The OSCAR PUBLISHING Services
American Journal of Applied Science and Technology

VOLUME Vol.06 Issue 04 2026
PAGE NO. 214-222
DOI 10.37547/ajast/Volume06Issue04-29

Mathematical Foundations of Elliptic Curve Primality Tests

Toshboyeva Feruza
Department of Higher and Applied Mathematics, Tashkent State University of Economics, Uzbekistan

Received: 18 March 2026; **Accepted:** 06 April 2026; **Published:** 30 April 2026

Abstract: Primality testing is a fundamental problem in computational number theory and modern cryptography, where the efficient generation and verification of large prime numbers are essential for the security of public-key cryptosystems. Among the numerous primality testing techniques, elliptic curve-based methods have emerged as one of the most powerful approaches due to their strong mathematical foundation and practical efficiency. This paper presents the mathematical foundations of elliptic curve primality tests by reviewing the algebraic structure of elliptic curves over finite fields, the group law, Hasse's theorem, and the properties of the Frobenius endomorphism that underpin these algorithms. The study examines the principles of major elliptic curve primality proving methods, including the Goldwasser–Kilian algorithm and the Atkin–Morain Elliptic Curve Primality Proving (ECPP) algorithm. Furthermore, elliptic curve-based techniques are compared with classical primality tests such as the Fermat, Solovay–Strassen, Miller–Rabin, and AKS algorithms in terms of computational complexity, accuracy, and practical applicability. The analysis demonstrates that elliptic curve methods provide a mathematically rigorous and computationally efficient framework for proving the primality of large integers, making them particularly suitable for cryptographic applications requiring certified prime numbers. The presented mathematical framework serves as a comprehensive reference for understanding the theoretical principles and practical significance of elliptic curve primality tests.

Keywords: Elliptic curves; Primality testing; Elliptic Curve Primality Proving (ECPP); Finite fields; Goldwasser–Kilian algorithm; Hasse theorem; Frobenius endomorphism; Computational number theory; Cryptography; Prime numbers.

INTRODUCTION:

Prime numbers constitute one of the fundamental objects of number theory and play an indispensable role in modern public-key cryptography. The security of widely deployed cryptographic systems, including RSA, Diffie–Hellman key exchange, and several digital signature schemes, depends on the generation and verification of sufficiently large prime numbers. As cryptographic key sizes continue to increase, efficient and mathematically rigorous primality testing algorithms have become an essential component of secure cryptographic infrastructures [1], [2].

Primality testing has been an active area of research for several decades. Early deterministic methods, such as trial division and Lucas-based tests, were computationally expensive for large integers and therefore unsuitable for practical cryptographic applications. The introduction of probabilistic algorithms, including the Fermat, Solovay–Strassen, and Miller–Rabin tests, significantly reduced computational complexity while providing a negligible probability of error [3], [4]. These algorithms remain among the most widely used methods for rapid primality testing because of their simplicity and efficiency.

Although probabilistic algorithms are highly efficient, they cannot provide an unconditional mathematical proof of primality. For applications requiring certified prime numbers, deterministic primality proving

algorithms are preferred. A major breakthrough was achieved with the development of the Agrawal–Kayal–Saxena (AKS) algorithm, which established that primality testing belongs to the complexity class P by providing the first deterministic polynomial-time primality test [5]. Despite its theoretical significance, the AKS algorithm is generally less efficient than practical primality proving methods when applied to cryptographic-size integers.

Elliptic curves have transformed computational number theory by providing powerful algebraic structures that can be exploited in various computational problems. Originally introduced in algebraic geometry, elliptic curves have found extensive applications in integer factorization, discrete logarithm problems, cryptographic protocol design, digital signatures, key exchange mechanisms, and primality proving [6], [7]. Their rich algebraic properties enable efficient arithmetic over finite fields while preserving strong mathematical guarantees.

The application of elliptic curves to primality testing was pioneered by Goldwasser and Kilian, who proposed a randomized primality proving algorithm based on the arithmetic of elliptic curves over finite fields [8]. Their work demonstrated that elliptic curves provide an elegant framework for constructing primality certificates whose correctness can be verified efficiently. This pioneering contribution opened a new research direction in computational number theory and inspired the development of more practical algorithms.

