© AUG 2024 | IRE Journals | Volume 8 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725643

Number Theory and Its Applications in Cryptography

Lavkush Pandey
MLK PG College, Balrampur, Uttar Pradesh, India

Abstract- ***Number theory provides the algebraic structure underlying many public-key cryptographic constructions. This paper examines how divisibility, congruences, prime numbers, Euler's theorem, the Chinese Remainder Theorem, finite groups, and elliptic curves support encryption, key exchange, and digital signatures. It adopts a mathematical review methodology, deriving principal identities and illustrating their use with reproducible small-integer examples. RSA correctness is established using modular arithmetic, while Diffie–Hellman and elliptic-curve systems are analyzed in terms of their respective computational hardness assumptions. The paper then distinguishes mathematical correctness from practical security, emphasizing randomness, authenticated protocols, padding, side-channel resistance, and suitable parameter selection. Finally, it evaluates the impact of quantum algorithms on factoring- and discrete-logarithm-based cryptography and briefly introduces lattice-based post-quantum alternatives. The synthesis shows that number theory is essential not only for constructing cryptographic algorithms but also for identifying their assumptions, implementation constraints, and long-term limitations.***

Index Terms- ***discrete logarithm, elliptic-curve cryptography, modular arithmetic, number theory, post-quantum cryptography, prime numbers, RSA***

I. INTRODUCTION

Cryptography protects information by supplying confidentiality, integrity, authentication, and, in appropriate signature settings, evidence of origin. Classical ciphers often manipulate symbols, whereas modern public-key systems operate on mathematical objects such as residue classes, cyclic groups, and elliptic-curve points. Number theory makes these operations precise. Its role is especially visible in RSA, whose trapdoor mechanism uses arithmetic modulo a composite number, and in Diffie–Hellman protocols, which exploit exponentiation in a finite group [1–3].

A cryptographic construction must satisfy two different requirements: *correctness*, meaning that authorized operations recover or verify the intended result, and *security*, meaning that a computationally bounded adversary cannot perform a prohibited task with meaningful advantage. Elementary number theory often proves correctness, but security additionally requires well-defined hardness assumptions and carefully designed protocols. For instance, the identity used to decrypt textbook RSA does not itself make deterministic RSA encryption secure [3].

Aim and scope. This review develops the number-theoretic foundations of major public-key methods, derives illustrative examples, compares hardness assumptions, and discusses contemporary implementation and quantum-era implications. Its contribution is an integrated mathematical exposition and comparative synthesis rather than a new cryptographic primitive or experimental performance claim.

II. METHODOLOGY AND ANALYTICAL FRAMEWORK

This is a theoretical, literature-based review. The analysis draws on foundational work on public-key cryptography [1, 2, 4], standard mathematical references [5, 6], and specifications and guidance published by the IETF and NIST [3, 7, 8]. Each application is examined using four criteria: (i) the underlying algebraic structure; (ii) a correctness argument; (iii) the computational problem presumed hard; and (iv) practical or security qualifications. Numerical demonstrations deliberately use small parameters to make calculations inspectable; they are pedagogical only and are not secure deployment choices.

III. MATHEMATICAL FOUNDATIONS

A. Divisibility and Greatest Common Divisors

For integers a and b, with $a \neq 0$, divisibility $a \mid b$ means $b = ak$ for some integer k. The greatest common divisor $gcd(a, b)$ is the largest positive integer dividing both numbers. The Euclidean algorithm repeatedly applies

$$a = bq + r, \quad 0 \leq r < |b| \qquad (1)$$

until the remainder becomes zero. The extended Euclidean algorithm also computes integers x, y such that

$$ax + by = \gcd(a, b) \qquad (2)$$

Consequently, if $gcd(a, n) = 1$, then $ax \equiv 1 \pmod{n}$ for some x. This x is the multiplicative inverse of a modulo n, a prerequisite for RSA private-exponent generation [5].

