ON THE APPLICATION OF ELLIPTIC CURVES IN CRYPTOGRAPHY

Anxhela Baraj [1*] , *Jonatan Lerga* [2]

[1] Canadian Institute of Technology, Tirane, Albania
[2] Faculty of Engineering, University of Rijeka, Rijeka, Croatia

Abstract

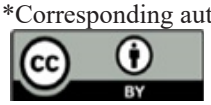
Elliptic Curve Cryptography (ECC) is a cornerstone of modern cryp- tography, providing strong security with much smaller key sizes com- pared to traditional algorithms such as RSA and Diffie–Hellman. This paper presents the mathematical foundations of ECC, including elliptic curves over finite fields, group operations, and scalar multiplication, which ensure security through the Elliptic Curve Discrete Logarithm Problem (ECDLP). Core protocols—Elliptic Curve Diffie–Hellman (ECDH), El- liptic Curve Digital Signature Algorithm (ECDSA), and Elliptic Curve Integrated Encryption Scheme (ECIES)—are examined, and a compara- tive analysis with RSA and DSA demonstrates ECC's superior efficiency in key generation, signing, and verification. Practical performance is il- lustrated through an experimental implementation using the NIST P-256 curve, highlighting fast key generation and shared secret computation. ECC's applications span secure internet communication, blockchain sys- tems, encrypted messaging, IoT networks, and digital identity verification. Advantages include computational efficiency, scalability, and suitability for resource-constrained devices, while challenges involve side-channel at- tacks, implementation complexity, interoperability, and potential quan- tum threats. The study confirms ECC as a secure, versatile, and practical framework for contemporary digital communications.

Keywords: Elliptic Curve Cryptography, Elliptic Curve Discrete Loga- rithm Problem, Scalar Multiplication, Digital Signatures, Key Exchange

1. INTRODUCTION

Elliptic Curve Cryptography (ECC) represents one of the most significant ad- vancements in modern cryptography, providing strong security while using much smaller key sizes compared to traditional algorithms such as RSA and classical Diffie–Hellman. This efficiency makes ECC particularly suitable for contempo- rary digital environments, where devices often have limited computational re- sources, memory, and battery life Miller1985,Washington2008. A 256-bit ECC key, for example, provides security equivalent to a 3072-bit RSA key, demon- strating its ability to achieve high levels of protection with reduced computa- tional overhead. ECC relies on the algebraic structure of elliptic curves defined over specific fields. While elliptic curves can be defined over many types of fields, cryp- tographic implementations typically use finite fields, either prime fields Fp or binary fields F2m . In these finite fields, the set of points on an elliptic curve forms a finite abelian group, which is either cyclic or a product of cyclic groups. This discrete group structure is fundamental to the security of ECC, particu- larly through the Elliptic Curve Discrete Logarithm Problem (ECDLP), which is computationally infeasible to solve for sufficiently large fields (Kyars2025). Elliptic curves over the real numbers R can provide intuitive insight into the group operations used in elliptic curve cryptography, such as point addition and point doubling. In practice, however, all cryptographic computations are performed over finite fields to ensure discrete, secure, and efficiently computable operations. The fundamental ECC operations—point addition, point doubling, and scalar multiplication—form the basis for cryptographic protocols including key exchange, digital signatures, and encryption schemes (Washington2008). Beyond its theoretical foundations, ECC has wide-ranging real-world appli- cations. It is employed in secure key exchange protocols such as Elliptic Curve Diffie–Hellman (ECDH), digital signature algorithms like ECDSA, and hybrid encryption schemes such as ECIES. ECC is also widely used in modern digital systems, including blockchain networks, encrypted messaging applications, In- ternet of Things (IoT) devices, cloud platforms, and digital identity verification systems (Achary2023; Tanksale2024; Stebila2025; Yu2022). The combination of strong mathematical foundations, computational effi- ciency, and broad applicability makes ECC a cornerstone of modern cryptog- raphy. At the same time, ECC faces challenges such as susceptibility to side- channel attacks, implementation complexity, interoperability with legacy systems, and potential vulnerabilities posed by future quantum computing (Benitez2025). Addressing these challenges requires careful implementation and ongoing re- search, but ECC's advantages make it a highly practical and versatile crypto- graphic framework for protecting digital communications in today's connected world.

