves over finite fields

- Elliptic curves over finite fields $\mathbb{F}_q$ (where q is a prime power) have a finite number of points
- The group of points on an elliptic curve over a finite field is a finite abelian group
- The order of the group, denoted by $\#E(\mathbb{F}_q)$, satisfies the Hasse-Weil bound: $|q + 1 - \#E(\mathbb{F}_q)| \leq 2\sqrt{q}$
- Elliptic curves over finite fields are the foundation for elliptic curve cryptography (e.g., ECDSA, ECDH)

Elliptic curves over p-adic fields

- p-adic fields $\mathbb{Q}_p$ are completions of the rational numbers with respect to the p-adic norm
- Elliptic curves over p-adic fields have a rich arithmetic structure and are studied in p-adic analysis and p-adic Hodge theory
- The group of points on an elliptic curve over a p-adic field is an infinite abelian group
- p-adic elliptic curves are used in cryptography (e.g., p-adic point counting algorithms) and in the study of rational points on elliptic curves

Reduction mod p

- For an elliptic curve defined over the rational numbers $\mathbb{Q}$, the reduction mod p is the elliptic curve obtained by reducing the coefficients modulo a prime p
- The reduction mod p is not always an elliptic curve; it may be singular (i.e., have a cusp or a node)
- Primes for which the reduction is singular are called bad primes; others are called good primes
- The reduction type (good, multiplicative, or additive) provides information about the arithmetic properties of the elliptic curve (e.g., the Néron-Ogg-Shafarevich criterion)

Torsion points on elliptic curves

- Torsion points are points of finite order on an elliptic curve, i.e., points P such that $nP = \mathcal{O}$ for some positive integer n
- The set of torsion points forms a subgroup of the elliptic curve group, called the torsion subgroup
- Studying torsion points and torsion subgroups is crucial for understanding the arithmetic structure of elliptic curves

Torsion subgroups

- The torsion subgroup of an elliptic curve E over a field K , denoted by $E(K)_{\text{tors}}$, is the subgroup of points of finite order
- Over the complex numbers, the torsion subgroup is isomorphic to a product of two cyclic groups: $E(\mathbb{C})_{\text{tors}} \cong \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$, where m divides n
- Over the rational numbers, the torsion subgroup is isomorphic to one of 15 possible groups (by Mazur's theorem)
- The structure of the torsion subgroup provides information about the arithmetic and geometric properties of the elliptic curve

Nagell-Lutz theorem

- The Nagell-Lutz theorem gives a criterion for a point on an elliptic curve over the rational numbers to be a torsion point
- For an elliptic curve $y^2 = x^3 + ax + b$ with $a, b \in \mathbb{Z}$, a rational point (x, y) is a torsion point if and only if: 1. $x, y \in \mathbb{Z}$, and 2. either $y = 0$ or y^2 divides the discriminant $\Delta = -16(4a^3 + 27b^2)$
- The Nagell-Lutz theorem is a useful tool for finding torsion points and determining the torsion subgroup of an elliptic curve over $\mathbb{Q}$

Mazur's theorem on torsion

- Mazur's theorem classifies the possible torsion subgroups of elliptic curves over the rational numbers
- The torsion subgroup of an elliptic curve over $\mathbb{Q}$ is isomorphic to one of the following 15 groups:
 - $\mathbb{Z}/n\mathbb{Z}$ for $n = 1, 2, \dots, 10, 12$
 - $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2n\mathbb{Z}$ for $n = 1, 2, 3, 4$
- Mazur's theorem is a deep result in arithmetic geometry and has important consequences for the study of rational points on elliptic curves

Rational torsion points

- Rational torsion points are torsion points whose coordinates are rational numbers
- The set of rational torsion points forms a subgroup of the torsion subgroup, denoted by $E(\mathbb{Q})_{\text{tors}}$
- Finding rational torsion points is a fundamental problem in the arithmetic of elliptic curves
- Rational torsion points

6.2 Elliptic curves and projective geometry

Elliptic curves and projective geometry form a powerful duo in mathematics. They combine to create a rich framework for studying geometric objects with algebraic properties. This union allows for deeper insights into curve behavior, point counting, and group structures.

Projective space provides a natural setting for elliptic curves, offering a more complete view of their properties. By adding points at infinity and using homogeneous coordinates, we can explore the full range of curve behavior and develop efficient algorithms for cryptography and other applications.

Elliptic curves in projective space

- Projective space provides a natural setting for studying elliptic curves, allowing for a more unified treatment of their geometry and arithmetic
- Elliptic curves in projective space have additional structure and properties compared to their affine counterparts, such as the presence of points at infinity and a well-defined group law

Projective plane vs affine plane

- The projective plane extends the affine plane by adding points at infinity, providing a more complete geometric picture

- In the affine plane, parallel lines never intersect, while in the projective plane, parallel lines intersect at a point at infinity
- The projective plane can be constructed from the affine plane by adding a line at infinity, which consists of points representing the directions of lines in the affine plane

Homogeneous coordinates for projective space

- Homogeneous coordinates are used to represent points in projective space, allowing for a consistent treatment of points at infinity
- A point in projective space is represented by a triple $[X : Y : Z]$, where X , Y , and Z are not all zero, and $[X : Y : Z]$ is considered equivalent to $[\lambda X : \lambda Y : \lambda Z]$ for any non-zero scalar λ
- The affine plane can be embedded in the projective plane by setting $Z = 1$, so that (x, y) in the affine plane corresponds to $[x : y : 1]$ in the projective plane

Points at infinity on elliptic curves

- Elliptic curves in projective space have a unique point at infinity, denoted by O or $[0 : 1 : 0]$, which serves as the identity element for the group law
- The point at infinity arises from the intersection of the elliptic curve with the line at infinity in the projective plane
- Adding the point at infinity to an elliptic curve completes its group structure and allows for a more unified treatment of the curve's geometry and arithmetic

Group law in projective coordinates

- The group law on an elliptic curve in projective space is defined geometrically using the chord-and-tangent process, similar to the affine case
- To add two points P and Q on an elliptic curve in projective space, draw a line through P and Q (or a tangent line if $P = Q$), and find the third point of intersection R with the curve. The sum $P + Q$ is defined as the point obtained by reflecting R across the X -axis
- The point at infinity serves as the identity element for the group law, so that $P + O = P$ for any point P on the curve

Projective equations of elliptic curves

- Elliptic curves in projective space can be defined by homogeneous equations of the form $Y^2Z = X^3 + aXZ^2 + bZ^3$, where a and b are constants
- The projective equation is obtained from the affine Weierstrass equation $y^2 = x^3 + ax + b$ by homogenizing the variables: $x = X/Z$ and $y = Y/Z$
- The projective equation allows for the inclusion of the point at infinity, which corresponds to $[0 : 1 : 0]$ and satisfies the equation

Intersection of curves in projective space

- The intersection of curves in projective space is a fundamental concept in algebraic geometry, with important applications to the study of elliptic curves

- Intersecting curves in projective space can be analyzed using tools such as Bézout's theorem and intersection multiplicity, which provide insights into the number and nature of intersection points

Bézout's theorem

- Bézout's theorem states that two projective plane curves of degrees m and n intersect in exactly mn points, counted with multiplicity, unless the curves have a common component
- The theorem provides an upper bound on the number of intersection points between two curves in projective space
- Bézout's theorem is a powerful tool for studying the intersection of elliptic curves with other curves, such as lines and conics

Intersection multiplicity

- Intersection multiplicity is a measure of the complexity of the intersection between two curves at a given point
- If two curves intersect at a point with multiplicity k , then the curves share k coincident points at that intersection
- The concept of intersection multiplicity is crucial for understanding the behavior of curves at singular points and for computing the total number of intersections between curves

Tangent lines and multiplicity

- A tangent line to a curve at a point is a line that intersects the curve with multiplicity at least 2 at that point
- The multiplicity of the intersection between a curve and its tangent line at a point is related to the singularity of the curve at that point
- For elliptic curves, the tangent line at a point P intersects the curve at exactly one other point, which is used to define the group law on the curve

Singular points on curves

- A singular point on a curve is a point where the curve fails to be smooth, i.e., where the tangent line is not well-defined or has higher multiplicity
- Singular points on elliptic curves are classified into two types: nodes (ordinary double points) and cusps
- The presence and nature of singular points on an elliptic curve provide important information about its geometry and arithmetic, such as its conductor and reduction type

Elliptic curves as cubic curves

- Elliptic curves are a special class of cubic curves in projective space, characterized by their smooth structure and the presence of a distinguished point (the point at infinity)
- The study of elliptic curves as cubic curves allows for a deeper understanding of their geometric properties and their connections to other areas of mathematics, such as algebraic geometry and number theory

Weierstrass equations in projective space

- Elliptic curves in projective space can be described by homogeneous Weierstrass equations of the form $Y^2Z = X^3 + aXZ^2 + bZ^3$, where a and b are constants
- The projective Weierstrass equation is obtained by homogenizing the affine Weierstrass equation $y^2 = x^3 + ax + b$, allowing for the inclusion of the point at infinity
- The coefficients a and b in the Weierstrass equation determine the geometric and arithmetic properties of the elliptic curve, such as its discriminant and j-invariant

Projective transformations of Weierstrass form

- Projective transformations can be applied to the Weierstrass equation of an elliptic curve to simplify its form or to study its isomorphism class
- A projective transformation is a change of variables of the form $X = uX' + rZ'$, $Y = vY' + sZ'$, $Z = wZ'$, where u, v, w, r, s are constants with $uvw \neq 0$
- Projective transformations preserve the geometric structure of the elliptic curve, including its group law and the number of rational points

Discriminant and j-invariant

- The discriminant Δ of an elliptic curve in Weierstrass form is a polynomial expression in the coefficients a and b , given by $\Delta = -16(4a^3 + 27b^2)$
- The discriminant determines the singularity of the elliptic curve: the curve is smooth if and only if $\Delta \neq 0$, and it has a node or a cusp if $\Delta = 0$
- The j-invariant of an elliptic curve is a rational expression in the coefficients a and b , given by $j = 1728 \frac{4a^3}{4a^3 + 27b^2}$ (when $\Delta \neq 0$), which characterizes the isomorphism class of the curve

Isomorphism classes of elliptic curves

- Two elliptic curves are said to be isomorphic if there exists a projective transformation that maps one curve onto the other, preserving the group structure
- Isomorphic elliptic curves share the same j-invariant, and over an algebraically closed field, the j-invariant completely determines the isomorphism class of the curve
- Understanding isomorphism classes of elliptic curves is important for classification purposes and for studying their moduli spaces

Rational points on elliptic curves

- The study of rational points on elliptic curves is a central topic in arithmetic geometry, with connections to diophantine equations and cryptography
- Rational points are points on an elliptic curve whose coordinates are rational numbers, forming a finitely generated abelian group under the curve's group law

Mordell-Weil theorem

- The Mordell-Weil theorem states that the group of rational points on an elliptic curve over a number field is finitely generated, i.e., it is isomorphic to $\mathbb{Z}^r \oplus T$, where r is the rank of the curve and T is the torsion subgroup
- The rank of an elliptic curve measures the "size" of its rational point group and is a key invariant in the study of elliptic curves
- Computing the rank of an elliptic curve is a difficult problem, related to the Birch and Swinnerton-Dyer conjecture, one of the Millennium Prize Problems

Torsion subgroup of rational points

- The torsion subgroup of an elliptic curve over a field K consists of the points of finite order under the group law, i.e., points P such that $nP = O$ for some positive integer n
- Over $\mathbb{Q}$, the torsion subgroup of an elliptic curve is completely classified by Mazur's theorem, which states that it is isomorphic to one of 15 possible groups
- Studying the torsion subgroup of an elliptic curve provides insights into its arithmetic structure and can be used to construct rational points of infinite order

Height functions and descent

- Height functions are tools used to measure the "size" or "complexity" of rational points on an elliptic curve, providing a way to study the arithmetic properties of the curve
- The canonical height is a quadratic form on the group of rational points, satisfying certain properties such as the parallelogram law, and is used to define the regulator of the curve
- Descent is a technique used to compute the rank of an elliptic curve by studying its rational points modulo various primes and using local-global principles to obtain information about the global structure of the rational point group

Weak Mordell-Weil theorem

- The weak Mordell-Weil theorem states that the group of rational points on an elliptic curve over a number field is a finitely generated abelian group
- The weak Mordell-Weil theorem is a key step in the proof of the full Mordell-Weil theorem, providing a finiteness result for the rational point group
- The proof of the weak Mordell-Weil theorem relies on techniques from diophantine geometry, such as height functions and the Néron-Tate pairing, to bound the size of the rational point group

Elliptic curves over finite fields

- The study of elliptic curves over finite fields has important applications in cryptography and coding theory, as well as connections to other areas of mathematics, such as number theory and algebraic geometry
- Over a finite field $\mathbb{F}_q$, an elliptic curve is a smooth projective curve of genus 1 with a distinguished point, and its rational points form a finite abelian group

Projective coordinates over finite fields

- Projective coordinates are particularly useful for studying elliptic curves over finite fields, as they allow for efficient computation and avoid the need for division operations
- In projective coordinates, an elliptic curve over $\mathbb{F}_q$ can be described by a homogeneous Weierstrass equation of the form $Y^2Z = X^3 + aXZ^2 + bZ^3$, where $a, b \in \mathbb{F}_q$
- Point addition and doubling formulas in projective coordinates involve only field operations and can be optimized for efficient implementation in cryptographic protocols

Group structure of elliptic curves over finite fields

- The group of $\mathbb{F}_q$ -rational points on an elliptic curve, denoted by $E(\mathbb{F}_q)$, is a finite abelian group with a well-defined group law
- The structure of $E(\mathbb{F}_q)$ is determined by the Hasse-Weil bound, which states that the order of the group satisfies $|E(\mathbb{F}_q)| = q + 1 - t$, where $|t| \leq 2\sqrt{q}$
- The group structure of $E(\mathbb{F}_q)$ can be either cyclic or the product of two cyclic groups, depending on the number of points and the presence of torsion points

Hasse's theorem on point counts

- Hasse's theorem provides a precise bound on the number of $\mathbb{F}_q$ -rational points on an elliptic curve, stating that $|E(\mathbb{F}_q)| = q + 1 - t$, where $|t| \leq 2\sqrt{q}$
- The value of t in Hasse's theorem is called the trace of Frobenius and is related to the zeta function of the elliptic curve and its L-function
- Hasse's theorem has important consequences for the security of elliptic curve cryptography, as it guarantees that the group of rational points has a sufficiently large size and a well-distributed structure

Supersingular vs ordinary elliptic curves

- Elliptic curves over finite fields can be classified into two types: supersingular and ordinary, based on their endomorphism ring structure and their behavior under the Frobenius endomorphism
- An elliptic curve E over $\mathbb{F}_q$ is called supersingular if the trace of Frobenius satisfies $t \equiv 0 \pmod{p}$, where $q = p^n$ for some prime p and integer n ; otherwise, it is called ordinary
- Supersingular elliptic curves have special properties, such as a larger automorphism group and a simpler group structure, and have applications in cryptography and coding theory, while ordinary curves are more commonly used in general elliptic curve cryptography

6.3 Elliptic curves and the Riemann-Roch theorem

Elliptic curves are smooth, projective algebraic curves of genus one with a specified basepoint. They have a rich structure that makes them useful in number theory, cryptography, and algebraic geometry.

The Riemann-Roch theorem is a fundamental result for elliptic curves. It relates the dimension of rational function spaces to divisor degrees, providing insights into the curve's geometry and arithmetic properties.

Definition of elliptic curves

- Elliptic curves are smooth, projective, algebraic curves of genus one with a specified basepoint

- They have a rich geometric and algebraic structure that makes them useful in various fields of mathematics, including number theory, cryptography, and algebraic geometry

Weierstrass equations

- Elliptic curves can be defined by Weierstrass equations of the form $y^2 = x^3 + ax + b$, where a and b are constants satisfying certain conditions
- The Weierstrass equation provides a canonical form for representing elliptic curves
- The discriminant $\Delta = -16(4a^3 + 27b^2)$ must be nonzero for the curve to be smooth

Group law on elliptic curves

- Elliptic curves have a natural group structure, where the group operation is defined geometrically
- The group law allows for the addition of points on the elliptic curve
- The group law is associative, has an identity element (the point at infinity), and every point has an inverse

Elliptic curves over finite fields

- Elliptic curves can be defined over finite fields $\mathbb{F}_q$, where q is a prime power
- The group of rational points on an elliptic curve over a finite field is a finite abelian group
- Elliptic curves over finite fields have applications in cryptography (elliptic curve cryptography) due to their discrete logarithm problem being hard to solve

Divisors on elliptic curves

- Divisors are formal sums of points on an elliptic curve, used to study the arithmetic and geometry of the curve
- They provide a way to describe the zeros and poles of rational functions and differentials on the curve

Definition of divisors

- A divisor on an elliptic curve E is a formal sum $D = \sum n_P P$, where P runs over the points of E and n_P are integers, with only finitely many n_P nonzero
- The set of divisors on E forms an abelian group under pointwise addition

Degree of divisors

- The degree of a divisor $D = \sum n_P P$ is defined as $\deg(D) = \sum n_P$
- The degree map is a homomorphism from the group of divisors to the integers
- The set of divisors of degree zero is a subgroup of the divisor group

Principal divisors

- A principal divisor is a divisor of the form $\text{div}(f) = \sum \text{ord}_P(f) P$, where f is a nonzero rational function on the elliptic curve and $\text{ord}_P(f)$ is the order of vanishing of f at P

- Principal divisors have degree zero and form a subgroup of the divisor group

Divisor class group

- The divisor class group (or Picard group) of an elliptic curve is the quotient group of the divisor group by the subgroup of principal divisors
- Two divisors are linearly equivalent if their difference is a principal divisor
- The divisor class group is a finite abelian group that encodes important information about the elliptic curve

Riemann-Roch theorem for elliptic curves

- The Riemann-Roch theorem is a fundamental result in the theory of algebraic curves, relating the dimension of the space of rational functions with prescribed poles to the degree of a divisor

Statement of Riemann-Roch theorem

- For an elliptic curve E and a divisor D on E , the Riemann-Roch theorem states that $\dim L(D) = \deg(D) + 1 - g + \dim L(K - D)$, where:
- $L(D)$ is the Riemann-Roch space associated to D
- g is the genus of the curve (which is 1 for elliptic curves)
- K is a canonical divisor

Proof of Riemann-Roch theorem

- The proof of the Riemann-Roch theorem for elliptic curves relies on the properties of the divisor class group and the Serre duality theorem
- Key steps involve showing the invariance of the Euler characteristic $\chi(D) = \dim L(D) - \dim L(K - D)$ and using the fact that the canonical divisor has degree $2g - 2$

Applications of Riemann-Roch theorem

- The Riemann-Roch theorem allows for the computation of the dimension of Riemann-Roch spaces
- It provides a way to study the behavior of rational functions and differentials on elliptic curves
- The theorem has applications in the theory of algebraic curves, algebraic geometry, and number theory (Mordell-Weil theorem)

Rational functions on elliptic curves

- Rational functions on elliptic curves are important objects in the study of the curve's geometry and arithmetic
- They are used to describe maps between elliptic curves and to study the divisor class group

Definition of rational functions

- A rational function on an elliptic curve E is a function $f: E \rightarrow \mathbb{P}^1$ that can be written as the ratio of two polynomials in the coordinates of the curve

- The set of rational functions on E forms a field, denoted by $K(E)$

Poles and zeros of rational functions

- The zeros of a rational function f are the points P on the elliptic curve where $f(P) = 0$
- The poles of f are the points P where f has a singularity (i.e., $f(P) = \infty$)
- The order of a zero or pole is the multiplicity of the point in the divisor of f

Riemann-Roch spaces

- For a divisor D on an elliptic curve E , the Riemann-Roch space $L(D)$ is the vector space of rational functions f on E such that $\text{div}(f) + D \geq 0$
- The dimension of $L(D)$ is given by the Riemann-Roch theorem
- Riemann-Roch spaces are key objects in the study of the geometry of elliptic curves

Canonical divisors

- Canonical divisors play a central role in the theory of algebraic curves and the Riemann-Roch theorem
- They are closely related to the concept of differentials on the curve

Definition of canonical divisors

- A canonical divisor on an elliptic curve E is a divisor K such that $L(K)$ is the space of holomorphic differentials on E
- On an elliptic curve, any divisor of the form $K = (P) - (O)$, where P is a point on E and O is the point at infinity, is a canonical divisor

Degree of canonical divisors

- The degree of a canonical divisor on an elliptic curve is always equal to $2g - 2$, where g is the genus of the curve
- For elliptic curves, the genus is 1, so the degree of a canonical divisor is 0

Canonical class

- The canonical class of an elliptic curve is the linear equivalence class of canonical divisors
- All canonical divisors on an elliptic curve are linearly equivalent
- The canonical class is an important invariant of the elliptic curve

Differentials on elliptic curves

- Differentials on elliptic curves are closely related to the concept of canonical divisors and play a role in the Riemann-Roch theorem
- They provide a way to study the geometry and arithmetic of the curve

Definition of differentials

- A differential on an elliptic curve E is a global section of the cotangent bundle of E

- In terms of the Weierstrass equation, a differential can be written as $\omega = dx/(2y)$

Holomorphic vs meromorphic differentials

- A holomorphic differential on an elliptic curve is a differential that has no poles
- A meromorphic differential is a differential that may have poles
- On an elliptic curve, there is a unique (up to scalar multiplication) holomorphic differential

Riemann-Roch theorem for differentials

- The Riemann-Roch theorem can be formulated in terms of differentials
- For a divisor D on an elliptic curve E , the theorem states that $\dim \Omega(D) = \deg(D) - g + 1 + \dim L(K - D)$, where:
- $\Omega(D)$ is the space of meromorphic differentials ω such that $\operatorname{div}(\omega) \geq -D$
- g is the genus of the curve (which is 1 for elliptic curves)
- K is a canonical divisor

Elliptic curves over $\mathbb{C}$

- The study of elliptic curves over the complex numbers $\mathbb{C}$ provides a rich interplay between complex analysis, algebraic geometry, and number theory
- Many properties of elliptic curves can be understood by studying their complex analytic structure

Lattices and elliptic curves

- An elliptic curve over $\mathbb{C}$ can be realized as the quotient of $\mathbb{C}$ by a lattice Λ
- A lattice is a discrete subgroup of $\mathbb{C}$ of the form $\Lambda = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$, where ω_1 and ω_2 are complex numbers linearly independent over $\mathbb{R}$
- The Weierstrass $\wp$ -function associated to a lattice Λ provides a parametrization of the elliptic curve $\mathbb{C}/\Lambda$

Isomorphisms between elliptic curves

- Two elliptic curves over $\mathbb{C}$ are isomorphic if and only if their associated lattices are homothetic (i.e., one is a scalar multiple of the other)
- The j -invariant of an elliptic curve is a complex number that characterizes the isomorphism class of the curve
- Elliptic curves with the same j -invariant are isomorphic over $\mathbb{C}$

Complex tori and elliptic curves

- A complex torus is a quotient of $\mathbb{C}^n$ by a lattice of rank $2n$
- Elliptic curves over $\mathbb{C}$ are precisely the complex tori of dimension 1
- The study of complex tori and their morphisms is a central topic in complex algebraic geometry

Elliptic curves over $\mathbb{Q}$

- The study of elliptic curves over the rational numbers $\mathbb{Q}$ is of great interest in number theory
- Many Diophantine equations can be reformulated as questions about rational points on elliptic curves

Mordell-Weil theorem

- The Mordell-Weil theorem states that the group of rational points on an elliptic curve over $\mathbb{Q}$ is finitely generated
- The group $E(\mathbb{Q})$ is isomorphic to $\mathbb{Z}^r \oplus E(\mathbb{Q})_{\text{tors}}$, where r is the rank of the curve and $E(\mathbb{Q})_{\text{tors}}$ is the torsion subgroup
- Computing the rank of an elliptic curve is a difficult problem, and there is no known algorithm for determining it in general

Torsion points on elliptic curves

- A torsion point on an elliptic curve E over $\mathbb{Q}$ is a point of finite order in the group $E(\mathbb{Q})$
- The torsion subgroup $E(\mathbb{Q})_{\text{tors}}$ is always finite, and its possible structures are known (Mazur's theorem)
- Torsion points play a role in the study of rational points on elliptic curves and in applications to cryptography

Height functions and descent

- Height functions are tools used to measure the "size" of rational points on an elliptic curve
- The canonical height is a quadratic form on $E(\mathbb{Q})$ that behaves well under the group law and can be used to prove the Mordell-Weil theorem
- Descent is a method for computing the rank of an elliptic curve by studying the curve's torsion points and isogenies
- The method of descent provides a way to bound the rank of an elliptic curve and to search for rational points

6.4 Elliptic curves and the Weierstrass $\wp$ -function

Elliptic curves are fascinating mathematical objects with deep connections to number theory and cryptography. Their study combines algebra, geometry, and complex analysis, revealing intricate relationships between seemingly disparate areas of mathematics.

The Weierstrass $\wp$ -function is a key tool for understanding elliptic curves from a complex analytic perspective. It parametrizes elliptic curves and bridges the gap between their algebraic and analytic properties, enabling powerful techniques from both fields.

Elliptic curves

- Elliptic curves are cubic equations that have fascinating geometric and algebraic properties making them important objects of study in number theory and cryptography

- The study of elliptic curves connects various branches of mathematics, including algebra, geometry, complex analysis, and number theory
- Understanding the key components and properties of elliptic curves is essential for grasping their significance and applications

Weierstrass equation

- The Weierstrass equation is the standard form of an elliptic curve given by $y^2 = x^3 + ax + b$, where a and b are constants
- For an elliptic curve to be non-singular, the discriminant $\Delta = 4a^3 + 27b^2$ must be non-zero
- The Weierstrass equation allows for a unified treatment of elliptic curves over various fields (rational numbers, complex numbers, finite fields)

Rational points

- Rational points on an elliptic curve are points (X, Y) where both X and Y are rational numbers satisfying the Weierstrass equation
- The set of rational points on an elliptic curve, together with a special point called the "point at infinity," forms an abelian group under the group law
- Studying the structure and properties of the group of rational points is a central theme in elliptic curve theory (Mordell-Weil theorem, torsion subgroup)

Group law

- The group law on an elliptic curve is a geometric operation that allows us to "add" points on the curve
- Given two points P and Q on an elliptic curve, we can find a third point $R = P + Q$ by: 1. Drawing a line through P and Q (if $P = Q$, take the tangent line at P) 2. Finding the third intersection point of this line with the curve, called $-R$ 3. Reflecting $-R$ across the X -axis to obtain R
- The group law satisfies the axioms of an abelian group (associativity, identity element, inverse elements, commutativity)

Discriminant

- The discriminant of an elliptic curve, denoted by Δ , is a quantity that determines whether the curve is singular or non-singular
- For the Weierstrass equation $y^2 = x^3 + ax + b$, the discriminant is given by $\Delta = 4a^3 + 27b^2$
- If $\Delta \neq 0$, the elliptic curve is non-singular and has distinct roots; if $\Delta = 0$, the curve is singular and has a node or cusp singularity

j-invariant

- The j-invariant of an elliptic curve is a quantity that characterizes the isomorphism class of the curve
- Two elliptic curves are isomorphic over an algebraically closed field if and only if they have the same j-invariant
- The j-invariant is given by $j = 1728 \frac{4a^3}{4a^3 + 27b^2}$ for the Weierstrass equation $y^2 = x^3 + ax + b$

Weierstrass $\wp$ -function

- The Weierstrass $\wp$ -function is a doubly periodic meromorphic function that parametrizes elliptic curves and plays a crucial role in the theory of elliptic functions
- Understanding the properties and connections of the $\wp$ -function is essential for studying elliptic curves from a complex analytic perspective
- The $\wp$ -function bridges the gap between elliptic curves and complex analysis, enabling the use of powerful tools from both fields

Lattices in complex plane

- A lattice in the complex plane is a discrete subgroup of $\mathbb{C}$ generated by two linearly independent complex numbers ω_1 and ω_2
- The lattice $\Lambda = \{n_1\omega_1 + n_2\omega_2 : n_1, n_2 \in \mathbb{Z}\}$ forms a periodic grid in the complex plane
- Lattices are fundamental objects in the study of elliptic functions and complex tori

Definition of $\wp$ -function

- The Weierstrass $\wp$ -function associated with a lattice Λ is defined as:
$$\wp(z; \Lambda) = \frac{1}{z^2} + \sum_{\omega \in \Lambda \setminus \{0\}} \left(\frac{1}{(z-\omega)^2} - \frac{1}{\omega^2} \right)$$
- The $\wp$ -function is an even meromorphic function with double poles at each lattice point
- The $\wp$ -function satisfies the differential equation $(\wp')^2 = 4\wp^3 - g_2\wp - g_3$, where g_2 and g_3 are lattice invariants

Periodicity and poles

- The $\wp$ -function is doubly periodic with periods ω_1 and ω_2 , meaning that $\wp(z + \omega_1) = \wp(z + \omega_2) = \wp(z)$ for all $z \in \mathbb{C}$
- The $\wp$ -function has double poles at each lattice point, and its Laurent series expansion around $z = 0$ is given by:

$$\wp(z) = \frac{1}{z^2} + \sum_{k=1}^{\infty} (2k+1)G_{2k+2}z^{2k}$$

, where G_k are the Eisenstein series

- The residue of $\wp$ at each pole is zero, and the sum of residues over a fundamental parallelogram of the lattice vanishes

Differential equation

- The $\wp$ -function satisfies the differential equation $(\wp')^2 = 4\wp^3 - g_2\wp - g_3$, where g_2 and g_3 are lattice invariants
- The lattice invariants g_2 and g_3 can be expressed in terms of the Eisenstein series G_4 and G_6 as:
 $g_2 = 60G_4$ and $g_3 = 140G_6$
- The differential equation connects the $\wp$ -function with the Weierstrass equation of an elliptic curve

Relation to Weierstrass equation

- The $\wp$ -function and its derivative $\wp'$ parametrize the elliptic curve given by the Weierstrass equation $y^2 = 4x^3 - g_2x - g_3$
- For any $z \in \mathbb{C} \setminus \Lambda$, the point $(\wp(z), \wp'(z))$ lies on the elliptic curve
- This parametrization establishes an isomorphism between the complex torus $\mathbb{C}/\Lambda$ and the elliptic curve

Homogeneity

- The $\wp$ -function satisfies the homogeneity property $\wp(\lambda z; \lambda \Lambda) = \lambda^{-2} \wp(z; \Lambda)$ for any non-zero complex number λ
- The lattice invariants g_2 and g_3 scale under homogeneity as $g_2(\lambda \Lambda) = \lambda^{-4} g_2(\Lambda)$ and $g_3(\lambda \Lambda) = \lambda^{-6} g_3(\Lambda)$
- Homogeneity allows for the normalization of the lattice and simplifies the study of elliptic functions and curves

Elliptic functions

- Elliptic functions are meromorphic functions that are doubly periodic with respect to a lattice in the complex plane
- The study of elliptic functions is closely related to the theory of elliptic curves and provides a rich set of tools for understanding their properties
- Elliptic functions have important applications in various areas of mathematics, including complex analysis, number theory, and mathematical physics

Definitions and examples

- An elliptic function is a meromorphic function f that satisfies $f(z + \omega_1) = f(z + \omega_2) = f(z)$ for all $z \in \mathbb{C}$ and some linearly independent $\omega_1, \omega_2 \in \mathbb{C}$
- Examples of elliptic functions include the Weierstrass $\wp$ -function, the Jacobi elliptic functions (sn, cn, dn), and the Weierstrass ζ -function
- Elliptic functions can be constructed using the Weierstrass $\wp$ -function and its derivative, as well as through infinite products and series

Relation to elliptic curves

- Elliptic functions are intimately connected to elliptic curves through the Weierstrass $\wp$ -function and its derivative
- The $\wp$ -function and $\wp'$ parametrize the elliptic curve given by the Weierstrass equation $y^2 = 4x^3 - g_2x - g_3$
- The group law on the elliptic curve corresponds to the addition of complex numbers modulo the lattice associated with the elliptic function

Poles and residues

- Elliptic functions have a finite number of poles in each fundamental parallelogram of the lattice
- The sum of the residues of an elliptic function at its poles in a fundamental parallelogram is always zero
- The number of poles (counting multiplicity) in a fundamental parallelogram is equal to the order of the elliptic function

Liouville's theorem

- Liouville's theorem states that every doubly periodic meromorphic function (elliptic function) is either constant or takes on each value in the complex plane (including ∞) the same number of times in each fundamental parallelogram
- The order of an elliptic function is equal to the number of times it takes on any value (counting multiplicity) in a fundamental parallelogram
- Liouville's theorem imposes strong constraints on the possible forms of elliptic functions and helps in their classification and study

Complex tori

- A complex torus is a quotient of the complex plane by a lattice, resulting in a compact Riemann surface with the structure of an abelian group
- Complex tori are fundamental objects in the study of elliptic curves and provide a geometric perspective on their properties
- The study of complex tori connects elliptic curves with the theory of Riemann surfaces and algebraic geometry

Lattices and parallelograms

- A lattice Λ in the complex plane is a discrete subgroup generated by two linearly independent complex numbers ω_1 and ω_2
- A fundamental parallelogram for the lattice Λ is a parallelogram with vertices 0 , ω_1 , ω_2 , and $\omega_1 + \omega_2$, which generates the entire lattice through translations
- The complex torus $\mathbb{C}/\Lambda$ is obtained by identifying opposite sides of the fundamental parallelogram, resulting in a compact Riemann surface

Doubly periodic functions

- Doubly periodic functions on a complex torus are functions that satisfy $f(z + \omega_1) = f(z + \omega_2) = f(z)$ for all $z \in \mathbb{C}$ and the lattice generators ω_1, ω_2
- Elliptic functions are examples of doubly periodic meromorphic functions on a complex torus
- The study of doubly periodic functions on complex tori is closely related to the theory of elliptic functions and provides insights into their properties

Isomorphism with elliptic curves

- There is a natural isomorphism between a complex torus $\mathbb{C}/\Lambda$ and the elliptic curve given by the Weierstrass equation associated with the lattice Λ
- The Weierstrass $\wp$ -function and its derivative provide an explicit isomorphism by mapping $z \in \mathbb{C}/\Lambda$ to the point $(\wp(z), \wp'(z))$ on the elliptic curve
- This isomorphism allows for the transfer of geometric and algebraic properties between complex tori and elliptic curves

Uniformization theorem

- The uniformization theorem states that every Riemann surface is conformally equivalent to either the Riemann sphere, the complex plane, or a complex torus
- In the case of elliptic curves, the uniformization theorem guarantees the existence of a lattice Λ such that the elliptic curve is isomorphic to the complex torus $\mathbb{C}/\Lambda$
- The uniformization theorem provides a powerful tool for studying elliptic curves and their related structures from a complex analytic perspective

Modular forms

- Modular forms are complex analytic functions defined on the upper half-plane that satisfy certain transformation properties under the action of the modular group $SL(2, \mathbb{Z})$
- The study of modular forms is a central topic in number theory and has deep connections with elliptic curves, complex analysis, and representation theory
- Modular forms play a crucial role in the proof of Fermat's Last Theorem and the Modularity Theorem for elliptic curves over rational numbers

Modular group $SL(2, \mathbb{Z})$

- The modular group $SL(2, \mathbb{Z})$ is the group of 2×2 matrices with integer entries and determinant 1
- Elements of $SL(2, \mathbb{Z})$ act on the upper half-plane $\mathbb{H} = \{\tau \in \mathbb{C} : \text{Im}(\tau) > 0\}$ by linear fractional transformations: $\tau \mapsto \frac{a\tau + b}{c\tau + d}$ for

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in SL(2, \mathbb{Z})$$

- The modular group is generated by the matrices

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

, which correspond to the transformations $\tau \mapsto \tau + 1$ and $\tau \mapsto -\frac{1}{\tau}$, respectively

Modular functions and forms

- A modular function is a meromorphic function f on the upper half-plane that satisfies $f\left(\frac{a\tau+b}{c\tau+d}\right) = f(\tau)$ for all

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in SL(2, \mathbb{Z})$$

and is meromorphic at the cusps (rational points on the real line and $i\infty$)

- A modular form of weight k is a holomorphic function f on the upper half-plane that satisfies $f\left(\frac{a\tau+b}{c\tau+d}\right) = (c\tau+d)^k f(\tau)$ for all

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in SL(2, \mathbb{Z})$$

and is holomorphic at the cusps

- Examples of modular functions include the j -invariant and the Hauptmodul for certain congruence subgroups, while examples of modular forms include the Eisenstein series and the Dedekind η -function

Eisenstein series

- The Eisenstein series G_{2k} of weight $2k$ ($k \geq 2$) are modular forms defined by:

$$G_{2k}(\tau) = \sum_{(m,n) \in \mathbb{Z}^2 \setminus (0,0)} \frac{1}{(m\tau+n)^{2k}}$$
- The Eisenstein series G_4 and G_6 are particularly important, as they appear in the Weierstrass equation of an elliptic curve and are related to the lattice invariants g_2 and g_3
- Eisenstein series have Fourier expansions with rational coefficients and satisfy certain congruence relations, making them useful in the study of modular forms and elliptic curves

Dedekind η -function

- The Dedekind η -function is a modular form of weight $1/2$ defined by:

$$\eta(\tau) = q^{1/24} \prod_{n=1}^{\infty} (1 - q^n)$$

, where $q = e^{2\pi i \tau}$

- The η -function satisfies

6.5 Elliptic curves and the group structure of the Picard group

Elliptic curves are fascinating mathematical objects with deep connections to number theory and geometry. They're defined by special equations and have a unique group structure that makes them powerful tools in cryptography and other fields.

