INTERNATIONAL JOURNAL OF PURE SCIENCE RESEARCH STUDIES (IJPSRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: ISSN Print
EDU SCHOOL
ijpsrs.com

Elliptic Curve Cryptography and the Discrete Logarithm Problem: An Algebraic Perspective

Renjith Varghese

Research Scholar, Department of Mathematics, St. Thomas College (Autonomous), Thrissur, India

Article information
Received: 2th August 2025
Received in revised form: 10th September 2025
Accepted: 17th October 2025
Available online: 14th November 2025

Volume:1
Issue:1
DOI: https://doi.org/10.5281/zenodo.17670077

Abstract

Elliptic Curve Cryptography (ECC) has emerged as a cornerstone of modern cryptographic systems, offering security levels comparable to traditional public-key cryptosystems with significantly reduced key sizes. This paper provides a comprehensive algebraic analysis of ECC, focusing on the mathematical foundations underlying the Elliptic Curve Discrete Logarithm Problem (ECDLP). We examine the group-theoretic properties of elliptic curves over finite fields, analyze the computational complexity of the discrete logarithm problem in this context, and evaluate current algorithmic approaches for solving ECDLP. The study presents a rigorous mathematical framework encompassing Weierstrass equations, point addition operations, scalar multiplication, and the algebraic structures that render ECDLP computationally intractable. We further investigate state-of-the-art attack methodologies including Pollard's rho algorithm, Baby-step Giant-step, and index calculus variants, demonstrating why ECC maintains its security advantage. The analysis concludes with implications for cryptographic protocol design and future directions in post-quantum cryptographic research. Our findings reinforce the robustness of ECC as a foundational technology for secure communication systems while identifying theoretical vulnerabilities that merit continued scrutiny.

Keywords: Elliptic Curve Cryptography, Discrete Logarithm Problem, Finite Fields, Group Theory, Cryptanalysis, Algebraic Geometry

I. INTRODUCTION

1.1. Motivation and Context

The exponential growth of digital communication systems and the proliferation of internet-connected devices have elevated cryptographic security from a specialized concern to a fundamental infrastructure requirement. Public-key cryptography, pioneered by Diffie and Hellman in 1976 and subsequently instantiated through RSA by Rivest, Shamir, and Adleman in 1977, established the paradigm for secure key exchange and digital signatures in open networks. However, the computational overhead and key size requirements of traditional public-key systems have motivated research into more efficient alternatives.

Elliptic Curve Cryptography, independently proposed by Neal Koblitz and Victor Miller in 1985, represents a fundamental shift in cryptographic methodology. By leveraging the algebraic structure of elliptic curves over finite fields, ECC achieves equivalent security to RSA with dramatically smaller key sizes—a 256-bit ECC key provides security comparable to a 3072-bit RSA key. This efficiency advantage translates directly into reduced computational requirements, lower power consumption, and decreased bandwidth utilization, making ECC particularly attractive for resource-constrained environments including mobile devices, embedded systems, and Internet of Things (IoT) applications.

1.2. The Discrete Logarithm Problem

The security of ECC fundamentally depends on the computational intractability of the Elliptic Curve Discrete Logarithm Problem. Given an elliptic curve E defined over a finite field F_q, a point P ∈ E (F_q) of order n, and another point Q ∈ ⟨P⟩ (the cyclic subgroup generated by P), the ECDLP requires finding the integer k ∈ [0, n-1] such that Q = kP, where kP denotes k-fold scalar multiplication of P.

The discrete logarithm problem in multiplicative groups of finite fields has been extensively studied since the inception of public-key cryptography. However, the ECDLP exhibits distinct characteristics that significantly impact its computational complexity. Unlike the classical DLP in (F_q)*, where sub-exponential algorithms such as the Number Field Sieve achieve complexity $O(exp((64/9)^{1/3}\ \ (logq)^{1/3}\ \ (loglogq)^{2/3}))$ the ECDLP resists such approaches due to the absence of a smooth homomorphism from the elliptic curve group to a more tractable algebraic structure.

1.3. Research Objectives and Contributions

This paper provides a comprehensive algebraic analysis of ECC and ECDLP with the following specific objectives:

- Establish rigorous mathematical foundations: We develop the complete algebraic framework for elliptic curves over finite fields, including detailed derivations of group operations and structural properties.
- Analyze computational complexity: We present a thorough examination of ECDLP complexity, demonstrating why current best algorithms require exponential time and establishing concrete security parameters.
- Evaluate cryptanalytic approaches: We systematically analyze existing algorithms for solving ECDLP, including generic algorithms applicable to arbitrary groups and specialized methods exploiting elliptic curve properties.
- Assess cryptographic implications: We examine how the algebraic properties of elliptic curves translate into practical security guarantees for cryptographic protocols.
- Identify research directions: We outline theoretical challenges and emerging threats, particularly from quantum computing, that will shape future cryptographic research.

1.4. Paper Organization

The remainder of this paper is structured as follows: Section II provides comprehensive background on elliptic curves, establishing the mathematical foundations necessary for subsequent analysis. Section III develops the algebraic theory of elliptic curves over finite fields, including group structure and point operations. Section IV presents a detailed examination of the ECDLP and its computational complexity. Section V analyzes algorithmic approaches for solving ECDLP. Section VI discusses cryptographic applications and protocol implementations. Section VII concludes with implications and future research directions.

II. MATHEMATICAL FOUNDATIONS

2.1. Elliptic Curves: Algebraic Definition

An elliptic curve E over a field K is the set of solutions (x, y) ∈ K × K to a generalized Weierstrass equation:

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

where $a_1, a_2, a_3, a_4, a_6 \in K$, together with a distinguished point O called the point at infinity. The curve must be non-singular, meaning it has no cusps or self-intersections. Formally, this non-singularity condition requires that the discriminant $\Delta \neq 0$.

For cryptographic applications, we typically work with elliptic curves over finite fields F_p (where p is a large prime) or F_2^m (binary fields). In the prime field case with characteristic p > 3, the Weierstrass equation simplifies to the short Weierstrass form:

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

where a, b ∈ F_p and the discriminant condition becomes $4a^3 + 27b^2 \neq 0 \pmod p$.