Subsequently, Atkin and Morain introduced the Elliptic Curve Primality Proving (ECPP) algorithm, which substantially improved the practical performance of elliptic curve-based primality proving while preserving rigorous mathematical correctness [9]. ECPP is currently regarded as one of the fastest and most reliable algorithms for proving the primality of very large integers and has become the standard approach in numerous computational number theory software packages.

The mathematical foundation of elliptic curve primality tests relies on several fundamental concepts, including finite field arithmetic, elliptic curve groups, Hasse's theorem, Frobenius endomorphisms, and point-counting algorithms [6], [10]. These mathematical tools establish the theoretical basis for constructing efficient primality certificates and analyzing the complexity of elliptic curve algorithms. Furthermore, advances in Schoof's point-counting algorithm and its improvements by Elkies and Atkin have significantly strengthened the computational framework supporting elliptic curve methods [11].

Compared with classical primality tests, elliptic curve-based methods offer an attractive balance between mathematical rigor and computational efficiency. While probabilistic tests are extremely fast and deterministic polynomial-time algorithms provide unconditional correctness, elliptic curve primality proving combines practical efficiency with rigorous mathematical certification, making it particularly suitable for cryptographic applications involving large integers [5], [9], [12].

The objective of this paper is to present the mathematical foundations underlying elliptic curve primality tests. The paper reviews the algebraic properties of elliptic curves over finite fields, discusses the principal mathematical theorems supporting elliptic curve algorithms, analyzes the Goldwasser–Kilian and Atkin–Morain primality proving methods, and compares elliptic curve techniques with classical primality testing algorithms in terms of theoretical foundations, computational complexity, and practical applicability in modern cryptography.

MATHEMATICAL BACKGROUND

1. Finite Fields

Finite fields, also referred to as Galois fields, provide the algebraic foundation for elliptic curve arithmetic. A finite field contains a finite number of elements and exists only when its order is equal to a prime power $q = p^m$, where p is a prime number and $m \geq 1$. In elliptic curve primality tests, computations are predominantly performed over prime fields, since they offer a simple yet powerful mathematical framework for defining elliptic curve groups [1], [6], [10].

A prime finite field is denoted by

$$\mathbb{F}_p = \{0,1,\dots,p-1\},$$

where all arithmetic operations are performed modulo the prime p. Addition, subtraction, multiplication, and division (except by zero) satisfy the field axioms, ensuring closure, associativity, commutativity, distributivity, and the existence of additive and multiplicative inverses [1], [10].

For any two elements $a, b \in \mathbb{F}_p$,

$$a + b \equiv (a + b) \ (\mathrm{mod} p),$$
$$a - b \equiv (a - b) \ (\mathrm{mod} p),$$
$$ab \equiv ab \ (\mathrm{mod} p).$$

Every nonzero element possesses a unique multiplicative inverse satisfying

$$aa^{-1} \equiv 1 \ (\mathrm{mod} p).$$

Using Fermat's Little Theorem, the inverse can be computed efficiently as

$$a^{-1} \equiv a^{p-2} \ (\mathrm{mod} p),$$

for every $a \neq 0$ [1].

The existence of multiplicative inverses is one of the principal reasons why elliptic curves are defined over finite fields rather than arbitrary modular rings. The point addition formulas involve division by coordinates differences, and these divisions are well-defined only when the underlying arithmetic forms a field. Consequently, finite fields provide the algebraic environment required for defining elliptic curve groups and for ensuring the correctness of elliptic curve primality proving algorithms [6], [10].

2. Elliptic Curves over Finite Fields

An elliptic curve is a smooth algebraic curve of genus one equipped with a distinguished point at infinity. Over a finite field $\mathbb{F}_p$, where $p > 3$, an elliptic curve is commonly represented by the short Weierstrass equation

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

where $a, b \in \mathbb{F}_p$.

Not every cubic equation defines an elliptic curve. To guarantee smoothness, the discriminant must be nonzero. Equivalently, the coefficients must satisfy

$$4a^3 + 27b^2 \not\equiv 0 \ (\mathrm{mod} p).$$

The discriminant of the curve is

$$\Delta = -16(4a^3 + 27b^2),$$

and the condition

$$\Delta \neq 0$$

ensures that the curve contains neither singular points nor self-intersections [6], [10].

The set of rational points on the elliptic curve is defined as

$$E(\mathbb{F}_p) = \{(x, y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\},$$

where $\mathcal{O}$ denotes the point at infinity. This additional point plays the role of the identity element in the elliptic curve group.