© AUG 2024 | IRE Journals | Volume 8 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725643

B. Congruences and Modular Arithmetic

Integers a and b are congruent modulo $n > 1$ when n divides their difference:

$$a \equiv b \pmod{n} \Leftrightarrow n \mid (a - b) \qquad (3)$$

Congruences are preserved under addition and multiplication. Reduction after every multiplication makes exponentiation with extremely large exponents computationally feasible through repeated squaring. Modular arithmetic is therefore both the conceptual language and the computational engine of many public-key systems.

C. Prime Numbers, Euler's Function, and Euler's Theorem

A prime $p > 1$ has exactly two positive divisors. For distinct primes p, q, the integer $n = pq$ satisfies

$$\varphi(n) = (p - 1)(q - 1) \quad (4)$$

where φ is Euler's totient function. Euler's theorem states that, whenever $gcd(a, n) = 1$,

$$a^{\varphi(n)} \equiv 1 \pmod{n} \qquad (5)$$

Fermat's little theorem is the prime-modulus special case:

$$a^{p-1} \equiv 1 \pmod{p} \quad (p \nmid a) \qquad (6)$$

These congruences justify the exponent relationships used in RSA and motivate finite-field exponentiation [5, 6].

D. Chinese Remainder Theorem

If m_1 and m_2 are coprime, the simultaneous congruences

$$x \equiv a_1 \pmod{m_1}, \quad x \equiv a_2 \pmod{m_2} \qquad (7)$$

have a unique solution modulo m_1m_2. More generally, pairwise coprime moduli produce a unique residue class modulo their product. The theorem allows large modular calculations to be decomposed into smaller ones, including optimized RSA private-key operations [3, 5].

E. Finite Groups and Discrete Logarithms

The invertible residue classes modulo p form the multiplicative group $F_p^{\times}$ of order $p - 1$. For a generator g of a subgroup of order r, the equation

$$h = g^x \pmod{p} \quad (8)$$

is easily evaluated for known x, but recovering x from g and h can be computationally difficult for suitable groups and parameters. This is the discrete logarithm problem. Hardness depends on the chosen group: not every modular group or subgroup yields adequate security [2, 6].

IV. RSA CRYPTOGRAPHY: CONSTRUCTION AND CORRECTNESS

RSA, introduced by Rivest, Shamir, and Adleman, uses a public modulus assembled from two secret primes [1]. In its elementary form, choose distinct large primes p and q, calculate $n = pq$ and $\varphi(n) = (p - 1)(q - 1)$, select e with $gcd(e, \varphi(n)) = 1$, and compute d satisfying

$$ed \equiv 1 \pmod{\varphi(n)} \qquad (9)$$

The public key is (n, e) and the private exponent is d (with the prime factors kept secret). For a message representative m in the permitted residue range, textbook operations are

$$c \equiv m^e \pmod{n}, \quad m' \equiv c^d \pmod{n} \qquad (10)$$

Correctness. Since $ed = 1 + k\varphi(n)$ for some integer k, Euler's theorem gives $m^{ed} \equiv m \pmod{n}$ when $gcd(m, n) = 1$. For all residue classes, including those not coprime to n, correctness follows by applying Fermat's little theorem separately modulo p and q, and combining the results with the Chinese Remainder Theorem [3].

Worked example (not secure). Let $p = 11$, $q = 17$, so $n = 187$ and $\varphi(n) = 160$. Choose $e = 7$ and $d = 23$, because $7 \cdot 23 = 161 \equiv 1 \pmod{160}$. For $m = 2$,

$$c = 2^7 \bmod 187 = 128 \quad (11)$$

and decryption recovers

$$128^{23} \bmod 187 = 2 \qquad (12)$$

The example establishes arithmetic correctness only. Real RSA encryption requires an approved encoding and padding scheme, such as RSAES-OAEP, while RSA signatures commonly use RSASSA-PSS; RFC 8017 explicitly distinguishes these schemes from the raw mathematical primitive [3]. Security additionally depends on key generation, implementation, and adequate key sizes [7].