Figure 1: Elliptic Curve Point Addition

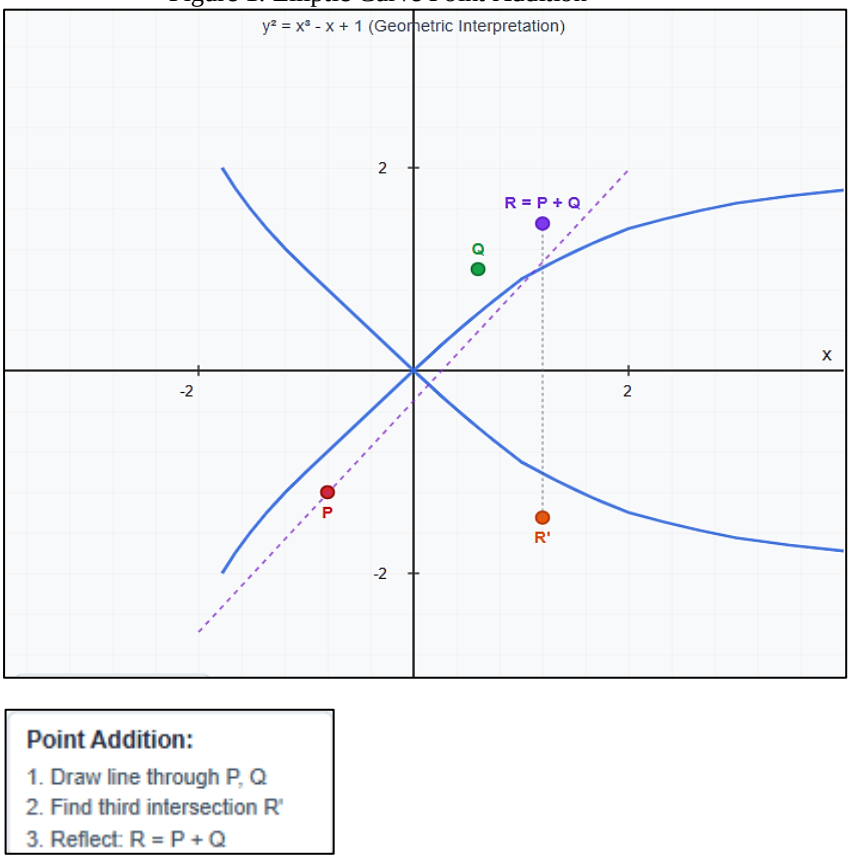

Figure 1 **:** Elliptic curve visualization showing the geometric structure of $y^2 = x^3 - x + 1$ over real numbers, illustrating the characteristic symmetric shape and point addition using the chord-and-tangent method

2.2. Group Structure on Elliptic Curves.

The set of points E(K) on an elliptic curve forms an abelian group under a geometric addition operation. For points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ on E, the sum $R = P + Q = (x_3, y_3)$ is defined through the following algebraic formulas:

2.2.1. Point Addition ($P \neq Q$):

If $x_1 \neq x_2$, the slope of the line through P and Q is: $\lambda = (y_2 - y_1) / (x_2 - x_1)$

$$\text{Then: } x_3 = \lambda^2 - x_1 - x_2 \; y_3 = \lambda(x_1 - x_3) - y_1$$

2.2.2. Point Doubling ($P = Q$):

For P = Q, where P has non-zero y-coordinate: $\lambda = (3x_1^2 + a) / (2y_1)$

$$\text{Then: } x_3 = \lambda^2 - 2x_1 \; y_3 = \lambda(x_1 - x_3) - y_1$$

2.2.3. Identity Element:

The point at infinity O serves as the identity element: $P + O = O + P = P$ for all $P \in E(K)$.

2.2.4. Inverse Element:

For P = (x, y), the inverse is -P = (x, -y), satisfying $P + (-P) = O$.

This group law satisfies all group axioms:

- Closure: For all $P, Q \in E(K)$, $P + Q \in E(K)$
- Associativity: $(P + Q) + R = P + (Q + R)$
- Identity: $O + P = P + O = P$
- Inverse: $P + (-P) = O$

The commutativity property P + Q = Q + P makes E(K) an abelian group, which is fundamental to the cryptographic properties we exploit.

2.3. Finite Field Arithmetic

Elliptic curve cryptography operates over finite fields to ensure computational feasibility and cryptographic security. The two primary finite field types used in ECC are:

2.3.1. Prime Fields F_P

These fields consist of integers modulo a large prime p. Arithmetic operations are performed modulo p:

- Addition: (a + b) mod p
- Multiplication: (a × b) mod p
- Inversion: a^{-1} such that $a \times a^{-1} \equiv 1 \pmod p$, computed via the Extended Euclidean Algorithm

2.3.2. Binary Fields F_2^m:

These fields consist of polynomials of degree less than m with binary coefficients, with arithmetic performed modulo an irreducible polynomial f(x) of degree m. Operations include:

- Addition: XOR operation
- Multiplication: Polynomial multiplication modulo f(x)
- Inversion: Extended Euclidean Algorithm for polynomials

The choice between prime and binary fields involves trade-offs between implementation efficiency and security considerations. Prime fields generally offer simpler theoretical analysis, while binary fields may provide implementation advantages in hardware.

2.4. Scalar Multiplication

Scalar multiplication is the fundamental operation in ECC, defined as repeated addition:

$$kP = P + P + ... + P \text{ (k times)}$$

For cryptographic applications, efficient computation of kP is essential. The binary method (double-and-add algorithm) computes kP in O(log k) point operations:

2.4.1. Algorithm 1: Binary Method for Scalar Multiplication

Input: k, P
Output: Q = kP

1. Write k in binary: $k = (k_t, k_{\{t-1\}}, ..., k_1, k_0)_2$
2. $Q \leftarrow O$
3. For i from t down to 0:
   a. $Q \leftarrow 2Q$
   b. If $k_i = 1$ then $Q \leftarrow Q + P$
4. Return Q

More sophisticated methods such as Non-Adjacent Form (NAF) representation and windowing techniques further optimize scalar multiplication, reducing the expected number of point additions by approximately 33% compared to the standard binary method.

III. ELLIPTIC CURVES OVER FINITE FIELDS

3.1. Point Counting and Group Order

The number of points on an elliptic curve E over F_q, denoted #E $(F_q)\cong$, is fundamental to cryptographic security. Hasse's theorem provides tight bounds on this value:

$$q + 1 - 2\sqrt{q} \leq \#E\ (F_q) \leq q + 1 + 2\sqrt{q}$$

The precise value can be computed using Schoof's algorithm, which runs in polynomial time $O((\log q)^8)$, and improved variants by Elkies and Atkin achieving $O((\log q)^4)$ complexity.