Unlike many algebraic curves, elliptic curves possess a natural group structure defined through geometric point addition. This remarkable property enables arithmetic operations to be performed directly on curve points and serves as the mathematical basis for numerous applications in computational number theory and cryptography, including elliptic curve cryptography (ECC), integer factorization, digital signatures, key exchange protocols, and primality proving [6], [7].

The number of rational points on an elliptic curve depends on both the defining coefficients and the underlying finite field. If the field changes, the corresponding set of rational points also changes. This dependence is of particular importance in elliptic curve primality tests because the order of the elliptic curve group determines whether the constructed curve satisfies the conditions required for generating a valid primality certificate [8], [9].

Therefore, the study of elliptic curves over finite fields provides the essential mathematical framework upon which modern elliptic curve primality proving algorithms are built. The algebraic properties of these curves, together with their finite abelian group structure, enable the construction of efficient algorithms whose correctness is supported by deep results from algebraic geometry and computational number theory [7], [10], [12].

3. Group Law on Elliptic Curves

The algebraic structure of an elliptic curve is established through a binary operation known as point addition, under which the set of points on the curve forms a finite abelian group. This group structure distinguishes elliptic curves from other algebraic curves and constitutes the mathematical foundation of elliptic curve cryptography and primality proving algorithms [6], [7], [10].

Let

$$\mathrm{P} = (\mathrm{x}_1, \mathrm{y}_1), \mathrm{Q} = (\mathrm{x}_2, \mathrm{y}_2)$$

be two distinct points on an elliptic curve $\mathrm{E}(\mathbb{F}_\mathrm{p})$. If $\mathrm{P} \neq \mathrm{Q}$ and $\mathrm{x}_1 \neq \mathrm{x}_2$, the slope of the secant line passing through the two points is

$$\lambda = \frac{\mathrm{y}_2 - \mathrm{y}_1}{\mathrm{x}_2 - \mathrm{x}_1} \ (\mathrm{modp}).$$

The coordinates of the resulting point

$$\mathrm{R} = \mathrm{P} + \mathrm{Q} = (\mathrm{x}_3, \mathrm{y}_3)$$

are computed as

$$\mathrm{x}_3 = \lambda^2 - \mathrm{x}_1 - \mathrm{x}_2 \ (\mathrm{modp}),$$
$$\mathrm{y}_3 = \lambda(\mathrm{x}_1 - \mathrm{x}_3) - \mathrm{y}_1 \ (\mathrm{modp})$$

When $P = Q$, the tangent line at P replaces the secant line, and the slope becomes

$$\lambda = \frac{3x_1^2 + a}{2y_1} \ (modp).$$

The corresponding point doubling formulas are

$$x_3 = \lambda^2 - 2x_1 \ (modp),$$
$$y_3 = \lambda(x_1 - x_3) - y_1 \ (modp).$$

For every point $P = (x, y)$, there exists an inverse point

$$-P = (x, -y),$$

which satisfies

$$P + (-P) = \mathcal{O},$$

where $\mathcal{O}$ denotes the point at infinity, serving as the identity element of the group. Consequently, the set $E(\mathbb{F}_p)$ satisfies the axioms of closure, associativity, commutativity, identity, and inverses, thereby forming a finite abelian group [6], [10].

Another fundamental operation is scalar multiplication, defined as repeated point addition:

$$kP = \underbrace{P + P + \cdots + P}_{k \text{ times}},$$

where k is a positive integer. Rather than performing repeated additions directly, scalar multiplication is efficiently implemented using algorithms such as the double-and-add method, whose computational complexity is $O(log\, k)$. This operation represents the computational core of elliptic curve cryptography as well as elliptic curve primality proving algorithms [7].

4. Number of Rational Points and Hasse's Theorem

The order of an elliptic curve over a finite field, denoted by

$$\#E(\mathbb{F}_p),$$

is defined as the total number of rational points on the curve, including the point at infinity. Determining this quantity is one of the central problems in computational number theory because the correctness and efficiency of elliptic curve primality tests rely heavily on the group order [8], [9].