The Picard group of an elliptic curve captures important information about its structure and properties. It's closely related to the group of rational points on the curve and plays a key role in understanding the arithmetic and geometric aspects of elliptic curves.

Definition of elliptic curves

- Elliptic curves are a fundamental object of study in number theory and algebraic geometry

- They have a rich structure and deep connections to various areas of mathematics
- Understanding the definition and basic properties of elliptic curves is crucial for this course

Weierstrass equation

- An elliptic curve can be defined by a Weierstrass equation of the form $y^2 = x^3 + ax + b$, where a and b are constants satisfying certain conditions
- The Weierstrass equation provides a convenient algebraic representation of elliptic curves
- The constants a and b determine the shape and properties of the curve (e.g., $y^2 = x^3 - x$ defines a specific elliptic curve)

Smooth projective curves

- Elliptic curves are smooth projective curves of genus one
- Smoothness means that the curve has no singular points or self-intersections
- Projective curves are defined in projective space, which allows for the inclusion of points at infinity
- The genus of a curve is a measure of its complexity, and elliptic curves have genus one

Rational points

- Rational points on an elliptic curve are points with rational coordinates that satisfy the Weierstrass equation
- The set of rational points on an elliptic curve forms a group under a certain operation called the group law
- Finding rational points on elliptic curves is a central problem in number theory (e.g., finding integer solutions to $y^2 = x^3 - 2$)

The group law

- The group law is a fundamental operation that defines the structure of rational points on an elliptic curve
- It allows us to add and subtract points on the curve, forming a group structure
- Understanding the group law is essential for studying the arithmetic and cryptographic applications of elliptic curves

Geometric definition

- The group law can be defined geometrically using the chord-and-tangent method
- To add two points P and Q on the curve, draw a line through them and find the third point of intersection with the curve, then reflect that point across the x-axis to obtain $P + Q$
- If $P = Q$, the line is taken to be the tangent line at P , and the process is similar

Algebraic formulas

- The group law can also be expressed using algebraic formulas derived from the Weierstrass equation

- These formulas allow for efficient computation of point addition and doubling
- The formulas involve the coordinates of the points and the coefficients of the elliptic curve equation

Associativity and commutativity

- The group law on elliptic curves is associative, meaning that $(P + Q) + R = P + (Q + R)$ for any points P , Q , and R
- It is also commutative, meaning that $P + Q = Q + P$ for any points P and Q
- These properties ensure that the set of rational points forms a well-defined group structure

Identity element

- The identity element of the group law is the point at infinity, often denoted as $\mathcal{O}$
- Adding any point P to the identity element results in P itself: $P + \mathcal{O} = P$
- The point at infinity serves as the neutral element in the group structure

Inverses

- For any point P on the elliptic curve, there exists an inverse point $-P$ such that $P + (-P) = \mathcal{O}$
- The inverse of a point can be found by reflecting it across the x-axis
- The existence of inverses ensures that every point has a unique "negative" point in the group

The Picard group

- The Picard group is a fundamental object associated with an algebraic curve, including elliptic curves
- It captures information about the divisors on the curve and their equivalence classes
- Understanding the Picard group is crucial for studying the arithmetic and geometric properties of elliptic curves

Divisors on curves

- A divisor on an algebraic curve is a formal sum of points on the curve with integer coefficients
- Divisors can be thought of as a way to describe the zeros and poles of functions on the curve
- The set of all divisors on an elliptic curve forms an abelian group under addition

Principal divisors

- A principal divisor is a divisor that arises from a rational function on the curve
- It is defined as the sum of the zeros and poles of the function, counted with multiplicities
- Principal divisors form a subgroup of the group of all divisors on the curve

Degree zero divisors

- A divisor is said to have degree zero if the sum of its coefficients is equal to zero
- The set of degree zero divisors on an elliptic curve is denoted by $\text{Div}^0(E)$

- Degree zero divisors play a crucial role in the definition of the Picard group

Equivalence of divisors

- Two divisors are said to be equivalent if their difference is a principal divisor
- Equivalent divisors can be thought of as differing by a rational function
- The equivalence of divisors defines an equivalence relation on the set of divisors

Structure of the Picard group

- The Picard group of an elliptic curve, denoted by $\text{Pic}(E)$, is the quotient group of degree zero divisors modulo principal divisors
- It captures important information about the curve and its rational points
- Understanding the structure of the Picard group is essential for studying the arithmetic and geometric properties of elliptic curves

Pic^0 vs Pic

- The Picard group can be decomposed into two parts: $\text{Pic}^0(E)$ and $\text{Pic}(E)$
- $\text{Pic}^0(E)$ consists of divisors of degree zero modulo principal divisors, and it is isomorphic to the group of rational points on the curve
- $\text{Pic}(E)$ includes divisors of any degree modulo principal divisors and contains $\text{Pic}^0(E)$ as a subgroup

Torsion points

- Torsion points on an elliptic curve are points of finite order in the Picard group
- They are points P such that $nP = \mathcal{O}$ for some positive integer n
- The set of torsion points forms a finite subgroup of the Picard group

Rank of elliptic curves

- The rank of an elliptic curve is the number of independent points of infinite order in the Picard group
- It measures the "size" of the infinite part of the group of rational points
- Determining the rank of an elliptic curve is a difficult problem in number theory

Mordell-Weil theorem

- The Mordell-Weil theorem states that the group of rational points on an elliptic curve (i.e., $\text{Pic}^0(E)$) is finitely generated
- It implies that the group can be written as a direct sum of a finite torsion subgroup and a free abelian group of rank equal to the rank of the curve
- The theorem is a fundamental result in the study of elliptic curves and their rational points

Isogenies

- Isogenies are morphisms between elliptic curves that preserve the group structure

- They provide a way to relate different elliptic curves and study their properties
- Understanding isogenies is important for various applications, including cryptography and the study of elliptic curve algorithms

Definition and properties

- An isogeny between two elliptic curves E_1 and E_2 is a non-constant rational map that preserves the group structure
- Isogenies are always surjective and have finite kernel
- The degree of an isogeny is the degree of the corresponding rational map

Dual isogenies

- For every isogeny $\phi : E_1 \rightarrow E_2$, there exists a unique dual isogeny $\hat{\phi} : E_2 \rightarrow E_1$
- The composition of an isogeny with its dual is the multiplication-by-degree map on the source curve
- Dual isogenies provide a way to "invert" isogenies and study their properties

Isogeny kernels

- The kernel of an isogeny $\phi : E_1 \rightarrow E_2$ is the set of points on E_1 that map to the identity element on E_2
- The kernel of an isogeny is always a finite subgroup of the source curve
- Studying isogeny kernels is important for understanding the structure of isogenies and their applications

Isogeny classes

- Two elliptic curves are said to be isogenous if there exists an isogeny between them
- The isogeny class of an elliptic curve consists of all curves isogenous to it
- Isogeny classes partition the set of elliptic curves and provide a way to classify them

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a modern approach to public-key cryptography based on the arithmetic of elliptic curves
- It offers similar security to traditional cryptosystems (e.g., RSA) with smaller key sizes
- Understanding the mathematical foundations of ECC is crucial for implementing and analyzing secure cryptographic protocols

Discrete logarithm problem

- The security of ECC relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)
- Given points P and Q on an elliptic curve, the ECDLP asks to find an integer k such that $Q = kP$
- The hardness of the ECDLP is the basis for the security of ECC protocols

Diffie-Hellman key exchange

- The Diffie-Hellman key exchange protocol can be adapted to work with elliptic curves
- Two parties can establish a shared secret key by performing scalar multiplications on a publicly known base point
- The elliptic curve Diffie-Hellman (ECDH) protocol is widely used for secure key establishment

Elliptic curve digital signatures

- Elliptic curve digital signature algorithms (ECDSA) provide a way to create and verify digital signatures using elliptic curves
- ECDSA signatures are compact and efficient, making them suitable for resource-constrained environments
- The security of ECDSA relies on the hardness of the ECDLP and the proper choice of elliptic curve parameters

6.6 Elliptic curves and the Mordell-Weil theorem

Elliptic curves are fascinating mathematical objects with deep connections to number theory and algebraic geometry. They're defined by special equations and have a unique group structure that allows for addition of points on the curve.

The Mordell-Weil theorem is a key result about elliptic curves. It states that the group of rational points on an elliptic curve over a number field is finitely generated, revealing important insights about the curve's structure.

Definition of elliptic curves

- Elliptic curves are a fundamental object of study in number theory and algebraic geometry
- They have a rich structure and deep connections to various areas of mathematics
- Understanding the basic definitions and properties of elliptic curves is essential for studying the Mordell-Weil theorem

Weierstrass equations

- Elliptic curves can be defined by Weierstrass equations of the form $y^2 = x^3 + ax + b$, where a and b are constants satisfying certain conditions
- The Weierstrass equation provides a standard way to represent elliptic curves algebraically
- The coefficients a and b determine the shape and properties of the curve (e.g., $y^2 = x^3 - x$ defines a curve with two components)

Smooth projective curves

- Elliptic curves are smooth projective curves of genus one with a specified base point
- Smoothness means the curve has no singularities or self-intersections
- Projective curves are defined in projective space, which allows for points at infinity (e.g., the point $[0 : 1 : 0]$ on the projective closure of the Weierstrass equation)

- The genus of a curve is a measure of its complexity, and elliptic curves have genus one

Rational points

- Elliptic curves are studied over various fields, but the most interesting case is often over the rational numbers $\mathbb{Q}$
- A rational point on an elliptic curve is a point whose coordinates are rational numbers
- The set of rational points on an elliptic curve forms a group under a certain operation called the group law (e.g., the points $(0, 0)$ and $(1, 0)$ on the curve $y^2 = x^3 - x$)

Group law on elliptic curves

- The set of points on an elliptic curve, together with a special point called the point at infinity, forms an abelian group under the group law
- The group law allows us to add points on the curve and obtain another point on the curve
- Understanding the group structure of elliptic curves is crucial for studying their arithmetic properties

Geometric description

- The group law on elliptic curves can be described geometrically using the chord-and-tangent process
- To add two points P and Q , draw a line through them and find the third point of intersection with the curve, then reflect that point across the X -axis to obtain $P + Q$
- If $P = Q$, draw the tangent line at P and follow the same process (e.g., adding the points $(0, 0)$ and $(1, 0)$ on the curve $y^2 = x^3 - x$)

Algebraic formulas

- The group law can also be described algebraically using explicit formulas in terms of the coordinates of the points
- These formulas involve rational functions and can be derived from the geometric description
- The algebraic formulas are useful for computations and implementing the group law in practice (e.g., the formula for the X -coordinate of $P + Q$ in terms of the coordinates of P and Q)

Associativity and identity element

- The group law on elliptic curves is associative, meaning that $(P + Q) + R = P + (Q + R)$ for any points P , Q , and R
- The point at infinity serves as the identity element of the group, denoted by $\mathcal{O}$
- For any point P on the curve, $P + \mathcal{O} = P$ and $P + (-P) = \mathcal{O}$, where $-P$ is the reflection of P across the X -axis

Mordell-Weil theorem

- The Mordell-Weil theorem is a fundamental result in the theory of elliptic curves, describing the structure of the group of rational points
- It states that the group of rational points on an elliptic curve over a number field is finitely generated

- The theorem has important consequences for the arithmetic and Diophantine properties of elliptic curves

Statement of theorem

- Let E be an elliptic curve defined over a number field K . Then the group $E(K)$ of K -rational points on E is a finitely generated abelian group
- This means that $E(K)$ is isomorphic to a direct product of a finite torsion subgroup and a free abelian group of finite rank
- The theorem was first proved by Louis Mordell for $K = \mathbb{Q}$ and later generalized by André Weil to arbitrary number fields

Finitely generated abelian groups

- A finitely generated abelian group is an abelian group that can be generated by a finite set of elements
- The structure theorem for finitely generated abelian groups states that every such group is isomorphic to a direct product of cyclic groups
- The Mordell-Weil theorem implies that the group of rational points on an elliptic curve has this structure (e.g., $E(\mathbb{Q}) \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}$)

Torsion subgroup

- The torsion subgroup of $E(K)$ consists of all points of finite order, i.e., points P such that $nP = \mathcal{O}$ for some positive integer n
- The torsion subgroup is always finite, and its structure is well-understood for many classes of elliptic curves
- For example, over $\mathbb{Q}$, the torsion subgroup is isomorphic to one of 15 possible groups (e.g., $\mathbb{Z}/2\mathbb{Z}$, $\mathbb{Z}/4\mathbb{Z}$, $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$)

Rank of elliptic curve

- The rank of an elliptic curve E over a field K is the rank of the free abelian part of $E(K)$
- In other words, it is the number of independent points of infinite order in $E(K)$
- Determining the rank of an elliptic curve is a difficult problem, and there is no known algorithm for computing it in general (e.g., the Birch and Swinnerton-Dyer conjecture relates the rank to the behavior of the L-function of the curve)

Proof of Mordell-Weil theorem

- The proof of the Mordell-Weil theorem is a major result in arithmetic geometry and involves several deep ideas and techniques
- The key steps in the proof include the use of height functions, the weak Mordell-Weil theorem, and the descent procedure
- Understanding the proof provides insight into the structure of rational points on elliptic curves and the methods used to study them

Height functions on elliptic curves

- Height functions are a way to measure the "size" or "complexity" of rational points on an elliptic curve
- The canonical height is a quadratic form on $E(K)$ that behaves like a logarithmic version of the naive height on projective space
- The height functions play a crucial role in the proof of the Mordell-Weil theorem, as they allow for the construction of a finite-index subgroup of $E(K)$

Descent procedure

- The descent procedure is a method for studying the rational points on an elliptic curve by relating them to points on simpler curves
- It involves constructing an auxiliary curve (often a twist of the original curve) and using information about its rational points to deduce information about the original curve
- The descent procedure is used in the proof of the weak Mordell-Weil theorem and can also be used to compute bounds on the rank of an elliptic curve

Weak Mordell-Weil theorem

- The weak Mordell-Weil theorem states that for any integer $n \geq 2$, the quotient group $E(K)/nE(K)$ is finite
- This means that the group of rational points modulo n -torsion is finite, which is a key step in proving the full Mordell-Weil theorem
- The proof of the weak Mordell-Weil theorem uses the descent procedure and the Kummer exact sequence to reduce the problem to studying the Galois cohomology of the curve

Applications of Mordell-Weil theorem

- The Mordell-Weil theorem has numerous applications in number theory, algebraic geometry, and cryptography
- It provides a powerful tool for studying the arithmetic properties of elliptic curves and related Diophantine equations
- Some of the most notable applications include the study of integral points, elliptic curve cryptography, and the resolution of certain Diophantine equations

Integral points on elliptic curves

- An integral point on an elliptic curve is a rational point whose coordinates are integers
- The Mordell-Weil theorem can be used to show that the set of integral points on an elliptic curve is finite
- Studying integral points is important for understanding the arithmetic of elliptic curves and has connections to Diophantine equations and the ABC conjecture (e.g., the equation $y^2 = x^3 + 1$ has only four integral solutions: $(0, \pm 1)$ and $(-1, 0)$)

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography system that uses the group of rational points on an elliptic curve as the underlying mathematical structure
- ECC relies on the difficulty of the elliptic curve discrete logarithm problem, which is believed to be harder than the discrete logarithm problem in finite fields
- The Mordell-Weil theorem ensures that the group of rational points is finite, which is necessary for the security of ECC (e.g., the curve $y^2 = x^3 - x + 1$ over a finite field is used in some ECC implementations)

Diophantine equations

- The Mordell-Weil theorem can be used to study certain types of Diophantine equations, which are polynomial equations in integers
- Many Diophantine equations can be transformed into questions about rational points on elliptic curves, allowing the theorem to be applied
- For example, the Fermat equation $x^n + y^n = z^n$ for $n \geq 3$ can be studied using elliptic curves, and the Mordell-Weil theorem played a role in Andrew Wiles' proof of Fermat's Last Theorem

Examples and computations

- Concrete examples and computations are essential for understanding the Mordell-Weil theorem and its applications
- By studying specific elliptic curves and their rational points, one can gain insight into the general theory and its practical implications
- In this section, we will look at some examples of curves with torsion points, curves of rank 0 and 1, and infinite families of curves

Curves with torsion points

- Torsion points on elliptic curves are points of finite order, i.e., points that generate a finite subgroup of the group of rational points
- The torsion subgroup of an elliptic curve over $\mathbb{Q}$ is well-understood and can be classified into 15 possible isomorphism classes
- Examples of curves with torsion points include:
 - $y^2 = x^3 - x$, which has torsion subgroup isomorphic to $\mathbb{Z}/4\mathbb{Z}$, generated by the point $(0, 0)$
 - $y^2 = x^3 + 1$, which has torsion subgroup isomorphic to $\mathbb{Z}/6\mathbb{Z}$, generated by the point $(0, 1)$

Curves of rank 0 and 1

- The rank of an elliptic curve is the number of independent points of infinite order in the group of rational points
- Curves of rank 0 have a finite number of rational points, while curves of rank 1 have infinitely many rational points that can be generated by a single point
- Examples of curves of rank 0 and 1 include:

- $y^2 = x^3 - x$, which has rank 0 and only four rational points: $(0, 0)$, $(1, 0)$, $(-1, 0)$, and the point at infinity
- $y^2 = x^3 - 2x + 1$, which has rank 1 and infinitely many rational points generated by the point $(1, 1)$

Infinite families of curves

- Some elliptic curves belong to infinite families with specific properties, such as having a certain torsion subgroup or rank
- Studying these families can provide insight into the behavior of elliptic curves and their rational points
- Examples of infinite families of curves include:
 - The Mordell curves, defined by the equation $y^2 = x^3 + k$ for integer values of k , which have been studied extensively due to their connection to the Mordell-Weil theorem
 - The Hessian family, defined by the equation $x^3 + y^3 + z^3 = dxyz$ for integer values of d , which includes curves with various torsion subgroups and ranks

Unit 7 – Elliptic curves and modular forms

7.1 Modular curves and modular forms

Modular curves and modular forms are key concepts in the study of elliptic curves. These mathematical objects provide a bridge between complex analysis, algebraic geometry, and number theory, offering powerful tools for understanding the properties of elliptic curves.

Modular curves arise as moduli spaces of elliptic curves with additional structure, while modular forms are complex analytic functions with specific transformation properties. Together, they form a rich framework for exploring deep connections between elliptic curves, number theory, and other areas of mathematics.

Definition of modular curves

- Modular curves are algebraic curves that arise as moduli spaces of elliptic curves with additional structure
- They play a central role in the study of elliptic curves and their connections to other areas of mathematics, such as modular forms and number theory
- Understanding the definition and properties of modular curves is essential for exploring their applications in the context of elliptic curves

Moduli spaces of elliptic curves

- Modular curves can be viewed as moduli spaces parametrizing isomorphism classes of elliptic curves with additional data (level structures)
- The moduli space of elliptic curves is a geometric object that classifies elliptic curves up to isomorphism
- Examples of level structures include torsion points, cyclic subgroups, or isogenies between elliptic curves

Connection to modular forms

- Modular curves have a deep connection to modular forms, which are complex analytic functions with certain transformation properties under the action of congruence subgroups of $SL(2, \mathbb{Z})$
- The Fourier coefficients of modular forms are related to the geometry and arithmetic of modular curves
- Modular forms can be used to construct functions and differentials on modular curves

Modular curves over $\mathbb{C}$

- Over the complex numbers, modular curves can be realized as quotients of the upper half-plane $\mathbb{H}$ by congruence subgroups of $SL(2, \mathbb{Z})$
- The resulting complex curves have a rich geometric structure, including cusps and elliptic points
- Examples of modular curves over $\mathbb{C}$ include the modular curve $X(1)$ (which is isomorphic to the j -line) and the modular curves $X_0(N)$ and $X_1(N)$ for various levels N

Modular curves over $\mathbb{Q}$

- Modular curves can also be defined over the rational numbers $\mathbb{Q}$ by considering elliptic curves with level structures defined over $\mathbb{Q}$
- The study of rational points on modular curves is of great interest in number theory, as it relates to Diophantine equations and the Birch and Swinnerton-Dyer conjecture
- Examples of modular curves over $\mathbb{Q}$ include the modular curves $X_0(N)$ and $X_1(N)$, which have models defined by explicit equations

Modular forms and functions

- Modular forms and functions are central objects in the study of modular curves and their applications to elliptic curves
- They are complex analytic functions that transform in a specific way under the action of certain subgroups of $SL(2, \mathbb{Z})$
- Understanding the properties and spaces of modular forms is crucial for exploring their connections to modular curves and elliptic curves

Modular forms for $SL(2, \mathbb{Z})$

- Modular forms for the full modular group $SL(2, \mathbb{Z})$ are holomorphic functions f on the upper half-plane $\mathbb{H}$ satisfying $f(\gamma z) = (cz + d)^k f(z)$ for all

$$\gamma = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in SL(2, \mathbb{Z})$$

and $z \in \mathbb{H}$, where k is the weight of the modular form

- Examples of modular forms for $SL(2, \mathbb{Z})$ include the Eisenstein series E_4 and E_6 , and the discriminant modular form Δ
- The space of modular forms for $SL(2, \mathbb{Z})$ of a given weight is a finite-dimensional vector space

Modular forms for congruence subgroups

- Modular forms can also be defined for congruence subgroups Γ of $SL(2, \mathbb{Z})$, such as $\Gamma_0(N)$ and $\Gamma_1(N)$
- These modular forms satisfy a similar transformation property under the action of elements in the congruence subgroup
- The spaces of modular forms for congruence subgroups are often larger than those for the full modular group and have additional structure

Spaces of modular forms

- The spaces of modular forms of a given weight and level form finite-dimensional vector spaces over the complex numbers
- These spaces have a rich algebraic and geometric structure, including Hecke operators, Petersson inner product, and connections to modular curves

- The dimension of the space of modular forms can be computed using the Riemann-Roch theorem or the Selberg trace formula

Fourier expansions of modular forms

- Modular forms have Fourier expansions

$$f(z) = \sum_{n=0}^{\infty} a_n e^{2\pi i n z}$$

, where the coefficients a_n encode arithmetic information

- The Fourier coefficients satisfy certain growth conditions and congruence relations
- The Fourier expansions of modular forms can be used to study their analytic and arithmetic properties, such as the valuation of the coefficients and the behavior at cusps

Hecke operators on modular forms

- Hecke operators T_n are linear operators acting on the spaces of modular forms, defined by averaging over certain cosets of congruence subgroups
- Hecke operators commute with each other and have a rich algebraic structure, including eigenforms and eigenvalues
- The action of Hecke operators on modular forms is closely related to the arithmetic of elliptic curves and the coefficients of L -functions

Modular curves as Riemann surfaces

- Modular curves can be viewed as Riemann surfaces, which are one-dimensional complex manifolds
- Studying the geometric properties of modular curves as Riemann surfaces provides insights into their structure and connections to other areas of mathematics
- The Riemann surface perspective allows for the application of tools from complex analysis and algebraic geometry to the study of modular curves

Modular curves as quotients of $\mathbb{H}$

- Over the complex numbers, modular curves can be realized as quotients of the upper half-plane $\mathbb{H}$ by the action of congruence subgroups of $SL(2, \mathbb{Z})$
- The quotient space $\Gamma \backslash \mathbb{H}$, where Γ is a congruence subgroup, has the structure of a Riemann surface
- The compactification of these quotient spaces by adding cusps leads to compact Riemann surfaces

Fundamental domains and cusps

- A fundamental domain for a congruence subgroup Γ is a connected region in $\mathbb{H}$ such that every orbit of Γ intersects the region exactly once, up to boundary identifications
- Cusps are points on the boundary of $\mathbb{H}$ that are fixed points of parabolic elements in Γ
- The cusps correspond to the points at infinity of the modular curve and play a crucial role in the study of modular forms and L -functions

Genus of modular curves

- The genus of a modular curve is a topological invariant that measures the complexity of the Riemann surface
- It can be computed using the Riemann-Hurwitz formula, which relates the genus to the index of the congruence subgroup and the number of elliptic points and cusps
- The genus of modular curves grows with the level of the congruence subgroup, reflecting the increasing complexity of the associated moduli spaces

Modular curves as algebraic curves

- Modular curves can also be viewed as algebraic curves, defined by polynomial equations in projective space
- The algebraic curve perspective allows for the application of tools from algebraic geometry, such as the study of rational points and the intersection theory
- Examples of modular curves as algebraic curves include the modular curves $X_0(N)$ and $X_1(N)$, which have explicit equations derived from the theory of modular forms

Rational points on modular curves

- The study of rational points on modular curves is a central problem in number theory, with connections to Diophantine equations and the Birch and Swinnerton-Dyer conjecture
- Rational points on modular curves correspond to elliptic curves with additional structure (such as isogenies or torsion points) defined over $\mathbb{Q}$
- The Modularity Theorem, proven by Wiles and others, establishes a correspondence between elliptic curves over $\mathbb{Q}$ and modular forms, highlighting the importance of rational points on modular curves

Modular parameterizations

- Modular parameterizations are maps from modular curves to elliptic curves that provide a way to study elliptic curves using the theory of modular forms
- They play a crucial role in the Modularity Theorem and have applications to Diophantine equations and the Birch and Swinnerton-Dyer conjecture
- Understanding modular parameterizations is essential for exploring the deep connections between elliptic curves and modular forms

Modular parameterizations of elliptic curves

- A modular parameterization of an elliptic curve E over $\mathbb{Q}$ is a surjective morphism $\phi : X_0(N) \rightarrow E$ from a modular curve $X_0(N)$ to the elliptic curve
- The existence of a modular parameterization implies that the elliptic curve is modular, meaning that its L -function corresponds to a modular form
- The modular parameterization provides a way to study the arithmetic and analytic properties of the elliptic curve using the theory of modular forms

Computation of modular parameterizations

- Modular parameterizations can be computed explicitly using the Fourier coefficients of the associated modular forms
- The computation involves finding a newform f of level N whose L -function matches the L -function of the elliptic curve, and then constructing a map from $X_0(N)$ to the elliptic curve using the periods of f
- Efficient algorithms have been developed to compute modular parameterizations, which have applications in number theory and cryptography

Applications to Diophantine equations

- Modular parameterizations have been used to solve certain Diophantine equations, such as the Fermat equation and the generalized Fermat equation
- The idea is to relate the solutions of the Diophantine equation to rational points on modular curves, and then use the modular parameterization to study these points
- The proof of Fermat's Last Theorem by Wiles and Taylor-Wiles relied heavily on the theory of modular parameterizations

Modularity theorem for elliptic curves

- The Modularity Theorem, proven by Wiles, Taylor-Wiles, and others, states that every elliptic curve over $\mathbb{Q}$ is modular, i.e., it admits a modular parameterization
- The theorem establishes a deep connection between elliptic curves and modular forms, and has numerous applications in number theory
- The proof of the Modularity Theorem was a major breakthrough in mathematics and led to the resolution of Fermat's Last Theorem

Generalized modular parameterizations

- The concept of modular parameterizations can be generalized to other contexts, such as abelian varieties and Shimura curves
- Generalized modular parameterizations provide a way to study higher-dimensional analogues of elliptic curves using the theory of automorphic forms
- These generalizations have applications in arithmetic geometry and the Langlands program

Modular forms and L -functions

- Modular forms and L -functions are closely related objects that play a central role in the study of elliptic curves and number theory
- L -functions are complex analytic functions associated to modular forms, elliptic curves, and other arithmetic objects, encoding important arithmetic and geometric information
- Understanding the properties and connections between modular forms and L -functions is crucial for exploring deep questions in number theory, such as the Birch and Swinnerton-Dyer conjecture

Hecke eigenforms and L-functions

- Hecke eigenforms are modular forms that are simultaneous eigenfunctions for all Hecke operators T_n
- Each Hecke eigenform f has an associated L -function $L(f, s)$, defined as an Euler product over primes involving the Hecke eigenvalues of f
- The L -functions of Hecke eigenforms have many remarkable properties, such as analytic continuation, functional equation, and conjectural connections to arithmetic objects

Analytic properties of L-functions

- The L -functions of modular forms and elliptic curves have several important analytic properties
- They admit analytic continuation to the entire complex plane and satisfy a functional equation relating values at s and $k - s$, where k is the weight of the modular form
- The analytic behavior of L -functions, such as the location and multiplicity of their zeros, has deep arithmetic significance

Special values of L-functions

- The values of L -functions at certain special points, such as integer or critical values, have arithmetic interpretations and are the subject of extensive research
- For example, the values of L -functions of modular forms at negative integers are related to the periods and rational structures on the associated modular curves
- The Birch and Swinnerton-Dyer conjecture, one of the Millennium Prize Problems, relates the rank of an elliptic curve to the order of vanishing of its L -function at $s = 1$

Birch and Swinnerton-Dyer conjecture

- The Birch and Swinnerton-Dyer (BSD) conjecture is a central open problem in number theory that connects the arithmetic of elliptic curves to the analytic behavior of their L -functions
- The conjecture states that the rank of an elliptic curve (the number of independent rational points of infinite order) is equal to the order of vanishing of its L -function at $s = 1$
- The BSD conjecture also relates the leading coefficient of the Taylor expansion of the L -function at $s = 1$ to other arithmetic invariants of the elliptic curve, such as the Tate-Shafarevich group and the regulator

Modular forms and Galois representations

- There is a deep connection between modular forms and Galois representations, which are continuous homomorphisms from the absolute Galois group of $\mathbb{Q}$ to $GL_n(\overline{\mathbb{Q}}_p)$
- The Langlands program, a vast generalization of the Modularity Theorem, predicts a correspondence between modular forms and certain Galois representations
- This correspondence has been established in many cases and has applications to the study of elliptic curves, Diophantine equations, and the Birch and Swinnerton-Dyer conjecture

7.2 Elliptic curves and the modular j-invariant

Elliptic curves are fascinating mathematical objects with deep connections to number theory, algebraic geometry, and cryptography. They're defined by specific equations and have a unique group structure that makes them powerful tools in various fields.

The modular j-invariant is a crucial concept in elliptic curve theory. It helps classify curves, connects them to complex analysis, and plays a key role in understanding their properties over different number fields. This chapter explores these ideas in depth.

Definition of elliptic curves

- Elliptic curves are a fundamental object of study in number theory and algebraic geometry
- They have a rich structure and deep connections to various areas of mathematics, including complex analysis, modular forms, and cryptography
- Understanding the basic definitions and properties of elliptic curves is crucial for exploring their applications and advanced theory

Weierstrass equations

- Elliptic curves can be defined by Weierstrass equations of the form $y^2 = x^3 + ax + b$, where a and b are constants satisfying certain conditions
- The Weierstrass equation provides a convenient algebraic representation of elliptic curves
- The coefficients a and b determine the shape and properties of the curve
- The discriminant $\Delta = -16(4a^3 + 27b^2)$ must be nonzero for the curve to be smooth

Smooth projective curves

- Elliptic curves are smooth projective curves of genus one with a distinguished point called the point at infinity
- Smoothness means that the curve has no singularities or self-intersections
- Projective curves are defined in projective space, which allows for the inclusion of points at infinity
- The genus of a curve is a measure of its complexity and determines many of its properties

Rational points on elliptic curves

- Rational points on an elliptic curve are points whose coordinates are rational numbers
- The set of rational points on an elliptic curve forms a group under a suitable addition law
- Studying the structure and properties of the group of rational points is a central problem in elliptic curve theory
- The Mordell-Weil theorem states that the group of rational points is finitely generated

Group law on elliptic curves

- The set of points on an elliptic curve, including the point at infinity, forms an abelian group under a well-defined addition operation called the group law

- The group law endows elliptic curves with a rich algebraic structure that is essential for their applications and theoretical study
- Understanding the geometric and algebraic properties of the group law is crucial for working with elliptic curves

Geometric definition of group law

- The group law on an elliptic curve can be defined geometrically using the chord-and-tangent process
- To add two points P and Q on the curve, draw a line through P and Q (or the tangent line if $P = Q$) and find the third point of intersection with the curve, then reflect this point across the X -axis to obtain the sum $P + Q$
- The point at infinity serves as the identity element of the group
- The geometric definition provides a visual and intuitive understanding of the group law

Algebraic formulas for group law

- The group law on an elliptic curve can also be expressed using algebraic formulas in terms of the coordinates of the points
- For points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$, the sum $P + Q = (x_3, y_3)$ is given by:
 - $x_3 = \lambda^2 - x_1 - x_2$
 - $y_3 = \lambda(x_1 - x_3) - y_1$
 - where $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ if $P \neq Q$, and $\lambda = \frac{3x_1^2 + a}{2y_1}$ if $P = Q$
- The algebraic formulas allow for efficient computation of the group law and are used in practical implementations

Associativity and identity element

- The group law on an elliptic curve satisfies the associativity property: $(P + Q) + R = P + (Q + R)$ for any points P , Q , and R on the curve
- The point at infinity, denoted by $\mathcal{O}$, serves as the identity element of the group: $P + \mathcal{O} = P$ for any point P on the curve
- The inverse of a point $P = (x, y)$ is given by $-P = (x, -y)$, satisfying $P + (-P) = \mathcal{O}$

Torsion points and subgroups

- Torsion points on an elliptic curve are points of finite order, meaning that repeated addition of the point to itself eventually yields the identity element
- The set of torsion points forms a subgroup of the group of rational points on the curve
- The structure of the torsion subgroup provides valuable information about the arithmetic properties of the elliptic curve
- The Nagell-Lutz theorem gives a criterion for determining the torsion points on an elliptic curve

Isomorphism classes of elliptic curves

- Elliptic curves can be classified up to isomorphism, which means identifying curves that have the same underlying structure and properties
- Isomorphism classes provide a way to study elliptic curves in a more abstract and unified manner
- Understanding isomorphisms and the properties that remain invariant under isomorphisms is important for the classification and study of elliptic curves

Isomorphisms of elliptic curves

- Two elliptic curves are said to be isomorphic if there exists a rational function (an algebraic map) that establishes a one-to-one correspondence between their points, preserving the group law
- Isomorphisms of elliptic curves are algebraic maps of the form $(X, Y) \mapsto (u^2X + r, u^3Y + u^2sX + t)$, where u, r, s, t are constants satisfying certain conditions
- Isomorphic curves have the same j -invariant and share many arithmetic and geometric properties

Short Weierstrass form

- Every elliptic curve is isomorphic to a curve in short Weierstrass form, given by the equation $y^2 = x^3 + ax + b$
- The short Weierstrass form provides a canonical representation of elliptic curves up to isomorphism
- Transforming an elliptic curve into short Weierstrass form simplifies its equation and facilitates the study of its properties
- The coefficients a and b in the short Weierstrass form are related to the invariants of the curve

j -invariant of elliptic curves

- The j -invariant is a fundamental invariant associated with an elliptic curve that characterizes its isomorphism class
- Two elliptic curves are isomorphic if and only if they have the same j -invariant
- The j -invariant is defined in terms of the coefficients of the elliptic curve and remains invariant under isomorphisms
- The j -invariant encodes important information about the structure and properties of the elliptic curve

Curves with same j -invariant

- Elliptic curves with the same j -invariant belong to the same isomorphism class and share many properties
- Curves with the same j -invariant have isomorphic torsion subgroups and similar Galois representations
- Studying families of elliptic curves with the same j -invariant can provide insights into their arithmetic and geometric behavior
- The moduli space of elliptic curves parameterizes isomorphism classes of elliptic curves and is closely related to the j -invariant

Modular j-invariant

- The modular j-invariant is a fundamental object in the theory of elliptic curves and modular forms
- It plays a central role in the study of elliptic curves over the complex numbers and their relationship to lattices and modular functions
- Understanding the properties and significance of the j-invariant is crucial for exploring the deep connections between elliptic curves, complex analysis, and number theory

Definition of j-invariant

- The j-invariant of an elliptic curve E given by the Weierstrass equation $y^2 = x^3 + ax + b$ is defined as:

$$j(E) = 1728 \frac{4a^3}{4a^3 + 27b^2}$$

- The j-invariant is a rational function of the coefficients a and b and remains invariant under isomorphisms of elliptic curves
- The j-invariant characterizes the isomorphism class of an elliptic curve: two curves are isomorphic if and only if they have the same j-invariant

j-invariant in terms of coefficients

- The j-invariant can be expressed in terms of the coefficients of the elliptic curve in various forms
- For a curve in short Weierstrass form $y^2 = x^3 + ax + b$, the j-invariant is given by:

$$j = 1728 \frac{4a^3}{4a^3 + 27b^2}$$

- In terms of the discriminant Δ and the invariants C_4 and C_6 , the j-invariant can be written as:

$$j = \frac{C_4^3}{\Delta}$$

- These expressions highlight the relationship between the j-invariant and the fundamental invariants of the elliptic curve

Properties of j-invariant

- The j-invariant satisfies several important properties that make it a key object in the study of elliptic curves:
- It is invariant under isomorphisms of elliptic curves
- It characterizes the isomorphism class of an elliptic curve
- It is a modular function with respect to the action of the modular group $SL_2(\mathbb{Z})$ on the upper half-plane
- The j-invariant has a pole at the cusp (i.e., at infinity) and takes on every complex value precisely once in the fundamental domain of the modular group

j-invariant as modular function

- The j-invariant can be viewed as a modular function, meaning that it is invariant under the action of the modular group $SL_2(\mathbb{Z})$ on the upper half-plane