For cryptographic purposes, we require #E (F_q) to have a large prime factor n, typically satisfying

#E (F_q)= hn where h (the cofactor) is small and n is a large prime. The subgroup of order n then provides the cryptographic group in which ECDLP must be solved.

Figure 2: Hasse's Theorem - Group Order Distribution

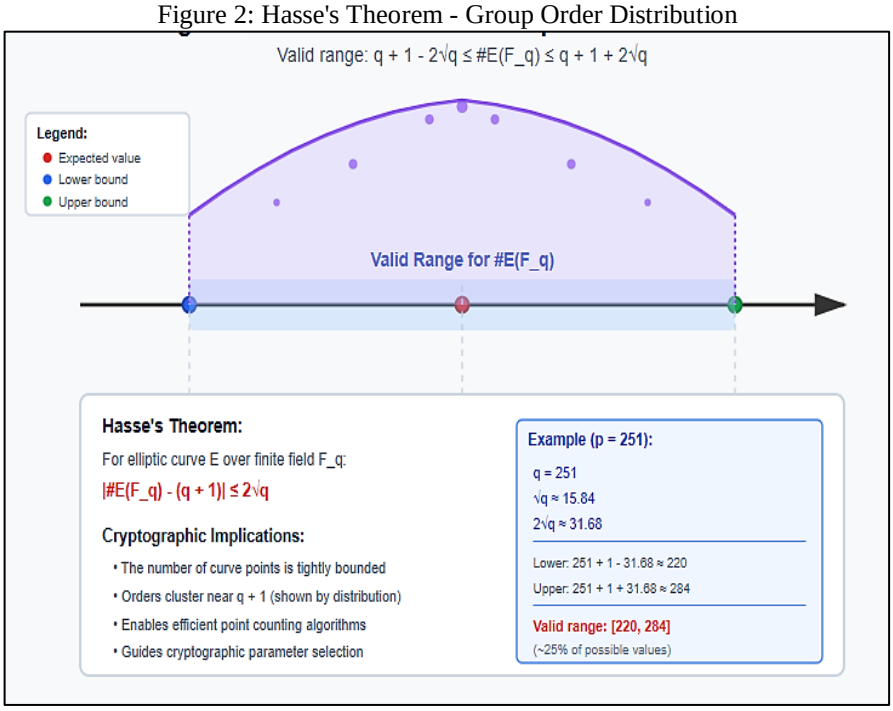

Figure 2 Schematic diagram illustrating the distribution of group orders for elliptic curves over F_p, showing how Hasse's theorem constrains possible values

3.2. Cyclic Subgroups and Generators

By Lagrange's theorem, the order of any point P divides #E(F_q). For cryptographic applications, we work within a cyclic subgroup G = ⟨P⟩ of prime order n, generated by a carefully chosen base point P.
The structure theorem for abelian groups guarantees that:

$$E(F_q) \cong Z_{n1} \times Z_{n2}$$

where n_2 divides both n_1 and q - 1. For cryptographically strong curves, we typically have #E(F_q)≅ = n (prime) or #E(F_q)≅ = 2n, 4n, 8n with n prime.

3.3. Embedding Degree and Complex Multiplication

The embedding degree k of an elliptic curve E over F_q with respect to a prime n dividing #E(F_q) is the smallest positive integer such that n q^{k-1} .This parameter determines the security against pairing-based attacks, as it defines the smallest extension field F_k^q into which E can be embedded while preserving the group structure.

For cryptographic security, we require k to be sufficiently large. Curves with small embedding degree are vulnerable to the MOV attack, which reduces ECDLP to DLP in F_k^q where sub-exponential algorithms apply.

Complex multiplication (CM) theory provides a method for constructing elliptic curves with prescribed group orders. Given a discriminant D, CM theory allows construction of curves over $F_p$with specific cryptographic properties, including resistance to known attacks and optimal efficiency for implementation.

3.4. Endomorphism Ring and Frobenius Map

The endomorphism ring End(E) of an elliptic curve consists of all rational maps E → E that preserve the group structure and the point at infinity. For ordinary curves over F_q, End(E) is isomorphic to an order in an imaginary quadratic field.

The Frobenius endomorphism φ_q: E → E defined by φ_q(x, y) = $x^q y^q$ plays a central role in the arithmetic of elliptic curves over finite fields. The Frobenius satisfies the characteristic equation:

$$\Psi_p^2 - t\Psi_q + q = 0$$

where t = q + 1 - #E(F_q) is the trace of Frobenius, with |t| ≤ 2√q by Hasse's theorem.

Understanding the endomorphism ring structure is crucial for assessing security, as certain endomorphisms can be exploited to accelerate cryptanalytic attacks. For instance, curves over F_{p^2} with efficiently computable endomorphisms are vulnerable to the GHS attack.

IV. THE ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM

4.1. Problem Formulation and Hardness

Definition 1 (ECDLP): Let E be an elliptic curve defined over a finite field F_q, let P ∈ E(F_q) be a point of prime order n, and let Q ∈ ⟨P⟩. The Elliptic Curve Discrete Logarithm Problem is to determine the unique integer k ∈ [0, n-1] such that Q = kP.

The computational hardness of ECDLP forms the foundation of ECC security. Unlike the classical discrete logarithm problem in multiplicative groups F_p*,, no sub-exponential algorithm is known for solving ECDLP on properly chosen elliptic curves. This asymmetry in computational complexity—easy to compute kP given k and P, but hard to recover k given P and kP—provides the one-way function property essential for public-key cryptography.