*Corresponding author: Anxhela Baraj, anxhela.baraj@cit.edu.al



2. MATHEMATICAL BACKGROUND

Elliptic Curve Cryptography (ECC) has gained prominence due to its ability to provide strong cryptographic security with relatively small key sizes, making it particularly suitable for resource-constrained environments such as IoT devices, embedded systems, and mobile platforms (Miller1985; Koblitz1987). The mathematical foundations of ECC rely on the rich algebraic structure of elliptic curves, which are defined over various fields, most notably the rational numbers and finite fields. This chapter introduces the essential mathematical concepts of elliptic curves, their group structure, and the computational problems that en- sure their cryptographic strength, following the formal definitions and notation used in **(Dujella2011)**.

2.1 Elliptic Curves over the Rationals

An elliptic curve E over the field of rational numbers Q is defined as the set of points (x, y) □ Q^2 satisfying the Weierstrass equation:

$$E : y2 = x3 + ax + b, \quad (1)$$

where a, b □ Q, and the discriminant

$$\Delta = -16(4a3 + 27b2) \quad (2)$$

must satisfy Δ = 0, ensuring that the curve is nonsingular.

The set of rational points E(Q), together with a distinguished point at in- finity O, forms an abelian group. The group operation is defined geometrically: the sum of two points P and Q is found by drawing a line through P and Q, determining its third intersection with the curve, and then reflecting that point across the x-axis. The inverse of a point P = (x, y) is P = (x, y), and the point at infinity O acts as the identity element.

According to the Mordell–Weil theorem, the group the group E(Q) is finitely generated:

$$E(\mathrm{Q}) \cong E_{\mathrm{tors}}(\mathrm{Q}) \times Z^r,$$

where Etors(Q) is the torsion subgroup (points of finite order) and r is the rank of the elliptic curve. Although curves over Q are not used directly in cryptography, their theoretical properties form the mathematical basis for elliptic curves over finite fields.

2.2 Elliptic Curves over Finite Fields

For cryptographic applications, the most relevant elliptic curves are defined over finite fields Fq, where q = pk for a prime p and integer k ≥ 1. The field Fq can be viewed as a k-dimensional vector space over the prime field Fp = Z/pZ, and can be constructed as

$$\mathrm{F}_q = \mathrm{F}_p[x]/(f(x)),$$

where f (x) is an irreducible polynomial of degree k over F_p (Dujella2011).

An elliptic curve E over F_q is typically defined by

$$E : y^2 = x^3 + ax + b,\ a, b \in F_q,$$

with the non-singularity condition 4a3 + 27b2 = 0. The set of Fq-rational points is

$$E(\mathrm{F}_q) = \{(x, y) \in \mathrm{F}^2 : y^2 = x^3 + ax + b\} \cup \{O\},$$

which forms a finite abelian group under the addition law. By Hasse's theorem:

$$|q + 1 - \#E(\mathrm{F}_q)| \le 2\sqrt{q}.$$

The group $E(F_q)$ has the structure

$$E(\mathrm{F}_q) \cong Z/n_1Z \times Z/n_2Z, \quad n_1 \mid n_2, n_1 \mid (q-1),$$

and in most cryptographic applications, n1 = 1, making $E(F_q)$ cyclic of prime order.

In binary fields (characteristic 2), elliptic curves take the modified form

$$E : y2 + xy = x3 + ax2 + b, \quad b \neq 0,$$

allowing efficient arithmetic suitable for hardware implementations.

2.3 Group Law and Scalar Multiplication

Let E be defined over Fq, and P = (x1, y1), Q = (x2, y2) ∈ E(Fq)

Point addition $(P \neq Q)$:

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}, \quad (10)$$

$$x_3 = \lambda^2 - x_1 - x_2, \quad (11)$$

$$y_3 = \lambda(x_1 - x_3) - y_1. \quad (12)$$

Point doubling $(P = Q)$:

$$\lambda = \frac{3x^2 + a}{2y_1} \quad (13)$$

$$x_3 = \lambda^2 - 2x_1 \quad (14)$$

$$y_3 = \lambda(x_1 - x_3) - y_1 \quad (15)$$

Scalar multiplication kP = P + + P (k times) is fundamental in ECC. Efficient algorithms include the ** double-and-add ** and ** Montgomery ladder ** methods, which optimize computation and resist side-channel attacks.