- As a modular function, the j -invariant satisfies certain transformation properties under the generators of the modular group
- The modular properties of the j -invariant are closely related to the theory of modular forms and elliptic modular curves
- The j -invariant provides a bridge between the arithmetic of elliptic curves and the analytic theory of modular forms

Elliptic curves over complex numbers

- Elliptic curves over the complex numbers have a rich structure and are closely connected to the theory of lattices and complex analysis
- Studying elliptic curves over $\mathbb{C}$ provides insights into their geometric and analytic properties and reveals deep connections to modular forms and complex multiplication
- The complex case serves as a foundation for understanding elliptic curves over other fields and their arithmetic behavior

Lattices and elliptic curves

- Every elliptic curve over $\mathbb{C}$ can be associated with a lattice Λ in the complex plane, which is a discrete subgroup of $\mathbb{C}$ of rank 2
- The lattice Λ determines the elliptic curve up to isomorphism, and conversely, every lattice gives rise to an elliptic curve
- The Weierstrass $\wp$ -function associated with the lattice Λ provides a parametrization of the elliptic curve and satisfies the differential equation $(\wp')^2 = 4\wp^3 - g_2\wp - g_3$, where g_2 and g_3 are invariants of the lattice

Periods and fundamental parallelogram

- The periods of an elliptic curve over $\mathbb{C}$ are the generators ω_1 and ω_2 of the associated lattice Λ
- The periods determine the shape and size of the fundamental parallelogram, which is a parallelogram in the complex plane with vertices 0 , ω_1 , ω_2 , and $\omega_1 + \omega_2$
- The ratio $\tau = \frac{\omega_2}{\omega_1}$ is called the modular parameter and lies in the upper half-plane $\mathbb{H}$
- The modular parameter τ determines the isomorphism class of the elliptic curve and is related to the j -invariant by a modular function

Uniformization theorem

- The uniformization theorem states that every elliptic curve over $\mathbb{C}$ is analytically isomorphic to a quotient of the complex plane by a lattice
- The isomorphism is given by the Weierstrass $\wp$ -function and its derivative, which provide a bijection between the complex plane modulo the lattice and the points on the elliptic curve
- The uniformization theorem establishes a deep connection between elliptic curves, complex analysis, and the geometry of lattices
- It allows for the study of elliptic curves using the tools and techniques of complex function theory

Complex multiplication

- Complex multiplication refers to the phenomenon where the endomorphism ring of an elliptic curve over $\mathbb{C}$ is larger than the integers
- Elliptic curves with complex multiplication have additional symmetries and special arithmetic properties
- The complex multiplication points on the modular curve parametrize elliptic curves with complex multiplication
- The theory of complex multiplication plays a crucial role in the study of elliptic curves over number fields and their Galois representations

Elliptic curves over finite fields

- Elliptic curves over finite fields have important applications in cryptography and number theory
- The study of elliptic curves over finite fields involves understanding their reduction modulo primes, point counting, and the distinction between supersingular and ordinary curves
- The arithmetic and geometric properties of elliptic curves over finite fields are closely related to their j -invariants and endomorphism rings

Reduction of elliptic curves mod p

- Given an elliptic curve defined over the rational numbers (or a number field), its reduction modulo a prime p is the curve obtained by reducing the coefficients of the Weierstrass equation modulo p
- The reduction of an elliptic curve modulo p can be smooth (if the discriminant is nonzero modulo p) or singular (if the discriminant is zero modulo p)
- The type of reduction (good, bad, or additive) provides information about the structure of the group of points on the reduced curve
- Studying the reduction of elliptic curves modulo primes is important for understanding their arithmetic properties and Galois representations

Number of points on elliptic curves

- The number of points on an elliptic curve over a finite field $\mathbb{F}_q$ (including the point at infinity) is denoted by $\#E(\mathbb{F}_q)$
- The Hasse-Weil bound states that the number of points satisfies the inequality $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- The number of points on an elliptic curve over $\mathbb{F}_q$ is related to its trace of Frobenius, which is defined as $a_q = q + 1 - \#E(\mathbb{F}_q)$
- Counting the number of points on elliptic curves over finite fields is a fundamental problem with applications in cryptography and coding theory

Hasse's theorem on point counts

- Hasse's theorem provides a precise characterization of the possible values for the number of points on an elliptic curve over a finite field

- It states that for an elliptic curve E over $\mathbb{F}_q$, the trace of Frobenius a_q satisfies $|a_q| \leq 2\sqrt{q}$
- Equivalently, the number of points $\#E(\mathbb{F}_q)$ satisfies the Hasse-Weil bound $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- Hasse's theorem imposes strong constraints on the possible point counts of elliptic curves over finite fields and is a key result in the arithmetic theory of elliptic curves

Supersingular vs ordinary curves

- Elliptic curves over finite fields can be classified as either supersingular or ordinary based on their endomorphism rings and j -invariants
- A curve E over $\mathbb{F}_q$ is supersingular if its trace of Frobenius satisfies $a_q \equiv 0 \pmod{p}$, where $q = p^n$ for some prime p and integer n
- Supersingular curves have a larger endomorphism ring than ordinary curves and have special properties that make them useful in certain cryptographic protocols
- Ordinary curves are those that are not supersingular and have endomorphism rings that are orders in imaginary quadratic fields
- The distinction between supersingular and ordinary curves is important for understanding their arithmetic and geometric properties

Applications of j -invariant

- The j -invariant of elliptic curves has numerous applications in various areas of mathematics, including complex analysis, algebraic geometry, and number theory
- It serves as a key tool for classifying elliptic curves, studying their moduli spaces, and investigating their connections

7.3 Elliptic curves and the Taniyama-Shimura conjecture

Elliptic curves are fascinating mathematical objects with deep connections to number theory and cryptography. They're defined by specific equations and have a unique group structure, making them powerful tools for solving complex problems.

The Taniyama-Shimura conjecture, now known as the modularity theorem, links elliptic curves to modular forms. This connection was crucial in proving Fermat's Last Theorem and continues to drive research in number theory and related fields.

Elliptic curves

- Elliptic curves are a fundamental object of study in number theory and algebraic geometry
- They have important applications in cryptography and the proof of Fermat's Last Theorem
- Understanding the properties and structure of elliptic curves is crucial for this course

Definition of elliptic curves

- An elliptic curve is a smooth, projective algebraic curve of genus one with a specified base point
- Can be defined over any field (rational numbers, real numbers, complex numbers, finite fields)
- Non-singular cubic curve in the projective plane

Weierstrass equation

- Every elliptic curve can be described by a Weierstrass equation of the form $y^2 = x^3 + ax + b$
- a and b are constants satisfying certain conditions to ensure smoothness
- Provides a standard way to represent and study elliptic curves
- Coefficients of the Weierstrass equation determine the specific shape and properties of the curve

Group law on elliptic curves

- Elliptic curves have a natural group structure defined geometrically
- Three collinear points on the curve (counting multiplicity) sum to the identity element (point at infinity)
- Group law allows for point addition and scalar multiplication
- Enables the use of elliptic curves in cryptographic protocols (ECDH, ECDSA)

Elliptic curves over finite fields

- Elliptic curves can be defined over finite fields $\mathbb{F}_q$, where q is a prime power
- Number of points on an elliptic curve over a finite field is finite and satisfies Hasse's theorem
- $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- Elliptic curve discrete logarithm problem (ECDLP) is believed to be hard, providing security for cryptographic applications

Elliptic curve cryptography

- Elliptic curves over finite fields are used in various cryptographic protocols
- Elliptic Curve Diffie-Hellman (ECDH) for key exchange
- Elliptic Curve Digital Signature Algorithm (ECDSA) for digital signatures
- Offers similar security to traditional public-key cryptography (RSA) with smaller key sizes
- Widely used in modern cryptographic applications (TLS, Bitcoin, etc.)

Modularity theorem

- The modularity theorem, previously known as the Taniyama-Shimura conjecture, establishes a profound connection between elliptic curves and modular forms
- It has significant consequences in number theory, including the proof of Fermat's Last Theorem
- Understanding the modularity theorem is essential for advanced studies in elliptic curves and related areas

Statement of Taniyama-Shimura conjecture

- Every elliptic curve over the rational numbers is modular
- Associated with a unique modular form of a specific level and weight

- Connects two seemingly distinct areas of mathematics: elliptic curves and modular forms
- Conjecture was posed in the 1950s and became a central problem in number theory

Elliptic curves and modular forms

- Modular forms are complex analytic functions with certain symmetry properties
- Defined on the upper half-plane and invariant under the action of congruence subgroups of $SL_2(\mathbb{Z})$
- Modularity theorem states that every elliptic curve over $\mathbb{Q}$ has an associated modular form
- L-function of the elliptic curve equals the L-function of the corresponding modular form

Frey curves and Fermat's last theorem

- Fermat's Last Theorem states that the equation $x^n + y^n = z^n$ has no non-zero integer solutions for $n > 2$
- In 1980s, Gerhard Frey constructed an elliptic curve (Frey curve) from a hypothetical counterexample to Fermat's Last Theorem
- Frey curve would have unusual properties incompatible with the modularity theorem

Wiles' proof of Fermat's last theorem

- Andrew Wiles, building on work by Richard Taylor, proved the modularity theorem for semistable elliptic curves in 1995
- Sufficient to prove Fermat's Last Theorem by contradiction using Frey curves
- Wiles' proof was a landmark achievement in number theory and relied on deep ideas from elliptic curves and modular forms

Consequences of modularity theorem

- Modularity theorem has numerous consequences beyond Fermat's Last Theorem
- Provides a powerful tool for studying the arithmetic of elliptic curves
- Enables the computation of elliptic curve L-functions and the investigation of their properties
- Modularity theorem has been generalized to other settings (e.g., elliptic curves over totally real fields)
- Continues to be an active area of research in number theory

L-functions

- L-functions are complex analytic functions associated with various mathematical objects, including elliptic curves
- They encode important arithmetic and geometric information about the underlying object
- L-functions play a central role in the study of elliptic curves and are related to deep conjectures in number theory

Definition of L-functions

- An L-function is a complex analytic function defined as an Euler product over primes
- $L(s) = \prod_p L_p(s)$, where $L_p(s)$ are local factors depending on the prime p
- L-functions are defined for various objects: elliptic curves, modular forms, number fields, etc.
- Convergence and analytic continuation of L-functions are important questions in analytic number theory

L-functions of elliptic curves

- Every elliptic curve E over a number field has an associated L-function $L(E, s)$
- Defined using the local zeta functions of E at each prime
- L-function encodes arithmetic information about the elliptic curve
- Coefficients of the L-function are related to the number of points on E over finite fields

Hasse-Weil conjecture

- The Hasse-Weil conjecture states that the L-function of an elliptic curve has an analytic continuation and satisfies a functional equation
- Relates values of $L(E, s)$ at s and $2 - s$
- Modularity theorem implies the Hasse-Weil conjecture for elliptic curves over $\mathbb{Q}$
- Follows from the corresponding properties of modular form L-functions

Birch and Swinnerton-Dyer conjecture

- The Birch and Swinnerton-Dyer (BSD) conjecture relates the rank of an elliptic curve to the behavior of its L-function at $s = 1$
- Rank of E equals the order of vanishing of $L(E, s)$ at $s = 1$
- BSD conjecture also predicts the leading coefficient of the Taylor expansion of $L(E, s)$ at $s = 1$
- Involves various arithmetic invariants of the elliptic curve (Tate-Shafarevich group, regulator, etc.)

Applications of L-functions

- L-functions have numerous applications in the study of elliptic curves and beyond
- BSD conjecture provides a way to compute the rank of an elliptic curve
- Special values of L-functions are related to important arithmetic quantities (e.g., Birch-Swinnerton-Dyer formula)
- L-functions are also used in the study of other objects (modular forms, number fields, etc.)
- Provide a unifying framework for various problems in number theory

Galois representations

- Galois representations are a fundamental tool in modern number theory, connecting arithmetic and geometric properties of objects
- They provide a way to study the action of Galois groups on various structures, including elliptic curves
- Galois representations are closely related to modularity and have important applications in the theory of elliptic curves

Galois groups and representations

- The Galois group G_K of a field extension $K/\mathbb{Q}$ is the group of automorphisms of K fixing $\mathbb{Q}$
- A Galois representation is a continuous homomorphism $\rho : G_K \rightarrow \mathrm{GL}_n(R)$, where R is a topological ring
- Encodes the action of G_K on an n -dimensional vector space over R
- Galois representations arise naturally in various contexts (étale cohomology, Tate modules, etc.)

Tate modules of elliptic curves

- The Tate module $T_\ell(E)$ of an elliptic curve E over a field K is the inverse limit of the ℓ -power torsion points of E
- $T_\ell(E) = \varprojlim E[\ell^n]$, where ℓ is a prime different from the characteristic of K
- Tate module is a free $\mathbb{Z}_\ell$ -module of rank 2, equipped with a continuous action of G_K
- Gives rise to a Galois representation $\rho_{E,\ell} : G_K \rightarrow \mathrm{GL}_2(\mathbb{Z}_\ell)$

Galois representations attached to elliptic curves

- The Tate module construction provides a way to attach a Galois representation to an elliptic curve
- Properties of the elliptic curve are reflected in the properties of the associated Galois representation
- Reduction type of E at primes related to the image of Frobenius elements under $\rho_{E,\ell}$
- Studying the Galois representations of elliptic curves is a powerful tool in understanding their arithmetic

Serre's conjecture

- Serre's conjecture, now a theorem (Khare-Wintenberger, Kisin), describes the Galois representations arising from modular forms
- Every irreducible, odd, continuous Galois representation $\rho : G_{\mathbb{Q}} \rightarrow \mathrm{GL}_2(\mathbb{F}_p)$ comes from a modular form
- Serre's conjecture provides a converse to the modularity theorem for Galois representations
- Establishes a bijection between certain Galois representations and modular forms

Modularity and Galois representations

- The modularity theorem has important consequences for Galois representations of elliptic curves

- Galois representation $\rho_{E, \ell}$ of a modular elliptic curve E is isomorphic to the Galois representation attached to the corresponding modular form
- Modularity allows for the study of elliptic curve Galois representations using the tools and techniques of modular forms
- Provides a rich interplay between the arithmetic of elliptic curves and the theory of modular forms

Elliptic surfaces

- Elliptic surfaces are a natural generalization of elliptic curves, combining aspects of algebraic geometry and complex geometry
- They provide a geometric framework for studying families of elliptic curves and their variations
- The theory of elliptic surfaces has important applications in the study of rational points and Diophantine equations

Elliptic fibrations

- An elliptic surface is a complex surface S equipped with a morphism $f: S \rightarrow C$ to a smooth curve C , such that almost all fibers are smooth elliptic curves
- f is called an elliptic fibration
- Elliptic surfaces can be described locally by Weierstrass equations with coefficients that are functions on the base curve C
- Coefficients determine the specific elliptic curve over each point of C

Kodaira-Néron classification

- The Kodaira-Néron classification describes the possible types of singular fibers in an elliptic fibration
- Singular fibers are classified into types $I_n, II, III, IV, I_n^*, II^*, III^*, IV^*$
- Each type corresponds to a specific configuration of the elliptic curve degenerating over a point of the base curve
- Classification is based on the local monodromy and the structure of the special fiber

Singular fibers and Euler characteristics

- The singular fibers of an elliptic fibration contribute to the Euler characteristic of the elliptic surface
- Each type of singular fiber has a specific contribution, given by the Ogg-Shafarevich formula
- The Euler characteristic is a topological invariant that measures the complexity of the elliptic surface
- Related to the geometry of the base curve and the singular fibers

Mordell-Weil theorem for elliptic surfaces

- The Mordell-Weil theorem states that the group of rational points (sections) of an elliptic surface over a number field is finitely generated
- Generalizes the Mordell-Weil theorem for elliptic curves

- The rank of the Mordell-Weil group measures the number of independent rational sections
- Relates to the geometry of the elliptic surface and the base curve

Rational points on elliptic surfaces

- The study of rational points on elliptic surfaces is a central problem in Diophantine geometry
- Relates to questions about the existence and density of rational points on the fibers (elliptic curves)
- Techniques from the theory of elliptic surfaces (Kodaira-Néron classification, Mordell-Weil theorem, etc.) are used to study rational points
- Provides a geometric approach to Diophantine problems
- The distribution of rational points on elliptic surfaces is a topic of ongoing research, with connections to arithmetic geometry and number theory

7.4 Elliptic curves and the Eichler-Shimura relation

Elliptic curves are smooth, projective algebraic curves of genus one with a specified base point. They have a rich algebraic structure and play a central role in number theory and algebraic geometry, with applications ranging from cryptography to the proof of Fermat's Last Theorem.

The Eichler-Shimura relation connects elliptic curves and modular forms, establishing an isomorphism between certain Hecke modules. This deep result has profound implications for understanding the arithmetic of elliptic curves and their L-functions, forming a crucial link in the modularity theorem.

Definition of elliptic curves

- Elliptic curves are smooth, projective algebraic curves of genus one with a specified base point
- They can be defined over various fields, including the complex numbers, rational numbers, and finite fields
- Elliptic curves have a rich algebraic structure and are central objects of study in number theory and algebraic geometry

Weierstrass equations

- Elliptic curves can be described by Weierstrass equations of the form $y^2 = x^3 + ax + b$, where a and b are constants satisfying certain conditions
- The Weierstrass equation provides a canonical way to represent elliptic curves and study their properties
- The discriminant $\Delta = -16(4a^3 + 27b^2)$ determines the singularity of the curve; if $\Delta \neq 0$, the curve is smooth

Group law on elliptic curves

- Elliptic curves admit a natural group structure, where the group operation is defined geometrically
- The group law is given by the chord-and-tangent process: given two points P and Q on the curve, the sum $P + Q$ is defined as the reflection of the third intersection point of the line through P and Q with the curve

- The group law satisfies the associative, identity, and inverse properties, making elliptic curves an example of an abelian group

Elliptic curves over finite fields

- Elliptic curves can be defined over finite fields $\mathbb{F}_q$, where q is a prime power
- The number of points on an elliptic curve over a finite field, denoted by $\#E(\mathbb{F}_q)$, is an important quantity in cryptography and coding theory
- Hasse's theorem bounds the number of points: $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$

Complex points on elliptic curves

- Over the complex numbers, elliptic curves can be viewed as complex tori $\mathbb{C}/\Lambda$, where Λ is a lattice in $\mathbb{C}$
- The complex points on an elliptic curve form a Lie group isomorphic to the additive group of $\mathbb{C}$ modulo the lattice Λ
- The Weierstrass $\wp$ -function and its derivative provide a parametrization of complex points on an elliptic curve

Modular forms and elliptic curves

- Modular forms are holomorphic functions on the upper half-plane satisfying certain transformation properties under the action of congruence subgroups of $SL_2(\mathbb{Z})$
- Elliptic curves and modular forms are intimately connected through the modularity theorem and the Eichler-Shimura relation

Modular curves and modular forms

- Modular curves are algebraic curves associated with congruence subgroups of $SL_2(\mathbb{Z})$, such as the modular curve $X_0(N)$
- Modular forms of weight k for a congruence subgroup Γ are holomorphic functions f on the upper half-plane satisfying $f(\gamma z) = (cz + d)^k f(z)$ for all

$$\gamma = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \Gamma$$

- The space of modular forms of weight k for Γ is a finite-dimensional vector space over $\mathbb{C}$

Hecke operators on modular forms

- Hecke operators T_n are linear operators acting on the space of modular forms of a given weight for a congruence subgroup
- For a prime p , the Hecke operator T_p is defined by

$$T_p f(z) = p^{k-1} f(pz) + \frac{1}{p} \sum_{i=0}^{p-1} f\left(\frac{z+i}{p}\right)$$

- Hecke operators are self-adjoint with respect to the Petersson inner product and commute with each other

Eigenforms and Hecke eigenvalues

- An eigenform is a modular form that is an eigenvector for all Hecke operators T_n
- The eigenvalues of T_n acting on an eigenform f are called the Hecke eigenvalues of f
- Normalized eigenforms are eigenforms whose Fourier coefficients are algebraic integers and the first coefficient is 1

L-functions of modular forms

- The L-function of a modular form

$$f(z) = \sum_{n=0}^{\infty} a_n e^{2\pi i n z}$$

is defined as

$$L(f, s) = \sum_{n=1}^{\infty} \frac{a_n}{n^s}$$

for $\text{Re}(s) > k + 1$

- L-functions of eigenforms have an Euler product expansion $L(f, s) = \prod_p (1 - a_p p^{-s} + p^{k-1-2s})^{-1}$
- The modularity theorem establishes a correspondence between L-functions of elliptic curves and L-functions of modular forms

Galois representations

- Galois representations are continuous homomorphisms from absolute Galois groups to matrix groups over various fields
- They provide a powerful tool to study the arithmetic properties of elliptic curves and their connection to modular forms

Galois groups and representations

- The absolute Galois group $G_K = \text{Gal}(\bar{K}/K)$ of a field K is the group of automorphisms of an algebraic closure $\bar{K}$ that fix K
- A Galois representation is a continuous homomorphism $\rho: G_K \rightarrow \text{GL}_n(F)$, where F is a field (e.g., $\mathbb{Q}_l$, $\mathbb{C}$)
- The study of Galois representations reveals arithmetic information about the underlying objects (e.g., elliptic curves, modular forms)

Tate modules of elliptic curves

- For an elliptic curve E over a field K and a prime l , the l -adic Tate module $T_l(E)$ is the inverse limit of the l -power torsion points $E[l^n]$
- The Tate module $T_l(E)$ is a free $\mathbb{Z}_l$ -module of rank 2, where $\mathbb{Z}_l$ is the ring of l -adic integers
- The absolute Galois group G_K acts on $T_l(E)$, giving rise to a Galois representation $\rho_l: G_K \rightarrow \text{GL}_2(\mathbb{Z}_l)$

ℓ -adic representations

- An ℓ -adic representation is a continuous homomorphism $\rho : G_K \rightarrow \mathrm{GL}_n(\mathbb{Q}_\ell)$, where $\mathbb{Q}_\ell$ is the field of ℓ -adic numbers
- The ℓ -adic representation associated with an elliptic curve E is obtained by extending the Tate module representation ρ_ℓ to $\mathbb{Q}_\ell$
- ℓ -adic representations encode arithmetic information about elliptic curves, such as the action of Frobenius elements on torsion points

Modularity of Galois representations

- The modularity of a Galois representation $\rho : G_{\mathbb{Q}} \rightarrow \mathrm{GL}_2(\mathbb{Q}_\ell)$ means that ρ arises from a modular form
- More precisely, ρ is modular if there exists a normalized eigenform f such that the characteristic polynomial of $\rho(\mathrm{Frob}_p)$ equals the Hecke polynomial $1 - a_p X + pX^2$ for almost all primes p
- The modularity theorem for elliptic curves states that the ℓ -adic representation associated with an elliptic curve over $\mathbb{Q}$ is modular

Eichler-Shimura relation

- The Eichler-Shimura relation is a fundamental result connecting the arithmetic of elliptic curves and modular forms
- It establishes an isomorphism between certain Hecke modules arising from elliptic curves and modular curves

Statement of Eichler-Shimura relation

- Let $\Gamma = \Gamma_0(N)$ be a congruence subgroup and $X = X_0(N)$ the corresponding modular curve
- The Eichler-Shimura relation states that there is an isomorphism of Hecke modules $H^1(X, \mathbb{Q}) \cong S_2(\Gamma) \oplus \overline{S_2(\Gamma)}$, where $S_2(\Gamma)$ is the space of cusp forms of weight 2 for Γ
- This isomorphism is compatible with the action of Hecke operators on both sides

Hecke correspondences on modular curves

- Hecke correspondences are algebraic curves T_n on $X \times X$ that generalize the Hecke operators on modular forms
- The Hecke correspondence T_n induces a map T_n^* on the cohomology group $H^1(X, \mathbb{Q})$
- The action of T_n^* on $H^1(X, \mathbb{Q})$ corresponds to the action of the Hecke operator T_n on $S_2(\Gamma) \oplus \overline{S_2(\Gamma)}$ under the Eichler-Shimura isomorphism

Jacobians of modular curves

- The Jacobian $J_0(N)$ of the modular curve $X_0(N)$ is an abelian variety that parametrizes degree zero divisors on $X_0(N)$
- The Hecke correspondences T_n induce endomorphisms of $J_0(N)$, which we also denote by T_n
- The Eichler-Shimura relation implies that the $\mathbb{Q}$ -vector space $\mathrm{Hom}(J_0(N), \mathbb{C})$ is isomorphic to $S_2(\Gamma_0(N)) \oplus \overline{S_2(\Gamma_0(N))}$ as a Hecke module

Isomorphism of Hecke modules

- The Eichler-Shimura isomorphism $H^1(X_0(N), \mathbb{Q}) \cong S_2(\Gamma_0(N)) \oplus \overline{S_2(\Gamma_0(N))}$ is an isomorphism of Hecke modules
- This means that the action of Hecke operators on the cohomology group $H^1(X_0(N), \mathbb{Q})$ corresponds to the action of Hecke operators on the space of cusp forms $S_2(\Gamma_0(N))$ and its complex conjugate
- The isomorphism is given by the period map, which associates to a cusp form f the cohomology class of the differential form $f(z)dz$ on $X_0(N)$

Consequences for L-functions

- The Eichler-Shimura relation has important consequences for the L-functions of elliptic curves and modular forms
- It implies that the L-function of an elliptic curve E of conductor N is equal to the L-function of a normalized eigenform $f \in S_2(\Gamma_0(N))$
- This connection between L-functions is a key ingredient in the proof of the modularity theorem for elliptic curves over $\mathbb{Q}$

Applications and examples

- The theory of elliptic curves and modular forms has numerous applications in number theory and beyond
- Some notable examples include the proof of Fermat's last theorem, the study of congruent numbers, and the Birch and Swinnerton-Dyer conjecture

Elliptic curves over $\mathbb{Q}$

- Elliptic curves over the rational numbers $\mathbb{Q}$ are of particular interest in number theory
- The Mordell-Weil theorem states that the group of rational points $E(\mathbb{Q})$ is finitely generated
- The rank of an elliptic curve, which is the rank of the free part of $E(\mathbb{Q})$, is a crucial invariant that measures the size of the rational point group

Modularity theorem for elliptic curves

- The modularity theorem, proved by Wiles, Taylor-Wiles, and others, states that every elliptic curve over $\mathbb{Q}$ is modular
- This means that for an elliptic curve E of conductor N , there exists a normalized eigenform $f \in S_2(\Gamma_0(N))$ such that the L-function of E equals the L-function of f
- The modularity theorem is a powerful result that connects the arithmetic of elliptic curves to the theory of modular forms

Fermat's last theorem

- Fermat's last theorem states that the equation $x^n + y^n = z^n$ has no non-zero integer solutions for $n > 2$

- The proof of Fermat's last theorem, completed by Wiles in 1995, relies crucially on the modularity theorem for semistable elliptic curves
- The key idea is to associate an elliptic curve to a hypothetical solution of Fermat's equation and derive a contradiction using the modularity theorem and Ribet's level-lowering result

Congruent number problem

- A positive integer n is called a congruent number if it is the area of a right triangle with rational side lengths
- The congruent number problem asks for a characterization of congruent numbers
- Tunnell's theorem provides a criterion for congruent numbers in terms of the Birch and Swinnerton-Dyer conjecture for certain elliptic curves

Birch and Swinnerton-Dyer conjecture

- The Birch and Swinnerton-Dyer (BSD) conjecture is a central open problem in the arithmetic of elliptic curves
- It relates the rank of an elliptic curve E over $\mathbb{Q}$ to the order of vanishing of its L-function $L(E, s)$ at $s = 1$
- The BSD conjecture also predicts a precise formula for the leading coefficient of the Taylor expansion of $L(E, s)$ at $s = 1$ in terms of arithmetic invariants of E
- The conjecture has been verified for specific classes of elliptic curves, but remains open in general

7.5 Elliptic curves and the Hecke operators

Elliptic curves are smooth, algebraic curves with unique properties that make them crucial in number theory and cryptography. Their structure as abelian groups allows for fascinating arithmetic operations, connecting geometry and algebra in profound ways.

Hecke operators are powerful tools for studying modular forms and elliptic curves. They reveal deep connections between these objects, helping mathematicians understand their arithmetic properties and L-functions. Hecke operators are key to many important results in modern number theory.

Definition of elliptic curves

- Elliptic curves are smooth, projective, algebraic curves of genus one with a specified basepoint
- They have a rich geometric and algebraic structure that makes them important objects of study in number theory and algebraic geometry
- Elliptic curves can be defined over various fields, including the complex numbers, rational numbers, and finite fields

Weierstrass equation

- An elliptic curve can be described by a Weierstrass equation of the form $y^2 = x^3 + ax + b$, where a and b are constants satisfying certain conditions
- The Weierstrass equation provides a canonical way to represent an elliptic curve algebraically

- The coefficients a and b determine the specific shape and properties of the elliptic curve ($y^2 = x^3 - x$ and $y^2 = x^3 - 4x + 1$)

Elliptic curves over fields

- Elliptic curves can be defined over various fields, such as the rational numbers $\mathbb{Q}$, the real numbers $\mathbb{R}$, or finite fields $\mathbb{F}_p$
- The choice of field affects the arithmetic and geometric properties of the elliptic curve
- Elliptic curves over finite fields have important applications in cryptography ($y^2 = x^3 + x + 1$ over $\mathbb{F}_5$)

Group law on elliptic curves

- Points on an elliptic curve form an abelian group under a geometric operation called the group law
- The group law is defined by drawing a line through two points on the curve and finding the third point of intersection, then reflecting across the x-axis
- The group law allows for point addition, subtraction, and scalar multiplication on elliptic curves ($P + Q = -R$ where R is the third point of intersection)

Modular forms and elliptic curves

- Modular forms are complex analytic functions with certain symmetry and growth properties that are deeply connected to elliptic curves
- The study of modular forms and their relationship to elliptic curves has led to significant advances in number theory, such as the proof of Fermat's Last Theorem
- Modular forms provide a powerful tool for understanding the arithmetic and geometric properties of elliptic curves

Modularity theorem

- The modularity theorem, also known as the Taniyama-Shimura-Weil conjecture, states that every elliptic curve over $\mathbb{Q}$ is modular
- This means that for every elliptic curve, there exists a corresponding modular form whose L-function matches the L-function of the elliptic curve
- The modularity theorem was a key ingredient in Andrew Wiles' proof of Fermat's Last Theorem

Elliptic curves over $\mathbb{C}$

- Over the complex numbers, an elliptic curve can be realized as a quotient of the complex plane by a lattice
- The complex uniformization of elliptic curves provides a link between the algebraic and analytic theories
- Elliptic functions, such as the Weierstrass $\wp$ -function, play a crucial role in the study of elliptic curves over $\mathbb{C}$ ($\mathbb{C}/\Lambda$ where Λ is a lattice)

Lattices and elliptic curves

- A lattice in the complex plane is a discrete subgroup of $\mathbb{C}$ of the form $\Lambda = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$ for linearly independent complex numbers ω_1 and ω_2
- Every elliptic curve over $\mathbb{C}$ can be obtained as the quotient $\mathbb{C}/\Lambda$ for some lattice Λ
- The periods ω_1 and ω_2 of the lattice determine the shape and properties of the corresponding elliptic curve ($\Lambda = \mathbb{Z} + \mathbb{Z}i$ and $\Lambda = \mathbb{Z} + \mathbb{Z}\sqrt{3}i$)

Hecke operators on modular forms

- Hecke operators are linear operators that act on the space of modular forms and play a central role in the study of modular forms and elliptic curves
- They are algebraic tools that help reveal the arithmetic properties of modular forms and their associated L-functions
- Hecke operators are indexed by positive integers and have important algebraic and geometric interpretations

Definition of Hecke operators

- For each positive integer n , the n -th Hecke operator T_n is defined as a certain double coset operator on the space of modular forms
- The action of T_n on a modular form f is given by a sum of the form

$$T_n(f) = \sum_{ad=n} a^{k-1} f|_k \begin{bmatrix} a & b \\ 0 & d \end{bmatrix}$$

, where k is the weight of the modular form

- Hecke operators preserve the space of modular forms and commute with each other ($T_2(f) = f(z) + 2^{k-1}f(2z)$)

Action on q-expansions

- Modular forms can be represented as power series in the variable $q = e^{2\pi iz}$, called q-expansions
- Hecke operators act on the q-expansions of modular forms in a natural way, allowing for explicit computations
- The action of T_n on the q-expansion

$$f(z) = \sum_{m=0}^{\infty} a_m q^m$$

is given by

$$T_n(f) = \sum_{m=0}^{\infty} b_m q^m$$

, where $b_m = \sum_{d|\gcd(m,n)} d^{k-1} a_{mn/d^2}$

Hecke eigenforms

- A modular form f is called a Hecke eigenform if it is an eigenvector for all Hecke operators T_n
- Hecke eigenforms have multiplicative Fourier coefficients, meaning that $a_{mn} = a_m a_n$ for coprime integers m and n
- Hecke eigenforms are of particular interest because their L-functions have an Euler product representation and satisfy a functional equation (

$$\Delta(z) = q \prod_{n=1}^{\infty} (1 - q^n)^{24}$$

is a Hecke eigenform)

Hecke operators on elliptic curves

- Hecke operators can also be defined on the space of elliptic curves, providing a geometric interpretation of their action
- The study of Hecke operators on elliptic curves leads to important results in arithmetic geometry and the theory of L-functions
- Hecke operators on elliptic curves are closely related to the concept of isogenies and the Tate module

Isogenies of elliptic curves

- An isogeny between two elliptic curves is a non-constant morphism that preserves the group structure
- Isogenies can be classified by their degree, which is the degree of the corresponding rational map
- The set of isogenies from an elliptic curve to itself forms a ring, called the endomorphism ring of the curve ($\phi(x, y) = (x^2, y^3)$ is an isogeny from $y^2 = x^3 - x$ to itself)

Tate modules

- For each prime p , the p -adic Tate module of an elliptic curve E is defined as the inverse limit of the p -power torsion points of E
- Tate modules are important tools in the study of elliptic curves and their Galois representations
- The action of Hecke operators on elliptic curves can be understood through their action on the Tate modules ($T_p(E) = \varprojlim E[p^n]$ where $E[p^n]$ denotes the p^n -torsion points)

Hecke correspondences

- Hecke operators on elliptic curves can be interpreted geometrically as correspondences on the moduli space of elliptic curves
- A Hecke correspondence associates to each elliptic curve E a sum of elliptic curves that are isogenous to E with a specified degree
- Hecke correspondences provide a way to study the arithmetic and geometric properties of elliptic curves ($T_n(E) = \sum_{C: \deg(C \rightarrow E) = n} C$ where the sum is over isogenies of degree n)

Eichler-Shimura relation

- The Eichler-Shimura relation is a fundamental result that relates the Hecke operators on modular forms to the Frobenius endomorphism on the reduction of elliptic curves mod p
- It states that for a prime p not dividing the level of the modular form, the Hecke operator T_p acts on the modular form as the trace of Frobenius on the corresponding elliptic curve
- The Eichler-Shimura relation provides a deep connection between the arithmetic of elliptic curves and the theory of modular forms ($a_p(f) = p + 1 - \#E(F_p)$ where f is the modular form associated to E)

L-functions of elliptic curves

- The L-function of an elliptic curve is a complex analytic function that encodes important arithmetic information about the curve
- L-functions of elliptic curves are defined using the Euler product and have a functional equation relating their values at s and $2 - s$
- The study of L-functions is central to modern number theory and has deep connections to the Birch and Swinnerton-Dyer conjecture

Definition of L-functions

- For an elliptic curve E over $\mathbb{Q}$, the L-function $L(E, s)$ is defined as an Euler product over primes p of local factors $L_p(E, s)$
- The local factors are defined using the number of points on the reduction of E mod p and the Frobenius endomorphism
- The L-function converges for $\text{Re}(s) > 3/2$ and is expected to have an analytic continuation to the entire complex plane ($L(E, s) = \prod_p L_p(E, s)^{-1}$)

Euler product

- The Euler product for the L-function of an elliptic curve has the form $L(E, s) = \prod_p (1 - a_p p^{-s} + p^{1-2s})^{-1}$ for primes p of good reduction
- The coefficients a_p are related to the number of points on the reduction of E mod p by $a_p = p + 1 - \#E(F_p)$
- The Euler product provides a way to study the L-function using local data at each prime ($L(E, s) = \prod_p (1 - a_p p^{-s} + p^{1-2s})^{-1}$)

Analytic continuation

- The L-function of an elliptic curve is expected to have an analytic continuation to the entire complex plane
- The analytic continuation is known to exist for elliptic curves that are modular, which includes all elliptic curves over $\mathbb{Q}$ by the modularity theorem
- The analytic continuation allows for the study of the L-function at special values, such as $s = 1$ ($L(E, s)$ is analytic on $\mathbb{C} - \{1\}$ and has a simple pole at $s = 1$)

Functional equation

- The L-function of an elliptic curve satisfies a functional equation relating its values at s and $2 - s$
- The functional equation involves the conductor N of the elliptic curve and has the form $\Lambda(E, s) = \pm \Lambda(E, 2 - s)$, where $\Lambda(E, s) = N^{s/2} (2\pi)^{-s} \Gamma(s) L(E, s)$
- The sign in the functional equation is related to the root number of the elliptic curve ($\Lambda(E, s) = \pm \Lambda(E, 2 - s)$ where the sign is $+1$ if E has even rank and -1 if E has odd rank)

Galois representations

- Galois representations are a fundamental tool in modern number theory that connect the arithmetic of elliptic curves to the theory of Galois groups
- The action of the absolute Galois group $\text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ on the torsion points of an elliptic curve gives rise to a Galois representation
- The study of Galois representations has led to important results in the theory of elliptic curves, such as the proof of the modularity theorem

Galois action on torsion points

- For each positive integer n , the n -torsion points of an elliptic curve E form a group $E[n]$ isomorphic to $(\mathbb{Z}/n\mathbb{Z})^2$
- The absolute Galois group $\text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ acts on $E[n]$ by its action on the coordinates of the torsion points
- The Galois action on torsion points provides a way to study the arithmetic of elliptic curves using Galois theory ($\sigma(P) = (\sigma(x), \sigma(y))$ for $\sigma \in \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q})$ and $P = (x, y) \in E[n]$)

Galois representations attached to elliptic curves

- The Galois action on the torsion points of an elliptic curve gives rise to a Galois representation $\rho_{E,n} : \text{Gal}(\overline{\mathbb{Q}}/\mathbb{Q}) \rightarrow \text{GL}_2(\mathbb{Z}/n\mathbb{Z})$
- The Galois representation $\rho_{E,n}$ encodes information about the arithmetic of the elliptic curve mod n
- The study of Galois representations attached to elliptic curves has led to important results in number theory, such as the proof of Serre's conjecture ($\rho_{E,n}$ is surjective for all but finitely many n)

Modularity and Galois representations

- The modularity theorem implies that the Galois representation attached to an elliptic curve over $\mathbb{Q}$ arises from a modular form
- More precisely, for a modular elliptic curve E , there exists a modular form f such that the Fourier coefficients of f are related to the traces of Frobenius in the Galois representation
- The connection between modularity and Galois representations has been a driving force in the development of modern number theory ($a_p(f) = \text{tr}(\rho_{E,p}(\text{Frob}_p))$ where f is the modular form associated to E)

Applications of Hecke operators

- Hecke operators have numerous applications in the study of modular forms, elliptic curves, and related areas of number theory
- They provide a powerful tool for computing and studying the arithmetic and geometric properties of these objects
- The theory of Hecke operators has led to significant advances in our understanding of diophantine equations and other classical problems in number theory

Computation of modular forms

- Hecke operators can be used to efficiently compute the Fourier coefficients of modular forms
- The action of Hecke operators on the q -expansions of modular forms allows for recursive computation of the coefficients
- Hecke operators also provide a way to decompose the space of modular forms into subspaces spanned by eigenforms ($T_n(f) = a_n(f)f$ for a Hecke eigenform f)

Computation of elliptic curves

- Hecke operators can be used to compute the conductor and the L -function of an elliptic curve
- The modularity theorem implies that every elliptic curve over $\mathbb{Q}$ corresponds to a modular form, which can be computed using Hecke operators
- The Fourier coefficients of the modular form determine the conductor and the L -function of the associated elliptic curve ($a_p(f) = p + 1 - \#E(F_p)$ where f is the modular form associated to E)

Diophantine equations

- Hecke operators have been used

7.6 Elliptic curves and the Modularity theorem

Elliptic curves are fascinating mathematical objects with deep connections to number theory and cryptography. They're defined by equations like $y^2 = x^3 + ax + b$ and form abelian groups under a geometric addition law. This structure makes them powerful tools in various fields.