Figure 3: The Elliptic Curve Discrete Logarithm Problem

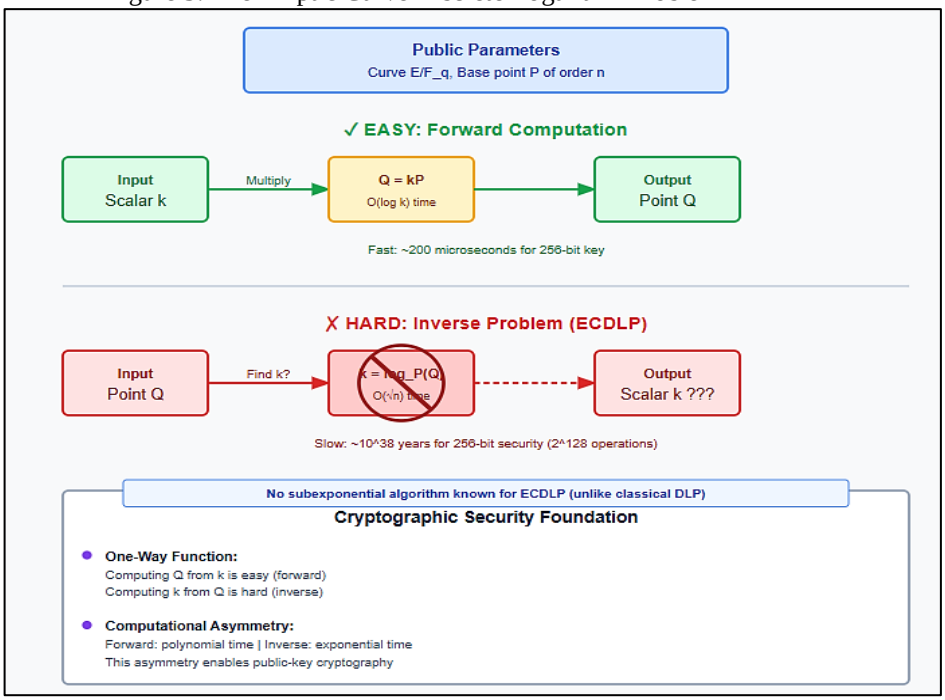

Figure 3: Flowchart illustrating the ECDLP: given base point P and target point Q = kP, the challenge of computing k despite efficient scalar multiplication

4.2. Complexity-Theoretic Analysis

The best known generic algorithms for solving ECDLP require O(√n) group operations, where n is the the order of the base point P. This exponential complexity in the bit length of n (approximately $O(2^{(log_2 n)\backslash 2})$) contrasts sharply with index calculus methods for the classical DLP, which achieve sub-exponential complexity.

The square-root complexity bound arises from the birthday paradox: in a group of order n, approximately √n random elements are required before a collision occurs with high probability. Both Pollard's rho and Baby-step Giant-step algorithms exploit this fundamental property.

For a security level of λ bits, we require n ≈ $2^{2\lambda}$ meaning a 256-bit elliptic curve provides 128-bit security. Current cryptographic standards recommend:

- 224-bit curves for 112-bit security
- 256-bit curves for 128-bit security
- 384-bit curves for 192-bit security
- 521-bit curves for 256-bit security

4.3. Reduction to Other Problems

ECDLP exhibits interesting relationships with other computational problems:

4.3.1. Relation to Computational Diffie-Hellman (CDH):

The CDH problem—given P, aP, bP, compute abP—is polynomial-time reducible to ECDLP. If ECDLP can be solved efficiently, CDH becomes trivial. However, the converse reduction is unknown, and CDH may be strictly easier than ECDLP.

4.3.2. Relation to Decisional Diffie-Hellman (DDH):

The DDH problem—given P, aP, bP, cP, decide if c = ab—is believed to be easier than CDH. On elliptic curves with efficiently computable pairings, DDH is easy due to bilinear maps, but ECDLP and CDH remain hard.

4.3.3. Relation to Inverse Problems:

The elliptic curve inverse problem—given P and kP, compute $k^{-1}P$ (where k^{-1} is the multiplicative inverse mod n)—is equivalent to ECDLP, as solving either immediately yields a solution to the other.

4.4. Weak Curve Classes

Certain elliptic curve configurations are vulnerable to attacks that reduce ECDLP to easier problems:

4.4.1. Supersingular Curves:

Curves with trace of Frobenius t = 0 have small embedding degree ($k \leq 6$), making them vulnerable to MOV/Frey-Rück attacks that transfer ECDLP to DLP in small extension fields.

4.4.2. Anomalous Curves:

Curves over F_p with exactly p points (#E F_p = p) are vulnerable to the Semaev-Smart-Satoh-Araki (SSSA) attack, which solves ECDLP in polynomial time using p-adic lifting techniques.

4.4.3. Curves with Small Embedding Degree:

If the embedding degree k is small relative to log q, pairing-based attacks reduce ECDLP to DLP in, F_q^k where index calculus methods may be applicable.

4.4.4. Curves Over F_2^m with Efficiently Computable Isogenies:

The GHS attack exploits efficiently computable isogenies to map ECDLP instances to hyperelliptic curves where index calculus variants are more effective.

Cryptographic standards explicitly prohibit these weak curve classes, requiring careful parameter selection during curve generation.

V. CRYPTANALYTIC ALGORITHMS

5.1. Generic Algorithms

Generic algorithms solve the discrete logarithm problem without exploiting specific properties of the group structure, making them applicable to ECDLP on any elliptic curve.

5.1.1. Baby-step Giant-step Algorithm:

Shanks' algorithm uses a meet-in-the-middle approach:

Algorithm 2: Baby-step Giant-step

Input: P, Q, n (where Q = kP for unknown k < n)

Output: k

1. m ← ⌈√n⌉
2. Compute baby steps: Store (iP, i) for i = 0, 1, ..., m-1
3. Compute mP
4. For j = 0, 1, ..., m-1:
   a. Compute Q - j(mP)
   b. If Q - j(mP) = iP for some stored baby step:
      Return k = i + jm

Time complexity: O(√n) group operations Space complexity: O(√n) group elements

5.1.2. Pollard's Rho Algorithm:

Pollard's rho uses pseudorandom walks to find collisions with minimal storage:

Algorithm 3: Pollard's Rho

Input: P, Q, n (where Q = kP)

Output: k

1. Define partition function f: G → G using random-walk steps
2. Initialize: $x_0 \leftarrow P$, $y_0 \leftarrow P$
3. Repeat:
   a. $x_{\{i+1\}} \leftarrow f(x_i)$
   b. $y_{\{i+1\}} \leftarrow f(f(y_i))$
   c. If $x_i = y_i$,collision found
4. Solve for k from collision relation

Time complexity: O(√n) expected group operations Space complexity: O(1) (constant storage)

The parallel version of Pollard's rho achieves near-linear speedup with multiple processors, making it the most practical generic attack for large-scale cryptanalysis.