A fundamental result governing the number of rational points is Hasse's theorem, which states that

$$\left|\#E(\mathbb{F}_p) - (p+1)\right| \leq 2\sqrt{p}.$$

Defining the trace of Frobenius by

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

the theorem can be expressed equivalently as

$$|t| \leq 2\sqrt{p}$$

This theorem demonstrates that the number of points on an elliptic curve always lies within a narrow interval centered at $p + 1$. The result provides an essential theoretical bound that makes efficient point-counting algorithms possible and forms one of the principal mathematical foundations of elliptic curve primality proving [10], [11].

5. Frobenius Endomorphism

The Frobenius endomorphism is one of the most important algebraic mappings associated with elliptic curves over finite fields. For an elliptic curve defined over $\mathbb{F}_p$, it is defined as

$$\pi(x, y) = (x^p, y^p)$$

Since every element of the finite field satisfies

$$a^p = a$$

the Frobenius mapping preserves the structure of the curve while encoding significant arithmetic information.

The Frobenius endomorphism satisfies the characteristic equation

$$\pi^2 - t\pi + p = 0$$

where t denotes the trace of Frobenius. Consequently,

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

The trace t therefore completely determines the order of the elliptic curve group. This relationship is fundamental to Schoof's point-counting algorithm and its improvements, including the Schoof–Elkies–

Atkin (SEA) algorithm, which are widely employed in modern implementations of elliptic curve primality proving [10], [11].

6. Point Counting Algorithms

Efficient determination of the group order is a prerequisite for elliptic curve primality proving. A naive enumeration of all points requires $O(p)$ arithmetic operations and is therefore impractical for large prime candidates encountered in cryptographic applications.

Schoof introduced the first deterministic polynomial-time algorithm for computing the order of elliptic curves over finite fields by evaluating the trace of Frobenius modulo several small primes and reconstructing the exact value using the Chinese Remainder Theorem [11]. Although theoretically efficient, the original algorithm incurs considerable computational overhead.

Subsequent improvements by Elkies and Atkin resulted in the Schoof-Elkies-Atkin (SEA) algorithm, which significantly accelerates point counting through the use of Elkies primes. Today, the SEA algorithm represents the standard method for computing elliptic curve orders and serves as a key component in practical implementations of elliptic curve primality proving systems [7], [10].
2.7. Mathematical Principle of Elliptic Curve Primality Tests

The mathematical principle underlying elliptic curve primality tests is based on the observation that if an integer n is prime, then arithmetic modulo n forms a finite field, and the associated elliptic curve possesses a well-defined finite abelian group satisfying Hasse's theorem. In contrast, if n is composite, the corresponding algebraic structure generally fails to satisfy these group-theoretic properties because arithmetic modulo a composite integer is no longer performed over a field [8], [9].

Elliptic curve primality proving algorithms exploit this distinction by constructing an elliptic curve whose group order contains a sufficiently large prime factor. The primality of the candidate integer is then established through the verification of specific scalar multiplication identities involving carefully selected points on the curve. The validity of these identities provides a mathematically rigorous certificate of primality that can be verified efficiently. This principle constitutes the theoretical foundation of the Goldwasser-Kilian algorithm and the Atkin-Morain Elliptic Curve Primality Proving (ECPP) algorithm, which are discussed in the following section [8], [9].

ELLIPTIC CURVE PRIMALITY TESTS

Elliptic curve primality tests establish the primality of an integer by exploiting the algebraic properties of elliptic curves defined over finite fields. Unlike classical primality tests, which are primarily based on modular exponentiation and congruence relations, elliptic curve methods utilize the group structure of elliptic curves to construct rigorous certificates of primality. These methods combine deep results from algebraic geometry with computational number theory, providing both mathematical certainty and practical efficiency for large integers [6], [8], [9].

The central idea underlying elliptic curve primality proving is based on the following observation. If an integer n is prime, then arithmetic modulo n defines the finite field $\mathbb{F}_n$, and every nonsingular elliptic curve over this field forms a finite abelian group. The order of this group satisfies Hasse's theorem,

$$|\#E(\mathbb{F}_n) - (n+1)| \leq 2\sqrt{n}.$$

Consequently, the group order lies within a relatively small interval around $n+1$, making it possible to identify large prime divisors that can be used to certify primality [10], [11].