The Modularity Theorem links elliptic curves over rational numbers to modular forms, complex analytic functions with special symmetries. This profound connection, proven in 2001, resolved Fermat's Last Theorem and opened new avenues in number theory and cryptography.

Elliptic curves over fields

- Elliptic curves are algebraic curves defined over various fields, including the real numbers, complex numbers, and finite fields
- The study of elliptic curves over fields is a fundamental aspect of algebraic geometry and number theory
- Understanding the properties and behavior of elliptic curves over different fields is crucial for applications in cryptography and solving Diophantine equations

Weierstrass equations

- Elliptic curves can be described using Weierstrass equations of the form $y^2 = x^3 + ax + b$, where a and b are constants from the field
- The Weierstrass equation provides a standard form for representing elliptic curves, making it easier to study their properties and perform computations
- The coefficients a and b in the Weierstrass equation determine the shape and characteristics of the elliptic curve
- The Weierstrass equation is used to define the group law for points on the elliptic curve

Group law for points

- The set of points on an elliptic curve, together with a special point called the "point at infinity," forms an abelian group under a well-defined group law
- The group law for points on an elliptic curve is defined geometrically using the chord-and-tangent method
- Given two points P and Q on the curve, draw a line through them and find the third point of intersection with the curve, then reflect that point across the X -axis to obtain the sum $P + Q$
- If $P = Q$, the line is taken to be the tangent line at P , and the sum is $2P$
- The group law satisfies the properties of associativity, identity (point at infinity), and inverses
- The group structure of elliptic curves is the foundation for their use in cryptography and other applications

Discriminant and j-invariant

- The discriminant Δ of an elliptic curve is a quantity that determines whether the curve is smooth (non-singular) or has singularities
- For a Weierstrass equation $y^2 = x^3 + ax + b$, the discriminant is given by $\Delta = -16(4a^3 + 27b^2)$
- If $\Delta \neq 0$, the curve is smooth; if $\Delta = 0$, the curve has a singularity (node or cusp)
- The j-invariant is a quantity that characterizes the isomorphism class of an elliptic curve
- Two elliptic curves are isomorphic over an algebraically closed field if and only if they have the same j-invariant
- The j-invariant is given by $j = 1728 \frac{4a^3}{4a^3 + 27b^2}$ for a Weierstrass equation $y^2 = x^3 + ax + b$

Isomorphism classes

- Elliptic curves over a field can be classified into isomorphism classes based on their j-invariants
- Isomorphic elliptic curves have the same j-invariant and share many properties, such as the same group structure and the same number of points over finite fields
- Understanding isomorphism classes is important for studying the moduli space of elliptic curves and for applications in cryptography

Torsion points and structure

- Torsion points on an elliptic curve are points of finite order under the group law, meaning that a multiple of the point by some integer yields the point at infinity
- The set of torsion points on an elliptic curve forms a subgroup of the curve's group structure
- The torsion subgroup of an elliptic curve over a field can have different structures depending on the characteristic of the field
- Over the complex numbers, the torsion subgroup is isomorphic to a product of two cyclic groups
- Over the rational numbers, the torsion subgroup is either cyclic or a product of two cyclic groups with specific possible orders (Mazur's theorem)
- Understanding the torsion structure of elliptic curves is important for studying their arithmetic properties and for applications in cryptography

Elliptic curves over complex numbers

- Elliptic curves over the complex numbers have a rich structure and deep connections to complex analysis and algebraic geometry
- The study of elliptic curves over the complex numbers reveals important relationships between elliptic functions, modular forms, and lattices

Lattices and isomorphism

- Every elliptic curve over the complex numbers is isomorphic to a complex torus, which is the quotient of the complex plane by a lattice
- A lattice is a discrete subgroup of the complex plane of the form $\Lambda = \{m\omega_1 + n\omega_2 : m, n \in \mathbb{Z}\}$, where ω_1 and ω_2 are complex numbers linearly independent over the real numbers
- The isomorphism between an elliptic curve and a complex torus is given by the Weierstrass $\wp$ -function and its derivative
- The period lattice of an elliptic curve determines its isomorphism class and j -invariant

Elliptic functions and invariants

- Elliptic functions are meromorphic functions on the complex plane that are periodic with respect to a lattice
- The Weierstrass $\wp$ -function is a fundamental elliptic function that satisfies the differential equation $(\wp')^2 = 4\wp^3 - g_2\wp - g_3$, where g_2 and g_3 are invariants of the lattice
- The Eisenstein series G_4 and G_6 are modular forms that can be used to express the invariants g_2 and g_3 in terms of the lattice
- The j -invariant of an elliptic curve can be expressed in terms of the Eisenstein series as
$$j = 1728 \frac{G_4^3}{G_4^3 - 27G_6^2}$$

Complex multiplication

- Some elliptic curves over the complex numbers have additional symmetries, known as complex multiplication (CM)

- An elliptic curve has complex multiplication if its endomorphism ring, the ring of isogenies from the curve to itself, is larger than the integers
- Elliptic curves with complex multiplication correspond to lattices that are invariant under certain imaginary quadratic fields
- The j -invariants of elliptic curves with complex multiplication are algebraic integers, and they play a crucial role in the theory of class fields and the construction of abelian extensions of imaginary quadratic fields

Modular curves and functions

- Modular curves are algebraic curves that parametrize isomorphism classes of elliptic curves together with additional structure, such as torsion points or isogenies
- The modular curve $X(1)$ parametrizes isomorphism classes of elliptic curves, and it is isomorphic to the complex plane quotiented by the modular group $SL(2, \mathbb{Z})$
- Modular functions are functions on the upper half-plane that are invariant under the action of subgroups of the modular group
- The j -invariant is an example of a modular function, and it provides an isomorphism between the modular curve $X(1)$ and the complex plane
- Modular curves and functions play a central role in the study of elliptic curves and their moduli spaces, and they have important applications in number theory and cryptography

Elliptic curves over finite fields

- Elliptic curves over finite fields have important applications in cryptography and coding theory
- The study of elliptic curves over finite fields involves understanding their point counts, endomorphism rings, and algorithms for efficient computation

Hasse's theorem on point counts

- Hasse's theorem provides bounds on the number of points on an elliptic curve over a finite field
- For an elliptic curve E over a finite field $\mathbb{F}_q$, the number of points $\#E(\mathbb{F}_q)$ satisfies the inequality $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$
- The quantity $a_q = q+1 - \#E(\mathbb{F}_q)$ is called the trace of Frobenius and plays a crucial role in the study of elliptic curves over finite fields
- Hasse's theorem is a consequence of the Riemann hypothesis for curves over finite fields, which was proved by André Weil

Supersingular vs ordinary curves

- Elliptic curves over finite fields can be classified into two types: supersingular and ordinary
- An elliptic curve E over a finite field $\mathbb{F}_q$ of characteristic p is called supersingular if the trace of Frobenius a_q is divisible by p ; otherwise, it is called ordinary
- Supersingular curves have special properties and are of particular interest in cryptography and coding theory

- They have a larger endomorphism ring than ordinary curves
- They are more resistant to certain attacks in elliptic curve cryptography
- The j -invariants of supersingular curves over a given finite field are roots of the supersingular polynomial, which has degree at most $\lfloor \frac{p}{12} \rfloor$

Endomorphism rings

- The endomorphism ring of an elliptic curve over a finite field is the ring of isogenies from the curve to itself
- For ordinary elliptic curves, the endomorphism ring is an order in an imaginary quadratic field
- For supersingular curves, the endomorphism ring is an order in a quaternion algebra
- The structure of the endomorphism ring provides information about the arithmetic and geometric properties of the elliptic curve
- Understanding the endomorphism ring is important for constructing efficient algorithms for point counting and other computations on elliptic curves over finite fields

Schoof's algorithm for point counting

- Schoof's algorithm is a polynomial-time algorithm for counting the number of points on an elliptic curve over a finite field
- The algorithm uses the action of the Frobenius endomorphism on the ℓ -torsion points of the curve for various small primes ℓ
- By computing the trace of Frobenius modulo several small primes and using the Chinese remainder theorem, Schoof's algorithm determines the trace of Frobenius and hence the point count
- Schoof's algorithm has a running time of $O(\log^8 q)$ for a curve over $\mathbb{F}_q$, which is polynomial in the size of the input
- Improvements to Schoof's algorithm, such as the Schoof-Elkies-Atkin (SEA) algorithm, have led to even faster point-counting methods for elliptic curves over finite fields

Modularity theorem

- The modularity theorem, also known as the Taniyama-Shimura conjecture or the Taniyama-Shimura-Weil conjecture, establishes a profound connection between elliptic curves over the rational numbers and modular forms
- The theorem states that every elliptic curve over the rational numbers is modular, meaning that it corresponds to a modular form of a specific level and weight
- The modularity theorem was initially conjectured in the 1950s and 1960s, and it was fully proved in 2001 by Breuil, Conrad, Diamond, and Taylor, building on the work of Andrew Wiles and others

Elliptic curves over rationals

- An elliptic curve over the rational numbers is a curve of the form $y^2 = x^3 + ax + b$, where a and b are rational numbers, and the curve has no singular points

- The set of rational points on an elliptic curve, together with the point at infinity, forms a finitely generated abelian group (Mordell-Weil theorem)
- The rank of an elliptic curve over the rationals is the number of free generators of its group of rational points
- Computing the rank of an elliptic curve is a difficult problem, and it is related to the Birch and Swinnerton-Dyer conjecture, one of the Millennium Prize Problems

Modular forms and Hecke operators

- Modular forms are complex analytic functions on the upper half-plane that satisfy certain transformation properties under the action of subgroups of the modular group $SL(2, \mathbb{Z})$
- A modular form f of weight k and level N is a holomorphic function on the upper half-plane that satisfies $f\left(\frac{az+b}{cz+d}\right) = (cz+d)^k f(z)$ for all

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \in \Gamma_0(N)$$

, and is holomorphic at the cusps

- Hecke operators are linear operators that act on the space of modular forms of a given weight and level
- Hecke operators are used to define Hecke eigenforms, which are simultaneous eigenfunctions of all Hecke operators
- The Fourier coefficients of a Hecke eigenform satisfy certain multiplicative properties and are related to the eigenvalues of the Hecke operators

Galois representations

- A Galois representation is a continuous homomorphism from the absolute Galois group of a field (usually the rational numbers) to the general linear group of a vector space over a field (usually a finite field or a p -adic field)
- Elliptic curves and modular forms both give rise to Galois representations
- The action of the absolute Galois group on the torsion points of an elliptic curve defines a Galois representation
- The Fourier coefficients of a Hecke eigenform generate a number field, and the action of the absolute Galois group on the roots of the characteristic polynomial of the Hecke operators defines a Galois representation
- The modularity theorem establishes a correspondence between the Galois representations arising from elliptic curves and those arising from modular forms

L-functions of elliptic curves

- The L-function of an elliptic curve over the rational numbers is a complex analytic function that encodes important arithmetic information about the curve
- The L-function of an elliptic curve E is defined as a product of local factors $L_p(E, s)$ over all primes p , where the local factors are determined by the number of points on the curve modulo p

- The modularity theorem implies that the L-function of an elliptic curve over the rationals coincides with the L-function of a modular form of weight 2 and a certain level
- The Birch and Swinnerton-Dyer conjecture relates the rank of an elliptic curve to the order of vanishing of its L-function at $s = 1$

Frey curves and Fermat's last theorem

- The modularity theorem played a crucial role in Andrew Wiles' proof of Fermat's last theorem
- Fermat's last theorem states that the equation $x^n + y^n = z^n$ has no non-zero integer solutions for $n > 2$
- Wiles' proof relied on showing that a certain class of elliptic curves, known as Frey curves, cannot exist
- A Frey curve is an elliptic curve constructed from a hypothetical solution to Fermat's equation
- The modularity theorem, combined with work on Galois representations and modular forms, led to a contradiction that proved the non-existence of Frey curves
- The proof of Fermat's last theorem is a testament to the power and importance of the modularity theorem in number theory

Applications of modularity

- The modularity theorem has numerous applications in number theory, including the study of Diophantine equations, congruent numbers, and the Birch and Swinnerton-Dyer conjecture
- The connection between elliptic curves and modular forms provided by the modularity theorem has led to new insights and techniques in solving long-standing problems

Congruent number problem

- A congruent number is a positive integer that is the area of a right triangle with rational side lengths
- The congruent number problem asks which integers are congruent numbers
- The congruent number problem can be reformulated in terms of the rank of elliptic curves
- An integer n is a congruent number if and only if the elliptic curve $E_n : y^2 = x^3 - n^2x$ has positive rank over the rational numbers
- The modularity theorem provides a powerful tool for studying the congruent number problem, as it relates the rank of the elliptic curve to the vanishing of the associated modular form's L-function

Sums of squares as congruent numbers

- The modularity theorem has been used to prove results about the representation of integers as sums of squares
- For example, a positive integer n is a congruent number if and only if the elliptic curve $E_n : y^2 = x^3 - n^2x$ has a rational point of infinite order
- Using the modularity theorem and the theory of modular forms, it has been shown that if n is a sum of two rational squares, then E_n has a rational point of

Unit 8 – Algorithms for elliptic curves

8.1 Elliptic curve point multiplication algorithms

Elliptic curve point multiplication is a crucial operation in cryptography. It involves computing multiples of points on elliptic curves, which forms the basis for key generation, encryption, and digital signatures in elliptic curve cryptography (ECC).

Efficient algorithms for point multiplication are essential for practical ECC implementations. This topic covers various methods to optimize point multiplication, including binary, window-based, and endomorphism-based approaches, as well as techniques to protect against side-channel attacks.

Basics of elliptic curve point multiplication

- Point multiplication is a fundamental operation in elliptic curve cryptography that involves computing the multiple of a point on an elliptic curve by a scalar value
- The operation is denoted as $[k]P$ where k is the scalar multiplier and P is a point on the elliptic curve
- Point multiplication is used in key generation, encryption, decryption, and digital signature algorithms in ECC
- The security of ECC relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), which involves finding the scalar multiplier given a point and its multiple

Efficient algorithms for point multiplication

- Efficient algorithms for point multiplication are essential for practical implementation of elliptic curve cryptography
- Naive approaches, such as repeated point addition, have a linear time complexity and are not suitable for real-world applications
- Several optimization techniques have been developed to reduce the number of point additions and improve the performance of point multiplication

Binary method for point multiplication

- The binary method is a simple and widely used algorithm for point multiplication
- It represents the scalar multiplier k in binary form and performs point doubling and addition based on the binary digits
- For each binary digit, the algorithm performs a point doubling, and if the digit is 1, it also performs a point addition
- The binary method has a time complexity of $O(\log k)$ point operations

Window-based methods for faster computation

- Window-based methods are an optimization of the binary method that processes multiple bits of the scalar multiplier at a time
- The scalar multiplier is divided into windows of a fixed or variable size, and pre-computed points are used to reduce the number of point additions

- Examples of window-based methods include the fixed-window method and the sliding-window method
- Window-based methods can significantly reduce the number of point operations compared to the binary method

Sliding window method vs fixed window method

- The sliding window method is an improvement over the fixed window method
- In the fixed window method, the window size is constant throughout the computation, while in the sliding window method, the window size varies based on the binary representation of the scalar multiplier
- The sliding window method skips consecutive zeros in the binary representation, reducing the number of point doublings
- The sliding window method requires more pre-computed points than the fixed window method but generally achieves better performance

Scalar multiplication in elliptic curve cryptography

- Scalar multiplication is the cornerstone of elliptic curve cryptography (ECC) and is used in various cryptographic protocols
- In ECC, the private key is a scalar value, and the corresponding public key is a point on the elliptic curve obtained by multiplying the base point by the private key
- Scalar multiplication is used in key exchange protocols (ECDH), digital signature algorithms (ECDSA), and encryption schemes (ECIES)

Importance of scalar multiplication in ECC

- The security of ECC relies on the hardness of the elliptic curve discrete logarithm problem (ECDLP)
- Given a point P on an elliptic curve and its multiple $[k]P$, it is computationally infeasible to determine the scalar multiplier k
- The hardness of ECDLP enables smaller key sizes in ECC compared to other public-key cryptosystems (RSA, DSA) while maintaining the same level of security
- Efficient and secure implementations of scalar multiplication are crucial for the performance and security of ECC-based systems

Hardness of elliptic curve discrete logarithm problem

- The elliptic curve discrete logarithm problem (ECDLP) is the problem of finding the scalar multiplier k given a point P and its multiple $[k]P$ on an elliptic curve
- The hardness of ECDLP is the foundation of security in elliptic curve cryptography
- The best known algorithms for solving ECDLP have a subexponential time complexity, making it infeasible to solve for properly chosen elliptic curves and key sizes
- The hardness of ECDLP enables ECC to achieve the same level of security as other public-key cryptosystems with smaller key sizes, leading to more efficient implementations

Projective coordinates for point multiplication

- Projective coordinates are an alternative representation of points on an elliptic curve that can improve the efficiency of point multiplication
- In affine coordinates, a point is represented by a pair of coordinates (x, y) satisfying the elliptic curve equation
- Projective coordinates introduce an additional coordinate (e.g., Z) to represent points, allowing for more efficient point arithmetic

Advantages of projective coordinates over affine

- Projective coordinates eliminate the need for expensive field inversions during point addition and doubling operations
- Field inversions are replaced by less expensive field multiplications, which can significantly speed up point multiplication
- Projective coordinates also provide a unified representation for both affine points and the point at infinity, simplifying the implementation of point arithmetic

Jacobian coordinates for efficient implementation

- Jacobian coordinates are a popular choice of projective coordinates for elliptic curve point multiplication
- In Jacobian coordinates, a point (X, Y, Z) corresponds to the affine point $(X/Z^2, Y/Z^3)$ on the elliptic curve
- Jacobian coordinates allow for efficient point doubling and addition formulas that minimize the number of field multiplications and squarings
- Many efficient implementations of ECC use Jacobian coordinates to achieve better performance compared to affine coordinates

López-Dahab coordinates for binary curves

- López-Dahab coordinates are a type of projective coordinates specifically designed for elliptic curves over binary fields
- In López-Dahab coordinates, a point (X, Y, Z) corresponds to the affine point $(X/Z, Y/Z^2)$ on the elliptic curve
- López-Dahab coordinates provide efficient point doubling and addition formulas for binary curves, which are commonly used in embedded systems and hardware implementations
- The use of López-Dahab coordinates can lead to faster point multiplication on binary curves compared to affine coordinates

Endomorphisms for accelerating point multiplication

- Endomorphisms are mappings from an elliptic curve to itself that preserve the group structure
- Certain classes of elliptic curves possess efficient endomorphisms that can be used to accelerate point multiplication

- Endomorphisms allow for the decomposition of the scalar multiplier into smaller components, reducing the number of point operations required for point multiplication

Frobenius endomorphism in characteristic p curves

- The Frobenius endomorphism is a special endomorphism available on elliptic curves defined over finite fields of characteristic p
- The Frobenius endomorphism maps a point (x, y) to (x^p, y^p) , where p is the characteristic of the field
- The Frobenius endomorphism can be efficiently computed and has a low cost compared to point addition and doubling operations
- Elliptic curves with efficient Frobenius endomorphisms, such as Koblitz curves, can benefit from faster point multiplication

Gallant-Lambert-Vanstone (GLV) method using endomorphisms

- The Gallant-Lambert-Vanstone (GLV) method is a technique that uses endomorphisms to accelerate point multiplication on certain classes of elliptic curves
- The GLV method decomposes the scalar multiplier k into two smaller components k_1 and k_2 using the endomorphism
- The point multiplication $[k]P$ is then computed as $[k_1]P + [k_2]\phi(P)$, where ϕ is the endomorphism
- The GLV method can significantly reduce the number of point operations required for point multiplication, leading to faster execution times

Guillevic-Ionica method for fast endomorphisms

- The Guillevic-Ionica method is an extension of the GLV method that allows for the use of endomorphisms on a wider range of elliptic curves
- The method uses a combination of the Frobenius endomorphism and a non-trivial endomorphism to decompose the scalar multiplier into four smaller components
- The point multiplication is then computed using a combination of the smaller scalar multiplications and the endomorphisms
- The Guillevic-Ionica method can achieve even faster point multiplication compared to the GLV method on suitable elliptic curves

Side-channel attacks on point multiplication

- Side-channel attacks are a class of attacks that exploit information leakage from the physical implementation of cryptographic algorithms
- In the context of elliptic curve cryptography, side-channel attacks can target the point multiplication operation to recover the secret scalar multiplier
- Common side-channel attacks on point multiplication include timing attacks, power analysis attacks, and electromagnetic emanation attacks

Timing attacks on scalar multiplication algorithms

- Timing attacks exploit the differences in execution time of point multiplication algorithms depending on the value of the scalar multiplier
- If the execution time of point multiplication depends on the bits of the scalar multiplier, an attacker can observe the timing variations and deduce information about the secret key
- Timing attacks can be mitigated by ensuring that the execution time of point multiplication is independent of the scalar multiplier, using techniques such as fixed-time algorithms or blinding

Power analysis attacks on ECC implementations

- Power analysis attacks measure the power consumption of a device during the execution of point multiplication to extract information about the scalar multiplier
- Simple power analysis (SPA) attacks observe a single power trace to deduce the sequence of point operations and recover the scalar multiplier
- Differential power analysis (DPA) attacks use statistical analysis of multiple power traces to recover the scalar multiplier, even in the presence of noise
- Power analysis attacks can be mitigated using techniques such as randomization, masking, or using constant-time algorithms

Countermeasures for side-channel attack resistance

- Countermeasures are techniques used to protect point multiplication implementations against side-channel attacks
- Randomization countermeasures introduce random values into the point multiplication algorithm to make the side-channel information unpredictable to the attacker
- Masking countermeasures split the sensitive data (scalar multiplier) into multiple shares and perform computations on the shares to prevent direct leakage of the sensitive data
- Constant-time algorithms ensure that the execution time and power consumption of point multiplication are independent of the scalar multiplier, making it difficult for an attacker to extract useful information
- A combination of countermeasures is often used to provide comprehensive protection against various side-channel attacks

Batch point multiplication techniques

- Batch point multiplication refers to the simultaneous computation of multiple point multiplications on an elliptic curve
- Batch point multiplication techniques aim to optimize the computation of $[k_1]P_1 + [k_2]P_2 + \dots + [k_n]P_n$, where k_i are scalar multipliers and P_i are points on the elliptic curve
- Batch point multiplication is used in scenarios where multiple point multiplications need to be computed efficiently, such as in signature verification or multi-party protocols

Straus-Shamir method for simultaneous point multiplication

- The Straus-Shamir method, also known as Shamir's trick, is a technique for computing the sum of multiple point multiplications
- The method is based on the observation that the sum of point multiplications can be computed more efficiently than individual point multiplications
- The Straus-Shamir method represents the scalar multipliers in binary form and performs point additions and doublings based on the binary representation
- The method can achieve a significant speedup compared to computing individual point multiplications and then adding the results

Lim-Lee method for fixed-base point multiplication

- The Lim-Lee method is a technique for efficiently computing multiple point multiplications with a fixed base point
- In the Lim-Lee method, the base point is pre-computed and stored in a table, allowing for faster computation of point multiplications
- The method splits the scalar multipliers into smaller windows and uses the pre-computed table to perform the point multiplications
- The Lim-Lee method is particularly useful in scenarios where multiple point multiplications with the same base point need to be computed, such as in signature verification

Bernstein's Batch Edwards point multiplication method

- Bernstein's Batch Edwards point multiplication method is a technique for efficient batch point multiplication on Edwards curves
- Edwards curves are a class of elliptic curves that provide fast and complete point addition formulas, making them suitable for efficient implementation
- Bernstein's method uses a combination of point addition and doubling formulas specific to Edwards curves to optimize the computation of batch point multiplication
- The method achieves high performance by exploiting the properties of Edwards curves and using optimized algorithms for point arithmetic
- Bernstein's Batch Edwards point multiplication method is widely used in modern elliptic curve cryptography implementations for its efficiency and security

8.2 Elliptic curve factorization methods

Elliptic curve factorization methods are powerful tools for finding small to medium-sized factors of large composite numbers. These techniques leverage the group structure of elliptic curves over finite fields, offering advantages like lower memory requirements and easy parallelization compared to other factorization algorithms.

Lenstra's original algorithm and Montgomery's improved method form the foundation of elliptic curve factorization. These approaches use carefully chosen curves and points, performing arithmetic operations to find factors. Optimizations in curve selection, parameter choice, and implementation have made ECM a crucial component in modern factorization efforts.

Elliptic curve method

- Efficient factorization method for finding small to medium-sized factors of large composite numbers
- Based on the properties of elliptic curves over finite fields
- Particularly effective when the number to be factored has one or more relatively small prime factors

Overview of ECM

- Utilizes the group structure of elliptic curves to find factors of a composite number
- Performs arithmetic operations on points of a randomly chosen elliptic curve
- Attempts to find a factor by constructing a point with a sufficiently large order on the curve
- If the factor is not found, the process is repeated with a different elliptic curve

Advantages vs other factorization methods

- Performs well for numbers with small prime factors, while other methods (like Quadratic Sieve and Number Field Sieve) are more efficient for numbers with large prime factors
- Has a lower memory requirement compared to other factorization algorithms
- Can be easily parallelized, allowing for efficient implementation on multiple processors or computers
- Serves as a useful preprocessing step to remove small factors before applying more complex factorization methods

Lenstra's elliptic curve factorization

- Developed by Hendrik Lenstra in 1987 as the first practical application of elliptic curves in factorization
- Generalizes the idea of Pollard's $p - 1$ method using elliptic curves

Lenstra's original algorithm

- Choose a random elliptic curve E over $\mathbb{Z}/n\mathbb{Z}$ and a random point P on E
- Compute the point $Q = kP$ for a suitable integer k , using elliptic curve arithmetic
- Calculate the greatest common divisor (GCD) of the x -coordinate of Q and n
- If the GCD is a non-trivial factor of n , then a factor has been found; otherwise, repeat the process with a new elliptic curve and point

Key steps in Lenstra's ECM

1. Curve selection: Choose a random elliptic curve E over $\mathbb{Z}/n\mathbb{Z}$
2. Point selection: Choose a random point P on the elliptic curve E
3. Scalar multiplication: Compute $Q = kP$ for a suitable integer k
4. GCD computation: Calculate $\gcd(x_Q, n)$, where x_Q is the x -coordinate of Q
5. Factor check: If the GCD is a non-trivial factor of n , output the factor; otherwise, go back to step 1

Complexity of Lenstra's method

- The expected running time of Lenstra's ECM is $O(L(p)^{\sqrt{2} + o(1)})$, where p is the smallest prime factor of n and $L(x) = \exp(\sqrt{\log x \log \log x})$
- The complexity depends on the size of the smallest prime factor, making it more effective for numbers with small factors

Improvements to Lenstra's algorithm

- Use of Montgomery curves for more efficient arithmetic operations
- Better strategies for selecting curves and initial points
- Optimization of curve parameters to increase the probability of finding factors
- Parallelization techniques to distribute the workload across multiple processors or computers

Montgomery's elliptic curve method

- Developed by Peter Montgomery as an improvement to Lenstra's ECM
- Uses a special form of elliptic curves (Montgomery curves) to optimize arithmetic operations

Montgomery curves

- Have the form $By^2 = x^3 + Ax^2 + x$, where $A, B \in \mathbb{Z}/n\mathbb{Z}$ and $B(A^2 - 4) \neq 0$
- Allow for efficient doubling and differential addition operations, which are the main building blocks of scalar multiplication
- Require fewer intermediate variables and modular inversions compared to general elliptic curves

Montgomery's ECM algorithm

1. Choose a random Montgomery curve E over $\mathbb{Z}/n\mathbb{Z}$ and a random point P on E
2. Compute the point $Q = kP$ using Montgomery's efficient scalar multiplication algorithm
3. Calculate the GCD of the x -coordinate of Q and n
4. If the GCD is a non-trivial factor of n , output the factor; otherwise, repeat the process with a new Montgomery curve and point

Advantages of Montgomery's method

- Faster arithmetic operations due to the special form of Montgomery curves
- Reduced memory requirements as fewer intermediate variables are needed
- More efficient scalar multiplication, which is the most time-consuming part of ECM
- Allows for easier implementation of optimizations and parallelization techniques

Stage 1 vs Stage 2 in Montgomery's ECM

- Stage 1: Performs scalar multiplication with a fixed bound B_1 to find factors up to a certain size

- Bound B_1 is chosen based on the desired trade-off between success probability and computational cost
- Typically accounts for the majority of the computational effort in ECM
- Stage 2: Extends the search for factors beyond the bound B_1 used in Stage 1
- Performs additional scalar multiplications with prime powers up to a larger bound B_2
- Increases the probability of finding factors at the cost of more computation
- Usually takes less time than Stage 1 but is still significant in the overall runtime of ECM

Choosing parameters for ECM

- The choice of parameters, such as the elliptic curve, initial point, and bounds, significantly impacts the performance and success probability of ECM
- Proper parameter selection is crucial for optimizing the efficiency of the factorization process

Importance of curve selection

- The selected elliptic curve determines the group structure and the efficiency of arithmetic operations
- Different curves have varying probabilities of yielding a factor for a given composite number
- Selecting curves with desirable properties (e.g., smooth group order) can increase the chances of finding factors

Strategies for selecting curves

- Random curve selection: Choose curves randomly from a predefined set or generate them on-the-fly
- Sieving for good curves: Precompute a set of curves with desirable properties and use them in the factorization process
- Curve recycling: Reuse curves that have been successful in previous factorizations, exploiting their proven effectiveness

Selecting initial points

- The choice of the initial point on the elliptic curve affects the probability of finding a factor
- Strategies for selecting initial points include:
 - Random point selection: Choose a random point on the curve
 - Torsion point selection: Select points with a specific torsion structure to optimize the factorization process
 - Linear combination of torsion points: Use a linear combination of torsion points to generate the initial point

Optimizing curve parameters

- Adjust the curve parameters (e.g., coefficients A and B for Montgomery curves) to optimize arithmetic operations
- Choose parameters that minimize the cost of point doubling and differential addition

- Consider the trade-off between the complexity of parameter selection and the efficiency of the resulting arithmetic operations

ECM implementations and optimizations

- Efficient implementations and optimizations are essential for the practical application of ECM in factoring large numbers
- Various techniques can be employed to improve the performance and scalability of ECM

Efficient arithmetic on elliptic curves

- Optimize the implementation of point addition, doubling, and scalar multiplication operations
- Use projective coordinates to avoid expensive modular inversions
- Employ efficient algorithms for modular arithmetic, such as Montgomery multiplication or Barrett reduction

Parallelization techniques for ECM

- Distribute the workload across multiple processors or computers to accelerate the factorization process
- Implement parallel versions of ECM, such as the Brent-Pollard Monte Carlo method or the Batch ECM algorithm
- Exploit SIMD (Single Instruction, Multiple Data) parallelism to perform multiple arithmetic operations simultaneously

Hardware acceleration of ECM

- Utilize specialized hardware, such as GPUs or FPGAs, to accelerate ECM computations
- Implement optimized algorithms and data structures for hardware architectures
- Take advantage of the parallel processing capabilities and high memory bandwidth of hardware accelerators

Comparison of popular ECM implementations

- GMP-ECM: A widely used implementation of ECM based on the GNU Multiple Precision Arithmetic Library (GMP)
- EECM-MPFQ: An efficient implementation of ECM using the MPFQ library for finite field arithmetic
- CADO-NFS: A complete implementation of the Number Field Sieve, including an optimized ECM component for factor base generation
- Msieve: A general-purpose factorization tool that includes an optimized implementation of ECM

Combining ECM with other methods

- ECM can be used in conjunction with other factorization methods to improve the overall efficiency of the factoring process

- Integrating ECM with more complex algorithms, such as the Quadratic Sieve or the Number Field Sieve, can lead to faster factorization times

ECM in the Quadratic Sieve

- Use ECM to find small factors of the number to be factored before applying the Quadratic Sieve
- Remove small factors to reduce the size of the number and simplify the sieving process
- Incorporate ECM in the Quadratic Sieve's factor base generation stage to identify and remove small prime factors

ECM in the Number Field Sieve

- Employ ECM to find small factors of the number to be factored before running the Number Field Sieve
- Use ECM in the polynomial selection stage to identify and remove small prime factors from the coefficients of the polynomials
- Apply ECM to the algebraic and rational factor bases to reduce their size and improve the efficiency of the sieving stage

ECM as a preprocessing step

- Run ECM as a preprocessing step before applying more complex factorization methods
- Eliminate small factors to reduce the size of the number and simplify subsequent factorization stages
- Combine ECM with other preprocessing techniques, such as trial division or Pollard's Rho method, to optimize the overall factorization process

Balancing ECM with other factorization stages

- Determine the optimal balance between the time spent on ECM and other factorization stages
- Adjust the ECM parameters (e.g., bounds, number of curves) based on the characteristics of the number to be factored
- Consider the trade-off between the success probability of ECM and the computational cost of the subsequent factorization stages
- Dynamically adapt the ECM parameters during the factorization process based on the progress and intermediate results obtained

8.3 Schoof's algorithm for point counting

Schoof's algorithm revolutionized point counting on elliptic curves over finite fields. It was the first polynomial-time solution, making it feasible to count points on curves used in cryptography. The algorithm's key insight is using the Frobenius endomorphism and division polynomials to compute the trace modulo small primes.

Schoof's approach paved the way for more efficient methods like the Schoof-Elkies-Atkin (SEA) algorithm. These techniques are crucial for selecting secure parameters in elliptic curve cryptography. Understanding Schoof's algorithm provides insight into the interplay between elliptic curves, finite fields, and computational number theory.