Figure 4: Pollard's Rho Algorithm for ECDLP

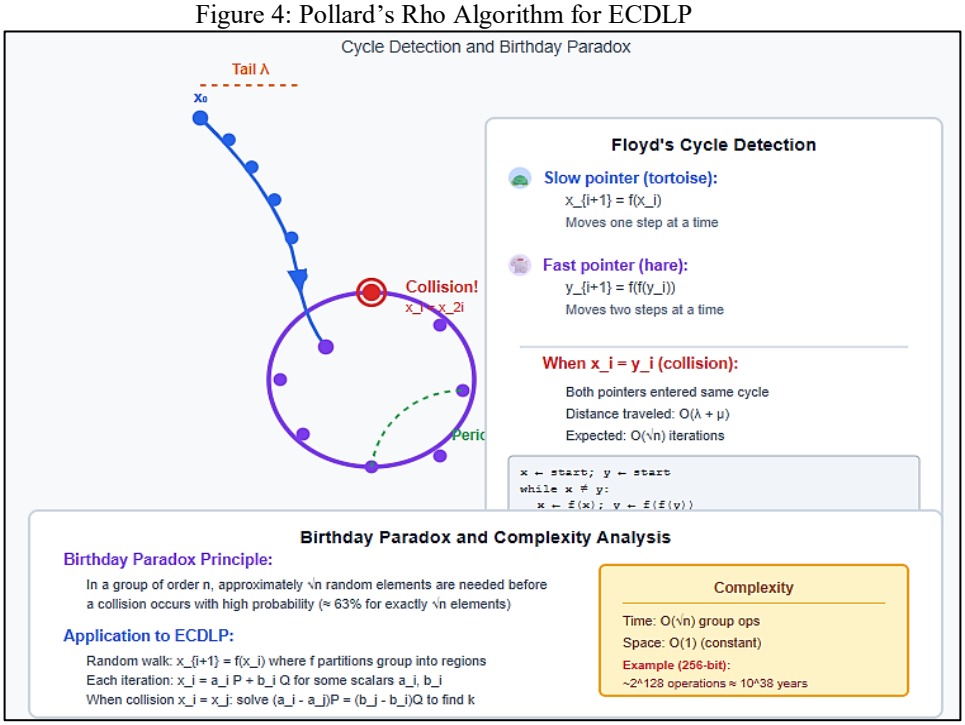

Figure 4 **:**Visualization of Pollard's rho algorithm showing the "rho-shaped" trajectory and cycle detection using Floyd's cycle-finding algorithm

5.2. Index Calculus Attempts

Index calculus has been extraordinarily successful for classical DLP in F_p^*, achieving $L_p[1/3]$ complexity. However, attempts to adapt this approach to elliptic curves have largely failed due to fundamental structural obstacles.

5.2.1 Classical Index Calculus (for reference):

The method relies on:

- Defining a factor base of "small" elements
- Finding relations among factor base elements
- Solving a sparse linear system
- Computing individual logarithms

5.2.2. Why Index Calculus Fails for ECDLP:

The critical obstacle is the absence of a natural notion of "small" or "smooth" points on elliptic curves. Unlike integers, which decompose into prime factors, elliptic curve points lack a canonical factorization structure. The group operation on $E(F_q)$ does not admit a smooth element base amenable to relation collection.

5.2.3. Gaudry's Index Calculus for Hyperelliptic Curves:

For hyperelliptic curves of genus $g \geq 3$ over small characteristic fields, Gaudry developed an index calculus variant with complexity O(q^(2-2/g)). However, this approach does not extend effectively to genus 1 curves (elliptic curves) with large q.

5.3. Specialized Attacks

5.3.1. MOV/Frey-Rück Attack:

For curves with small embedding degree k, the Weil or Tate pairing provides an efficiently computable homomorphism:

$$e: E(F_q)[n] \times E((F_{q^k})[n] \rightarrow F^*_{q^k}$$

Given Q = kP, we can compute:

$$e(Q, R) = e(kP, R) = e(P, R)^k$$

Solving DLP in F_q^k* yields k, reducing ECDLP to classical DLP. This attack is polynomial-time when k = O(log q).

5.3.2. SSSA Attack on Anomalous Curves:

For curves over F_p with #E(F_p) = p, the attack uses p-adic lifting:

Given Q = kP, lift points to E(Q_p) (curve over p-adic numbers). The formal group logarithm provides a group isomorphism:

$$\log: E_1(pZ_p) \rightarrow pZ_p$$

Computing logarithms modulo increasingly high powers of p eventually recovers k in polynomial time.

5.3.3. GHS Attack:

For certain curves over F_2^m, the GHS attack constructs a covering curve C (typically hyperelliptic) over a subfield such that ECDLP on E(F_2^m) reduces to DLP on the Jacobian of C, where index calculus variants may apply.

5.3.4. Fault Analysis and Side-Channel Attacks:

While not directly solving ECDLP, implementation attacks exploit physical characteristics:

- Timing attacks: Measuring computation time to infer secret bits
- Power analysis: Analyzing power consumption during scalar multiplication
- Fault injection: Inducing computational errors to reveal information

Countermeasures include constant-time implementations, point randomization, and anomaly detection.

5.4. Quantum Algorithms

Shor's quantum algorithm fundamentally threatens ECDLP security. Given a quantum computer with

sufficient qubits and coherence time, Shor's algorithm solves ECDLP in polynomial time $O((\log n)^3)$ using quantum Fourier transforms.

5.4.1. Shor's Algorithm for ECDLP:

- Prepare quantum superposition over scalars k
- Apply quantum scalar multiplication: $|k\rangle|0\rangle \rightarrow |k\rangle|kP\rangle$
- Quantum Fourier Transform on the first register
- Measure to obtain information about the order
- Classical post-processing recovers discrete logarithm

The algorithm requires O(log n) qubits and O((log n)³) quantum gates. Current quantum computers remain far from the scale required to threaten cryptographic instances (thousands of logical qubits after error correction), but the theoretical threat has motivated post-quantum cryptography research.

VI. CRYPTOGRAPHIC APPLICATIONS

6.1. Elliptic Curve Diffie-Hellman (ECDH)

ECDH extends the Diffie-Hellman key exchange protocol to elliptic curves:

6.1.1. Protocol:

- Public parameters: Elliptic curve E over F_q, base point P of prime order n
- Alice chooses private key a ∈ [1, n-1], computes public key A = aP
- Bob chooses private key b ∈ [1, n-1], computes public key B = bP
- Alice computes shared secret: $K_A = a(B) = abP$
- Bob computes shared secret: $K_B = b(A) = baP$
- Shared secret: $K = K_A = K_B$

Security relies on the computational Diffie-Hellman assumption: given P, aP, bP, computing abP is hard without knowing a or b.