Suppose that an elliptic curve E is constructed over the ring $\mathbb{Z}_n$, where n is the integer to be tested. Let

$$m = \#E(\mathbb{F}_n)$$

denote the expected order of the elliptic curve group. If a sufficiently large prime divisor q of m can be identified together with a point

$$\mathrm{P} \in \mathrm{E}(\mathbb{Z}_\mathrm{n}),$$

such that

$$\mathrm{mP} = \mathcal{O},$$

while

$$\frac{\mathrm{m}}{\mathrm{q}}\mathrm{P} \neq \mathcal{O},$$

then the point P has order divisible by q. Under appropriate size conditions on q, this property implies that n must be prime. If n were composite, the required group structure would generally fail to exist,

and the above relations would not hold simultaneously [8], [9].

Compared with classical primality tests, elliptic curve methods possess several important advantages. First, they generate explicit primality certificates that can be verified independently with relatively small computational effort. Second, their practical running time is considerably better than that of general deterministic algorithms such as AKS for integers encountered in cryptographic applications. Finally, the probability of incorrectly proving a composite integer to be prime is mathematically eliminated once a valid certificate has been constructed [5], [9].

Several elliptic curve primality tests have been proposed during the past four decades. Among these, two algorithms have had the greatest theoretical and practical impact. The first is the Goldwasser-Kilian algorithm, which introduced the concept of primality proving using elliptic curves and established the theoretical foundation of the field [8]. The second is the Atkin-Morain Elliptic Curve Primality Proving (ECPP) algorithm, which significantly improved computational efficiency and has become the standard algorithm implemented in modern computational number theory software [9].

Although both algorithms rely on the arithmetic of elliptic curves, their approaches to curve construction and point-counting differ substantially. Goldwasser-Kilian constructs random elliptic curves and determines their group orders through point-counting techniques, whereas ECPP employs complex multiplication theory to generate curves whose orders are known in advance. This distinction enables ECPP to achieve considerably better practical performance while preserving rigorous mathematical correctness [9], [10].

The following subsections present the mathematical principles and algorithmic procedures of the Goldwasser-Kilian primality test and the Atkin-Morain ECPP algorithm, followed by an analysis of their computational complexity and practical performance.

1. Goldwasser-Kilian Primality Test

The first practical application of elliptic curves to primality proving was proposed by Goldwasser and Kilian in 1986 [8]. Their algorithm demonstrated that elliptic curves could be used to generate efficiently verifiable certificates of primality. This work established the theoretical foundation of elliptic curve primality proving and introduced a fundamentally different approach from classical congruence-based primality tests.

The algorithm considers an odd integer $n > 2$ whose primality is to be determined. Instead of relying on modular exponentiation, an elliptic curve is constructed over the ring

$$\mathbb{Z}_n.$$

Random coefficients $a, b \in \mathbb{Z}_n$ are selected, and the elliptic curve

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

is defined, provided that the discriminant

$$\Delta = -16(4a^3 + 27b^2)$$

satisfies

$$\gcd(\Delta, n) = 1.$$

This condition guarantees that the curve is nonsingular whenever n is prime [6], [8].

After constructing the elliptic curve, the next objective is to determine its group order $m = \#E(\mathbb{F}_n)$.

According to Hasse's theorem, $|m - (n+1)| \leq 2\sqrt{n}$, which considerably restricts the possible values of m [10]. The order m is then factored as $m = qr$ where q is a sufficiently large prime factor and r is the remaining cofactor.
A random point $P \in E(\mathbb{Z}_n)$ is selected, and two scalar multiplications are performed.
First, $mP = \mathcal{O}$ must hold. Second, $rP \neq \mathcal{O}$
Equivalently, $\frac{m}{q}P \neq \mathcal{O}$.

These conditions imply that the order of P is divisible by the prime factor q. If $q > (\sqrt[4]{n} + 1)^2$ then the existence of such a point guarantees that n is prime [8].
The correctness of the algorithm follows from the algebraic structure of elliptic curve groups. If n were composite, arithmetic modulo n would not define a field, and the necessary group-theoretic properties would generally fail. Consequently, the probability of simultaneously satisfying all required conditions for a composite integer is negligible. Once the recursive verification of the large prime factor q is completed, a complete certificate of primality is obtained [8], [9].

The Goldwasser-Kilian algorithm can be summarized as follows.

Algorithm 1. Goldwasser-Kilian Primality Test

Input: An odd integer $n > 2$.

Output: Prime or Composite.

Step 1. Select random coefficients $a, b \in \mathbb{Z}_n$.