3. CRYPTOGRAPHIC OPERATIONS AND PROTOCOLS

The algebraic structure of elliptic curves over finite fields enables a range of cryptographic operations that underpin modern secure communication systems. The

robustness of these mechanisms relies primarily on the computational in- tractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP), which forms the cornerstone of elliptic curve security (Koblitz1987; Miller1985). This chapter presents the principal cryptographic mechanisms based on ellip- tic curves, including key exchange (ECDH), digital signatures (ECDSA), and public-key encryption (ECIES) — each contributing to confidentiality, authen- tication, and data integrity in modern cryptosystems.

3.1 Elliptic Curve Discrete Logarithm Problem (ECDLP)

The ECDLP serves as the foundation for the security of all elliptic curve cryp- tographic protocols. Given an elliptic curve E(Fq), a base point $P \in E(Fq)$, and another point $Q = kP$, the goal is to determine the scalar
$k \in Z$ such that: $Q = kP$

Despite extensive research, no efficient classical algorithm has been found to solve this problem when q and the group order of E(Fq) are sufficiently large (Silverman2009). This intractability allows ECC to achieve equivalent security to RSA or DLP systems with much smaller key sizes — for example, a
256-bit ECC key provides comparable security to a 3072-bit RSA key **(Barker2019)**

3.2 Elliptic Curve Diffie–Hellman (ECDH)

The Elliptic Curve Diffie–Hellman (ECDH) protocol enables two parties to es- tablish a shared secret over an insecure channel. Each participant selects a pri- vate key kA, kB and computes the corresponding public key KA = kAP, KB = kBP . After exchanging public keys, both compute the same shared secret:

$$S = k_A K_B = k_B K_A = k_A k_B P$$

This shared secret is subsequently used to derive symmetric encryption keys (Hankerson2004). The protocol inherits its security directly from the hardness of the ECDLP, as an eavesdropper observing P, KA, KB cannot feasibly compute S without solving the discrete logarithm problem. ECDH is widely implemented in protocols such as TLS 1.3, SSH, and Signal due to its efficiency and forward secrecy properties **(Rescorla2018).**

3.3 Elliptic Curve Digital Signature Algorithm (ECDSA)

The Elliptic Curve Digital Signature Algorithm (ECDSA) provides message authentication and integrity verification. A signer holding a private key d generates a signature (r, s) for a message m using a random nonce k, while the verifier uses the public key Q = dP to confirm validity. The verification process ensures that the signature corresponds to the original message without revealing the private key. The security of ECDSA depends both on the unpredictability of the nonce and the hardness of the ECDLP (Johnson2001).
Modern blockchain platforms such as Bitcoin and Ethereum employ ECDSA extensively for transaction verification due to its strong security and computa- tional efficiency (Bonneau2015).

3.4 Elliptic Curve Integrated Encryption Scheme (ECIES)

The Elliptic Curve Integrated Encryption Scheme (ECIES) combines asymmet- ric key exchange with symmetric encryption and message authentication codes (MACs). The sender uses the recipient's public key to compute a shared secret via ECDH, derives a symmetric key using a key derivation function (KDF), encrypts the message, and generates a MAC for integrity verification. The re- cipient repeats the ECDH operation with their private key to derive the same shared secret and decrypts the message. This hybrid approach balances per- formance and security, providing both confidentiality and integrity in a single scheme. ECIES has become a preferred encryption mechanism in lightweight cryp- tographic systems, including smart cards, IoT devices, and embedded systems, where computational and energy efficiency are critical.

4. COMPARATIVE ANALYSIS OF CRYPTOGRAPHIC SYSTEMS

Elliptic Curve Cryptography (ECC) has become a cornerstone of modern se- cure communication due to its ability to achieve strong security with relatively small key sizes. This efficiency makes ECC particularly suitable for constrained environments such as IoT platforms, embedded systems, and mobile devices (Hankerson2004).
This chapter provides a comparative analysis between ECC and classical public-key systems—namely RSA and DSA—in terms of key size, computational performance, and efficiency. The goal is to demonstrate ECC's advantages both theoretically and experimentally.