Figure 5: Elliptic Curve Diffie-Hellman (ECDH) Key Exchange

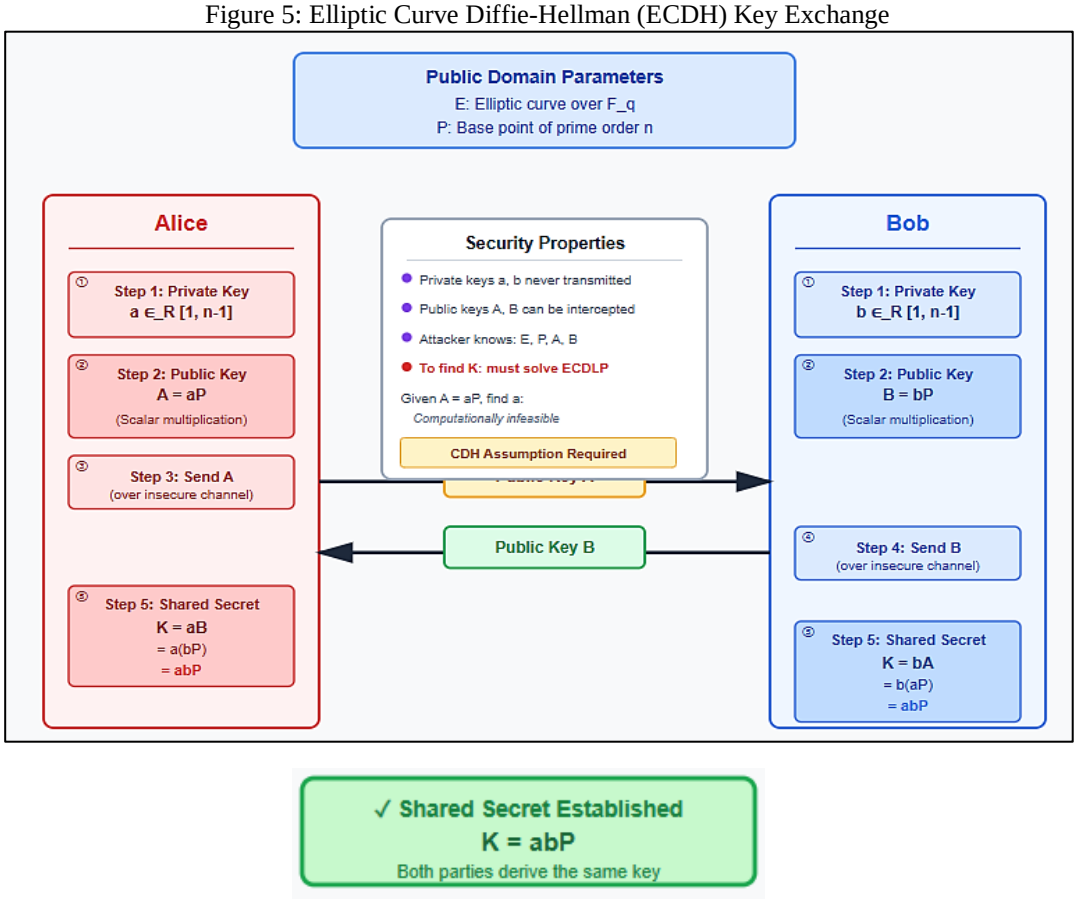

Figure 5 Protocol diagram illustrating ECDH key exchange between Alice and Bob, showing the flow of public key exchanges and shared secret computation

6.2. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA provides authentication and non-repudiation:

6.2.1. Signature Generation:

- Message m, private key d
- Hash h = H(m) where H is a cryptographic hash function
- Choose random k ∈ [1, n-1]

- Compute $(x_1, y_1) = kP$
- Compute $r = x_1 \bmod n$ (if $r = 0$, choose new k)
- Compute $s = k^{-1}(h + dr) \bmod n$ (if $s = 0$, choose new k)
- Signature: (r, s)

6.2.2. Signature Verification:

- Verify $r, s \in [1, n-1]$
- Compute $h = H(m)$
- Compute $u_1 = hs^{-1} \bmod n$, $u_2 = rs^{-1} \bmod n$
- Compute $(x_1, y_1) = u_1P + u_2Q$ (where $Q = dP$ is public key)
- Accept if $r \equiv x_1 \pmod n$

Security depends on both ECDLP hardness and hash function collision resistance. The randomness of k is critical—reusing k or using predictable k values completely breaks the scheme, as demonstrated by the 2010 PlayStation 3 signing key compromise.

6.3. Elliptic Curve Integrated Encryption Scheme (ECIES)

ECIES combines ECDH with symmetric encryption and MAC for public-key encryption:

6.3.1. Encryption (Alice encrypts to Bob):

- Generate ephemeral key pair (k, R = kP)
- Compute shared secret $S = k(Q_B)$ where Q_B is Bob's public key
- Derive encryption and MAC keys: $(K_E, K_M) = KDF(S)$
- Encrypt: $c = E_{K_E}(m)$
- Compute MAC: $MAC_{K_M}(c)$
- Ciphertext: (R, c, t)

6.3.2. Decryption (Bob decrypts):

- Compute S = d_B(R) using private key d_B
- Derive keys: $(K_E, K_M) = KDF(S)$
- Verify MAC: $MAC_{KM}(c) = t$
- Decrypt: $m = D_{K_E}(c)$

ECIES provides IND-CCA2 security under appropriate assumptions, making it suitable for general-purpose public-key encryption.

6.4. Performance Comparison

Table 1: Key Size Comparison:

Security Level	ECC (bits)	RSA (bits)	Ratio
80-bit	160	1024	1:6.4
112-bit	224	2048	1:9.1
128-bit	256	3072	1:12
192-bit	384	7680	1:20
256-bit	521	15360	1:29

6.4.1. Computational Performance:

On a modern processor (Intel Core i7, 3.4 GHz):

- ECDH key generation (P-256): ~0.8 ms
- ECDH shared secret: ~0.8 ms
- ECDSA signing (P-256): ~1.2 ms
- ECDSA verification (P-256): ~2.1 ms

Compare to RSA-2048:

- Key generation: ~50-100 ms (significantly slower)
- Signature generation: ~1.5 ms (comparable)
- Signature verification: ~0.1 ms (faster due to small public exponent)

ECC's efficiency advantage becomes more pronounced at higher security levels and in bandwidth-constrained environments.