Step 2. Compute $\Delta = -16(4a^3 + 27b^2)$. If $\gcd(\Delta, n) \neq 1$, terminate and declare Composite.

Step 3. Construct the elliptic curve $E: y^2 = x^3 + ax + b$.

Step 4. Determine the group order $m = \#E(\mathbb{F}_n)$.

Step 5. Factor $m = qr$, where q is a large prime divisor.

Step 6. Select a random point $P \in E(\mathbb{Z}_n)$.

Step 7. Verify $mP = \mathcal{O}$. If the equality fails, choose another curve.

Step 8. Verify $rP \neq \mathcal{O}$. If this condition fails, select another point or another elliptic curve.

Step 9. Recursively prove the primality of the prime factor q.

Step 10. If all recursive verifications succeed, output Prime.

The expected running time of the Goldwasser-Kilian algorithm is polynomial under heuristic assumptions, making it one of the first practical algorithms capable of generating rigorous primality certificates. Nevertheless, its efficiency is largely determined by the difficulty of computing the elliptic curve order and by the probability of finding suitable curves satisfying the required conditions. These limitations motivated the development of more efficient methods, most notably the Atkin-Morain Elliptic Curve Primality Proving (ECPP) algorithm, which substantially improves the practical performance while preserving the same mathematical guarantees [8], [9].

2. Atkin-Morain Elliptic Curve Primality Proving (ECPP)

To improve the practical efficiency of elliptic curve primality proving, Atkin and Morain introduced the Elliptic Curve Primality Proving (ECPP) algorithm in 1993 [9]. Unlike the Goldwasser-Kilian algorithm, which requires explicit point counting on randomly generated elliptic curves, ECPP employs the theory of complex multiplication (CM) to construct elliptic curves whose group orders are known in advance. This significantly reduces the computational cost of the primality proving process.

Given an integer n, ECPP first selects a suitable discriminant D and computes the corresponding Hilbert class polynomial. Using complex multiplication theory, an elliptic curve $E: y^2 = x^3 + ax + b$ is constructed over $\mathbb{Z}_n$ with a predetermined group order $m = \#E(\mathbb{F}_n)$.

The order is then expressed as $m = qr$, where q is a large prime factor. A point $P \in E(\mathbb{Z}_n)$ is chosen, and the following conditions are verified: $mP = \mathcal{O}$, and $rP \neq \mathcal{O}$. If these conditions are satisfied and the primality of q is established recursively, then n is proved to be prime [9].

COMPARISON OF CLASSICAL AND ELLIPTIC CURVE PRIMALITY TESTS

Primality testing algorithms differ significantly in their mathematical foundations, computational complexity, and practical performance. Classical algorithms such as the Fermat, Solovay–Strassen, and Miller–Rabin tests rely on modular arithmetic and probabilistic principles, whereas elliptic curve methods exploit the algebraic structure of elliptic curve groups over finite fields. Deterministic algorithms, including AKS and ECPP, provide mathematically rigorous proofs of primality but employ fundamentally different approaches [3]–[10].

The Fermat primality test is computationally efficient and easy to implement; however, it is vulnerable to Carmichael numbers, which satisfy Fermat's congruence despite being composite [3]. The Solovay–Strassen algorithm improves reliability by incorporating the Jacobi symbol and Euler's criterion, thereby reducing the probability of incorrectly classifying composite numbers [4]. The Miller–Rabin test further enhances efficiency and accuracy and remains one of the most widely used probabilistic primality tests in cryptographic applications because the probability of error decreases exponentially with the number of independent rounds [1], [2].

The AKS algorithm represented a major theoretical breakthrough by proving that primality testing belongs to the complexity class **P**. Although it guarantees deterministic polynomial-time execution,

its practical performance is generally inferior to probabilistic and elliptic curve-based methods when testing large integers used in modern cryptography [5].

Elliptic curve primality proving algorithms provide a different balance between mathematical rigor and computational efficiency. The Goldwasser–Kilian algorithm introduced the concept of generating explicit primality certificates using elliptic curves, while the Atkin–Morain ECPP algorithm significantly improved its practical performance through the use of complex multiplication theory [8], [9]. Unlike probabilistic tests, ECPP produces a certificate that can be verified independently, making it particularly suitable for applications requiring mathematically certified prime numbers.

Table 1 summarizes the principal characteristics of the most widely used primality testing algorithms.