Overview of Schoof's algorithm

- Schoof's algorithm is a polynomial-time algorithm for counting the number of points on an elliptic curve over a finite field
- Developed by René Schoof in 1985, it was the first algorithm to solve the point counting problem in polynomial time
- The algorithm relies on the properties of the Frobenius endomorphism and the division polynomials associated with the elliptic curve

Key ideas of Schoof's approach

- Schoof's algorithm reduces the point counting problem to computing the trace of the Frobenius endomorphism modulo small primes
- The algorithm exploits the relationship between the trace of Frobenius and the number of points on the curve
- By computing the trace modulo sufficiently many small primes, the actual trace can be recovered using the Chinese Remainder Theorem

Reduction to torsion points

- Schoof's algorithm focuses on computing the Frobenius action on torsion points of the elliptic curve
- Torsion points are points of finite order, i.e., points P such that $nP = O$ for some positive integer n
- The algorithm computes the Frobenius action on ℓ -torsion points for small primes ℓ , which helps determine the trace of Frobenius modulo ℓ

Frobenius endomorphism

- The Frobenius endomorphism ϕ is a special endomorphism of an elliptic curve over a finite field
- For an elliptic curve E over a finite field $\mathbb{F}_q$, the Frobenius endomorphism is defined as $\phi(x, y) = (x^q, y^q)$
- The Frobenius endomorphism satisfies a characteristic equation that relates its trace to the number of points on the curve

Characteristic equation

- The Frobenius endomorphism ϕ satisfies the characteristic equation $\phi^2 - t\phi + q = 0$, where t is the trace of Frobenius and q is the size of the finite field
- The number of points on the elliptic curve E over $\mathbb{F}_q$ is given by $\#E(\mathbb{F}_q) = q + 1 - t$
- Schoof's algorithm aims to compute the trace t modulo small primes to recover the actual value of t and, consequently, the number of points on the curve

Steps in Schoof's algorithm

- Schoof's algorithm consists of several key steps that work together to compute the trace of Frobenius and the number of points on the elliptic curve

- The main steps include defining the division polynomials, computing the torsion points, and determining the trace of Frobenius modulo small primes
- The algorithm iterates over a set of small primes, performing these steps for each prime until enough information is gathered to recover the actual trace

Defining the division polynomials

- Division polynomials are a sequence of polynomials associated with an elliptic curve that characterize the torsion points
- The ℓ -th division polynomial $\psi_\ell(x, y)$ vanishes precisely at the ℓ -torsion points of the curve
- Schoof's algorithm uses division polynomials to identify the ℓ -torsion points and compute the Frobenius action on them

Computing the torsion points

- For each small prime ℓ , Schoof's algorithm computes the ℓ -torsion points of the elliptic curve
- The ℓ -torsion points are the solutions to the equation $\psi_\ell(x, y) = 0$
- The algorithm efficiently computes the ℓ -torsion points by exploiting the structure of the division polynomials and using polynomial arithmetic

Determining the trace of Frobenius

- Once the ℓ -torsion points are computed, Schoof's algorithm determines the action of the Frobenius endomorphism on these points
- The algorithm computes the Frobenius action on each ℓ -torsion point P by evaluating $\phi(P)$ and comparing it to the scalar multiples of P
- By examining the Frobenius action on the ℓ -torsion points, the algorithm determines the trace of Frobenius modulo ℓ

Complexity of Schoof's algorithm

- Schoof's algorithm is notable for its polynomial time complexity, which was a significant improvement over previous point counting methods
- The complexity of the algorithm depends on several factors, including the size of the finite field and the number of small primes used
- While polynomial, the algorithm still faces some bottlenecks that have led to further optimizations and improvements

Polynomial time complexity

- Schoof's algorithm has a time complexity of $O(\log^8 q)$, where q is the size of the finite field
- This polynomial time complexity is achieved by iterating over a set of small primes ℓ and performing computations modulo ℓ
- The number of primes required is proportional to $\log q$, and the computations for each prime take polynomial time

Bottlenecks and optimizations

- The main bottleneck in Schoof's algorithm is the computation of the ℓ -torsion points, which involves solving high-degree polynomial equations
- The division polynomials used in the algorithm have degrees that grow quadratically with ℓ , leading to increased computational complexity
- Optimizations to Schoof's algorithm have focused on reducing the complexity of computing the ℓ -torsion points and the Frobenius action on them

Extensions of Schoof's algorithm

- Since its introduction, Schoof's algorithm has been extended and improved by various researchers
- These extensions aim to reduce the time complexity of the algorithm and make it more practical for larger finite fields
- The most notable extension is the Schoof-Elkies-Atkin (SEA) algorithm, which combines ideas from Schoof's algorithm with the work of Noam Elkies and A.O.L. Atkin

Schoof-Elkies-Atkin (SEA) algorithm

- The SEA algorithm improves upon Schoof's algorithm by incorporating the use of modular polynomials and isogenies
- Elkies and Atkin independently discovered that for certain primes ℓ , called Elkies primes, the ℓ -torsion points can be computed more efficiently using modular polynomials
- The SEA algorithm uses a combination of Elkies primes and Atkin primes (for which modular polynomials are not available) to compute the trace of Frobenius

Improvements in time complexity

- The SEA algorithm has a time complexity of $O(\log^6 q)$, which is a significant improvement over Schoof's original algorithm
- The use of modular polynomials and isogenies allows for more efficient computation of the ℓ -torsion points and the Frobenius action
- Further optimizations, such as the use of volcano structures and the Chinese Remainder Theorem, have led to even faster implementations of the SEA algorithm

Applications of point counting

- Point counting algorithms, such as Schoof's algorithm and its extensions, have important applications in elliptic curve cryptography
- Elliptic curve cryptography relies on the difficulty of the discrete logarithm problem on elliptic curves, which is closely related to the number of points on the curve
- Accurate point counting is crucial for selecting secure parameters and assessing the security of elliptic curve cryptosystems

Cryptographic security

- The security of elliptic curve cryptography depends on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP)
- The difficulty of the ECDLP is directly related to the number of points on the elliptic curve
- Point counting algorithms provide a way to determine the exact number of points, allowing for accurate security assessments and parameter selection

Parameter selection for curves

- When designing elliptic curve cryptosystems, it is essential to choose curves with desirable properties, such as a prime or near-prime number of points
- Point counting algorithms enable the efficient determination of the number of points on candidate curves
- By using point counting algorithms, cryptographers can select curves with optimal security properties and avoid curves with known weaknesses or vulnerabilities

Comparisons to other point counting methods

- Schoof's algorithm was a groundbreaking development in point counting, but it is not the only method available
- Other notable point counting methods include the Shanks-Mestre algorithm and p-adic methods
- Each method has its own strengths and weaknesses, and the choice of algorithm often depends on the specific characteristics of the elliptic curve and the finite field

Schoof vs Shanks-Mestre

- The Shanks-Mestre algorithm is an older point counting method that predates Schoof's algorithm
- It has a time complexity of $O(q^{1/4})$, which is exponential in the size of the finite field
- While less efficient than Schoof's algorithm asymptotically, the Shanks-Mestre algorithm can be faster for certain small finite fields due to its simpler implementation

Schoof vs p-adic methods

- p-adic methods for point counting, such as the Satoh-Skjernaa-Taguchi (SST) algorithm, work over p-adic fields instead of finite fields
- These methods have a time complexity of $O(\log^3 q)$, which is better than Schoof's algorithm asymptotically
- However, p-adic methods are more complex to implement and have higher memory requirements compared to Schoof's algorithm and its extensions
- In practice, the choice between Schoof-based methods and p-adic methods depends on the specific requirements of the application and the available computational resources

8.4 SEA (Schoof-Elkies-Atkin) algorithm

The SEA algorithm is a game-changer for counting points on elliptic curves over finite fields. It combines Schoof's algorithm with Elkies and Atkin primes, achieving polynomial-time complexity and making it practical for cryptographic use.

SEA's efficiency comes from its clever use of modular polynomials and isogenies for Elkies primes, and ruling out possible traces for Atkin primes. This powerful combo enables secure curve selection and discrete logarithm computation in elliptic curve cryptography.

Overview of SEA algorithm

- The Schoof-Elkies-Atkin (SEA) algorithm is a polynomial-time algorithm for counting the number of points on an elliptic curve over a finite field
- It combines techniques from Schoof's algorithm, Elkies primes, and Atkin primes to efficiently compute the trace of the Frobenius endomorphism
- SEA has significant applications in elliptic curve cryptography, enabling the secure selection of curves and computation of discrete logarithms

Polynomial time complexity

- SEA achieves a polynomial time complexity of $O(\log^8 p)$ for counting points on an elliptic curve over a finite field of characteristic p
- This is a significant improvement over exponential-time algorithms, making SEA practical for cryptographic purposes
- The polynomial time complexity allows SEA to be used for curves over large prime fields, which are commonly used in cryptography

Applicability to cryptography

- SEA is a fundamental tool in elliptic curve cryptography for determining the security and suitability of elliptic curves
- It enables the computation of the group order of an elliptic curve, which is essential for selecting cryptographically secure curves
- SEA is used in the implementation of elliptic curve cryptosystems, such as the Elliptic Curve Digital Signature Algorithm (ECDSA) and Elliptic Curve Diffie-Hellman (ECDH) key exchange

Key components

- SEA combines three main techniques: Schoof's algorithm, Elkies primes, and Atkin primes
- Each component contributes to the efficiency and practicality of the overall algorithm
- The interplay between these components allows SEA to achieve its polynomial time complexity

Schoof's algorithm

- Schoof's algorithm is the foundation of SEA and provides a deterministic method for counting points on elliptic curves

- It uses division polynomials to compute the trace of the Frobenius endomorphism modulo small primes ℓ
- Schoof's algorithm has a complexity of $O(\log^8 p)$, which is polynomial but still relatively high for practical purposes

Elkies primes

- Elkies primes are a subset of primes ℓ for which the characteristic polynomial of the Frobenius endomorphism splits modulo ℓ
- For Elkies primes, the trace of Frobenius can be computed efficiently using modular polynomials and isogenies of elliptic curves
- Elkies primes allow for a significant speedup compared to Schoof's algorithm alone

Atkin primes

- Atkin primes are the complement of Elkies primes and are used to rule out possible traces of the Frobenius endomorphism
- For Atkin primes, the characteristic polynomial of Frobenius does not split modulo ℓ
- Atkin primes provide additional information that helps narrow down the possible values of the trace

Schoof's algorithm

- Schoof's algorithm is a deterministic point counting method that computes the trace of the Frobenius endomorphism modulo small primes ℓ
- It relies on the properties of division polynomials, which are polynomials that encode information about the ℓ -torsion points of an elliptic curve
- Schoof's algorithm computes the trace by solving a system of equations derived from the action of Frobenius on the ℓ -torsion points

Computation of trace of Frobenius

- The trace of the Frobenius endomorphism, denoted by t , satisfies the equation $\#E(\mathbb{F}_p) = p + 1 - t$, where $\#E(\mathbb{F}_p)$ is the number of points on the elliptic curve E over the finite field $\mathbb{F}_p$
- Schoof's algorithm computes t modulo small primes ℓ by evaluating the action of Frobenius on the ℓ -torsion points
- The computed values of t modulo different primes are combined using the Chinese Remainder Theorem to obtain the actual value of t

Division polynomials

- Division polynomials are a sequence of polynomials $\psi_n(x, y)$ associated with an elliptic curve E
- The roots of the n -th division polynomial $\psi_n(x, y)$ are the x -coordinates of the n -torsion points of E
- Schoof's algorithm uses division polynomials to characterize the action of Frobenius on the ℓ -torsion points and derive equations for computing the trace modulo ℓ

Complexity and limitations

- Schoof's algorithm has a complexity of $O(\log^8 p)$, which is polynomial in the size of the characteristic p of the finite field
- While polynomial, the complexity is still relatively high for practical purposes, especially for large prime fields used in cryptography
- Schoof's algorithm alone is not sufficient for efficient point counting, and further optimizations are needed, leading to the development of the Elkies and Atkin primes techniques

Elkies primes

- Elkies primes are a subset of primes ℓ for which the characteristic polynomial of the Frobenius endomorphism splits modulo ℓ
- For Elkies primes, the action of Frobenius on the ℓ -torsion points can be efficiently computed using modular polynomials and isogenies
- Elkies primes provide a significant speedup compared to Schoof's algorithm alone, making SEA practical for cryptographic purposes

Modular polynomials

- Modular polynomials $\Phi_\ell(x, y)$ are bivariate polynomials that describe the relationships between the j -invariants of ℓ -isogenous elliptic curves
- For Elkies primes ℓ , the modular polynomial $\Phi_\ell(x, j(E))$ splits completely over the base field, where $j(E)$ is the j -invariant of the elliptic curve E
- The roots of $\Phi_\ell(x, j(E))$ correspond to the j -invariants of elliptic curves that are ℓ -isogenous to E

Isogenies of elliptic curves

- An isogeny between two elliptic curves E_1 and E_2 is a non-constant rational map that preserves the group structure
- For Elkies primes ℓ , there exist ℓ -isogenies between E and the elliptic curves corresponding to the roots of $\Phi_\ell(x, j(E))$
- Isogenies allow for the efficient computation of the action of Frobenius on the ℓ -torsion points, leading to a faster determination of the trace modulo ℓ

Eigenvalues of Frobenius

- For Elkies primes ℓ , the characteristic polynomial of the Frobenius endomorphism splits completely over the base field
- The eigenvalues of Frobenius, denoted by λ_1 and λ_2 , satisfy the equation $\lambda_1 + \lambda_2 \equiv t \pmod{\ell}$, where t is the trace of Frobenius
- The eigenvalues can be efficiently computed using the ℓ -isogenies between E and the elliptic curves corresponding to the roots of $\Phi_\ell(x, j(E))$

Efficient computation of trace

- For Elkies primes ℓ , the trace of Frobenius modulo ℓ can be computed efficiently using the eigenvalues of Frobenius
- The eigenvalues λ_1 and λ_2 are computed using the ℓ -isogenies derived from the modular polynomial $\Phi_\ell(x, j(E))$
- The trace modulo ℓ is then obtained as $t \equiv \lambda_1 + \lambda_2 \pmod{\ell}$, significantly faster than Schoof's algorithm alone

Atkin primes

- Atkin primes are the complement of Elkies primes and are used to rule out possible traces of the Frobenius endomorphism
- For Atkin primes ℓ , the characteristic polynomial of Frobenius does not split modulo ℓ
- Atkin primes provide additional information that helps narrow down the possible values of the trace, complementing the information obtained from Elkies primes

Complementary approach to Elkies

- While Elkies primes allow for the efficient computation of the trace modulo ℓ , Atkin primes provide a different approach
- For Atkin primes, the characteristic polynomial of Frobenius does not split modulo ℓ , and the eigenvalues cannot be directly computed
- Instead, Atkin primes are used to rule out possible values of the trace, narrowing down the search space

Ruling out possible traces

- For Atkin primes ℓ , the trace of Frobenius modulo ℓ satisfies certain conditions based on the splitting behavior of the characteristic polynomial
- By evaluating these conditions, it is possible to rule out certain values of the trace modulo ℓ
- Ruling out possible traces helps reduce the number of candidates for the actual value of the trace, improving the efficiency of the overall algorithm

Chinese Remainder Theorem

- The information obtained from Elkies and Atkin primes is combined using the Chinese Remainder Theorem (CRT)
- CRT allows for the reconstruction of the trace of Frobenius modulo the product of the primes used in the computation
- By combining the trace values modulo different primes, the actual value of the trace can be determined, leading to the computation of the number of points on the elliptic curve

Combining Schoof, Elkies, and Atkin

- SEA combines the techniques of Schoof's algorithm, Elkies primes, and Atkin primes to achieve a polynomial-time point counting algorithm
- The choice of primes and the balance between Elkies and Atkin primes play a crucial role in optimizing the performance of SEA
- Parallelization strategies can be employed to further improve the efficiency of the algorithm

Heuristics for prime selection

- The selection of primes for use in SEA is guided by heuristics that balance the contribution of Elkies and Atkin primes
- Heuristics aim to maximize the number of Elkies primes while ensuring a sufficient number of Atkin primes for effective trace ruling
- Factors such as the size of the primes, the splitting behavior of the characteristic polynomial, and the computational cost of each technique are considered in the prime selection process

Optimizing overall performance

- SEA's performance depends on various factors, including the choice of primes, the implementation of the underlying arithmetic, and the use of pre-computed data
- Optimization techniques, such as the use of modular arithmetic, efficient polynomial arithmetic, and precomputation of frequently used values, can significantly improve the running time of SEA
- Balancing the computation between Elkies and Atkin primes, as well as the choice of the prime bound, are crucial for achieving optimal performance

Parallelization strategies

- SEA is well-suited for parallelization, as the computation of the trace modulo different primes can be performed independently
- Parallelization strategies, such as distributing the workload across multiple processors or cores, can significantly reduce the overall running time of the algorithm
- Load balancing techniques can be employed to ensure an even distribution of work among the available computing resources, maximizing the utilization of parallel processing capabilities

SEA in practice

- SEA has been widely implemented in various libraries and software packages for elliptic curve cryptography
- It has practical applications in the secure selection of elliptic curves and the computation of discrete logarithms
- SEA's performance is often compared to other point counting methods to assess its efficiency and suitability for different scenarios

Implementations and libraries

- SEA has been implemented in several popular libraries and software packages for elliptic curve cryptography
- Examples include the PARI/GP library, which provides an efficient implementation of SEA for point counting on elliptic curves over prime fields
- Other implementations can be found in specialized libraries such as libecc, Crypto++, and Miracl, which offer optimized versions of SEA for various cryptographic applications

Cryptographic applications

- SEA is a fundamental tool in elliptic curve cryptography, enabling the secure selection of elliptic curves for cryptographic protocols
- It is used to determine the group order of an elliptic curve, which is essential for ensuring the security of cryptographic schemes such as ECDSA and ECDH
- SEA is also employed in the computation of discrete logarithms on elliptic curves, a problem that underlies the security of many cryptographic protocols

Comparison to other point counting methods

- SEA is one of several point counting methods available for elliptic curves, each with its own advantages and limitations
- Other methods include the baby-step giant-step algorithm, the Pollard's rho algorithm, and the Schoof-Pila algorithm
- SEA is often compared to these methods in terms of its complexity, performance, and applicability to different types of elliptic curves and field sizes
- The choice of the most suitable point counting method depends on factors such as the characteristics of the elliptic curve, the size of the underlying field, and the specific requirements of the cryptographic application

8.5 Elliptic curve primality proving (ECP)

Elliptic curve primality proving (ECP) is a powerful method for determining if a number is prime. It uses the mathematical properties of elliptic curves over finite fields to efficiently prove primality with a lower error probability than other tests.

ECP relies on the group structure of elliptic curves and operations like point addition and multiplication. The Atkin-Morain variant, which uses Suyama curves, is a popular implementation that improves efficiency and simplifies computations in the primality test.

Elliptic curve primality proving (ECP) overview

- ECP is a probabilistic primality testing algorithm that uses elliptic curves to determine whether a given integer is prime
- Relies on the mathematical properties of elliptic curves over finite fields to efficiently prove primality
- ECP has a lower error probability compared to other primality tests, making it suitable for applications requiring high confidence in primality

Mathematical foundations of ECPP

Elliptic curves over finite fields

- Elliptic curves are defined by the Weierstrass equation $y^2 = x^3 + ax + b$ over a finite field $\mathbb{F}_p$, where p is a prime number
- The set of points (x, y) satisfying the elliptic curve equation, along with a special point called the "point at infinity," form an abelian group
- The number of points on an elliptic curve over a finite field is called the order of the curve and is denoted by $\#E(\mathbb{F}_p)$

Group law on elliptic curves

- The group law defines the addition operation on elliptic curve points, allowing for the computation of point addition and point doubling
- Point addition: Given two distinct points P and Q on an elliptic curve, the sum $P + Q$ is defined as the point R obtained by drawing a line through P and Q and finding the third point of intersection with the curve, then reflecting that point across the X -axis
- Point doubling: When adding a point to itself ($P + P$), the tangent line to the curve at P is used to find the third point of intersection, which is then reflected across the X -axis

Elliptic curve point multiplication

- Point multiplication is the operation of adding a point P to itself k times, denoted as kP
- Efficient point multiplication algorithms, such as the double-and-add method or the window method, are used to compute kP in logarithmic time
- Point multiplication is a fundamental operation in elliptic curve cryptography and is used extensively in the ECPP algorithm

ECPP algorithm

Atkin-Morain ECPP variant

- The Atkin-Morain variant of ECPP is a widely used implementation that improves the efficiency of the original ECPP algorithm
- It uses a special class of elliptic curves called "Suyama curves" to simplify the computations involved in the primality test
- The Atkin-Morain ECPP consists of two main stages: the construction of a probable prime chain and the verification of primality using elliptic curve operations

Suyama curves in ECPP

- Suyama curves are elliptic curves of the form $y^2 = x^3 + ax$, where $a = (v^2 - 5)/4$ and v is a small integer
- These curves have desirable properties that simplify the computations in the ECPP algorithm, such as a known point of order 3 and a simple formula for the curve order

- Using Suyama curves reduces the complexity of the ECPP algorithm and improves its performance

Complexity analysis of ECPP

- The complexity of the ECPP algorithm is determined by the size of the input integer n and the desired error probability
- The Atkin-Morain ECPP variant has an expected running time of $O((\log n)^5)$ for proving the primality of an integer n
- The space complexity of ECPP is $O((\log n)^2)$, as it requires storing intermediate values and elliptic curve points during the computation

ECPP implementation considerations

Selecting suitable elliptic curves

- The choice of elliptic curves used in ECPP can significantly impact the performance and security of the algorithm
- Suyama curves are commonly used due to their desirable properties and simplicity
- Other elliptic curve models, such as Montgomery curves or Edwards curves, may also be used in ECPP implementations

Efficient point multiplication algorithms

- Implementing efficient point multiplication algorithms is crucial for the performance of ECPP
- The double-and-add method is a simple and widely used point multiplication algorithm, but more advanced techniques like the window method or the Montgomery ladder can provide better performance
- Parallelization and hardware acceleration can further optimize point multiplication in ECPP implementations

Handling large integer arithmetic

- ECPP involves computations with large integers, often hundreds or thousands of bits in size
- Efficient large integer arithmetic libraries, such as GMP (GNU Multiple Precision Arithmetic Library) or MPIR (Multiple Precision Integers and Rationals), are essential for implementing ECPP
- Optimizing large integer operations, such as modular arithmetic and integer division, can significantly improve the performance of ECPP

ECPP vs other primality tests

ECPP vs Fermat primality test

- The Fermat primality test is a simple and fast probabilistic primality test based on Fermat's Little Theorem
- However, the Fermat test has a higher error probability compared to ECPP, especially for Carmichael numbers (composite numbers that pass the Fermat test for all bases coprime to the number)

- ECPP provides a more rigorous and reliable primality test than the Fermat test, making it suitable for applications requiring high confidence in primality

ECPP vs Miller-Rabin primality test

- The Miller-Rabin primality test is another widely used probabilistic primality test based on the properties of strong pseudoprimes
- While the Miller-Rabin test is generally faster than ECPP, it has a higher error probability, especially for large integers
- ECPP offers a lower error probability than the Miller-Rabin test, making it a more reliable choice for primality testing in certain scenarios

Advantages and limitations of ECPP

- Advantages of ECPP include its low error probability, making it suitable for applications requiring high confidence in primality, and its deterministic nature when combined with a suitable bound on the input size
- Limitations of ECPP include its relatively higher computational complexity compared to other primality tests and the need for efficient elliptic curve arithmetic implementations
- ECPP may not be the most efficient choice for all primality testing scenarios, and the choice of primality test depends on the specific requirements of the application

Applications of ECPP

Prime number generation for cryptography

- ECPP can be used to generate large prime numbers for use in cryptographic protocols, such as RSA or elliptic curve cryptography
- Generating prime numbers with high confidence is essential for ensuring the security of cryptographic systems
- ECPP's low error probability makes it a suitable choice for generating prime numbers in cryptographic applications

Verifying primality in mathematical proofs

- Primality testing is often required in mathematical proofs, particularly in number theory and cryptography
- ECPP can be used to verify the primality of numbers in mathematical proofs, providing a rigorous and reliable primality test
- The deterministic nature of ECPP (when combined with a suitable bound on the input size) makes it attractive for use in mathematical proofs

Current research on ECPP

Improvements to ECPP algorithms

- Researchers continue to investigate ways to improve the efficiency and performance of ECPP algorithms

- Recent advances include the use of alternative elliptic curve models, such as Montgomery curves or Edwards curves, which can simplify computations and improve performance
- Parallelization and hardware acceleration techniques are also being explored to further optimize ECPP implementations

Alternative elliptic curve models for ECPP

- Alternative elliptic curve models, such as Montgomery curves or Edwards curves, have been proposed for use in ECPP
- These curve models can simplify computations and improve the efficiency of point multiplication, which is a critical operation in ECPP
- Research is ongoing to evaluate the suitability and performance of these alternative curve models in ECPP implementations

Combining ECPP with other primality tests

- Researchers are investigating ways to combine ECPP with other primality tests to create hybrid primality testing algorithms
- Combining ECPP with faster probabilistic tests, such as the Miller-Rabin test, can potentially improve the overall efficiency of primality testing while maintaining a low error probability
- Hybrid approaches may offer a balance between the reliability of ECPP and the speed of other primality tests, making them attractive for various applications

8.6 Elliptic curve method for integer factorization

The elliptic curve method (ECM) is a powerful integer factorization algorithm that uses elliptic curves to find factors of large composite numbers. It's particularly effective for finding small factors, making it a great preprocessing step before applying other factoring methods like the quadratic sieve or number field sieve.

ECM exploits the group structure of elliptic curves over finite fields to detect factors of the input number. Introduced by Hendrik Lenstra in 1987, the algorithm works by randomly selecting elliptic curves and points, then computing scalar multiples to find factors. It's especially useful for factors up to 30-40 digits long.

Elliptic curve method overview

- The elliptic curve method (ECM) is a fast integer factorization algorithm that uses elliptic curves to find factors of large composite numbers
- ECM is particularly effective for finding small factors and is often used as a preprocessing step before applying other factoring methods like the quadratic sieve or number field sieve
- The key idea behind ECM is to exploit the group structure of elliptic curves over finite fields to detect factors of the input number

Lenstra's elliptic curve factorization algorithm

- Hendrik Lenstra introduced the elliptic curve factorization algorithm in 1987 as a generalization of Pollard's $p - 1$ method

- Lenstra's algorithm works by randomly selecting elliptic curves and points on those curves, then computing scalar multiples of the points in an attempt to find a factor of the input number
- If a non-trivial factor is found during the scalar multiplication process, the algorithm terminates and returns the factor
- If no factor is found after a certain number of iterations, a new elliptic curve and point are chosen and the process repeats

Comparison vs other factoring methods

- ECM is particularly effective for finding small factors (up to around 30-40 digits) of large composite numbers
- For larger factors, other methods like the quadratic sieve and number field sieve are generally more efficient
- ECM has a heuristic runtime of $O(e^{\sqrt{2 \log p \log \log p}})$ for finding a prime factor p , which is subexponential in the size of p
- In practice, ECM is often used in conjunction with other factoring methods, serving as a preprocessing step to remove small factors before applying more powerful algorithms

Elliptic curves over finite fields

- To apply elliptic curves to integer factorization, we work with elliptic curves defined over finite fields, specifically the ring of integers modulo n (where n is the number we want to factor)
- An elliptic curve over a finite field is a set of points (x, y) satisfying an equation of the form $y^2 = x^3 + ax + b$, along with a special point called the "point at infinity"
- The coefficients a and b are chosen such that the discriminant $\Delta = 4a^3 + 27b^2$ is nonzero modulo n to ensure the curve is smooth

Elliptic curves modulo n

- When working with elliptic curves modulo n , the points on the curve are represented by pairs (x, y) where x and y are integers in the range $[0, n - 1]$
- The point at infinity serves as the identity element for the group law on the elliptic curve
- If n is composite, the elliptic curve modulo n may have a different structure than the curves over the prime factors of n , which is exploited by the ECM algorithm

Group law for elliptic curves mod n

- The points on an elliptic curve form an abelian group under a well-defined group law
- The group law for elliptic curves is defined geometrically:
 - To add two points P and Q , draw a line through P and Q and find the third point of intersection with the curve, then reflect that point across the x -axis to obtain $P + Q$
 - If $P = Q$, the line is taken to be the tangent line at P
- The point at infinity serves as the identity element, so $P + O = P$ for any point P on the curve

- The group law can be expressed algebraically using explicit formulas involving the coordinates of the points

Calculating multiples of points

- A key operation in ECM is computing scalar multiples of points on the elliptic curve, i.e., calculating kP for a positive integer k and a point P
- Scalar multiplication can be efficiently computed using the double-and-add algorithm, which is analogous to the square-and-multiply algorithm for modular exponentiation
- The double-and-add algorithm works by expressing k in binary and iteratively doubling and adding points based on the binary representation of k
- Optimizations like the sliding window method and signed-digit representations can further speed up scalar multiplication

Factorization process

- The ECM algorithm attempts to find a factor of a composite number n by randomly selecting elliptic curves and points, then computing scalar multiples of the points in the hope of encountering a non-trivial factor

Choosing random elliptic curves and points

- To start the factorization process, a random elliptic curve E and a random point P on that curve are selected
- The curve coefficients a and b are typically chosen uniformly at random from the range $[0, n - 1]$, subject to the constraint that the discriminant is nonzero modulo n
- The point P is selected by choosing a random X -coordinate in the range $[0, n - 1]$ and solving for the corresponding Y -coordinate using the curve equation

Computing scalar multiples

- Once a curve E and a point P have been selected, the algorithm computes the scalar multiple kP for a suitably chosen integer k
- The choice of k is critical to the success of the algorithm; it should be smooth (i.e., have only small prime factors) and not too large
- Common choices for k include prime powers, factorials, and products of small primes up to a certain bound
- The scalar multiple kP is computed using the double-and-add algorithm or one of its optimized variants

Detecting non-trivial factors

- As the scalar multiple kP is being computed, the algorithm checks for the occurrence of a non-trivial factor of n
- This is done by computing the greatest common divisor (GCD) of the difference of the X -coordinates of certain intermediate points encountered during the scalar multiplication process

- If the GCD is not 1 or n , then a non-trivial factor has been found, and the algorithm terminates successfully

Handling trivial factors and failure cases

- If the GCD computed during the scalar multiplication process is 1, then no factor has been found, and the algorithm proceeds to the next iteration with a new curve and point
- If the GCD is equal to n , then the chosen curve was singular modulo n , and a new curve must be selected
- After a certain number of iterations (determined by a heuristic based on the size of the smallest prime factor), if no factor has been found, the algorithm terminates unsuccessfully

Optimizing curve and point selection

- The success of ECM depends heavily on the choice of elliptic curves and points
- Various strategies have been proposed to optimize the selection process, such as:
 - Using curves with small coefficients to reduce the cost of curve arithmetic
 - Selecting curves with a large number of points modulo small primes to increase the chances of finding a factor
 - Using a "second phase" with a different choice of k to find factors that may have been missed in the first phase
- Implementing these optimizations can significantly improve the performance of ECM in practice

ECM runtime analysis

- The runtime of ECM depends on various factors, including the size of the input number, the size of the smallest prime factor, and the choice of parameters used in the algorithm

Heuristic runtime of ECM

- The heuristic runtime of ECM for finding a prime factor p of a composite number n is $O(e^{\sqrt{2 \log p \log \log p}})$
- This runtime is subexponential in the size of p , making ECM particularly effective for finding small prime factors
- The heuristic assumes that the elliptic curve group orders modulo the prime factors of n behave like random integers, which is supported by empirical evidence

Comparison vs quadratic sieve and NFS

- For finding small prime factors (up to around 30-40 digits), ECM is generally faster than other factoring methods like the quadratic sieve and number field sieve
- However, for larger prime factors, the quadratic sieve and number field sieve become more efficient due to their subexponential runtime in the size of n
- In practice, ECM is often used as a preprocessing step to remove small prime factors before applying more powerful factoring algorithms

Asymptotic complexity of ECM

- The asymptotic complexity of ECM is subexponential in the size of the smallest prime factor P , but exponential in the size of the input number n
- This means that ECM is not a polynomial-time algorithm for integer factorization, and its effectiveness diminishes as the size of the prime factors increases
- Despite its exponential worst-case complexity, ECM remains a valuable tool in practice due to its efficiency in finding small prime factors

Optimal ECM parameters for various sizes

- The choice of parameters, such as the bound on the prime factors of k and the number of iterations, can significantly impact the performance of ECM
- Optimal parameter choices depend on the size of the input number and the desired trade-off between runtime and success probability
- Heuristics and empirical data are used to determine suitable parameter values for different input sizes
- In general, larger input numbers require larger bounds on the prime factors of k and more iterations to achieve a high success probability

ECM implementation considerations

- Implementing ECM efficiently requires careful consideration of various aspects, such as the representation of elliptic curve points, the choice of coordinate system, and the optimization of curve arithmetic

Representing points and curves

- The choice of coordinate system can significantly impact the performance of ECM implementations
- Common coordinate systems include:
 - Affine coordinates: Points are represented as (X, Y) pairs satisfying the curve equation
 - Projective coordinates: Points are represented as $(X : Y : Z)$ triples, allowing for more efficient curve arithmetic
 - Montgomery coordinates: A special form of projective coordinates that simplifies the scalar multiplication process
- The choice of coordinate system depends on the specific requirements of the implementation, such as memory constraints and the relative costs of different field operations

Optimizing curve arithmetic

- Efficient implementation of elliptic curve arithmetic is crucial for the performance of ECM
- Various techniques can be used to optimize curve operations, such as:
 - Using fast reduction algorithms for modular arithmetic, like Montgomery reduction or Barrett reduction
 - Employing specialized field arithmetic algorithms for specific prime moduli, such as primes of the form $2^k \pm c$

- Exploiting the structure of the curve equation to simplify point addition and doubling formulas
- Implementing these optimizations can significantly reduce the cost of curve arithmetic and improve the overall performance of ECM

Techniques for speeding up scalar multiplication

- Scalar multiplication is the most computationally expensive part of ECM, so optimizing this operation is critical for efficient implementations
- Various techniques can be used to speed up scalar multiplication, such as:
 - Using window-based methods, like the sliding window or fixed window algorithms, to reduce the number of point additions
 - Employing signed-digit representations, such as the non-adjacent form (NAF), to minimize the weight of the scalar
 - Precomputing frequently used scalar multiples to avoid redundant computations
 - Combining these techniques can lead to significant speedups in the scalar multiplication process

Parallelization of ECM

- ECM is well-suited for parallelization, as the curve and point selection process can be performed independently across multiple threads or processors
- Parallelization strategies for ECM include:
 - Distributing the selection of curves and points across multiple threads, with each thread performing its own scalar multiplications
 - Using a master-slave model, where a central master process distributes curves and points to slave processes and collects the results
 - Employing a sieving approach, where multiple threads cooperate to find suitable curves and points based on certain criteria
 - Effective parallelization can significantly reduce the runtime of ECM, especially for large input numbers

ECM in practice vs theory

- While the theoretical analysis of ECM provides valuable insights into its asymptotic behavior, the practical performance of the algorithm can vary significantly depending on the implementation and the specific input numbers
- In practice, ECM implementations often incorporate various optimizations and heuristics that are not captured by the theoretical analysis, such as:
 - Using a second phase with a different choice of k to find factors missed in the first phase
 - Employing early abort strategies to detect singular curves or unlikely success cases
 - Adapting the parameter choices based on the size and structure of the input number
 - Practical implementations of ECM also need to account for factors like memory constraints, cache effects, and the relative costs of different field operations, which can impact the optimal choice of

parameters and implementation strategies

- Despite these differences between theory and practice, ECM remains a powerful and widely-used factoring algorithm, particularly for finding small prime factors of large composite numbers

Unit 9 – Elliptic Curves in Coding Theory

9.1 Goppa codes and algebraic-geometric codes

Goppa codes and algebraic-geometric codes are powerful error-correcting codes with applications in cryptography. These codes use finite fields and algebraic structures to achieve strong error-correction capabilities. They're especially important in post-quantum cryptography due to their resistance to attacks by quantum computers.

Understanding Goppa and AG codes requires knowledge of finite fields, algebraic geometry, and coding theory. These codes offer advantages in terms of security and error-correction, but come with challenges in implementation and efficiency. Their study is crucial for developing secure cryptographic systems in the quantum era.