VII. STANDARDIZATION AND CURVE SELECTION

7.1. NIST Recommended Curves

The National Institute of Standards and Technology (NIST) recommends five prime field curves (P-192, P-224, P-256, P-384, P-521) and five binary field curves, defined in FIPS 186-4. These curves use the equation:

$$y^2 = x^3 - 3x + b \pmod p$$

with specially chosen primes p and curve parameter b for computational efficiency. The coefficient a = -3 is selected to optimize point doubling operations.

7.1.1. P-256 Parameters (secp256r1):

- $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
- $a = -3$
- b = 0x5AC635D8AA3A93E7B3EBBD55769886BC651D06B0CC53B0F63BCE3C3E27D2604B
- Generator G with order n (prime)

7.2. Alternative Curves

Concerns about potential backdoors in NIST curves have motivated development of alternative standards:

7.2.1. Curve25519:

Developed by Daniel Bernstein, using Montgomery curve form:

$$By^2 = x^3 + Ax^2 + x$$

over F_p where $p = 2^{255} - 19$. Designed for high performance and side-channel resistance, widely deployed in TLS, Signal Protocol, and other applications.

7.2.2. Curve448 (Goldilocks):

A 448-bit curve offering 224-bit security, designed with similar principles to Curve25519 but providing higher security margin.

7.2.3. Brainpool Curves:

Specified by ECC Brainpool consortium, generated using verifiably random parameters to address transparency concerns. Uses fully random coefficients rather than special forms.

7.3. Security Considerations in Curve Selection

Cryptographically strong curves must satisfy:

- Large prime order subgroup: $\#E(F_q) = hn$ where n is prime and h is small (typically $h \in \{1, 2, 4, 8\}$)
- Resistance to MOV attack: Large embedding degree $k \geq (\log q) / 6$
- Resistance to anomalous curve attacks: $\#E(F_q) \neq q$
- Twist security: The quadratic twist $\tilde{E}$ should also have large prime order, protecting against invalid curve attacks
- Transfer to weak curves: No efficiently computable isogenies to curves with weaker security properties
- Constant-time implementation: Curve arithmetic should permit constant-time implementations resistant to timing attacks
- Efficient arithmetic: Parameters chosen to optimize field arithmetic and point operations

VIII. THEORETICAL ADVANCES AND OPEN PROBLEMS

8.1. Lower Bounds on ECDLP Complexity

Establishing rigorous lower bounds for ECDLP remains an open problem. While generic algorithms require $\Omega(\sqrt{n})$ group operations, proving that no substantially faster algorithm exists for general elliptic curves remains beyond current theoretical techniques.

8.1.1. Generic Group Model:

Shoup proved that any generic algorithm for discrete logarithm in a group of order n requires $\Omega(\sqrt{n})$ group operations. This provides conditional security assuming the attacker treats the group as a black box without exploiting specific structural properties.

8.1.2. Algebraic Group Model:

Recent work explores the algebraic group model, an intermediate notion between generic groups and standard complexity assumptions, providing stronger security arguments for certain protocols.

8.2. Isogeny-Based Cryptography

Supersingular isogeny cryptography represents an alternative approach using isogenies (rational maps between elliptic curves) rather than point multiplication:

8.2.1. SIDH/SIKE Protocol:

The Supersingular Isogeny Diffie-Hellman protocol achieves post-quantum security by replacing scalar multiplication with isogeny computation. However, the 2022 cryptanalysis by Castryck and Decru broke SIKE using higher-dimensional isogenies, demonstrating the nascent nature of this field.

8.3. Quantum-Resistant Variants

While Shor's algorithm breaks ECDLP, research explores quantum-resistant constructions:

8.3.1. Hash-Based Signatures:

Schemes like SPHINCS+ use hash functions rather than number-theoretic problems, providing quantum resistance at the cost of larger signatures.

8.3.2. Lattice-Based Cryptography:

Post-quantum schemes based on Learning With Errors (LWE) and related lattice problems offer quantum resistance with reasonable efficiency.

8.3.3. Code-Based Cryptography:

McEliece cryptosystem and variants resist known quantum attacks but require large public keys.

IX. CONCLUSION

9.1. Summary of Findings

This paper has presented a comprehensive algebraic analysis of Elliptic Curve Cryptography and the Discrete Logarithm Problem. We established rigorous mathematical foundations, demonstrating how the group-theoretic properties of elliptic curves over finite fields enable efficient cryptographic constructions while resisting cryptanalytic attacks.

The computational intractability of ECDLP, requiring exponential time $O(\sqrt{n})$ with best known algorithms compared to polynomial-time scalar multiplication, provides the asymmetry essential for public-key cryptography. The absence of sub-exponential algorithms for properly chosen elliptic curves—contrasting sharply with classical DLP vulnerability to index calculus—establishes ECC as the most efficient classical cryptographic paradigm.

Our analysis of cryptanalytic algorithms confirmed that generic methods (Pollard's rho, Baby-step Giant-step) represent the practical threat to ECC, while specialized attacks (MOV, SSSA, GHS) apply only to weak curve classes excluded by cryptographic standards. The requirement for careful parameter selection and the existence of weak curves underscore the importance of rigorous standards adherence in cryptographic implementations.

9.2. Practical Implications

ECC has become the dominant public-key cryptographic technology for modern applications:

- Efficiency advantage: 256-bit ECC provides security equivalent to 3072-bit RSA with dramatically reduced computational and bandwidth requirements
- Mobile and IoT deployment: Reduced resource requirements make ECC ideal for constrained environments
- Protocol integration: ECDH and ECDSA are widely deployed in TLS 1.3, secure messaging (Signal Protocol), blockchain systems (Bitcoin, Ethereum), and certificate infrastructure
- Implementation considerations: Side-channel resistance, constant-time implementations, and proper randomness generation are critical for security

- Standards evolution: Transition from NIST curves to alternatives like Curve25519 reflects concerns about transparency and implementation efficiency

9.3. Future Research Directions

Several critical research areas merit continued investigation:

9.3.1. Post-Quantum Transition:

The development of practical quantum computers represents an existential threat to ECC. Research priorities include:

- Developing efficient post-quantum alternatives
- Hybrid classical-quantum schemes for transition period
- Quantum-resistant protocol designs maintaining ECC's efficiency advantages