A. RSA Acceleration via the Chinese Remainder Theorem

For a private-key computation, let $d_p = d \bmod (p - 1)$ and $d_q = d \bmod (q - 1)$. Evaluate

$$m_p = c^{d_p} \bmod p, \quad m_q = c^{d_q} \bmod q \qquad (13)$$

then reconstruct m modulo pq by the Chinese Remainder Theorem. Because both exponentiations use approximately half-size moduli, this can reduce computation substantially. Implementations must nevertheless address timing, fault, and other side-channel risks [3].

© AUG 2024 | IRE Journals | Volume 8 Issue 2 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725643

V. DIFFIE–HELLMAN KEY AGREEMENT

Diffie and Hellman showed that participants can establish a shared secret over a public channel using exponentiation in a suitable finite group [2]. Let g generate a selected subgroup modulo a prime p. Alice chooses a private integer a and publishes $A = g^a \bmod p$; Bob chooses b and publishes $B = g^b \bmod p$. They compute

$$K_A = B^a \bmod p = g^{ab} \bmod p \qquad (14)$$

$$K_B = A^b \bmod p = g^{ab} \bmod p \qquad (15)$$

Equality follows because exponent multiplication commutes. An eavesdropper sees (p, g, A, B) but is not given the private exponents. The relevant assumptions concern the difficulty of discrete-logarithm-related problems in the specified group; their exact form depends on the protocol's security definition.

Worked example (not secure). With $p = 23$, $g = 5$, $a = 6$, and $b = 15$, the public values are $A = 8$ and $B = 19$. The two parties obtain

$$19^6 \bmod 23 = 8^{15} \bmod 23 = 2 \qquad (16)$$

Basic unauthenticated Diffie–Hellman is vulnerable to a man-in-the-middle attack: a secure deployment requires authentication and a key-derivation procedure rather than treating the raw group element as a ready-to-use encryption key [2, 6].

VI. ELLIPTIC-CURVE CRYPTOGRAPHY

Over a prime field F_p with $p > 3$, an elliptic curve may be written as

$$E: y^2 \equiv x^3 + ax + b \pmod p \qquad (17)$$

subject to the nonsingularity condition

$$4a^3 + 27b^2 \not\equiv 0 \pmod p \qquad (18)$$

The curve's points, together with the point at infinity O, form an abelian group. The group operation is point addition, and scalar multiplication repeatedly adds a point to itself. Given a base point G of large prime order r, a private scalar k generates a public point

$$Q = [k]G \qquad (19)$$

The elliptic-curve discrete logarithm problem asks for k given G and Q. Elliptic-curve Diffie–Hellman takes public points $A = [a]G$ and $B = [b]G$, with the common point

$$[a]B = [ab]G = [b]A \qquad (20)$$

Curve choice, point validation, subgroup handling, and protocol specification are critical. Elliptic curves can offer strong classical security with shorter public keys than traditional finite-field or RSA systems at comparable security levels; key-size comparisons depend on the exact algorithms and strength targets [6, 7].

VII. COMPARATIVE ANALYSIS

Table 1. Comparison of number-theoretic public-key constructions

Construction	Number-theoretic structure	Central computational challenge	Typical cryptographic use	Key qualification
RSA	Composite-modulus arithmetic and modular inverses	Integer factorization / RSA inversion assumptions	Encryption and digital signatures	Secure padding and sufficient key sizes are essential
Finite-field Diffie–Hellman	Cyclic subgroups of $F_p^\times$	Discrete-logarithm-related problems	Establishing shared keys	Authenticate exchanged values
Elliptic-curve protocols	Abelian groups of curve points over finite fields	Elliptic-curve discrete logarithm	Key exchange and signatures	Validate curve points and parameters
Lattice-based ML-KEM	Polynomial modules and modular arithmetic	Module Learning with Errors	Post-quantum key encapsulation	Uses a different hardness basis from factoring and discrete log

The comparison reveals a common design pattern: a mathematically efficient forward operation is coupled to a recovery task believed to be hard without privileged information. However, an informal "one-way" description cannot replace a formal security reduction, a precise adversarial model, or implementation testing. Number theory provides the mechanism; cryptographic engineering determines whether the complete system meets its intended security goals.