Goppa codes

- Goppa codes are a family of linear error-correcting codes named after Valerii Denisovich Goppa
- Goppa codes have found applications in various areas, including cryptography and post-quantum cryptography due to their strong security properties
- Understanding Goppa codes is essential for the study of algebraic-geometric codes and their applications in elliptic curve cryptography

Definition of Goppa codes

- Goppa codes are defined using a generating polynomial $g(x)$ over a finite field $\mathbb{F}_q$ and a set of distinct elements $L = \{\alpha_1, \alpha_2, \dots, \alpha_n\} \subset \mathbb{F}_q$
- The generating polynomial $g(x)$ must be monic, irreducible, and have degree t
- The Goppa code $\Gamma(L, g)$ consists of all vectors $c = (c_1, c_2, \dots, c_n)$ such that

$$\sum_{i=1}^n \frac{c_i}{x - \alpha_i} \equiv 0 \pmod{g(x)}$$

Construction of Goppa codes

- To construct a Goppa code, first choose a finite field $\mathbb{F}_q$, a set of distinct elements $L \subset \mathbb{F}_q$, and a generating polynomial $g(x)$ of degree t
- Compute the parity-check matrix H using the generating polynomial and the set L
- The Goppa code is the null space of the parity-check matrix H
- The generator matrix G can be obtained from the parity-check matrix H using linear algebra techniques

Parameters of Goppa codes

- The length of a Goppa code is n , which is the number of elements in the set L
- The dimension of a Goppa code is $k \geq n - mt$, where $m = \log_q(n)$
- The minimum distance of a Goppa code is at least $2t + 1$, providing strong error-correcting capabilities

- The rate of a Goppa code is $\frac{k}{n}$, which represents the efficiency of the code

Decoding of Goppa codes

- Decoding Goppa codes involves finding the error locations and error values in a received word
- The Patterson algorithm is a widely used decoding algorithm for Goppa codes
- It involves computing syndromes, finding an error locator polynomial, and solving for error locations and values
- Other decoding algorithms for Goppa codes include the Berlekamp-Massey algorithm and the Euclidean algorithm

Binary Goppa codes

- Binary Goppa codes are a special case of Goppa codes where the finite field is $\mathbb{F}_2$
- Binary Goppa codes have additional properties and are often used in cryptographic applications
- The parity-check matrix of a binary Goppa code has a specific structure, which can be exploited for efficient encoding and decoding

Distinguishers for Goppa codes

- Distinguishers are algorithms that can distinguish between a Goppa code and a random linear code
- The existence of efficient distinguishers can have implications for the security of cryptographic schemes based on Goppa codes
- Some distinguishers for Goppa codes include the support splitting algorithm and the Schur product distinguisher
- Developing secure Goppa codes resistant to known distinguishers is an active area of research

Algebraic-geometric codes

- Algebraic-geometric (AG) codes are a generalization of Goppa codes that use algebraic curves over finite fields
- AG codes have the potential to achieve better parameters (length, dimension, minimum distance) compared to Goppa codes
- Understanding AG codes requires knowledge of algebraic geometry concepts such as curves, rational points, and divisors

Motivation for algebraic-geometric codes

- AG codes were introduced to overcome the limitations of Goppa codes in terms of code parameters
- By using algebraic curves, AG codes can achieve longer code lengths and better error-correction capabilities
- AG codes have the potential to provide improved performance and security in various applications, including cryptography

Goppa's construction from curves

- Goppa's construction of AG codes uses an algebraic curve $\mathcal{X}$ over a finite field $\mathbb{F}_q$ and a set of rational points $P_1, P_2, \dots, P_n$ on the curve
- A divisor D is chosen such that it has support disjoint from the rational points
- The AG code is defined as the image of the Riemann-Roch space $\mathcal{L}(D)$ under the evaluation map at the rational points

Rational points and divisors

- Rational points on an algebraic curve are points whose coordinates are in the base field $\mathbb{F}_q$
- Divisors are formal sums of points on the curve, representing the zeros and poles of functions
- The Picard group of a curve is the group of divisors modulo principal divisors
- Understanding rational points and divisors is crucial for constructing AG codes and analyzing their properties

Riemann-Roch theorem

- The Riemann-Roch theorem is a fundamental result in algebraic geometry that relates the dimension of the Riemann-Roch space $\mathcal{L}(D)$ to the degree of the divisor D and the genus of the curve
- The theorem states that $\dim \mathcal{L}(D) = \deg(D) - g + 1 + \dim \mathcal{L}(K - D)$, where K is the canonical divisor and g is the genus of the curve
- The Riemann-Roch theorem is used to determine the parameters of AG codes and to prove the existence of codes with desired properties

Construction of algebraic-geometric codes

- To construct an AG code, first choose an algebraic curve $\mathcal{X}$ over a finite field $\mathbb{F}_q$ and a set of rational points $P_1, P_2, \dots, P_n$ on the curve
- Choose a divisor D with support disjoint from the rational points and compute the Riemann-Roch space $\mathcal{L}(D)$
- The AG code is the image of $\mathcal{L}(D)$ under the evaluation map at the rational points
- The generator matrix of the AG code can be obtained by evaluating a basis of $\mathcal{L}(D)$ at the rational points

Parameters of algebraic-geometric codes

- The length of an AG code is n , which is the number of rational points used in the construction
- The dimension of an AG code is $k = \dim \mathcal{L}(D) = \deg(D) - g + 1$, where g is the genus of the curve
- The minimum distance of an AG code is at least $n - \deg(D)$, providing good error-correction capabilities
- The Singleton bound for AG codes is $d \leq n - k + 1$, where d is the minimum distance

Hermitian codes

- Hermitian codes are a special class of AG codes constructed using Hermitian curves over finite fields
- Hermitian curves have the form $y^q + y = x^{q+1}$ over $\mathbb{F}_{q^2}$ and have many rational points
- Hermitian codes have good parameters and efficient decoding algorithms, making them attractive for practical applications
- The Hermitian function field has been extensively studied in the context of AG codes and their properties

Decoding of algebraic-geometric codes

- Decoding AG codes involves finding the error locations and error values in a received word
- The Berlekamp-Massey-Sakata (BMS) algorithm is a widely used decoding algorithm for AG codes
- It is a generalization of the Berlekamp-Massey algorithm used for decoding Goppa codes
- Other decoding algorithms for AG codes include the Guruswami-Sudan algorithm and the Kötter algorithm
- Decoding AG codes can be more complex compared to decoding Goppa codes due to the additional algebraic geometry concepts involved

Comparison vs Goppa codes

- AG codes can achieve better parameters (length, dimension, minimum distance) compared to Goppa codes
- AG codes require more advanced algebraic geometry tools and concepts for their construction and analysis
- Goppa codes have more efficient decoding algorithms compared to AG codes, which can be more complex to decode
- AG codes have the potential for improved performance and security, but their practical implementation can be more challenging

Applications of Goppa and AG codes

- Goppa codes and AG codes have found various applications, particularly in the field of cryptography
- The strong error-correction capabilities and security properties of these codes make them suitable for constructing secure cryptographic schemes
- Understanding the applications of Goppa and AG codes is important for appreciating their significance in modern cryptography

McEliece cryptosystem

- The McEliece cryptosystem is a public-key encryption scheme based on the hardness of decoding random linear codes
- It uses Goppa codes as the underlying error-correcting codes due to their strong security properties

- The private key consists of the Goppa code parameters, while the public key is a disguised generator matrix
- Encryption involves encoding a message with errors, while decryption requires decoding the Goppa code to recover the original message

Code-based cryptography

- Code-based cryptography refers to cryptographic schemes that rely on the hardness of decoding problems in coding theory
- Goppa codes and AG codes are among the most commonly used codes in code-based cryptography
- Code-based cryptosystems offer advantages such as fast encryption and decryption, and resistance to quantum attacks
- Some examples of code-based cryptosystems include the McEliece cryptosystem, the Niederreiter cryptosystem, and the HyMES scheme

Post-quantum cryptography

- Post-quantum cryptography aims to develop cryptographic algorithms that are secure against attacks by quantum computers
- Goppa codes and AG codes are considered promising candidates for post-quantum cryptography
- The security of code-based cryptosystems relies on the difficulty of decoding random linear codes, which is believed to be hard even for quantum computers
- Code-based cryptography is one of the main areas of research in post-quantum cryptography, along with lattice-based cryptography and multivariate cryptography

Advantages vs other code families

- Goppa codes and AG codes have several advantages compared to other code families used in cryptography
- They have strong error-correction capabilities, which allows for efficient handling of errors in cryptographic schemes
- Goppa codes and AG codes have a rich mathematical structure, which enables the construction of secure cryptographic primitives
- The security of code-based cryptosystems relies on well-studied problems in coding theory, providing a solid foundation for their security analysis

Challenges in implementation

- Implementing Goppa codes and AG codes in practical cryptographic schemes comes with several challenges
- The key sizes of code-based cryptosystems can be large compared to traditional cryptosystems like RSA, which can impact efficiency
- Efficient encoding and decoding algorithms are required to ensure fast encryption and decryption times

- Side-channel attacks and other implementation-specific vulnerabilities need to be carefully considered and mitigated
- Balancing security, efficiency, and key size is an ongoing challenge in the implementation of code-based cryptography

9.2 Elliptic curves and linear codes

Elliptic curves over finite fields are powerful tools in cryptography and coding theory. They provide a discrete algebraic structure that enables secure encryption and efficient error correction. Understanding their properties and arithmetic is crucial for implementing robust cryptographic systems and error-correcting codes.

Linear error-correcting codes add redundancy to data, allowing detection and correction of transmission errors. These codes, including Hamming and Goppa codes, use mathematical structures to achieve optimal error-correcting capabilities. Their applications span telecommunications, data storage, and cryptography.

Elliptic curves over finite fields

- Elliptic curves over finite fields play a fundamental role in cryptography and coding theory
- Finite fields provide a discrete and bounded algebraic structure for defining elliptic curves
- Understanding the properties and arithmetic of elliptic curves over finite fields is crucial for their applications

Definitions and examples

- An elliptic curve over a finite field $\mathbb{F}_q$ is defined by an equation of the form $y^2 = x^3 + ax + b$ where $a, b \in \mathbb{F}_q$ and the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$
- The set of points (x, y) satisfying the equation along with a special point at infinity denoted $\mathcal{O}$ form the elliptic curve
- Example: The curve $y^2 = x^3 + x + 1$ over the field $\mathbb{F}_5 = \{0, 1, 2, 3, 4\}$ consists of the points $\{(0, 1), (0, 4), (2, 1), (2, 4), (3, 1), (3, 4), \mathcal{O}\}$

Group law and point addition

- The set of points on an elliptic curve over a finite field forms an abelian group under a well-defined addition operation
- The group law is based on the chord-and-tangent method for adding points geometrically
- Given two points P and Q on the curve, their sum $P + Q$ is defined as the reflection across the x-axis of the third point of intersection of the line through P and Q with the curve
- The point at infinity $\mathcal{O}$ serves as the identity element of the group
- Example: On the curve $y^2 = x^3 + x + 1$ over $\mathbb{F}_5$, the sum of points $(2, 1)$ and $(3, 4)$ is $(0, 4)$

Finite field arithmetic

- Elliptic curve operations involve arithmetic in the underlying finite field $\mathbb{F}_q$
- Finite field elements are represented as integers modulo the field size q

- Addition, subtraction, multiplication, and division in the finite field are performed modulo q
- Efficient algorithms exist for finite field arithmetic, such as Montgomery multiplication and fast modular reduction techniques
- Example: In $\mathbb{F}_5$, the sum of 2 and 3 is 0 (since $2 + 3 \equiv 0 \pmod{5}$), and the product of 2 and 3 is 1 (since $2 \times 3 \equiv 1 \pmod{5}$)

Singular vs non-singular curves

- A non-singular elliptic curve over a finite field has a non-zero discriminant and forms a smooth curve without any cusps or self-intersections
- Singular curves have a vanishing discriminant and possess singularities or points of self-intersection
- Cryptographic applications typically use non-singular curves to ensure desirable group properties and security
- Example: The curve $y^2 = x^3$ over $\mathbb{F}_5$ is singular at the point $(0, 0)$, while the curve $y^2 = x^3 + x + 1$ is non-singular

Linear error-correcting codes

- Linear error-correcting codes are mathematical constructions used for detecting and correcting errors in data transmission or storage
- They provide a systematic way to add redundancy to information symbols to enable error correction
- Linear codes are widely used in various applications, including telecommunications, data storage, and cryptography

Generator and parity check matrices

- A linear code C of length n and dimension k over a finite field $\mathbb{F}_q$ is a k -dimensional subspace of the vector space $\mathbb{F}_q^n$
- The generator matrix G is a $k \times n$ matrix whose rows form a basis for the code C
- The parity check matrix H is an $(n - k) \times n$ matrix such that $GH^T = 0$, where H^T denotes the transpose of H
- Codewords in C are obtained by multiplying a message vector $m \in \mathbb{F}_q^k$ with the generator matrix G , i.e., $c = mG$
- Example: The generator matrix

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix}$$

over $\mathbb{F}_3$ generates a $(4, 2)$ linear code

Hamming codes and perfect codes

- Hamming codes are a family of linear codes with parameters $[2^r - 1, 2^r - r - 1, 3]$ for any positive integer r
- They are single-error-correcting codes, capable of correcting any single-bit error in a codeword

- Perfect codes are codes that achieve the Hamming bound, meaning they have the maximum possible minimum distance for a given code length and dimension
- Hamming codes are perfect codes for the case of single-error correction
- Example: The $[7, 4, 3]$ Hamming code can correct any single-bit error in a 7-bit codeword

Bounds on minimum distance

- The minimum distance d of a linear code determines its error-correcting capability
- The Singleton bound states that $d \leq n - k + 1$ for an $[n, k]$ linear code
- The Hamming bound provides an upper bound on the size of a code with a given minimum distance
- Codes meeting the Singleton bound are called maximum distance separable (MDS) codes
- Example: The $[7, 4, 3]$ Hamming code meets the Singleton bound and is an MDS code

Decoding algorithms for linear codes

- Decoding algorithms aim to correct errors in received codewords and recover the original message
- Syndrome decoding is a common technique that computes the syndrome of a received word and uses it to determine the error pattern
- Maximum likelihood decoding finds the codeword closest to the received word in terms of Hamming distance
- Algebraic decoding methods, such as the Berlekamp-Massey algorithm, use the algebraic structure of the code for efficient decoding
- Example: Syndrome decoding of the $[7, 4, 3]$ Hamming code can correct any single-bit error by comparing the syndrome with a pre-computed syndrome table

Goppa codes from algebraic curves

- Goppa codes are a class of linear error-correcting codes constructed from algebraic curves over finite fields
- They provide a geometric approach to coding theory and offer advantages over classical linear codes in terms of code parameters and decoding efficiency
- Goppa codes have found applications in cryptography, such as the McEliece cryptosystem, due to their resistance to certain cryptanalytic attacks

Goppa's construction from curves

- Goppa codes are constructed from a smooth, projective, absolutely irreducible algebraic curve $\mathcal{X}$ over a finite field $\mathbb{F}_q$
- Given a set of n distinct $\mathbb{F}_q$ -rational points $P_1, \dots, P_n$ on $\mathcal{X}$ and a divisor D on $\mathcal{X}$ with support disjoint from the P_i , the Goppa code $\mathcal{C}(P_1, \dots, P_n, D)$ is defined as the image of the evaluation map from the Riemann-Roch space $\mathcal{L}(D)$ to $\mathbb{F}_q^n$
- The generator matrix of the Goppa code is constructed from the basis elements of $\mathcal{L}(D)$ evaluated at the points P_i

- Example: The Hermitian curve $y^q + y = x^{q+1}$ over $\mathbb{F}_{q^2}$ can be used to construct Goppa codes with good parameters

Parameters of Goppa codes

- The length n of a Goppa code is determined by the number of $\mathbb{F}_q$ -rational points used in the construction
- The dimension k of the code is related to the dimension of the Riemann-Roch space $\mathcal{L}(D)$ and can be computed using the Riemann-Roch theorem
- The minimum distance d of the code is lower bounded by the degree of the divisor D used in the construction
- Example: A Goppa code constructed from the Hermitian curve over $\mathbb{F}_{16}$ with $n = 64$ and $\deg(D) = 24$ has parameters $[64, 18, \geq 25]$

Decoding Goppa codes

- Decoding Goppa codes involves finding the closest codeword to a received word in terms of Hamming distance
- Efficient decoding algorithms for Goppa codes exploit the algebraic structure of the code and the underlying curve
- The Patterson algorithm is a widely used decoding method for Goppa codes, which reduces the decoding problem to solving a key equation and finding roots of an error locator polynomial
- Example: The Patterson algorithm can decode up to $\lfloor (\deg(D) - 1)/2 \rfloor$ errors in a Goppa code constructed from a divisor D

Advantages vs classical linear codes

- Goppa codes often achieve better parameters (length, dimension, minimum distance) compared to classical linear codes of the same length
- The geometric structure of Goppa codes provides additional algebraic properties that can be exploited for efficient decoding
- Goppa codes have a higher error-correcting capability for a given code rate compared to classical linear codes
- The security of certain cryptographic schemes, such as the McEliece cryptosystem, relies on the difficulty of decoding random Goppa codes, which is believed to be a hard problem

Elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptography approach based on the algebraic structure of elliptic curves over finite fields
- ECC provides high security with smaller key sizes compared to other public-key cryptosystems, such as RSA
- The security of ECC relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP)

Diffie-Hellman key exchange on curves

- The Diffie-Hellman key exchange protocol can be adapted to work on elliptic curves over finite fields
- Two parties, Alice and Bob, agree on a public elliptic curve E over a finite field $\mathbb{F}_q$ and a base point P on the curve
- Alice chooses a secret integer a and computes aP , while Bob chooses a secret integer b and computes bP
- Alice and Bob exchange their computed points aP and bP over a public channel
- Both parties can then compute the shared secret point abP using their respective secret integers
- The shared secret can be used to derive a symmetric key for secure communication
- Example: Using the curve $y^2 = x^3 + 2x + 2$ over $\mathbb{F}_{17}$ with base point $P = (5, 1)$, Alice chooses $a = 3$ and Bob chooses $b = 7$, resulting in the shared secret point $(10, 6)$

Elliptic curve digital signatures

- Elliptic curve digital signature algorithms (ECDSA) provide a way to create and verify digital signatures using elliptic curves
- The signer has a private key d and a corresponding public key $Q = dP$, where P is a base point on the curve
- To sign a message m , the signer chooses a random integer k , computes $kP = (x_1, y_1)$ and $r = x_1 \bmod n$, where n is the order of P
- The signer then computes $s = k^{-1}(H(m) + dr) \bmod n$, where H is a cryptographic hash function
- The signature is the pair (r, s)
- To verify a signature, the verifier computes $w = s^{-1} \bmod n$, $u_1 = H(m)w \bmod n$, $u_2 = rw \bmod n$, and $u_1P + u_2Q = (x_0, y_0)$
- The signature is valid if $r = x_0 \bmod n$
- Example: Using the curve $y^2 = x^3 + ax + b$ over $\mathbb{F}_p$ with base point P and a private key d , a message can be signed and verified using ECDSA

Security based on discrete log problem

- The security of elliptic curve cryptography relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)
- Given an elliptic curve E over a finite field $\mathbb{F}_q$, a base point P on the curve, and a point Q on the curve, the ECDLP is to find an integer k such that $Q = kP$
- The best known algorithms for solving the ECDLP have exponential time complexity, making it infeasible for properly chosen curve parameters
- The hardness of the ECDLP ensures the security of elliptic curve cryptographic schemes
- Example: On the curve $y^2 = x^3 + 2x + 2$ over $\mathbb{F}_{17}$, given points $P = (5, 1)$ and $Q = (3, 6)$, finding the integer k such that $Q = kP$ is an instance of the ECDLP

Comparisons vs RSA cryptosystem

- Elliptic curve cryptography offers several advantages compared to the RSA cryptosystem
- ECC achieves the same level of security as RSA with significantly smaller key sizes, reducing storage and transmission requirements
- ECC is more computationally efficient than RSA, especially for higher security levels
- The absence of efficient quantum algorithms for the ECDLP makes ECC a candidate for post-quantum cryptography
- Example: A 256-bit elliptic curve key provides similar security to a 3072-bit RSA key, demonstrating the key size advantage of ECC

Elliptic curve primality proving

- Elliptic curve primality proving (ECPP) is a probabilistic algorithm for determining the primality of a given integer using elliptic curves
- ECPP is based on the properties of elliptic curves over finite fields and provides an efficient and practical method for primality testing

Goldwasser-Kilian algorithm

- The Goldwasser-Kilian algorithm is a foundational ECPP algorithm that laid the groundwork for subsequent improvements
- It relies on the fact that the order of an elliptic curve over a finite field $\mathbb{F}_p$ is close to $p + 1$ for most curves
- The algorithm starts by randomly selecting an elliptic curve E over $\mathbb{F}_p$ and a point P on the curve
- It then computes the order m of P and checks if m divides $p + 1$ and if $(p + 1)/m$ is a prime power
- If the conditions are satisfied, the algorithm recursively proves the primality of $(p + 1)/m$ using the same process
- The recursion continues until a small enough prime is reached, at which point the primality of p is established
- Example: To prove the primality of $p = 31$, the Goldwasser-Kilian algorithm may select the curve $y^2 = x^3 + x + 1$ and the point $P = (0, 1)$, which has order $m = 5$, and then recursively prove the primality of $(31 + 1)/5 = 6$

Atkin-Morain improvements

- The Atkin-Morain ECPP algorithm improves upon the Goldwasser-Kilian algorithm by using complex multiplication (CM) methods to construct elliptic curves with known group orders
- Instead of randomly selecting curves, the Atkin-Morain algorithm uses curves with CM by imaginary quadratic fields with small discriminants
- The use of CM curves ensures that the group order of the curve can be efficiently computed and eliminates the need for recursive primality testing

- The Atkin-Morain algorithm also employs various optimizations, such as the use of isogenies and the computation of Hilbert class polynomials, to speed up the primality proving process

9.3 Elliptic curves and cyclic codes

Elliptic curves are a fascinating area of mathematics with applications in cryptography and coding theory. These smooth, projective algebraic curves of genus one have unique properties that make them ideal for creating secure cryptographic protocols and efficient error-correcting codes.

Cyclic codes derived from elliptic curves, like Goppa codes, offer powerful error correction capabilities. By evaluating functions on elliptic curve points, we can construct codes with desirable properties. These codes have found use in data protection, satellite communications, and even post-quantum cryptography.

Elliptic curves

- Elliptic curves are a type of algebraic curve that have been studied extensively in mathematics and have found numerous applications in cryptography and coding theory
- Elliptic curves are defined over a field and have a specific geometric shape that gives them unique properties
- The study of elliptic curves is a central topic in algebraic geometry and number theory, and their use in cryptography has led to the development of secure and efficient cryptographic protocols

Definition of elliptic curves

- An elliptic curve is a smooth, projective algebraic curve of genus one with a specified base point
- Elliptic curves can be defined over any field, including the real numbers, complex numbers, and finite fields
- The most common form of an elliptic curve is given by the Weierstrass equation: $y^2 = x^3 + ax + b$, where a and b are constants that satisfy certain conditions to ensure the curve is smooth
- Every elliptic curve has a distinguished point called the "point at infinity" which serves as the identity element for the group law

Weierstrass equation

- The Weierstrass equation is a canonical form for representing elliptic curves
- For an elliptic curve over a field K , the Weierstrass equation is given by $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, where $a_1, a_2, a_3, a_4, a_6 \in K$
- The coefficients of the Weierstrass equation must satisfy certain conditions to ensure the curve is smooth and has no singular points
- The Weierstrass equation can be simplified by a change of variables to eliminate some of the coefficients (e.g., the short Weierstrass form: $y^2 = x^3 + ax + b$)

Elliptic curve group law

- Elliptic curves have a natural group structure under a geometric operation called the "group law"
- The group law defines a way to "add" points on the elliptic curve, giving the set of points on the curve the structure of an abelian group

- The group law is defined geometrically using the chord-and-tangent method:
- To add two points P and Q , draw a line through P and Q (if $P = Q$, take the tangent line at P)
- This line intersects the curve at a third point R
- The sum $P + Q$ is defined as the reflection of R across the x-axis
- The group law satisfies the usual group axioms (associativity, identity element, inverses)

Point at infinity

- The point at infinity, denoted $\mathcal{O}$, is a distinguished point on an elliptic curve that serves as the identity element for the group law
- In the projective plane, the point at infinity is the point where all vertical lines intersect
- Adding any point P on the curve to the point at infinity results in P itself: $P + \mathcal{O} = P$
- The point at infinity is the neutral element for the group law, playing a role analogous to zero in ordinary addition

Elliptic curve arithmetic

- Elliptic curve arithmetic involves performing operations (addition, subtraction, multiplication by a scalar) on points of an elliptic curve using the group law
- Adding points on an elliptic curve is a fundamental operation in elliptic curve cryptography (used in key generation, encryption, and digital signatures)
- Scalar multiplication, i.e., adding a point P to itself k times (denoted kP), is another important operation in elliptic curve cryptography
- Efficient algorithms (e.g., double-and-add, sliding window method) have been developed to perform scalar multiplication quickly
- The security of many elliptic curve cryptographic schemes relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP): given points P and Q on an elliptic curve, find an integer k such that $Q = kP$

Cyclic codes from elliptic curves

- Cyclic codes are a class of linear error-correcting codes that have a cyclic structure (i.e., any cyclic shift of a codeword is also a codeword)
- Elliptic curves can be used to construct a special class of cyclic codes called Goppa codes, which have good error-correcting properties
- The construction of cyclic codes from elliptic curves involves evaluating functions on the points of the curve and using the resulting values to define the codewords
- The algebraic structure of elliptic curves provides a rich source of functions that can be used to construct cyclic codes with desirable properties

Goppa codes

- Goppa codes are a family of linear error-correcting codes that were introduced by Valerii Denisovich Goppa in 1970

- Goppa codes are constructed using a generator polynomial $g(x)$ and a set of distinct elements $\alpha_1, \dots, \alpha_n$ from a finite field $\mathbb{F}_q$
- The generator polynomial is chosen to be monic, square-free, and relatively prime to

$$\prod_{i=1}^n (x - \alpha_i)$$

- Goppa codes have good error-correcting capabilities and can be efficiently decoded using Patterson's algorithm or the Berlekamp-Massey algorithm

Construction of Goppa codes

- To construct a Goppa code from an elliptic curve E over a finite field $\mathbb{F}_q$, we follow these steps: 1. Choose a set of distinct $\mathbb{F}_q$ -rational points $P_1, \dots, P_n$ on the curve E 2. Choose a divisor

$$D = \sum_{i=1}^n m_i P_i$$

with support disjoint from the chosen points 3. The Goppa code is defined as the set of codewords $(f(P_1), \dots, f(P_n))$, where f runs over the functions in the Riemann-Roch space $\mathcal{L}(D)$

- The Riemann-Roch space $\mathcal{L}(D)$ consists of all rational functions on the curve whose divisor of poles is bounded by D
- The dimension and minimum distance of the resulting Goppa code can be estimated using the Riemann-Roch theorem

Generator matrix of Goppa codes

- The generator matrix of a Goppa code constructed from an elliptic curve can be obtained by evaluating a basis of the Riemann-Roch space $\mathcal{L}(D)$ at the chosen points $P_1, \dots, P_n$
- Let $f_1, \dots, f_k$ be a basis for $\mathcal{L}(D)$, then the generator matrix G is given by:

$$G = \begin{bmatrix} f_1(P_1) & f_1(P_2) & \cdots & f_1(P_n) \\ f_2(P_1) & f_2(P_2) & \cdots & f_2(P_n) \\ \vdots & \vdots & \ddots & \vdots \\ f_k(P_1) & f_k(P_2) & \cdots & f_k(P_n) \end{bmatrix}$$

- The generator matrix can be used to encode messages into codewords by taking linear combinations of the rows
- The structure of the generator matrix depends on the choice of the divisor D and the basis for $\mathcal{L}(D)$

Parity check matrix of Goppa codes

- The parity check matrix of a Goppa code is a matrix H such that a vector C is a codeword if and only if $HC^T = 0$
- For a Goppa code constructed from an elliptic curve, the parity check matrix can be obtained using the differential of the functions in the Riemann-Roch space $\mathcal{L}(D)$
- Let ω be a differential form on the curve with divisor D , then the parity check matrix H is given by:

$$H = \begin{bmatrix} \text{res}_{P_1}(f_1\omega) & \text{res}_{P_2}(f_1\omega) & \cdots & \text{res}_{P_n}(f_1\omega) \\ \text{res}_{P_1}(f_2\omega) & \text{res}_{P_2}(f_2\omega) & \cdots & \text{res}_{P_n}(f_2\omega) \\ \vdots & \vdots & \ddots & \vdots \\ \text{res}_{P_1}(f_k\omega) & \text{res}_{P_2}(f_k\omega) & \cdots & \text{res}_{P_n}(f_k\omega) \end{bmatrix}$$

where $\text{res}_{P_i}(f_j\omega)$ denotes the residue of the differential $f_j\omega$ at the point P_i

- The parity check matrix is used in syndrome-based decoding algorithms for Goppa codes

Decoding of Goppa codes

- Decoding a Goppa code involves finding the most likely codeword given a received word that may contain errors
- Two common decoding algorithms for Goppa codes are Patterson's algorithm and the Berlekamp-Massey algorithm
- Patterson's algorithm is based on the Euclidean algorithm for polynomials and can correct up to $\lfloor (d-1)/2 \rfloor$ errors, where d is the minimum distance of the code
- The Berlekamp-Massey algorithm is an efficient algorithm for finding the error locator polynomial, which can be used to locate and correct errors in the received word
- Both algorithms exploit the algebraic structure of Goppa codes and the properties of the underlying elliptic curve to achieve efficient decoding

Applications of elliptic curve cyclic codes

- Elliptic curve cyclic codes, such as Goppa codes, have found various applications in cryptography and error correction
- The algebraic structure and geometric properties of elliptic curves provide a rich source of mathematical tools for constructing secure and efficient cryptographic schemes and error-correcting codes
- The use of elliptic curves in these applications offers several advantages over traditional approaches, including shorter key sizes, increased security, and better error-correcting capabilities

Cryptography using elliptic curves

- Elliptic curve cryptography (ECC) is a public-key cryptography approach that uses the algebraic structure of elliptic curves over finite fields
- ECC schemes, such as the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Elliptic Curve Integrated Encryption Scheme (ECIES), provide secure key exchange, digital signatures, and encryption
- The security of ECC relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP), which is believed to be harder than the discrete logarithm problem used in traditional cryptosystems (e.g., DSA)
- ECC offers equivalent security with shorter key sizes compared to RSA and DSA, making it suitable for resource-constrained environments (e.g., smartphones, IoT devices)

Error correction with Goppa codes

- Goppa codes constructed from elliptic curves have good error-correcting capabilities and can be used to protect data transmitted over noisy channels
- The minimum distance of a Goppa code can be estimated using the Riemann-Roch theorem, which allows for the construction of codes with a desired error-correcting capability
- Goppa codes have been used in various applications, such as:
 - Storage systems to protect against data corruption
 - Satellite communications to correct errors caused by atmospheric noise
 - Post-quantum cryptography to provide security against quantum computer attacks

Advantages vs traditional cyclic codes

- Elliptic curve cyclic codes, such as Goppa codes, offer several advantages over traditional cyclic codes (e.g., BCH codes, Reed-Solomon codes):
- Goppa codes can be constructed with a wide range of parameters (length, dimension, minimum distance) by choosing different elliptic curves and divisors
- The algebraic structure of elliptic curves provides a rich source of mathematical tools for constructing codes with good properties
- Goppa codes have a compact representation using the generator matrix or parity check matrix, which can be efficiently computed from the underlying elliptic curve
- The decoding algorithms for Goppa codes (e.g., Patterson's algorithm, Berlekamp-Massey algorithm) are efficient and can correct a large number of errors

Challenges in implementation

- Despite the advantages of elliptic curve cyclic codes, there are some challenges in their practical implementation:
- The construction of Goppa codes from elliptic curves requires a good understanding of algebraic geometry and the theory of function fields, which may be a barrier for some practitioners
- The decoding algorithms for Goppa codes are more complex than those for traditional cyclic codes, which may result in higher computational costs and latency
- The choice of the underlying elliptic curve and the divisor used in the construction of the code can have a significant impact on the performance and security of the resulting code
- Implementing elliptic curve arithmetic and the decoding algorithms efficiently requires careful optimization and attention to details, especially when targeting resource-constrained environments

Advanced topics

- The study of elliptic curve cyclic codes has led to the development of several advanced topics and generalizations that explore the connections between coding theory, algebraic geometry, and number theory

- These advanced topics offer new opportunities for constructing codes with improved properties and for studying the fundamental limits of error correction and cryptography

Hermitian codes

- Hermitian codes are a generalization of Goppa codes that use Hermitian curves instead of elliptic curves
- A Hermitian curve is a type of algebraic curve defined over a finite field of square order, given by the equation $y^q + y = x^{q+1}$
- Hermitian codes can be constructed using a similar approach to Goppa codes, by evaluating functions from the Riemann-Roch space of a divisor on the Hermitian curve
- Hermitian codes have good error-correcting properties and can achieve the Singleton bound in some cases, making them optimal codes

Algebraic geometry codes

- Algebraic geometry codes are a broad class of error-correcting codes that use tools from algebraic geometry to construct codes with good properties
- These codes are constructed by evaluating functions or differentials on algebraic curves or higher-dimensional varieties over finite fields
- Goppa codes and Hermitian codes are special cases of algebraic geometry codes, but the general construction allows for the use of a wider range of algebraic curves and geometric objects
- Algebraic geometry codes have been used to construct codes with better parameters than classical codes (e.g., Reed-Solomon codes) and to study the fundamental limits of coding theory

Elliptic curve discrete logarithm problem

- The security of many elliptic curve cryptographic schemes relies on the difficulty of the elliptic curve discrete logarithm problem (ECDLP)
- The ECDLP is the problem of finding an integer k such that $Q = kP$, given points P and Q on an elliptic curve
- The best known algorithms for solving the ECDLP have exponential complexity, which makes it a suitable basis for cryptographic schemes
- However, the ECDLP is not believed to be hard for quantum computers, which has led to the development of post-quantum cryptographic schemes

Post-quantum cryptography considerations

- The development of quantum computers poses a threat to the security of many classical cryptographic schemes, including those based on the ECDLP
- Post-quantum cryptography aims to develop cryptographic schemes that are secure against attacks by both classical and quantum computers
- Some post-quantum cryptographic schemes based on elliptic curve cyclic codes have been proposed, such as the McEliece cryptosystem and its variants

- These schemes rely on the difficulty of decoding a general linear code, which is believed to be hard even for quantum computers
- However, the security and efficiency of these post-quantum schemes are still active areas of research, and standardization efforts are ongoing to identify the most promising candidates for future use

9.4 Elliptic curves and quantum error-correcting codes

Elliptic curves play a crucial role in modern cryptography, offering strong security with smaller key sizes. They're used in key exchange protocols like ECDH and digital signatures like ECDSA, relying on the difficulty of solving the elliptic curve discrete logarithm problem.

However, quantum computers pose a significant threat to elliptic curve cryptography. Shor's algorithm can efficiently solve the ECDLP, prompting research into quantum-resistant cryptography. Quantum error-correcting codes, including those based on elliptic curves, are essential for building reliable quantum computers and secure quantum communication systems.