9.3.2. Advanced Cryptanalysis:

Theoretical advances may reveal new vulnerabilities:

- Rigorous complexity lower bounds for ECDLP
- Understanding structural properties that might enable faster algorithms
- Quantum algorithms intermediate between classical and Shor's complexity

9.3.3. Novel Curve Constructions:

Exploration of alternative curve families:

- Curves with special endomorphism structures for accelerated arithmetic
- Higher-genus curves balancing security and efficiency
- Curves optimized for specific applications or zero-knowledge proofs

9.3.4. Implementation Security:

Continued development of side-channel resistant implementations:

- Formally verified cryptographic libraries
- Hardware-accelerated secure implementations
- Fault injection countermeasures

9.4. Concluding Remarks

Elliptic Curve Cryptography represents one of the most significant achievements in modern cryptography, demonstrating how deep mathematical structures can be leveraged for practical security. The elegance of the underlying algebra—abelian groups with rich structure yet resisting efficient algorithmic exploitation—provides both theoretical beauty and engineering utility.

As digital infrastructure continues its exponential growth, ECC's efficiency advantages become increasingly critical. However, the quantum computing threat ensures that ECC cannot be viewed as a permanent solution. The transition to post-quantum cryptography will be one of the defining challenges for cryptographic research over the coming decades.

Understanding the algebraic foundations of ECC and ECDLP remains essential not only for current system security but also for developing the next generation of cryptographic technologies. The mathematical principles explored in this paper—group theory, finite field arithmetic, computational complexity—will continue to guide cryptographic innovation as we navigate the transition to quantum-resistant security.

REFERENCES

1. Koblitz N. 1987. Elliptic curve cryptosystems. Mathematics of Computation. 48(177):203–209.
2. Miller VS. 1986. Use of elliptic curves in cryptography. In: Williams HC, editor. Advances in Cryptology—CRYPTO '85 Proceedings. (Lecture Notes in Computer Science; vol. 218). Berlin (Germany): Springer-Verlag. p. 417–426.
3. Hankerson D, Menezes AJ, Vanstone S. 2004. Guide to Elliptic Curve Cryptography. New York (NY): Springer-Verlag.
4. Washington LC. 2008. Elliptic Curves: Number Theory and Cryptography. 2nd ed. Boca Raton (FL): Chapman & Hall/CRC.
5. Silverman JH. 2009. The Arithmetic of Elliptic Curves. 2nd ed. (Graduate Texts in Mathematics; vol. 106). New York (NY): Springer.
6. Blake I, Seroussi G, Smart N. 2005. Advances in Elliptic Curve Cryptography. (London Mathematical Society Lecture Note Series; vol. 317). Cambridge (UK): Cambridge University Press.
7. Menezes AJ, van Oorschot PC, Vanstone SA. 1996. Handbook of Applied Cryptography. Boca Raton (FL): CRC Press.
8. Stinson DR, Paterson MB. 2018. Cryptography: Theory and Practice. 4th ed. Boca Raton (FL): CRC Press.

9. Pollard JM. 1978. Monte Carlo methods for index computation (mod p). Mathematics of Computation. 32(143):918–924.
10. Shoup V. 1997. Lower bounds for discrete logarithms and related problems. In: Fumy W, editor. Advances in Cryptology—EUROCRYPT '97. (Lecture Notes in Computer Science; vol. 1233). Berlin (Germany): Springer. p. 256–266.
11. Shor PW. 1997. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing. 26(5):1484–1509.
12. Joux A, Lercier R. 2002. The function field sieve is quite special. In: Fieker C, Kohel DR, editors. Algorithmic Number Theory. (Lecture Notes in Computer Science; vol. 2369). Berlin (Germany): Springer. p. 431–445.
13. Gaudry P. 2009. Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem. Journal of Symbolic Computation. 44(12):1690–1702.
14. Menezes A, Okamoto T, Vanstone S. 1993. Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory. 39(5):1639–1646.
15. Satoh T, Araki K. 1998. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. Commentarii Mathematici Universitatis Sancti Pauli. 47:81–92.
16. Semaev IA. 1998. Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p. Mathematics of Computation. 67(221):353–356.
17. Smart NP. 1999. The discrete logarithm problem on elliptic curves of trace one. Journal of Cryptology. 12(3):193–196.
18. Gaudry P, Hess F, Smart NP. 2002. Constructive and destructive facets of Weil descent on elliptic curves. Journal of Cryptology. 15(1):19–46.
19. Bernstein DJ. 2006. Curve25519: New Diffie-Hellman speed records. In: Yung M, Dodis Y, Kiayias A, Malkin T, editors. Public Key Cryptography—PKC 2006. (Lecture Notes in Computer Science; vol. 3958). Berlin (Germany): Springer. p. 207–228.
20. National Institute of Standards and Technology. 2013. Digital Signature Standard (DSS). Federal Information Processing Standards Publication 186-4.
21. Johnson D, Menezes A, Vanstone S. 2001. The elliptic curve digital signature algorithm (ECDSA). International Journal of Information Security. 1(1):36–63.
22. Shoup V. 2001. A proposal for an ISO standard for public key encryption. IACR Cryptology ePrint Archive. Report 2001/112 [Internet]. [cited 2025 Nov 3]. Available from: https://eprint.iacr.org/2001/112
23. Elgamal T. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory. 31(4):469–472.
24. Schoof R. 1985. Elliptic curves over finite fields and the computation of square roots mod p. Mathematics of Computation. 44(170):483–494.
25. Schoof R. 1995. Counting points on elliptic curves over finite fields. Journal de Théorie des Nombres de Bordeaux. 7(1):219–254.
26. Atkin AOL, Morain F. 1993. Elliptic curves and primality proving. Mathematics of Computation. 61(203):29–68.
27. Mestre JF. 1986. La méthode des graphes. Exemples et applications. In: Proceedings of the International Conference on Class Numbers and Fundamental Units of Algebraic Number Fields; 1986 Jun; Katata (Japan). p. 217–242.
28. Elkies ND. 1992. Explicit isogenies. Manuscript. Boston University.
29. Möller B. 2001. Algorithms for multi-exponentiation. In: Vaudenay S, Youssef AM, editors. Selected Areas in Cryptography. (Lecture Notes in Computer Science; vol. 2259). Berlin (Germany): Springer. p. 165–180.
30. Gordon DM. 1998. A survey of fast exponentiation methods. Journal of Algorithms. 27(1):129–146.