4.1 Key Size Comparison for Equivalent Security Levels

One of ECC's most significant advantages is its superior key size–to–security ra- tio. Because ECC's security relies on the hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), it provides the same cryptographic strength as much larger RSA or DSA keys based on integer factorization or discrete

logarithms **(Bernstein2017).**

For example, an ECC key of 256 bits offers approximately the same security level as an RSA key of 3072 bits or a DSA key of 3072 bits, while a 384-bit ECC key corresponds to roughly 7680-bit RSA security. This substantial reduc- tion in key length directly translates to lower storage, reduced communication bandwidth, and faster computations.

These equivalences are visually illustrated in Figure 1, which presents a com- parative plot of cryptographic key sizes across systems offering similar security levels. The figure clearly shows that ECC achieves comparable cryptographic strength with keys that are an order of magnitude smaller than those required by RSA and DSA, reinforcing its suitability for lightweight and resource-limited implementations.

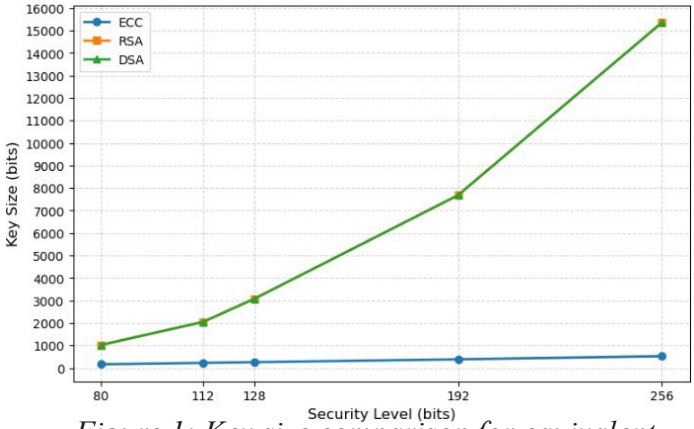


Figure 1: Key size comparison for equivalent security levels for ECC, RSA and DSA

4.2 Performance Evaluation of Cryptographic Operations

In addition to smaller key sizes, ECC demonstrates superior performance in computational operations. Table 1 summarizes the measured execution times for key generation, digital signing, and verification for ECC, RSA, and DSA, obtained through Python-based experimental evaluation.

	Key Generation	Signing	Verification
ECC	0.004196	0.062078	0.000583
RSA	1.422051	0.006053	0.000624
DSA	0.434125	0.000844	0.000937

Table 1: Cryptographic operation timings (seconds) for ECC, RSA, and DSA

The results indicate that ECC significantly outperforms RSA and DSA in key generation, being over 300× faster than RSA and more than 100× faster than DSA. Although the ECC signing phase is slower than DSA, its verification process is the fastest, which is critical in real-world systems where signature validation occurs frequently.

5. APPLICATIONS OF ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Cryptography (ECC) has transitioned from a theoretical concept to a fundamental pillar of modern cryptographic systems. Its appeal lies in the ability to deliver strong security with smaller key sizes, reducing computational load and power consumption across a wide range of environments — from high- performance servers to low-power embedded systems.

6. CORE APPLICATION DOMAINS

6.1.1 Secure Internet Communication

ECC forms the cryptographic foundation of the Transport Layer Security (TLS 1.3) protocol, which safeguards the majority of secure internet traffic. The El- liptic Curve Diffie–Hellman Ephemeral (ECDHE) mechanism ensures forward secrecy by establishing a unique session key for every connection. Widely de- ployed curves such as NIST P-256 and X25519 are supported by all major browsers and servers, offering both high security and efficient key exchange op- erations (Rescorla2018).

6.1.2 Blockchain and Digital Assets

Modern blockchain systems—including Bitcoin, Ethereum, and numerous de- centralized finance (DeFi) platforms—rely on ECC for digital signatures and address generation. The secp256k1 curve and the Elliptic Curve Digital Sig- nature Algorithm (ECDSA) provide authentication, transaction integrity, and non-repudiation. ECC's compact key representation reduces storage overhead on distributed ledgers while preserving cryptographic strength.

6.1.3 Secure Messaging and End-to-End Encryption

In privacy-oriented communication platforms such as Signal and WhatsApp, ECC enables forward-secure message encryption. Using Curve25519 for key ex- change and Ed25519 for signatures, each conversation employs ephemeral keys that prevent retrospective decryption of intercepted data. ECC's efficiency al- lows these operations to run smoothly on mobile devices with limited resources.