VIII. SECURITY, COMPLEXITY, AND QUANTUM COMPUTING

The security of classical public-key algorithms is not measured solely by the number of bits in a modulus or coordinate. Attack complexity, algorithmic

advances, subgroup order, parameter generation, and attacker capabilities all matter. NIST's key-management guidance, for example, associates a 3072-bit RSA modulus with roughly 128 bits of classical security strength; a 2048-bit RSA modulus is associated with approximately 112 bits [7]. These values are strength estimates under stated assumptions, not guarantees for an arbitrary implementation.

Shor's quantum algorithm gives polynomial-time approaches to integer factorization and discrete logarithms on a sufficiently capable fault-tolerant quantum computer [9]. Therefore, RSA, traditional finite-field Diffie–Hellman, and elliptic-curve discrete-logarithm systems face a longterm quantum threat. This does not imply that every present-day quantum device can break deployed public-key systems; the resources needed for cryptographically relevant attacks remain an important engineering consideration.

Post-quantum cryptography investigates different computational foundations. In 2024, NIST standardized ML-KEM in FIPS 203, building key encapsulation on the Module Learning with Errors problem [8]. Such systems still use modular arithmetic and algebraic structures, but they do not depend on integer factorization or the classical discrete-logarithm problems. Number theory thus remains relevant even as cryptography moves beyond its traditional public-key hardness assumptions.

IX. DISCUSSION AND RESEARCH DIRECTIONS

Three findings follow from the synthesis. First, elementary congruence theory connects directly to practical public-key algorithms: Bézout's identity supplies RSA inverses, Euler's theorem establishes exponent relationships, and the Chinese Remainder Theorem accelerates decryption. Second, abstract group structure is central to key agreement: Diffie–Hellman correctness requires compatible group operations, whereas its security depends on assumptions about information recoverable from public elements. Third, practical cryptography is a protocol discipline as much as a mathematical one. Weak randomness, unauthenticated exchanges, unsafe padding, or leaking side-channel information can defeat a system even when its underlying number-theoretic theorem is correct.

Useful future research includes formally verified cryptographic implementations; quantitative comparisons of classical, hybrid, and post-quantum key establishment; efficient constant-time modular arithmetic; side-channel-resistant implementations; and cryptographic agility during standards migration. Empirical studies should specify platforms, threat models, parameter sets, metrics, and reproducible benchmarks before making performance or security claims.

X. CONCLUSION

Number theory is a foundational language of modern cryptography. Prime factorization, modular inverses, Euler's theorem, and the Chinese Remainder Theorem explain the arithmetic of RSA, while cyclic groups and elliptic curves enable discrete-logarithm-based exchange and signatures. Mathematical proofs establish correctness, but secure use also demands trustworthy parameters, authenticated protocols, appropriate padding, constant-time implementation, and ongoing evaluation of computational threats. As post-quantum designs gain prominence, the field continues to draw on modular arithmetic and abstract algebra while broadening the hard problems used to protect communications.

REFERENCES

[1] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM, 21*(2), 120–126. https://doi.org/10.1145/359340.359342

[2] Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. *IEEE Transactions on Information Theory, 22*(6), 644–654. https://doi.org/10.1109/TIT.1976.1055638

[3] Moriarty, K., Kaliski, B., Jonsson, J., & Rusch, A. (2016). *PKCS #1: RSA cryptography specifications version 2.2* (RFC 8017). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc8017

[4] Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation, 48*(177), 203–209. https://doi.org/10.1090/S0025-5718-1987-0866109-5

[5] Rosen, K. H. (2011). *Elementary number theory and its applications* (6th ed.). Pearson.

[6] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). *Handbook of applied cryptography*. CRC Press.

[7] Barker, E. (2020). *Recommendation for key management: Part 1—General* (NIST SP 800-57 Part 1, Rev. 5). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-57pt1r5

[8] National Institute of Standards and Technology. (2024). *Module-lattice-based key-encapsulation mechanism standard* (FIPS 203). https://doi.org/10.6028/NIST.FIPS.203

[9] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing, 26*(5), 1484–1509. https://doi.org/10.1137/S0097539795293172