Elliptic curves in cryptography

- Elliptic curves have become a fundamental tool in modern cryptography due to their unique mathematical properties and ability to provide strong security with smaller key sizes compared to other cryptosystems
- Cryptographic protocols based on elliptic curves rely on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP), which is believed to be much harder than the discrete logarithm problem over finite fields

Elliptic curve Diffie-Hellman (ECDH)

- ECDH is a key agreement protocol that allows two parties to establish a shared secret key over an insecure channel
- The protocol involves each party generating a private-public key pair on an agreed-upon elliptic curve and exchanging their public keys
- The shared secret is computed by each party using their own private key and the other party's public key, resulting in a common value that can be used for symmetric encryption
- ECDH provides forward secrecy, meaning that compromise of a private key does not compromise previously established shared secrets

Elliptic curve digital signature algorithm (ECDSA)

- ECDSA is a widely used digital signature scheme based on elliptic curves, providing authentication, non-repudiation, and integrity
- The signer generates a private-public key pair on an elliptic curve and uses their private key to sign a message digest (hash) of the document or transaction
- The signature consists of two components, r and S , computed using the signer's private key, the message digest, and a random nonce
- Verification involves using the signer's public key to validate the signature against the message digest, confirming the signer's identity and the integrity of the signed data

- ECDSA is used in various applications, including cryptocurrencies (Bitcoin), SSL/TLS certificates, and secure software updates

Security of elliptic curve cryptography

- The security of elliptic curve cryptography relies on the presumed intractability of the ECDLP, which states that given points P and Q on an elliptic curve, it is computationally infeasible to find an integer k such that $Q = kP$
- The best-known classical algorithms for solving the ECDLP have exponential time complexity, making elliptic curve cryptosystems secure against conventional computers when using appropriate key sizes and well-chosen curves
- Cryptographic standards, such as NIST and Brainpool curves, provide recommended parameters for secure implementation of elliptic curve cryptography
- Proper implementation and protection against side-channel attacks (timing, power analysis) are crucial for maintaining the security of elliptic curve-based systems

Quantum computing and elliptic curves

- Quantum computers, which exploit quantum mechanical phenomena like superposition and entanglement, pose a significant threat to the security of classical cryptosystems, including those based on elliptic curves
- While quantum computers of sufficient scale to break current cryptographic schemes do not yet exist, their potential development has prompted research into quantum-resistant cryptography and the impact of quantum algorithms on existing systems

Shor's algorithm for elliptic curves

- Shor's algorithm is a quantum algorithm that can efficiently solve the discrete logarithm problem, including the ECDLP, by reducing it to the problem of finding the period of a function
- The algorithm consists of two main steps: 1. A quantum subroutine that creates a superposition of states and applies a quantum Fourier transform to find the period of a function related to the ECDLP 2. A classical post-processing step that uses the period to compute the discrete logarithm
- Shor's algorithm for elliptic curves has a polynomial time complexity, meaning that it can solve the ECDLP in a number of steps that grows polynomially with the size of the problem, rendering current elliptic curve cryptosystems vulnerable to sufficiently powerful quantum computers

Impact on elliptic curve cryptography

- The existence of Shor's algorithm implies that elliptic curve cryptosystems, such as ECDH and ECDSA, will become insecure once quantum computers with enough qubits and low error rates are available
- This has led to the development of post-quantum cryptography, which seeks to design cryptographic algorithms that are resistant to attacks by both classical and quantum computers
- Candidates for post-quantum cryptography include lattice-based, code-based, hash-based, and multivariate cryptosystems, some of which are being standardized by organizations like NIST
- Migration to post-quantum cryptography will be necessary to maintain the long-term security of communication and data protection systems, particularly for applications with extended security lifetime

requirements

Quantum error-correcting codes

- Quantum error-correcting codes are essential for building reliable quantum computers and ensuring the integrity of quantum information in the presence of noise and decoherence
- These codes encode logical qubits into a larger number of physical qubits, introducing redundancy that allows for the detection and correction of errors without disturbing the encoded quantum state

Stabilizer codes

- Stabilizer codes are a broad class of quantum error-correcting codes that are defined using the stabilizer formalism, which describes the code space as the simultaneous eigenspace of a set of commuting Pauli operators called the stabilizer generators
- The most well-known examples of stabilizer codes are Shor's 9-qubit code, which can correct any single-qubit error, and the 7-qubit Steane code, which is a perfect code that saturates the quantum Hamming bound
- Stabilizer codes have a compact description in terms of the stabilizer generators and can be efficiently encoded and decoded using circuits consisting of Clifford gates and measurements
- Concatenated stabilizer codes, where the qubits of one code are themselves encoded using another code, can achieve arbitrarily low error rates at the cost of increased overhead

CSS codes

- CSS (Calderbank-Shor-Steane) codes are a subclass of stabilizer codes that are constructed from two classical linear codes, one for correcting X (bit-flip) errors and another for correcting Z (phase-flip) errors
- The stabilizer generators of a CSS code consist of tensor products of only X or only Z Pauli operators, which allows for a simpler encoding and decoding procedure compared to general stabilizer codes
- Notable examples of CSS codes include the 7-qubit Steane code and the 15-qubit Reed-Muller code
- CSS codes have a close connection to classical coding theory and can be used to construct quantum LDPC (low-density parity-check) codes with sparse parity-check matrices and efficient decoding algorithms

Topological quantum codes

- Topological quantum codes are a family of quantum error-correcting codes that protect quantum information by encoding it into the global properties of a many-body quantum system with a topological order
- The most prominent examples are the surface code and the color code, which are defined on a 2D lattice of qubits with local stabilizer generators associated with plaquettes (faces) and vertices of the lattice
- Topological codes have a high error threshold, meaning they can tolerate a significant level of noise before error correction fails, making them promising candidates for fault-tolerant quantum computation
- These codes have a local structure that allows for efficient error syndrome measurement and a simple decoding procedure based on minimum-weight perfect matching of error chains

- Higher-dimensional generalizations of topological codes, such as the 3D toric code, can achieve even better performance and fault-tolerance properties

Quantum error correction with elliptic curves

- Elliptic curves have found applications in the construction of quantum error-correcting codes, leveraging their rich mathematical structure and properties
- These codes combine the advantages of classical algebraic coding theory with the principles of quantum error correction, offering good performance and efficient decoding algorithms

Elliptic curve Goppa codes

- Goppa codes are a class of classical error-correcting codes that are constructed using polynomials over finite fields and have good minimum distance properties
- Elliptic curve Goppa codes are a variant of Goppa codes where the underlying polynomial is replaced by a rational function on an elliptic curve over a finite field
- These codes have better parameters (length, dimension, minimum distance) compared to classical Goppa codes and can be used as a starting point for constructing quantum error-correcting codes using the CSS construction
- The decoding of elliptic curve Goppa codes involves solving a key equation on the elliptic curve, which can be done efficiently using the Berlekamp-Massey-Sakata algorithm or the Sugiyama algorithm

Elliptic curve quantum codes

- Elliptic curve quantum codes are quantum error-correcting codes that are constructed from classical elliptic curve codes, such as elliptic curve Goppa codes or elliptic curve LDPC codes
- These codes inherit the good properties of their classical counterparts, such as high minimum distance and efficient decoding algorithms, while providing quantum error correction capabilities
- One approach to constructing elliptic curve quantum codes is to use a pair of elliptic curve codes with suitable properties (such as self-orthogonality) to form a CSS code
- Another approach is to use a single elliptic curve code and a suitable automorphism of the curve to define a quantum code with a stabilizer structure
- Elliptic curve quantum codes have been shown to achieve good parameters and performance, making them a promising avenue for quantum error correction research

Advantages vs classical error correction

- Quantum error correction is fundamentally different from classical error correction due to the nature of quantum information and the constraints imposed by the laws of quantum mechanics
- Quantum errors are continuous and can affect both the amplitude and phase of a qubit, requiring codes that can handle both bit-flip and phase-flip errors simultaneously
- Quantum error correction needs to protect against decoherence and entanglement with the environment, which have no classical analog
- Quantum codes must satisfy the no-cloning theorem, which prevents the creation of independent copies of arbitrary quantum states, limiting the use of repetition codes

- Despite these challenges, quantum error correction is essential for realizing the potential of quantum computing and enabling fault-tolerant quantum computation
- Elliptic curve-based quantum codes offer advantages such as good parameters, efficient decoding, and the potential for integrating with classical post-quantum cryptography based on elliptic curves

Applications of elliptic curve quantum codes

- Elliptic curve quantum codes have several potential applications in the field of quantum information processing and communication, where protecting quantum states from errors and ensuring the integrity of quantum operations is crucial

Quantum key distribution

- Quantum key distribution (QKD) is a secure communication protocol that uses quantum mechanics to establish a shared secret key between two parties, which can then be used for encrypting and decrypting classical messages
- QKD protocols, such as BB84 and E91, rely on the principles of quantum superposition, entanglement, and the no-cloning theorem to detect eavesdropping attempts and ensure the security of the shared key
- Elliptic curve quantum codes can be used to improve the efficiency and security of QKD by providing error correction and privacy amplification techniques that are tailored to the specific requirements of quantum communication
- By encoding the quantum states used in QKD with elliptic curve quantum codes, the protocol can tolerate higher levels of noise and channel errors, increasing the achievable key rates and communication distances

Quantum secure communication

- Quantum secure communication refers to a broader class of protocols that use quantum mechanics to ensure the confidentiality, integrity, and authenticity of communication between two or more parties
- In addition to QKD, these protocols may include quantum secret sharing, quantum digital signatures, and quantum secure direct communication
- Elliptic curve quantum codes can be applied to these protocols to provide error correction and security enhancements, similar to their use in QKD
- For example, in quantum secret sharing, elliptic curve quantum codes can be used to encode the shared quantum state in a way that is robust against errors and ensures that only authorized subsets of parties can reconstruct the secret

Fault-tolerant quantum computation

- Fault-tolerant quantum computation is a framework for building reliable quantum computers that can perform arbitrary quantum computations in the presence of noise and errors
- This is achieved by using quantum error correction to encode logical qubits and quantum gates in a way that suppresses the propagation of errors and allows for the detection and correction of errors at the physical level
- Elliptic curve quantum codes can be used as building blocks for fault-tolerant quantum computation, providing good error correction properties and efficient decoding algorithms

- By concatenating elliptic curve quantum codes with other codes or using them in a topological setting (such as the surface code), it is possible to construct fault-tolerant quantum circuits with high error thresholds and low overhead
- The use of elliptic curve quantum codes in fault-tolerant quantum computation can help in the design of practical quantum computers and the implementation of quantum algorithms for various applications, such as quantum chemistry, optimization, and machine learning

9.5 Elliptic curve cryptography in coding theory

Elliptic curve cryptography (ECC) is a powerful public-key cryptosystem based on the algebraic structure of elliptic curves over finite fields. It offers smaller key sizes and faster computations compared to traditional systems like RSA, making it ideal for resource-constrained environments.

ECC's security relies on the elliptic curve discrete logarithm problem, which is believed to be computationally infeasible to solve. This chapter covers the basics of elliptic curves, key exchange protocols, digital signatures, and practical applications of ECC in modern cryptography.

Basics of elliptic curve cryptography

- Elliptic curve cryptography (ECC) is a public-key cryptosystem based on the algebraic structure of elliptic curves over finite fields
- ECC relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP) for its security
- Elliptic curves used in cryptography are defined over finite fields, which are mathematical structures with a finite number of elements

Elliptic curves over finite fields

- An elliptic curve over a finite field is a set of points (x, y) that satisfy a specific cubic equation, along with a special point called the "point at infinity"
- The most common finite fields used in ECC are prime fields $\mathbb{F}_p$ and binary fields $\mathbb{F}_{2^m}$
- The choice of finite field affects the efficiency and security of the cryptographic operations performed on the elliptic curve

Elliptic curve equation

- The general form of an elliptic curve equation over a field K is $y^2 = x^3 + ax + b$, where $a, b \in K$ and the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$
- For cryptographic purposes, the elliptic curve equation is often simplified to $y^2 = x^3 + ax + b \pmod{p}$ for prime fields or $y^2 + xy = x^3 + ax^2 + b$ for binary fields
- The coefficients a and b determine the shape of the elliptic curve and its cryptographic properties

Point addition and doubling

- Elliptic curves have a group structure under a point addition operation, which allows for the construction of cryptographic schemes
- Given two points P and Q on an elliptic curve, their sum $P + Q$ is defined as the point obtained by drawing a line through P and Q and finding the third point of intersection with the curve

- Point doubling is a special case of point addition where $P = Q$, and the tangent line at P is used to find the result of $2P$
- The formulas for point addition and doubling involve arithmetic operations in the underlying finite field

Cyclic subgroups and generators

- A cyclic subgroup of an elliptic curve is a subset of points that can be generated by repeatedly adding a specific point to itself
- The point used to generate the cyclic subgroup is called the generator or base point
- The order of a cyclic subgroup is the number of points in the subgroup, and the order of a point is the smallest positive integer n such that $nP = \mathcal{O}$ (the point at infinity)
- Cryptographic protocols often rely on cyclic subgroups of prime order to ensure the security of the system

Elliptic curve discrete logarithm problem

- The security of elliptic curve cryptography relies on the difficulty of solving the elliptic curve discrete logarithm problem (ECDLP)
- The ECDLP is the foundation for various cryptographic schemes, such as key exchange and digital signatures
- Solving the ECDLP is believed to be computationally infeasible for properly chosen elliptic curves and key sizes

Computational intractability

- The ECDLP states that given an elliptic curve E over a finite field, a point $P \in E$, and a point $Q = kP$ for some unknown integer k , it is computationally infeasible to determine k
- The best-known algorithms for solving the ECDLP have a running time that is exponential in the size of the underlying field
- The computational intractability of the ECDLP is what makes elliptic curve cryptography secure against attacks

Security of elliptic curve cryptography

- The security of ECC depends on the choice of the underlying finite field, the elliptic curve parameters, and the key size
- Cryptographically secure elliptic curves are chosen to resist known attacks, such as the Pollard's rho algorithm and the index calculus method
- Key sizes in ECC are typically smaller than those in other public-key cryptosystems (RSA) while providing the same level of security

Comparison vs integer factorization problem

- The integer factorization problem (IFP) is the basis for the security of the RSA cryptosystem
- The IFP involves finding the prime factors of a large composite number, which is believed to be computationally difficult

- For the same level of security, elliptic curve cryptography requires significantly smaller key sizes compared to RSA
- ECC is more efficient in terms of computational resources and bandwidth compared to cryptosystems based on the IFP

Elliptic curve key exchange

- Elliptic curve key exchange protocols allow two parties to establish a shared secret key over an insecure channel
- The shared secret key can be used for symmetric encryption, message authentication, or other cryptographic purposes
- Elliptic curve key exchange is based on the Diffie-Hellman key exchange protocol adapted to elliptic curves

Diffie-Hellman key exchange on elliptic curves

- The Diffie-Hellman key exchange protocol can be implemented using the group structure of elliptic curves
- Instead of using the multiplicative group of integers modulo a prime, the protocol uses the group of points on an elliptic curve
- The security of the elliptic curve Diffie-Hellman (ECDH) protocol relies on the difficulty of solving the ECDLP

Elliptic curve Diffie-Hellman (ECDH)

- In ECDH, the two parties agree on a common elliptic curve E over a finite field, a base point $P \in E$ of order n , and a cryptographic hash function H
- Each party generates a private key d_A and d_B (integers less than n) and computes their public keys $Q_A = d_A P$ and $Q_B = d_B P$
- The parties exchange their public keys and compute the shared secret point $S = d_A Q_B = d_B Q_A = d_A d_B P$
- The shared secret key is derived by hashing the x-coordinate of S using the agreed-upon hash function H

Key generation and exchange process

1. Alice and Bob agree on the elliptic curve parameters (E, P, n, H)
2. Alice generates her private key d_A and computes her public key $Q_A = d_A P$
3. Bob generates his private key d_B and computes his public key $Q_B = d_B P$
4. Alice sends her public key Q_A to Bob, and Bob sends his public key Q_B to Alice
5. Alice computes the shared secret point $S = d_A Q_B$, and Bob computes the shared secret point $S = d_B Q_A$
6. Alice and Bob derive the shared secret key $K = H(x_S)$, where x_S is the x-coordinate of S

Shared secret derivation

- The shared secret point S computed by both parties is the same, as $d_A Q_B = d_A(d_B P) = d_B(d_A P) = d_B Q_A$
- To obtain the shared secret key, the x-coordinate of S is hashed using the agreed-upon cryptographic hash function H
- The hash function ensures that the shared secret key has a uniform distribution and a fixed length
- The resulting shared secret key can be used for symmetric encryption, message authentication, or key derivation for other purposes

Elliptic curve digital signature algorithm

- The Elliptic Curve Digital Signature Algorithm (ECDSA) is a variant of the Digital Signature Algorithm (DSA) that uses elliptic curve cryptography
- ECDSA is used for generating and verifying digital signatures, which provide authentication, integrity, and non-repudiation
- ECDSA signatures are more compact and computationally efficient compared to DSA signatures

ECDSA vs DSA

- ECDSA and DSA have similar signature generation and verification processes, but ECDSA uses elliptic curve operations instead of modular arithmetic
- ECDSA requires smaller key sizes than DSA for the same level of security, making it more suitable for resource-constrained environments
- ECDSA is based on the ECDLP, while DSA is based on the discrete logarithm problem in the multiplicative group of integers modulo a prime

Key generation and verification

- In ECDSA, the signer generates a key pair consisting of a private key d (an integer less than the order n of the base point) and a public key $Q = dP$
- The public key Q is used for signature verification, while the private key d is used for signature generation
- The verifier obtains the signer's public key Q through a trusted channel or a public key infrastructure (PKI)

Signature generation and verification process

1. Signature generation: - The signer selects a random integer k (less than n) and computes the point $R = kP$ - The signer calculates $r = x_R \bmod n$, where x_R is the x-coordinate of R - The signer computes $s = k^{-1}(H(m) + dr) \bmod n$, where $H(m)$ is the hash of the message m - The signature is the pair (r, s)
2. Signature verification: - The verifier computes $w = s^{-1} \bmod n$ and $u_1 = H(m)w \bmod n$ - The verifier computes $u_2 = rw \bmod n$ and the point $V = u_1P + u_2Q$ - The verifier calculates $v = x_V \bmod n$, where x_V is the x-coordinate of V - The signature is valid if $v = r$

Security and efficiency of ECDSA

- The security of ECDSA relies on the difficulty of solving the ECDLP and the collision resistance of the hash function used
- Properly implemented ECDSA with secure elliptic curve parameters and key sizes provides strong protection against forgery and tampering
- ECDSA is more efficient than DSA and RSA in terms of key sizes and computational costs, making it suitable for resource-constrained devices and high-volume signature applications

Elliptic curve cryptography in practice

- Elliptic curve cryptography has been widely adopted in various industry standards and protocols
- ECC is used in secure communication protocols, digital certificates, cryptocurrency systems, and embedded devices
- Standardized elliptic curves and domain parameters have been developed to ensure interoperability and security

Standardized elliptic curves

- Several organizations, such as NIST, SECG, and Brainpool, have published standardized elliptic curves for cryptographic use
- Examples of standardized curves include NIST P-256, secp256k1 (used in Bitcoin), and Curve25519 (used in Signal and WhatsApp)
- Standardized curves are chosen based on their security properties, efficiency, and resistance to known attacks

Curve parameters and domain parameters

- Elliptic curve domain parameters consist of the finite field, the elliptic curve equation, the base point, the order of the base point, and the cofactor
- Domain parameters are used to ensure that different implementations of ECC are interoperable and secure
- Curve parameters, such as the coefficients of the elliptic curve equation, are part of the domain parameters and determine the security and efficiency of the cryptographic operations

ECC in TLS and SSL

- Elliptic curve cryptography is used in the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols for secure web communication
- ECC cipher suites in TLS/SSL provide forward secrecy and improved performance compared to RSA-based cipher suites
- Examples of ECC cipher suites include ECDHE-ECDSA-AES256-GCM-SHA384 and ECDHE-RSA-AES128-GCM-SHA256

ECC in cryptocurrency and blockchain

- Elliptic curve cryptography is the foundation for many cryptocurrency systems and blockchain platforms
- Bitcoin and Ethereum use ECDSA with the secp256k1 curve for generating and verifying transaction signatures
- ECC enables compact and efficient signatures, which is crucial for scalability and performance in blockchain applications

Attacks on elliptic curve cryptography

- While elliptic curve cryptography is considered secure when properly implemented, various attacks have been proposed to exploit vulnerabilities in specific implementations or curve parameters
- Side-channel attacks and fault injection attacks target the physical implementation of ECC, while mathematical attacks aim to solve the ECDLP more efficiently
- Cryptographers and security researchers continuously study and improve ECC to mitigate the risk of attacks

Side-channel attacks

- Side-channel attacks exploit information leakage from the physical implementation of ECC, such as timing, power consumption, or electromagnetic emanations
- Examples of side-channel attacks include timing attacks, power analysis attacks, and cache attacks
- Countermeasures against side-channel attacks involve constant-time implementations, randomization techniques, and physical shielding

Fault injection attacks

- Fault injection attacks intentionally introduce errors into the computation of ECC operations to reveal sensitive information or bypass security checks
- Attackers may use techniques such as voltage glitches, laser pulses, or electromagnetic pulses to induce faults
- Countermeasures against fault injection attacks include error detection and correction, redundant computations, and physical tamper resistance

Pollard's rho algorithm

- Pollard's rho algorithm is a probabilistic method for solving the ECDLP in a cyclic subgroup of an elliptic curve
- The algorithm uses a pseudo-random walk on the elliptic curve to find a collision, which reveals the discrete logarithm
- The expected running time of Pollard's rho algorithm is $O(\sqrt{n})$, where n is the order of the subgroup
- Selecting elliptic curves with a large prime order and implementing countermeasures like point blinding can mitigate the risk of Pollard's rho attack

Isogenies and supersingular curves

- Isogenies are mappings between elliptic curves that preserve the group structure
- Supersingular elliptic curves have a special structure that makes them vulnerable to attacks based on isogenies
- The supersingular isogeny Diffie-Hellman (SIDH) key exchange protocol, later known as SIKE, was proposed as a post-quantum cryptographic scheme
- However, recent research has shown that SIDH/SIKE is vulnerable to attacks and is no longer considered secure for post-quantum cryptography

Advantages and limitations of ECC

- Elliptic curve cryptography offers several advantages over other public-key cryptosystems, but it also has some limitations and challenges
- Understanding the strengths and weaknesses of ECC is essential for selecting the appropriate cryptographic scheme for a given application
- Ongoing research aims to improve the efficiency, security, and usability of ECC and address its limitations

Shorter key sizes vs RSA

- One of the main advantages of ECC is that it requires shorter key sizes compared to RSA for the same level of security
- For example, a 256-bit ECC key provides similar security to a 3072-bit RSA key
- Shorter key sizes result in reduced storage, transmission, and computation costs, making ECC suitable for resource-constrained environments

Computational efficiency

- ECC operations, such as point multiplication and addition, are computationally efficient compared to the modular exponentiation used in RSA
- The efficiency of ECC enables faster key generation, encryption, decryption, signing, and verification processes
- ECC is particularly advantageous for devices with limited processing power and battery life, such as smartphones, smart cards, and IoT devices

Implementation challenges

- Implementing ECC securely and efficiently requires careful attention to various aspects, such as curve selection, point representation, and side-channel protection
- ECC implementations must be thoroughly tested and validated to ensure they are free from vulnerabilities and conform to standards
- Interoperability between different ECC implementations can be challenging due to the variety of curve parameters and domain parameters in use

Future developments and research areas

- Research in elliptic curve cryptography continues to explore new techniques for improving efficiency, security, and functionality
- Areas of active research include post-quantum cryptography, pairing-based cryptography, and secure multiparty computation
- The development of quantum computers poses a threat to the security of ECC, as Shor's algorithm can solve the ECDLP in polynomial time
- Post-quantum cryptographic schemes based on other mathematical problems, such as lattices and isogenies, are being investigated as potential alternatives to ECC

9.6 Elliptic curve-based secret sharing schemes

Elliptic curve-based secret sharing schemes divide sensitive information among multiple parties using elliptic curve math. This approach offers enhanced security and efficiency compared to traditional methods, making it valuable for protecting cryptographic keys and other confidential data.

These schemes adapt Shamir's secret sharing to elliptic curves, leveraging their algebraic properties. They enable threshold-based access control, verifiable sharing, and proactive refreshing of shares. Applications include distributed key generation, threshold cryptography, and secure multi-party computation.

Elliptic curve-based secret sharing

- Elliptic curve-based secret sharing involves dividing a secret into multiple shares and distributing them among participants using the mathematical properties of elliptic curves
- Provides secure and efficient methods for sharing sensitive information, such as cryptographic keys, among multiple parties
- Enables threshold-based access control, where a minimum number of participants must collaborate to reconstruct the secret

Shamir's secret sharing scheme

- Shamir's secret sharing is a fundamental scheme that allows dividing a secret into n shares, where any t (threshold) or more shares can reconstruct the secret
- Based on polynomial interpolation, where the secret is encoded as the constant term of a polynomial and shares are generated by evaluating the polynomial at different points
- Provides information-theoretic security, meaning that any fewer than t shares reveal no information about the secret

Elliptic curve adaptation

- Elliptic curve secret sharing adapts Shamir's scheme to the context of elliptic curve cryptography
- Utilizes the algebraic structure of elliptic curves over finite fields to generate shares and reconstruct the secret
- Leverages the security and efficiency properties of elliptic curves, such as smaller key sizes and faster computations compared to traditional schemes

Dealer vs participants

- In secret sharing, there are two main roles: the dealer and the participants
- The dealer is responsible for generating the shares and distributing them securely to the participants
- Participants receive their respective shares and collaborate to reconstruct the secret when needed (threshold number of participants required)

Threshold schemes

- Threshold secret sharing schemes require a minimum number of participants (threshold) to collaborate to reconstruct the secret
- Provides resilience against the loss or unavailability of a subset of shares
- Enhances security by preventing individual participants from accessing the secret alone

Verifiable secret sharing

- Verifiable secret sharing adds an additional layer of security to detect and prevent malicious behavior by the dealer or participants
- Enables participants to verify the consistency and validity of their received shares without revealing the secret
- Uses cryptographic techniques, such as commitments and zero-knowledge proofs, to ensure the integrity of the shares

Proactive secret sharing

- Proactive secret sharing involves periodically refreshing the shares to maintain the security of the scheme over time
- Addresses the issue of long-term storage of shares and potential compromise of individual participants
- Enables the scheme to tolerate a mobile adversary who can corrupt different participants over time

Security of EC-based secret sharing

- The security of elliptic curve-based secret sharing relies on the hardness of mathematical problems specific to elliptic curves
- Provides strong security guarantees based on well-established assumptions in elliptic curve cryptography
- Resistant to various attacks, such as collusion among malicious participants or attempts to recover the secret from a subset of shares

Discrete logarithm problem

- The discrete logarithm problem (DLP) on elliptic curves is the foundation of security in EC-based secret sharing
- Given a point P on an elliptic curve and a scalar multiple $Q = kP$, finding the value of k is computationally infeasible for large elliptic curves

- The hardness of the DLP ensures that an adversary cannot efficiently recover the secret from the shares

Decisional Diffie-Hellman assumption

- The decisional Diffie-Hellman (DDH) assumption is another important security assumption in EC-based secret sharing
- States that given three points P , aP , and bP on an elliptic curve, it is computationally infeasible to distinguish between the point abP and a random point on the curve
- The DDH assumption is used to prove the security of various EC-based secret sharing schemes

Malicious participants

- EC-based secret sharing schemes must consider the presence of malicious participants who may attempt to disrupt the secret reconstruction process
- Malicious participants may provide incorrect shares or manipulate the communication channels to mislead other participants
- Verifiable secret sharing techniques are employed to detect and mitigate the impact of malicious behavior

Verifiability of shares

- Verifiability ensures that participants can validate the correctness of their received shares without revealing the secret
- Achieved through the use of commitments, where the dealer publishes cryptographic commitments of the shares
- Participants can verify the consistency of their shares against the commitments, detecting any tampering or malicious behavior

Applications of EC-based secret sharing

- Elliptic curve-based secret sharing finds applications in various domains where secure and distributed management of sensitive information is crucial
- Enables the implementation of advanced cryptographic protocols and secure multi-party computation
- Particularly relevant in decentralized systems and threshold cryptography scenarios

Distributed key generation

- EC-based secret sharing is used for distributed key generation, where multiple parties collaborate to generate a shared cryptographic key
- Each party contributes a share of the key, ensuring that no single party has complete control over the key
- Enables secure and decentralized generation of keys for various cryptographic purposes (encryption, signing)

Threshold cryptography

- Threshold cryptography leverages EC-based secret sharing to distribute cryptographic operations among multiple parties
- Allows for the secure decryption or signing of messages only when a threshold number of parties cooperate
- Enhances security by eliminating single points of failure and protecting against key compromise

Multi-party computation

- EC-based secret sharing is a fundamental building block for secure multi-party computation protocols
- Enables multiple parties to jointly compute a function on their private inputs without revealing the inputs to each other
- Preserves privacy and allows for secure collaborative computations in various domains (financial, medical, etc.)

Decentralized systems

- EC-based secret sharing is employed in decentralized systems, such as blockchain networks and distributed ledgers
- Enables secure and distributed storage of sensitive data, such as private keys or confidential information
- Provides resilience against single points of failure and enhances the overall security and trustworthiness of the system

Efficiency considerations

- Efficiency is a critical factor in the practical deployment of elliptic curve-based secret sharing schemes
- Involves trade-offs between security, share size, computational complexity, and communication overhead
- Careful design and parameter selection are necessary to achieve an optimal balance between security and efficiency

Share size vs security level

- The share size in EC-based secret sharing schemes directly impacts the security level and storage requirements
- Larger share sizes provide higher security by making it harder for an adversary to guess or recover the shares
- However, larger shares also increase the storage overhead and communication bandwidth required for share distribution and reconstruction

Computational complexity

- The computational complexity of EC-based secret sharing operations, such as share generation and secret reconstruction, is an important consideration

- Efficient algorithms and optimizations are employed to minimize the computational burden on participants
- The choice of elliptic curve parameters and underlying field arithmetic affects the computational complexity

Communication overhead

- EC-based secret sharing involves the distribution of shares among participants, which incurs communication overhead
- The communication cost depends on the number of participants, the size of the shares, and the network topology
- Efficient communication protocols and compression techniques can be utilized to reduce the overhead and improve the overall performance

Pairing-based schemes

- Pairing-based cryptography, which utilizes bilinear maps on elliptic curves, can be leveraged in EC-based secret sharing schemes
- Pairing-based schemes offer additional functionalities and security properties, such as identity-based encryption and attribute-based access control
- However, pairing operations are computationally expensive, and their use requires careful consideration of efficiency trade-offs

Variants and extensions

- Elliptic curve-based secret sharing has been extended and adapted to various scenarios and requirements
- Variants and extensions address specific challenges, such as hierarchical access control, dynamic participant sets, and enhanced security properties
- These advancements expand the applicability and flexibility of EC-based secret sharing in different domains

Hierarchical secret sharing

- Hierarchical secret sharing introduces a hierarchical structure among participants, where different levels have different access privileges
- Enables fine-grained access control and reflects organizational hierarchies or security clearance levels
- Achieved by using multiple layers of secret sharing and assigning different thresholds to each level

Dynamic secret sharing

- Dynamic secret sharing allows for the addition or removal of participants without reconstructing the entire scheme
- Enables flexible management of participant sets and accommodates changes in the system over time
- Achieved through techniques such as share re-randomization and update protocols

Compartmented secret sharing

- Compartmented secret sharing divides the secret into multiple compartments, each with its own access policy
- Allows for the separation of sensitive information based on different security domains or access requirements
- Enables granular access control and enhances the overall security of the scheme

Publicly verifiable secret sharing

- Publicly verifiable secret sharing extends the verifiability property to allow external parties to verify the correctness of shares
- Enables public scrutiny and enhances transparency in applications where trust in the dealer or participants is limited
- Achieved through the use of public-key cryptography and publicly verifiable commitments or proofs

Unit 10 – Elliptic Curves in Primality Testing & Factoring

10.1 Elliptic curve primality proving (ECP)

Elliptic curve primality proving (ECP) is a powerful method for verifying if a number is prime. It uses elliptic curves over finite fields to construct a sequence of curves with specific properties, ultimately proving primality.

ECP relies on the relationship between elliptic curve group orders and field primality. It's more reliable than probabilistic tests and can be faster than other deterministic methods for large numbers, making it valuable in cryptography and number theory.

Elliptic curve primality proving (ECP)

- Probabilistic primality testing method based on elliptic curves over finite fields
- Proves primality of integers by constructing a sequence of elliptic curves with certain properties
- Relies on the relationship between the order of elliptic curve groups and the primality of the underlying field

Atkin-Goldwasser-Kilian-Morain certificates

- Primality certificates used in ECP to prove an integer is prime
- Consist of a sequence of elliptic curves and isogenies between them
- Each curve has a number of points that is almost prime, differing from a prime by a small cofactor
- Isogenies connect the curves, preserving the prime order up to the cofactor

Elliptic curve method for factorization

- Algorithm for integer factorization using elliptic curves (ECM)
- Finds a non-trivial factor of a composite number by constructing random elliptic curves over the ring modulo the number
- Hopes to find a curve with smooth order, leading to a factor via the greatest common divisor (GCD)
- Can be used as a subroutine in ECP to prove primality by ruling out small factors

Relationship between ECP and ECM

- ECP uses the failure of ECM to provide evidence of primality
- If ECM fails to find factors after sufficiently many attempts, the number is likely prime
- ECP constructs a certificate of primality by linking curves via isogenies
- ECM is a Monte Carlo algorithm, while ECP is a Las Vegas algorithm (always correct, but running time varies)

Proving primality vs proving compositeness

- Primality testing determines whether a number is prime or composite
- Proving compositeness is generally easier, as finding a non-trivial factor suffices
- Proving primality requires showing no non-trivial factors exist
- ECPP provides a deterministic proof of primality, while most other methods are probabilistic

Probabilistic primality tests

- Algorithms that determine primality with a certain probability of error
- Examples include Fermat, Miller-Rabin, and Solovay-Strassen tests
- Rely on properties that hold for all primes but are unlikely for composites
- Can be fooled by pseudoprimes, composites that pass the test
- Probability of error can be reduced by repeating the test with different parameters

Deterministic primality proofs

- Algorithms that prove primality with certainty, no probability of error
- Typically slower than probabilistic tests, but provide a certificate of primality
- Examples include ECPP, AKS test, and cyclotomic methods
- Often impractical for very large numbers due to computational complexity

Goldwasser-Kilian algorithm

- First ECPP algorithm, developed by Shafi Goldwasser and Joe Kilian in 1986
- Constructs a sequence of elliptic curves over finite fields of increasing size
- Each curve has an almost prime number of points, certified by a recursively constructed primality proof
- Isogenies connect the curves, preserving the prime order up to a small cofactor
- Final curve has a prime number of points equal to the input, proving its primality

Atkin-Morain improvements

- Enhancements to the Goldwasser-Kilian algorithm by A. O. L. Atkin and François Morain
- Use complex multiplication to construct curves with a known number of points
- Employ isogenies of degree ℓ to connect curves, reducing the recursion depth
- Implement fast polynomial arithmetic to speed up computations
- Lead to a more practical and efficient ECPP algorithm

Suyama parametrization

- Method for constructing elliptic curves with a known number of points

- Developed by Hiromi Suyama in 1985
- Parametrizes curves by a single integer, simplifying the search space
- Guarantees the curve order is divisible by 12, useful for ECPP
- Allows efficient generation of curves with a desired number of points

Choosing curves for ECPP

- Selecting suitable elliptic curves is crucial for the efficiency of ECPP
- Curves should have an almost prime number of points, with a small cofactor
- Complex multiplication constructions (such as Suyama parametrization) help find such curves
- Curves with special properties (e.g., Montgomery form) can speed up computations
- Balancing security (curve size) and performance is important for practical implementations

Checking isogenies in ECPP

- Verifying the correctness of isogenies is a key step in ECPP
- Isogenies must preserve the prime order of the curves up to the cofactor
- Explicit formulas for isogenies of small degree can be used for efficiency
- Vélu's formulas provide a general method for computing isogenies
- Checking isogenies ensures the primality proof is valid

Complexity analysis of ECPP

- ECPP has a polynomial-time complexity, but the degree is relatively high
- The Goldwasser-Kilian algorithm has a running time of $O((\log n)^{10+\epsilon})$ for proving n is prime
- Atkin-Morain improvements reduce the complexity to around $O((\log n)^{6+\epsilon})$
- Heuristic arguments suggest the average-case complexity may be lower
- Practical implementations have subexponential running times for large inputs

ECPP vs other primality tests

- ECPP is deterministic, providing a proof of primality, unlike probabilistic tests
- It is slower than probabilistic methods (e.g., Miller-Rabin) for most inputs
- ECPP is more complex to implement than other primality tests
- For very large numbers, ECPP may be more efficient than the AKS test
- ECPP has better worst-case complexity than cyclotomic methods

Implementing ECPP

- Efficient implementation of ECPP requires careful optimization
- Key components include curve construction, isogeny computation, and polynomial arithmetic

- Libraries for elliptic curve cryptography (e.g., MIRACL, Crypto++) can be adapted for ECPP
- Parallel and distributed implementations can help tackle large inputs
- Specialized hardware (e.g., FPGAs) can significantly speed up ECPP

Applications of ECPP in cryptography

- ECPP is used to generate prime numbers for cryptographic purposes
- Elliptic curve cryptography (ECC) relies on the difficulty of the discrete logarithm problem in elliptic curve groups
- Generating prime-order curves is crucial for the security of ECC
- ECPP can be used to prove the primality of the curve order and the underlying field size
- Other applications include generating RSA primes and testing the primality of Mersenne numbers

10.2 Atkin-Morain ECPP

The Atkin-Morain ECPP algorithm is a powerful method for proving primality using elliptic curves. It combines complex multiplication techniques with efficient point counting to generate primality certificates for large numbers.

This algorithm showcases the practical applications of elliptic curves in number theory. By constructing curves with known properties and exploiting their structure, it offers a probabilistic approach to primality testing with high confidence.