6.1.4 Internet of Things (IoT) and Embedded Security

ECC's reduced key size makes it particularly attractive

for IoT environments, where processing power and energy are limited. Implementations such as TinyECC and MicroECC enable secure communication and firmware authentication in sensors, smart home devices, and autonomous systems. These lightweight cryptographic frameworks demonstrate ECC's suitability for real-world embedded systems.

6.1.5 Digital Identification and E-Government Systems

Governments and institutions increasingly employ ECC-based digital signatures (ECDSA, EdDSA) for identity verification, e-passports, and document signing.

6.2 Implementation and Practical Demonstration

To illustrate ECC's real-world applicability, a practical demonstration was con- ducted using the Elliptic Curve Diffie–Hellman (ECDH) protocol over the NIST P-256 (secp256r1) curve. This implementation aimed to evaluate the computa- tional efficiency of key generation and shared secret computation — two fundamental operations in many ECC-based applications.
The experiment was executed in Python using the cryptography library. The results presented in Table 2 summarize the average execution time of the primary ECDH operations, measured over multiple runs.

Operation	Average Time (seconds)
Key Pair Generation	0.004215
Shared Secret Computation	0.001258
Total Average Time	0.005473

Table 2: Execution time of ECC operations using NIST= P-256

The total execution time of approximately 5.47 milliseconds demonstrates ECC's computational efficiency and its practicality for real-time applications. This level of performance supports its widespread use in secure internet pro- tocols, IoT authentication, and blockchain transaction validation, where rapid cryptographic exchanges are required.

7. ADVANTAGES AND CHALLENGES OF ELLIPTIC CURVE CRYPTOGRAPHY

Following its extensive applications in secure communication, blockchain, IoT, and digital identity systems, it is important to evaluate the strengths and limi- tations of Elliptic Curve Cryptography (ECC) to understand its practical suit- ability.
ECC provides strong cryptographic security with substantially smaller keys than classical public-key systems such as RSA and DSA. For example, a 256- bit ECC key offers security equivalent to a 3072-bit RSA key (Barker2019; Koblitz1987). This reduction in key length decreases storage requirements, communication overhead, and computational load, making ECC particularly attractive for resource-constrained environments like IoT devices, mobile plat- forms, and embedded systems.
Smaller keys also enable faster cryptographic operations, including key gen- eration, signing, and signature verification. As illustrated in the experimental evaluation of NIST P-256 operations (see Section 5), ECC demonstrates high efficiency in both key establishment and shared secret computation, support- ing real-time applications such as secure messaging and blockchain transaction processing.
The compact key sizes and efficient arithmetic operations reduce memory and energy consumption, which is critical for battery-powered or low-performance devices. ECC's efficiency also supports scalable deployment across large net- works, where frequent cryptographic operations are required.
ECC's adoption in widely used protocols—including TLS/HTTPS, secure messaging applications, blockchain platforms, and digital identity management—underscores its versatility and practical relevance (Rescorla2018). Its compatibility with hybrid encryption schemes, forward secrecy mechanisms, and signature stan- dards further enhances its applicability across diverse operational contexts.

7.1 Challenges and Limitations

However, ECC presents several implementation challenges. Its arithmetic oper- ations are mathematically complex, and careful implementation is necessary to prevent vulnerabilities in scalar multiplication, point addition, and side-channel protections. Errors in implementation, even when using secure curves and pa- rameters, can compromise security.
ECC operations are also vulnerable to side-channel attacks, including timing attacks, power analysis, and fault injection. Countermeasures such as constant- time algorithms, randomization, and blinding techniques are essential to miti- gate these threats (Hankerson2004; Bernstein2017).
Interoperability between different cryptographic libraries, curves, and stan- dards can introduce additional challenges. Ensuring cross-platform compatibil- ity requires strict adherence to standardized curves and protocol specifications. Finally, the rise of quantum computing poses a potential long-term risk.
Shor's algorithm could efficiently solve the Elliptic Curve Discrete Logarithm Problem on a sufficiently powerful quantum computer, undermining ECC se- curity. Research into post-quantum cryptography and hybrid approaches is actively addressing this emerging threat **(Chen2016).**