Algorithm	Type	Mathematical Guarantee	Certificate	Practical Performance
Fermat	Probabilistic	No	No	Very High
Solovay-Strassen	Probabilistic	No	No	High
Miller-Rabin	Probabilistic	No	No	Very High
AKS	Deterministic	Yes	Implicit	Moderate
Goldwasser-Kilian	Deterministic (certificatebased)	Yes	Yes	Moderate
Atkin-Morain ECPP	Deterministic (certificatebased)	Yes	Yes	Very High

Among the algorithms considered, the Miller–Rabin test remains the preferred choice for rapid probabilistic testing due to its simplicity and excellent practical performance. However, when a formal proof of primality is required, elliptic curve methods-particularly ECPP-provide the best compromise between computational efficiency and mathematical certainty. Consequently, ECPP has become the standard algorithm for proving the primality of cryptographic-size integers in modern computational number theory software and cryptographic libraries [9], [10], [12].

CONCLUSION

This paper has presented the mathematical foundations and algorithmic principles of elliptic curve primality tests. The study reviewed the essential concepts of finite fields, elliptic curve groups, Hasse's theorem, the Frobenius endomorphism, and point-counting algorithms that form the theoretical basis of elliptic curve primality proving. These mathematical tools provide a rigorous framework for constructing efficient algorithms capable of certifying the primality of large integers.

The paper further examined the two most influential elliptic curve primality proving algorithms, namely the Goldwasser–Kilian algorithm and the Atkin–Morain Elliptic Curve Primality Proving (ECPP) algorithm. Although both methods rely on the arithmetic of elliptic curves, ECPP achieves significantly better practical performance by employing complex multiplication techniques, making it the preferred algorithm for proving the primality of cryptographic-size integers.

A comparative analysis with classical primality tests demonstrated that probabilistic algorithms such as Fermat, Solovay–Strassen, and Miller–Rabin offer excellent computational efficiency but do not produce mathematical certificates of primality. In

contrast, deterministic methods based on elliptic curves provide rigorous proofs while maintaining practical efficiency that substantially exceeds general deterministic algorithms such as AKS for large integers.

Overall, elliptic curve primality tests represent one of the most significant achievements in computational number theory, combining deep mathematical theory with practical applicability. Their ability to generate efficiently verifiable primality certificates has made them an indispensable tool in modern cryptography, particularly in the generation and validation of secure cryptographic parameters.

Future research may focus on improving the efficiency of elliptic curve point-counting algorithms, optimizing ECPP implementations for large-scale computations, and investigating the integration of elliptic curve primality proving with emerging cryptographic systems and post-quantum security frameworks.

REFERENCES

1. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL, USA: CRC Press, 1996.
2. D. R. Stinson and M. B. Paterson, Cryptography: Theory and Practice, 4th ed. Boca Raton, FL, USA: CRC Press, 2019.
3. M. O. Rabin, "Probabilistic algorithm for testing primality," Journal of Number Theory, vol. 12, no. 1, pp. 128–138, 1980.
4. R. Solovay and V. Strassen, "A fast Monte-Carlo test for primality," SIAM Journal on Computing, vol. 6, no. 1, pp. 84–85, 1977.
5. M. Agrawal, N. Kayal, and N. Saxena, "PRIMES is in P," Annals of Mathematics, vol. 160, no. 2, pp. 781–793, 2004.
6. J. H. Silverman and J. Tate, Rational Points on Elliptic Curves, 2nd ed. New York, NY, USA: Springer, 2015.
7. L. C. Washington, Elliptic Curves: Number Theory and Cryptography, 2nd ed. Boca Raton, FL, USA: CRC Press, 2008.
8. S. Goldwasser and J. Kilian, "Almost all primes can be quickly certified," Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (STOC), pp. 316–329, 1986.
9. A. O. L. Atkin and F. Morain, "Elliptic curves and primality proving," Mathematics of Computation, vol. 61, no. 203, pp. 29–68, 1993.
10. J. H. Silverman, The Arithmetic of Elliptic Curves, 2nd ed. New York, NY, USA: Springer, 2009.
11. R. Schoof, "Elliptic curves over finite fields and the computation of square roots mod p," Mathematics of Computation, vol. 44, no. 170, pp. 483–494, 1985.
12. H. Cohen, A Course in Computational Algebraic Number Theory. Berlin, Germany: Springer, 1993.