Atkin-Morain ECPP algorithm

- Probabilistic primality proving algorithm based on elliptic curves over finite fields
- Utilizes the complex multiplication method to efficiently construct elliptic curves with suitable properties
- Generates primality certificates that can be used to prove the primality of large numbers

Elliptic curve primality proving

- Primality testing approach that relies on the properties of elliptic curves over finite fields
- Exploits the relationship between the number of points on an elliptic curve and the primality of the underlying field
- Offers a probabilistic method for proving primality with high confidence

Complex multiplication method

- Technique used to construct elliptic curves with known number of points
- Involves finding suitable discriminants and computing the Hilbert class polynomial
- Enables the efficient generation of elliptic curves for primality proving purposes

Constructing elliptic curves

- Process of finding elliptic curves with desired properties for primality proving

- Involves selecting suitable discriminants and computing the Hilbert class polynomial
- Aims to generate elliptic curves with a known number of points over a finite field

Suitable discriminants

- Negative integers that satisfy certain conditions for constructing elliptic curves
- Must be congruent to 0 or 1 modulo 4 and have a small class number
- Examples of suitable discriminants include -3, -4, -7, -8, -11, -19, -43, -67, -163

Hilbert class polynomial

- Polynomial whose roots are the j -invariants of elliptic curves with complex multiplication by a given discriminant
- Computed using the complex multiplication method and plays a crucial role in constructing suitable elliptic curves
- Has integer coefficients and can be efficiently evaluated modulo a prime number

Roots and j -invariants

- Roots of the Hilbert class polynomial correspond to the j -invariants of elliptic curves with complex multiplication
- j -invariants characterize isomorphism classes of elliptic curves over a given field
- Used to construct elliptic curves with a known number of points for primality proving

Elliptic curve generation

- Process of constructing elliptic curves using the j -invariants obtained from the Hilbert class polynomial
- Involves selecting a suitable j -invariant and computing the coefficients of the elliptic curve equation
- Generates elliptic curves with a known number of points over a finite field

Point counting

- Determining the number of points on an elliptic curve over a finite field
- Crucial step in the Atkin-Morain ECPP algorithm to assess the primality of the underlying field
- Efficient point counting algorithms, such as the Schoof-Elkies-Atkin (SEA) algorithm, are employed

Schoof-Elkies-Atkin (SEA) algorithm

- Deterministic polynomial-time algorithm for counting the number of points on an elliptic curve over a finite field
- Combines the approaches of Schoof's algorithm, Elkies' method, and Atkin's improvement
- Achieves a complexity of $O(\log^4 p)$ for a prime p , making it practical for large primes

Primality certificates

- Proofs that demonstrate the primality of a number using the properties of elliptic curves
- Generated by the Atkin-Morain ECPP algorithm based on the number of points on the constructed elliptic curves
- Can be verified efficiently to confirm the primality of the tested number

Proving primality

- Process of establishing the primality of a number using the Atkin-Morain ECPP algorithm
- Involves constructing suitable elliptic curves, counting the number of points, and generating primality certificates
- Provides a probabilistic proof of primality with a high level of confidence

Probable prime curves

- Elliptic curves constructed by the Atkin-Morain ECPP algorithm that satisfy certain conditions
- Have a number of points that is likely to be prime based on the properties of the underlying field
- Used to generate primality certificates and prove the primality of large numbers

Recursion and factorization

- Atkin-Morain ECPP algorithm employs recursion to handle composite numbers encountered during the primality testing process
- If the number of points on an elliptic curve is composite, the algorithm recursively applies the primality test to its prime factors
- Factorization of the number of points is performed to reduce the problem to smaller instances

Improvements and optimizations

- Various enhancements have been proposed to improve the efficiency and practicality of the Atkin-Morain ECPP algorithm
- Techniques such as the use of isogenies, optimized curve construction, and parallel computation have been explored
- Aim to reduce the computational complexity and speed up the primality proving process

Complexity analysis

- Atkin-Morain ECPP algorithm has a polynomial-time complexity, making it efficient for primality testing
- The complexity depends on the size of the tested number and the efficiency of the underlying operations (point counting, curve construction)
- Heuristic analysis suggests an expected running time of $O(\log^4 n)$ for a number n , assuming the Generalized Riemann Hypothesis (GRH)

Atkin-Morain vs other ECPP methods

- Atkin-Morain ECPP algorithm is one of the most well-known and widely used ECPP methods
- Other ECPP methods include the Goldwasser-Kilian algorithm and the Adleman-Huang algorithm
- Atkin-Morain ECPP offers a balance between efficiency and practicality, making it suitable for primality proving of large numbers
- Comparisons in terms of complexity, implementation, and performance have been studied in the literature

10.3 Lenstra's elliptic curve factorization method

Lenstra's elliptic curve factorization method is a powerful algorithm for breaking down large composite numbers. It uses elliptic curves over finite fields to find factors efficiently, making it useful for cryptanalysis and number theory research.

The method works by selecting random curves and points, then computing multiples using the group law. If a non-trivial factor is found during calculations, the algorithm stops. This process repeats with different curves until the number is fully factored.

Overview of Lenstra's method

- Lenstra's elliptic curve factorization method is a powerful algorithm for factoring large composite integers
- It utilizes the properties of elliptic curves over finite fields to efficiently find non-trivial factors of a given integer
- The method has important applications in cryptanalysis and computational number theory

Factoring integers using elliptic curves

- Lenstra's method works by randomly selecting elliptic curves and points on those curves
- The algorithm computes multiples of the chosen point using the group law on the elliptic curve
- If a non-trivial factor is found during the computation, the algorithm terminates and returns the factor
- The process is repeated with different random curves and points until a complete factorization is obtained

Key concepts

- Understanding Lenstra's method requires familiarity with several key concepts from elliptic curve theory and abstract algebra
- These concepts include elliptic curves over finite fields, the group law, torsion points, and orders of points
- Mastering these foundational ideas is essential for grasping the workings and efficiency of the algorithm

Elliptic curves over finite fields

- Elliptic curves can be defined over finite fields $\mathbb{F}_p$, where p is a prime number

- The curve equation takes the form $y^2 = x^3 + ax + b$ with coefficients $a, b \in \mathbb{F}_p$
- Points on the curve are pairs (x, y) satisfying the equation, along with a special point at infinity $\mathcal{O}$
- The set of points forms a finite abelian group under a well-defined addition operation

Group law on elliptic curves

- The group law on an elliptic curve defines how to add two points to obtain a third point on the curve
- Geometrically, addition involves drawing a line through the two points and finding the third intersection with the curve
- The group law formulas can be derived using the chord-and-tangent method or algebraic manipulations
- The group operation is associative, commutative, and has identity element $\mathcal{O}$ and inverses for each point

Torsion points and orders

- A torsion point on an elliptic curve is a point of finite order, meaning that a finite number of self-additions yields the identity $\mathcal{O}$
- The order of a point P is the smallest positive integer n such that $nP = \mathcal{O}$, where nP denotes the n -fold addition of P to itself
- The set of all torsion points forms a subgroup of the elliptic curve group, called the torsion subgroup
- Understanding torsion points and their orders is crucial for analyzing the behavior of point multiples in Lenstra's method

Lenstra's algorithm

- Lenstra's algorithm is an iterative procedure that attempts to find non-trivial factors of a given integer N using elliptic curves
- The main steps involve choosing random curves and points, computing multiples, and detecting factors
- The efficiency of the method relies on the properties of elliptic curve groups and the probability of hitting a factor in a reasonable number of iterations

Choosing random elliptic curves

- In each iteration, a random elliptic curve E over $\mathbb{F}_N$ is selected by choosing random coefficients a, b
- The curve should be non-singular, meaning that its discriminant $\Delta = -16(4a^3 + 27b^2)$ is non-zero modulo N
- A random point P on the curve is also chosen to serve as the starting point for computing multiples

Computing multiples of points

- The algorithm computes successive multiples of the chosen point P using the group law on the elliptic curve
- The multiples are computed as $P, 2P, 3P, \dots, kP$, where k is incremented in each step

- The computation is performed using efficient algorithms for point addition and doubling, such as the double-and-add method
- The process continues until either a non-trivial factor is found or a maximum number of iterations is reached

Detecting non-trivial factors

- During the computation of point multiples, the algorithm checks for the occurrence of a non-trivial factor of N
- This happens when a point addition or doubling step fails due to a division by a non-invertible element modulo N
- The greatest common divisor (GCD) of the non-invertible element and N reveals a non-trivial factor
- If a factor is found, the algorithm terminates and returns the factor; otherwise, it proceeds with the next multiple

Complexity and runtime analysis

- The runtime of Lenstra's method depends on the size of the smallest prime factor p of the input integer N
- Heuristically, the algorithm is expected to find a factor in $O(\exp(\sqrt{\log p \log \log p}))$ bit operations
- This subexponential complexity makes Lenstra's method more efficient than trial division and Pollard's p method for large inputs
- However, the actual performance can vary depending on the choice of random curves and the distribution of prime factors

Variations and optimizations

- Several variations and optimizations of Lenstra's original method have been proposed to improve its performance and applicability
- These include the use of special curve families, combination with other factoring techniques, and parallel implementations
- Researchers continue to explore ways to enhance the efficiency and practicality of elliptic curve factorization

Suyama curves for better performance

- Suyama curves are a family of elliptic curves that exhibit favorable properties for factoring purposes
- They are defined by equations of the form $y^2 = x^3 + (\sigma^2 - 5)x + (10 - 4\sigma^2)$, where σ is a parameter
- Suyama curves have a high probability of having a point of smooth order, which can speed up the factoring process
- Using Suyama curves instead of random curves can significantly improve the performance of Lenstra's method

Combining with other factoring methods

- Lenstra's method can be combined with other factoring algorithms to create hybrid approaches
- For example, it can be used as a preprocessing step to remove small factors before applying more expensive methods like the quadratic sieve or number field sieve
- Combining Lenstra's method with the elliptic curve method (ECM) can also yield better results for certain types of inputs
- Hybrid approaches aim to leverage the strengths of different algorithms and optimize the overall factoring process

Distributed and parallel implementations

- Lenstra's method is well-suited for distributed and parallel implementations due to its iterative nature
- Different processors or nodes can work on distinct random curves and points simultaneously, sharing any factors found
- Parallel implementations can take advantage of multi-core CPUs, GPUs, or cluster computing environments
- Distributed computing projects like the ECM Factorization App allow volunteers to contribute their computing resources to factor large integers collaboratively

Applications

- Lenstra's elliptic curve factorization method has found various applications in cryptography, number theory, and computer algebra
- Its ability to efficiently factor large integers has implications for the security of certain cryptographic systems
- The method has also contributed to advances in computational number theory and the development of powerful algebraic software

Cryptanalysis of RSA and other systems

- Lenstra's method can be used to attack the security of the widely-used RSA cryptosystem
- RSA relies on the difficulty of factoring large composite numbers, typically the product of two large primes
- If an attacker can factor the RSA modulus using Lenstra's method or other techniques, they can compute the private key and break the encryption
- The method has also been applied to analyze the security of other cryptographic schemes based on integer factorization

Factoring in number theory research

- Integer factorization is a fundamental problem in number theory with connections to various mathematical concepts and conjectures
- Lenstra's method has been used to factor large numbers arising in number-theoretic investigations

- Researchers have employed the method to study the properties of prime numbers, test conjectures, and explore computational aspects of number theory
- The development of efficient factoring algorithms like Lenstra's method has stimulated further research in this area

Use in computational algebra systems

- Lenstra's method has been implemented in several computational algebra systems and mathematical software packages
- These systems provide tools for solving problems in algebra, number theory, and cryptography
- Examples include Mathematica, Maple, Sage, and specialized libraries like GMP-ECM and FLINT
- The integration of Lenstra's method in these systems allows users to efficiently factor integers and perform related computations

Comparisons

- Lenstra's elliptic curve factorization method can be compared with other integer factoring algorithms in terms of efficiency, suitability, and limitations
- Understanding the relative strengths and weaknesses of different methods helps in selecting the most appropriate approach for a given factoring problem
- Researchers continue to study and compare factoring algorithms to advance the state of the art in this field

Lenstra's method vs other factoring algorithms

- Lenstra's method compares favorably to other factoring algorithms like trial division, Pollard's ρ method, and the quadratic sieve for certain input sizes
- It has a subexponential runtime complexity, which makes it more efficient than exponential-time methods like trial division for large inputs
- However, for very large numbers (hundreds of digits or more), the general number field sieve (GNFS) is typically faster than Lenstra's method
- The performance of Lenstra's method can also be influenced by the size and distribution of the prime factors of the input integer

Advantages and limitations

- Lenstra's method has several advantages, including its subexponential complexity, parallelizability, and the ability to find small factors efficiently
- It is particularly effective when the input integer has a small prime factor, as the method is more likely to find it quickly
- However, the method also has some limitations and drawbacks
- Its performance can be sensitive to the choice of random curves and points, and it may require many iterations to find all factors

- The method is not guaranteed to find all factors in a specified number of steps, and its runtime can vary significantly depending on the input

Suitability for different problem sizes

- The suitability of Lenstra's method for a given factoring problem depends on the size of the input integer and the available computational resources
- For integers up to around 100 digits, Lenstra's method can be a good choice, especially if small factors are expected
- For larger inputs (100-200 digits), the quadratic sieve or other subexponential methods may be more efficient
- Beyond 200 digits, the general number field sieve is usually the fastest known method, although it has a high memory requirement
- In practice, the choice of factoring algorithm often involves a trade-off between speed, memory usage, and implementation complexity

10.4 Montgomery's elliptic curve factorization method

Montgomery's elliptic curve factorization method is a powerful tool for breaking down large composite numbers. It uses the unique properties of elliptic curves over finite fields to find factors efficiently, outperforming many traditional factoring methods.

This technique is crucial in cryptography, especially for assessing the security of systems like RSA. By exploiting the group structure of elliptic curves, Montgomery's method offers a clever approach to the age-old problem of integer factorization.

Overview of Montgomery's method

- Montgomery's elliptic curve factorization method utilizes the mathematical properties of elliptic curves to efficiently factor large composite integers
- Relies on the idea that finding certain points on carefully chosen elliptic curves can reveal factors of the target integer
- Offers advantages over other factoring methods in terms of computational efficiency and the size of numbers it can handle

Key ideas

Elliptic curves over finite fields

- Elliptic curves are defined by specific cubic equations and can be studied over various fields, including finite fields ($\mathbb{F}_p$ where p is prime)
- Over finite fields, elliptic curves have a finite number of points that form a group under a well-defined addition operation
- Properties of elliptic curves over finite fields, such as the number of points and the structure of the group, play a crucial role in Montgomery's method

Factoring with elliptic curves

- Montgomery's method exploits the connection between the group structure of elliptic curves and the factorization of integers
- By mapping an integer to a point on a carefully chosen elliptic curve and performing operations on that point, it is possible to obtain information about the factors of the original integer
- The method relies on the fact that if a point on the curve has an order that divides the target integer, it can lead to a non-trivial factor

Advantages vs other factoring methods

- Montgomery's method is particularly efficient for factoring large integers (hundreds of digits) compared to other methods like trial division or Pollard's rho algorithm
- It has a subexponential running time, making it faster than exponential-time algorithms for sufficiently large inputs
- Requires less memory compared to other subexponential factoring algorithms like the quadratic sieve or number field sieve

Mathematical foundations

Elliptic curve equations

- An elliptic curve over a field K is defined by an equation of the form $y^2 = x^3 + ax + b$, where $a, b \in K$ and the discriminant $\Delta = 4a^3 + 27b^2 \neq 0$
- Montgomery's method uses a specific form of elliptic curve equation: $By^2 = x^3 + Ax^2 + x$, where $A, B \in K$ and $B(A^2 - 4) \neq 0$
- The choice of the parameters A and B affects the properties of the curve and its suitability for factoring

Group law on elliptic curves

- Points on an elliptic curve (including a special point at infinity) form an abelian group under a geometric addition operation
- The group law is defined by a chord-and-tangent process: given two points P and Q , the sum $P + Q$ is obtained by drawing a line through P and Q and finding the third point of intersection with the curve, then reflecting that point across the x-axis
- The group law allows for scalar multiplication of points, i.e., computing kP for a positive integer k by repeated addition

Torsion points and orders

- A point P on an elliptic curve is called a torsion point if there exists a positive integer k such that $kP = \mathcal{O}$ (the point at infinity)
- The smallest such k is called the order of the point P
- The set of all torsion points on an elliptic curve forms a subgroup, and the order of this subgroup is called the torsion order

- Montgomery's method relies on finding points with certain orders to obtain information about the factors of the target integer

Montgomery's algorithm

Choosing suitable elliptic curves

- The choice of elliptic curve is crucial for the success of Montgomery's method
- The curve must be chosen so that its order (number of points) has certain properties relative to the target integer N
- Typically, curves are selected so that their order is smooth, i.e., composed of small prime factors
- The parameters A and B of the curve equation are chosen to satisfy specific conditions that make the curve suitable for factoring

Mapping integers to curve points

- To factor an integer N , Montgomery's method starts by mapping N to a point on the chosen elliptic curve
- This mapping is done by computing $x \equiv N \pmod{p}$ for a prime p and then solving for y in the curve equation
- If a solution exists, the point (x, y) is used as the starting point for the factoring process
- If no solution exists, a different prime p or a different curve is chosen

Calculating multiples of points

- The core of Montgomery's method involves calculating multiples of the starting point P on the elliptic curve
- Multiples are computed using the group law, i.e., $kP = P + P + \dots + P$ (k times)
- Efficient algorithms, such as the double-and-add method or the Montgomery ladder, are used to compute multiples quickly
- The goal is to find a multiple kP such that its order divides the target integer N

Detecting non-trivial factors

- If a multiple kP is found such that its order divides N , it means that $kP = \mathcal{O} \pmod{q}$ for some prime factor q of N
- This information can be used to obtain a non-trivial factor of N by computing $\gcd(x_k - x_P, N)$, where x_k and x_P are the x -coordinates of kP and P , respectively
- If the GCD is not 1 or N , it is a non-trivial factor of N
- The process is repeated with different curves and starting points until all factors of N are found

Computational aspects

Pseudocode of core algorithm

```
function MontgomeryFactorization(N):
  while N is not factored:
    Choose an elliptic curve E and a point P on E
    k ← 1
    while k ≤ B:
      Q ← kP
      g ← gcd(x_Q - x_P, N)
      if 1 < g < N:
        return g
      k ← k + 1
  return "N is prime"
```

Complexity analysis

- Montgomery's method has a subexponential running time, typically expressed as $O(e^{c(\log N)^{1/2}(\log \log N)^{1/2}})$ for some constant C
- The complexity depends on the choice of the bound B for the multiples and the smoothness of the curve orders
- In practice, the method is faster than exponential-time algorithms like trial division for sufficiently large inputs

Optimizations and speedups

- Various optimizations can be applied to Montgomery's method to improve its efficiency
- The choice of elliptic curves can be optimized by using curves with particularly smooth orders or by precomputing suitable curves
- The computation of multiples can be sped up using windowing techniques or by exploiting the structure of the Montgomery ladder
- Early abort strategies can be employed to detect useless curves or points quickly

Comparison vs quadratic sieve

- Montgomery's method and the quadratic sieve are both subexponential factoring algorithms
- The quadratic sieve has a better asymptotic complexity than Montgomery's method, with a running time of approximately $O(e^{c(\log N)^{1/2}(\log \log N)^{1/4}})$
- However, Montgomery's method has the advantage of requiring less memory and being easier to parallelize
- In practice, the performance of the two methods depends on the specific implementation and the size of the numbers being factored

Applications and significance

Integer factorization in cryptography

- Integer factorization is a fundamental problem in cryptography, as the security of many cryptographic systems relies on the difficulty of factoring large numbers
- Public-key cryptosystems like RSA base their security on the assumption that factoring the modulus (a product of two large primes) is computationally infeasible
- Efficient factoring methods, such as Montgomery's method, are of great interest in cryptanalysis and in assessing the security of cryptographic protocols

Impact on RSA security

- The security of the widely-used RSA cryptosystem depends on the hardness of factoring large integers
- Montgomery's method, along with other advanced factoring algorithms, poses a potential threat to RSA if the key sizes are not chosen appropriately
- As factoring methods improve, the recommended key sizes for RSA have increased to maintain a sufficient level of security
- Understanding the capabilities of factoring algorithms is crucial for setting appropriate key lengths and ensuring the long-term security of RSA-based systems

Role in number theory research

- Montgomery's method is not only of practical importance but also of theoretical interest in number theory
- The development and analysis of elliptic curve factorization methods have led to new insights and connections between different areas of mathematics
- Studying the properties of elliptic curves and their application to factoring has contributed to the growth of computational number theory
- Montgomery's method and related techniques have inspired further research on the use of elliptic curves in cryptography and on the design of efficient algorithms for various number-theoretic problems

10.5 Elliptic curve method for discrete logarithms

Elliptic curve discrete logarithms are a crucial concept in cryptography. They involve finding an integer k such that $Q = kP$ on an elliptic curve. This problem's difficulty underpins the security of many cryptographic protocols.

Various methods exist to tackle this problem, including index calculus, Pollard's rho, and Pohlig-Hellman algorithms. However, quantum algorithms like Shor's pose a significant threat, driving research into post-quantum cryptography alternatives.

Elliptic curve discrete logarithm problem

- The elliptic curve discrete logarithm problem (ECDLP) is a fundamental computational problem in elliptic curve cryptography

- Given an elliptic curve E over a finite field, a point P on E , and another point Q on E , the ECDLP is to find an integer k such that $Q = kP$
- The security of many elliptic curve cryptographic protocols relies on the presumed intractability of the ECDLP for properly chosen curves

Discrete logarithms on elliptic curves

- The discrete logarithm problem on elliptic curves is defined analogously to the classical discrete logarithm problem in finite fields
- Instead of working with elements of a finite field, the ECDLP deals with points on an elliptic curve
- The group operation on elliptic curves is point addition, which gives the set of points on the curve a group structure

Difficulty vs classical discrete logarithms

- The ECDLP is believed to be significantly harder than the classical discrete logarithm problem in finite fields of comparable size
- The best known algorithms for solving the ECDLP have exponential complexity, while subexponential algorithms exist for the classical case
- This increased difficulty allows elliptic curve cryptosystems to use smaller key sizes while maintaining the same level of security as classical systems (RSA)

Index calculus algorithms

- Index calculus algorithms are a class of subexponential algorithms for solving the classical discrete logarithm problem in finite fields
- These algorithms work by expressing the target element as a product of smaller elements, called a "smooth" representation
- The logarithms of the smaller elements are then computed and combined to obtain the desired discrete logarithm

Idea behind index calculus methods

- The key idea behind index calculus methods is to build a database of logarithms of small elements (the factor base) in the finite field
- The target element is then expressed as a product of factor base elements using a process called the "sieving stage"
- By solving a linear system of equations, the logarithm of the target element can be computed from the logarithms of the factor base elements

Challenges for elliptic curves

- Index calculus methods are not directly applicable to elliptic curves due to the lack of a suitable notion of "smoothness" for points
- The group operation on elliptic curves (point addition) does not correspond to a simple arithmetic operation like multiplication in finite fields

- Attempts to adapt index calculus methods to elliptic curves have not yielded subexponential algorithms so far

Elliptic curve method

- The elliptic curve method (ECM) is a factorization algorithm that uses elliptic curves to find small prime factors of large integers
- While not directly related to the ECDLP, ECM shares some similarities with the index calculus approach and provides insight into the structure of elliptic curves

Overview of ECM algorithm

- The ECM algorithm starts by choosing a random elliptic curve E and a random point P on E
- The point P is then multiplied by a large power k to obtain the point $Q = kP$
- If the prime factor p divides the integer kP , the computation of Q will fail, revealing the factor p

Sieving stage

- In the sieving stage of ECM, the point P is multiplied by a product of small primes (the factor base) raised to some powers
- This stage is analogous to the sieving stage in index calculus methods, where the target element is expressed as a product of factor base elements
- The sieving stage in ECM helps to increase the probability of finding a factor by effectively testing multiple points simultaneously

Linear algebra stage

- If the sieving stage fails to find a factor, the ECM algorithm proceeds to the linear algebra stage
- In this stage, a system of linear equations is constructed using the points obtained from the sieving stage
- By solving this system, the algorithm attempts to find a non-trivial factor of the input integer

Complexity analysis of ECM

- The complexity of ECM depends on the size of the smallest prime factor p of the input integer
- The expected running time of ECM is $O(e^{c\sqrt{\log p \log \log p}})$, where c is a constant
- ECM is particularly effective for finding small prime factors, but its performance deteriorates for larger factors

Pollard's rho algorithm

- Pollard's rho algorithm is a probabilistic algorithm for solving the discrete logarithm problem, including the ECDLP
- The algorithm works by simulating a random walk in the group and exploiting the birthday paradox to find a collision

Idea behind Pollard's rho

- The key idea behind Pollard's rho is to define a pseudo-random sequence of group elements using a function f that maps group elements to group elements
- The sequence starts from a random initial point and is defined by $X_{i+1} = f(X_i)$
- Due to the birthday paradox, the sequence will eventually cycle and a collision $X_i = X_j$ will occur

Application to elliptic curves

- To apply Pollard's rho to the ECDLP, the function f is defined as a random combination of the point P and the target point Q
- The sequence of points is computed using the group operation (point addition) on the elliptic curve
- When a collision is found, it yields a linear relation between the logarithms of P and Q , which can be solved to obtain the discrete logarithm

Parallelization of Pollard's rho

- Pollard's rho algorithm can be parallelized efficiently using the distinguished points method
- Each parallel instance of the algorithm starts from a different initial point and follows the pseudo-random sequence until a distinguished point (a point satisfying a certain property) is encountered
- The distinguished points are then collected and checked for collisions, which can be used to solve the ECDLP

Complexity of Pollard's rho on elliptic curves

- The expected running time of Pollard's rho on elliptic curves is $O(\sqrt{n})$ group operations, where n is the order of the group
- This complexity is based on the birthday paradox, which states that a collision is likely to occur after approximately $\sqrt{n}$ steps
- Pollard's rho is the best known algorithm for solving the ECDLP on general elliptic curves

Pohlig-Hellman algorithm

- The Pohlig-Hellman algorithm is a method for solving the discrete logarithm problem in groups where the order of the group is a smooth number (a product of small primes)
- The algorithm works by reducing the discrete logarithm problem in the full group to discrete logarithm problems in smaller subgroups

Idea of Pohlig-Hellman

- The key idea behind the Pohlig-Hellman algorithm is to express the discrete logarithm as a sum of discrete logarithms in subgroups of prime power order
- If the order of the group is $n = p_1^{e_1} \cdots p_k^{e_k}$, the algorithm computes the discrete logarithm modulo each prime power $p_i^{e_i}$ separately

- The results are then combined using the Chinese Remainder Theorem to obtain the discrete logarithm in the full group

Adaptation for elliptic curves

- The Pohlig-Hellman algorithm can be applied to the ECDLP when the order of the elliptic curve group is a smooth number
- The algorithm computes the discrete logarithm in subgroups of prime power order using a suitable algorithm (Pollard's rho)
- The results are then combined to obtain the discrete logarithm in the full elliptic curve group

Complexity of Pohlig-Hellman on elliptic curves

- The complexity of the Pohlig-Hellman algorithm depends on the size of the largest prime factor P of the group order n
- The running time is $O(\sqrt{P} \log n)$ group operations, where P is the largest prime factor of n
- If the group order is a smooth number (all prime factors are small), the Pohlig-Hellman algorithm can solve the ECDLP efficiently

Attacks on anomalous curves

- Anomalous curves are elliptic curves over finite fields with certain special properties that make them vulnerable to attacks
- These curves have group orders that are equal to the size of the underlying finite field, which allows for a reduction of the ECDLP to the classical discrete logarithm problem

Definition of anomalous curves

- An elliptic curve E over a finite field $\mathbb{F}_q$ is called anomalous if the group order $\#E(\mathbb{F}_q)$ is equal to q
- Anomalous curves have a unique structure that makes them susceptible to specific attacks
- The term "anomalous" refers to the fact that these curves have a different number of points than what is expected from the Hasse bound

Reduction to finite field discrete logarithms

- For anomalous curves, the ECDLP can be efficiently reduced to the classical discrete logarithm problem in the underlying finite field
- This reduction is based on the fact that the group of points on an anomalous curve is isomorphic to the additive group of the finite field
- By mapping the points on the curve to elements of the finite field, the ECDLP can be solved using algorithms for the classical discrete logarithm problem (index calculus methods)

Implications for cryptography

- The reduction of the ECDLP to the classical discrete logarithm problem for anomalous curves has significant implications for elliptic curve cryptography

- Anomalous curves should be avoided in cryptographic applications, as they do not provide the desired level of security
- Cryptographic protocols based on elliptic curves must ensure that the chosen curves are not anomalous to prevent potential attacks

Pairing-based attacks

- Pairing-based attacks are a class of attacks on elliptic curve cryptosystems that exploit the existence of bilinear pairings on certain elliptic curves
- These attacks can reduce the ECDLP on a pairing-friendly curve to the classical discrete logarithm problem in a finite field

MOV attack using pairings

- The Menezes-Okamoto-Vanstone (MOV) attack is a pairing-based attack that reduces the ECDLP to the discrete logarithm problem in a finite field
- The attack uses the Weil pairing, a bilinear pairing defined on the ℓ -torsion points of an elliptic curve, where ℓ is a prime divisor of the group order
- By computing the Weil pairing of the points involved in the ECDLP and solving the resulting discrete logarithm problem in the finite field, the MOV attack can solve the ECDLP efficiently

Frey-Rück attack using pairings

- The Frey-Rück attack is another pairing-based attack that uses the Tate pairing instead of the Weil pairing
- The Tate pairing is a bilinear pairing that can be computed more efficiently than the Weil pairing
- Similar to the MOV attack, the Frey-Rück attack reduces the ECDLP to the discrete logarithm problem in a finite field by computing the Tate pairing of the points involved

Consequences for pairing-friendly curves

- Pairing-based attacks have important consequences for the selection of elliptic curves in cryptographic protocols
- Curves that are susceptible to these attacks, known as pairing-friendly curves, must be avoided in most cryptographic applications
- However, pairing-friendly curves have found applications in various cryptographic constructions, such as identity-based encryption and short signature schemes, where the existence of pairings is exploited constructively

Quantum algorithms

- Quantum algorithms are algorithms designed to run on quantum computers, which exploit quantum-mechanical phenomena to perform computations
- These algorithms can solve certain problems, including the discrete logarithm problem, more efficiently than classical algorithms

Shor's algorithm for discrete logarithms

- Shor's algorithm is a quantum algorithm for solving the discrete logarithm problem, including the ECDLP
- The algorithm uses the quantum Fourier transform and the period-finding technique to compute the discrete logarithm efficiently
- Shor's algorithm has a polynomial running time, which means that it can solve the ECDLP in polynomial time on a quantum computer

Impact on elliptic curve cryptography

- The existence of Shor's algorithm has significant implications for the security of elliptic curve cryptography
- If large-scale quantum computers become available, the ECDLP can be solved efficiently, rendering current elliptic curve cryptosystems insecure
- This potential threat has motivated the development of post-quantum cryptography, which seeks to design cryptographic systems that are resistant to quantum attacks

Post-quantum alternatives to elliptic curves

- Post-quantum cryptography focuses on developing cryptographic primitives that are secure against both classical and quantum attacks
- Some notable post-quantum alternatives to elliptic curve cryptography include: 1. Lattice-based cryptography: Cryptosystems based on the hardness of lattice problems (LWE, NTRU) 2. Code-based cryptography: Cryptosystems based on the difficulty of decoding random linear codes (McEliece, BIKE) 3. Multivariate cryptography: Cryptosystems based on the hardness of solving systems of multivariate polynomial equations (Rainbow, UOV) 4. Hash-based signatures: Digital signature schemes based on the security of hash functions (XMSS, SPHINCS+)
- These post-quantum alternatives are actively being researched and standardized to ensure the long-term security of cryptographic systems in the presence of quantum computers

10.6 Suyama's parametrization for ECM

Suyama's parametrization is a clever way to generate elliptic curves for the Elliptic Curve Method of factorization. It creates curves with specific properties that make them well-suited for finding factors of large numbers efficiently.

By using a single parameter, Suyama's method produces curves with known torsion groups and small ranks. This saves computation time and increases the chances of successfully factoring integers compared to using random curves.

Suyama's parametrization

- Suyama's parametrization is a method for generating elliptic curves with desirable properties for use in the Elliptic Curve Method (ECM) of integer factorization
- It allows for the efficient construction of curves with known rank and torsion group, which can improve the performance of ECM
- Suyama's parametrization is based on a specific family of elliptic curves known as Suyama curves

Motivation for parametrization

- Elliptic curves used in ECM should have certain properties to maximize the chances of finding a factor of the integer being factored
- Randomly selecting elliptic curves may not yield curves with the desired properties, leading to suboptimal performance
- Parametrizations like Suyama's allow for the targeted generation of curves with known characteristics that are beneficial for ECM

Definition of Suyama curves

- Suyama curves are a family of elliptic curves defined by the equation $y^2 = x^3 + ax + b$, where a and b satisfy certain conditions
- The coefficients a and b are parameterized by a single parameter σ , such that $a = -(\sigma^2 + 3\sigma + 3)$ and $b = \sigma^2 + 3\sigma + 3$
- The parameter σ is chosen to ensure that the resulting curve has a known torsion group and rank

Properties of Suyama curves

- Suyama curves have a torsion group isomorphic to $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$, meaning they have a point of order 2 and a point of order 4
- The rank of Suyama curves is typically small (often 0 or 1), which is desirable for ECM as it limits the size of the group of rational points
- Suyama curves have a simple and efficient parametrization, allowing for easy generation and computation

Advantages vs general elliptic curves

- Suyama curves offer several advantages over randomly selected elliptic curves in the context of ECM
- The known torsion group and small rank of Suyama curves can lead to faster convergence and higher success rates in finding factors
- The efficient parametrization reduces the computational overhead of generating and working with these curves compared to general elliptic curves

Suyama's theorem

- Suyama's theorem is a mathematical result that underlies the parametrization of Suyama curves
- It establishes a connection between the parameter σ and the properties of the resulting elliptic curve

Statement of theorem

- Suyama's theorem states that for any integer σ , the elliptic curve defined by $y^2 = x^3 - (\sigma^2 + 3\sigma + 3)x + \sigma^2 + 3\sigma + 3$ has a torsion group isomorphic to $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/4\mathbb{Z}$
- Furthermore, the theorem provides explicit formulas for the coordinates of the points of order 2 and 4 on the curve in terms of σ

Proof of theorem

- The proof of Suyama's theorem involves algebraic manipulations and the application of known results about elliptic curves
- It relies on the characterization of torsion groups of elliptic curves over $\mathbb{Q}$ and the use of the Nagell-Lutz theorem
- The proof demonstrates that the given parametrization always yields curves with the desired torsion group structure

Generating Suyama curves

- Suyama's parametrization provides a simple and efficient method for generating elliptic curves suitable for ECM
- By choosing appropriate values for the parameter σ , one can construct Suyama curves with desired properties

Suyama's parametrization

- Suyama's parametrization defines the coefficients a and b of the elliptic curve equation $y^2 = x^3 + ax + b$ in terms of the parameter σ
- The coefficients are given by $a = -(\sigma^2 + 3\sigma + 3)$ and $b = \sigma^2 + 3\sigma + 3$
- By substituting these expressions into the elliptic curve equation, one obtains a Suyama curve for a given value of σ

Choosing parameters

- The choice of the parameter σ determines the specific Suyama curve generated
- In practice, σ is often chosen to be a small integer to simplify computations and reduce the size of the coefficients
- Different values of σ lead to different Suyama curves, all sharing the same torsion group structure but varying in other properties such as the rank

Examples of Suyama curves

- For $\sigma = 1$, the resulting Suyama curve is defined by $y^2 = x^3 - 7x + 7$
- Setting $\sigma = 2$ yields the Suyama curve $y^2 = x^3 - 15x + 17$
- Other examples include the curves $y^2 = x^3 - 28x + 41$ ($\sigma = 3$) and $y^2 = x^3 - 46x + 71$ ($\sigma = 4$)

Applications in ECM

- Suyama curves find significant applications in the Elliptic Curve Method (ECM) of integer factorization
- Their special properties make them well-suited for use in various stages of the ECM algorithm

Suyama curves in stage 1

- Stage 1 of ECM involves the selection of an elliptic curve and a point on the curve to perform scalar multiplication

- Suyama curves are often used in stage 1 due to their known torsion group and small rank
- The efficient parametrization of Suyama curves allows for quick generation of suitable curves for this stage

Benefits for curve selection

- Using Suyama curves in ECM offers several benefits in terms of curve selection
- The known torsion group structure eliminates the need to compute the torsion group for each curve, saving computational resources
- The small rank of Suyama curves increases the likelihood of finding a factor in stage 1, as the group of rational points is typically smaller compared to random curves

Limitations of Suyama curves

- While Suyama curves have advantages in ECM, they also have some limitations
- The restricted family of curves may not always yield the best possible performance for all integers being factored
- In some cases, using a wider range of elliptic curves or different parametrizations may be more effective, especially for hard-to-factor integers

Variants of Suyama's parametrization

- Several variants and generalizations of Suyama's parametrization have been proposed to extend its applicability and improve its performance
- These variants aim to address some of the limitations of the original parametrization and provide additional flexibility in curve selection

Generalizations of theorem

- Researchers have explored generalizations of Suyama's theorem to encompass a broader class of elliptic curves
- These generalizations often involve relaxing the conditions on the torsion group or considering different parameterizations of the coefficients
- Generalized versions of Suyama's theorem can lead to the discovery of new families of curves with desirable properties for ECM

Alternative parametrizations

- Alternative parametrizations have been developed to generate elliptic curves with specific characteristics
- For example, the Montgomery parametrization focuses on curves of the form $by^2 = x^3 + ax^2 + x$, which have efficient arithmetic and are well-suited for ECM
- Other parametrizations, such as the Edwards parametrization, have been studied for their potential benefits in curve selection and computation

Comparison of variants

- Different variants of Suyama's parametrization offer trade-offs in terms of efficiency, flexibility, and performance
- Some variants may generate a wider range of curves but require more complex computations, while others prioritize simplicity and speed
- The choice of parametrization depends on the specific requirements of the ECM implementation and the characteristics of the integers being factored

Implementation considerations

- When implementing Suyama's parametrization or its variants in ECM, several practical considerations need to be taken into account
- Efficient implementation techniques can significantly impact the performance and scalability of the algorithm

Efficient parameter selection

- Selecting suitable values for the parameter σ is crucial for generating effective Suyama curves
- Strategies for efficient parameter selection include precomputing a set of good values, using heuristics to identify promising candidates, or employing randomized selection techniques
- The choice of parameter selection method depends on the desired balance between computational efficiency and the quality of the generated curves

Optimizing curve arithmetic

- Efficient arithmetic operations on elliptic curves are essential for the performance of ECM
- Implementing optimized algorithms for point addition, doubling, and scalar multiplication can significantly reduce the computational overhead
- Techniques such as projective coordinates, Montgomery ladder, and window-based methods can be employed to speed up curve arithmetic

Integration with ECM algorithm

- Suyama's parametrization should be seamlessly integrated into the overall ECM algorithm implementation
- This involves incorporating the curve generation process into the stage 1 and stage 2 procedures of ECM
- Efficient data structures and memory management techniques should be used to handle the storage and manipulation of Suyama curves throughout the factorization process
- Parallelization and distributed computing approaches can be explored to further enhance the performance of ECM when using Suyama curves