8. CONCLUSIONS

Elliptic Curve Cryptography (ECC) offers a compelling combination of strong security, efficiency, and versatility, making it highly suitable for modern digi- tal environments. By leveraging the algebraic structure of elliptic curves and the computational hardness of the Elliptic Curve Discrete Logarithm Problem (ECDLP), ECC achieves security levels comparable to traditional public-key systems while using significantly smaller keys. This paper demonstrated both the theoretical foundations and practical im- plementation of ECC, highlighting its core protocols—ECDH, ECDSA, and ECIES—and showing superior computational performance in key generation, signing, and shared secret computation using the NIST P-256 curve. ECC's applicability spans a wide range of domains, including secure internet commu- nication, blockchain, encrypted messaging, IoT, and digital identity verification. While challenges remain, such as implementation complexity, side-channel vulnerabilities, interoperability, and potential quantum threats, ECC's advan- tages in efficiency, scalability, and suitability for resource-constrained devices underscore its continued relevance. Overall, ECC provides a robust, flexible, and practical framework for securing communications in today's interconnected digital world.

9. REFERENCES

Achary, R., Shelke, C. J., Marx, K., & Rajesh, A. (2023). Security implementation on IoT using CoAP and elliptic curve cryptography. Procedia Computer Science, 216, 104–115. https://doi.org/10.1016/j.procs.2023.12.105

Barker, E., & Roginsky, A. (2019). Transitioning the use of cryp- tographic algorithms and key lengths (NIST Special Publication 800- 131A Rev. 2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-131Ar2

Benitez, C. (2025). Mapping quantum threats: An engineering inventory of cryptographic dependencies. arXiv. https://arxiv.org/abs/2509.24623

Bernstein, D. J., & Lange, T. (2017). SafeCurves: Choosing safe curves for elliptic-curve cryptography. Journal of Cryptographic Engineering, 7, 233–246. https://doi.org/10.1007/s13389-017-0167-3

Bonneau, J., et al. (2015). SoK: Research perspectives and challenges for Bitcoin and cryptocurrencies. In Proceedings of the IEEE Symposium on Security and Privacy (pp. 104–121). https://doi.org/10.1109/SP.2015.14

Chen, L. K., et al. (2016). Report on post-quantum cryptography (NIST IR 8105). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8105

Dujella, A. (2011). Elliptic curves. Basque Center for Applied Mathematics and Universidad del Pa´ıs Vasco.

Hankerson, D., Menezes, A. J., & Vanstone, S. (2004). Guide to elliptic curve cryptography. Springer. https://doi.org/10.1007/b97644

Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve dig- ital signature algorithm (ECDSA). International Journal of Information Security, 1 (1), 36–63. https://doi.org/10.1007/s102070100002

Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of Com- putation, 48 (177), 203–209. https://doi.org/10.1090/S0025-5718-1987-0866083-5

Kyars, K. (2025). Elliptic curve cryptography: Secure. Columbia Journal of Mathematics.

Marouan, A., et al. (2023). Elliptic curve cryptography signing algo- rithms behind Blockchain 2.0. In Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security (NISS 2023). https://doi.org/10.1145/3607720.3607747

Miller, V. S. (1985). Use of elliptic curves in cryptography. In H. C. Williams (Ed.), Advances in cryptology—CRYPTO '85 (pp. 417–426). Springer. https://doi.org/10.1007/3-540-39799-X31

National Institute of Standards and Technology. (2020). Recom- mendation for key management (NIST Special Publication 800-57). https://doi.org/10.6028/NIST.SP.800-57

Rescorla, E. (2018). The transport layer security (TLS) proto- col version 1.3 (RFC 8446). Internet Engineering Task Force. https://doi.org/10.17487/RFC8446

Silverman, J. H. (2009). The arithmetic of elliptic curves. Springer. https://doi.org/10.1007/978-0-387-09494-6

Stebila, D., Fluhrer, S., & Gueron, S. (2025). Hybrid key exchange in TLS 1.3. IETF Internet-Draft. https://www.ietf.org/archive/id/draft-ietf-tls-hybrid-design-12.html

Tanksale, V. (2024). Efficient elliptic curve Diffie–Hellman key ex- change for resource-constrained IoT devices. Electronics, 13 (18). https://doi.org/10.3390/electronics13183631

Washington, L. C. (2008). Elliptic curves: Number theory and cryptography (2nd ed.). CRC Press. https://doi.org/10.1201/9781420071474

Yu, H., & Wang, H. (2022). Elliptic curve threshold signature scheme for blockchain. Journal of Information Security and Applications, 63, 103445. https://doi.org/10.1016/j.jisa.2022.103445