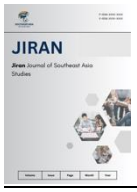

Journal of Southeast Asia Studies
Vol 7 No 2 July 2026
ISSN: XXXX-XXXX (Print) ISSN: XXXX-XXXX (Electronic)
Open Access: https://jiran.unaim-wamena.ac.id/jiran

Symmetric and Asymmetric Cryptography: Basic Principles of DES, AES, RSA, and Elliptic Curve Encryption in Information Security

Nia Novianti[1], Syurahbil[2], Rani Robetty[3]
[1,2]Politeknik Hasan Abdi
[3]Universitas Deztron Indonesia
Email: nia.politeknikhasanabdi@gmail.com[1], syurahbil@gmail.com[2], robetrani6@gmail.com[3]

Article Info :
Received:
25/01/2026
Revised:
29/01/2026
Accepted:
05/02/2026

ABSTRACT

In the digital era, information security has become a critical priority for organizations and individuals. Cryptography serves as the foundation of modern security systems by providing confidentiality, integrity, and authentication mechanisms. This research comprehensively examines the fundamental principles and implementation of symmetric cryptography (DES and AES) and asymmetric cryptography (RSA and Elliptic Curve). The study analyzes the mathematical foundations, encryption mechanisms, security strengths, and practical applications of each algorithm. DES, despite being deprecated due to its 56-bit key vulnerability, remains relevant for understanding block cipher evolution. AES has emerged as the global standard with robust 128, 192, and 256-bit key options. RSA provides secure public-key infrastructure through factorization complexity, while Elliptic Curve Cryptography offers equivalent security with significantly smaller key sizes, making it ideal for resource-constrained environments. The research employs comparative analysis methodology to evaluate algorithm performance, security levels, and implementation efficiency. Results demonstrate that hybrid cryptosystems combining symmetric and asymmetric approaches provide optimal security-performance balance. The findings provide practical guidance for security architects in selecting appropriate cryptographic solutions based on specific requirements, computational resources, and threat models.

Keywords: *Symmetric Cryptography, Asymmetric Cryptography, DES, AES, RSA, Elliptic Curve, Information Security, Encryption Algorithms*

1. Introduction

The exponential growth of digital communication and data storage has created unprecedented challenges in information security. With billions of sensitive transactions occurring daily across global networks, the need for robust cryptographic protection has never been more critical (Stallings, 2017). Cryptography, derived from the Greek words 'kryptos' (hidden) and 'graphein' (writing), provides the mathematical foundation for securing information in the digital age. Modern cryptographic systems must address multiple security objectives: confidentiality to prevent unauthorized access, integrity to detect tampering, authentication to verify identity, and non-repudiation to ensure accountability.

Cryptographic algorithms are broadly classified into two fundamental categories: symmetric and asymmetric encryption. Symmetric cryptography, also known as secret-key cryptography, uses the same key for both encryption and decryption operations. This approach offers high computational efficiency and speed, making it suitable for bulk data encryption. The Data Encryption Standard (DES) and Advanced Encryption Standard (AES) represent the evolution of symmetric block ciphers, from the pioneering but now vulnerable DES to the contemporary AES that secures everything from wireless communications to government classified information (Daemen & Rijmen, 2002).

Asymmetric cryptography, or public-key cryptography, revolutionized digital security by introducing the concept of key pairs: a public key for encryption and a private key for decryption. This innovation solved the critical key distribution problem that plagued symmetric systems (Diffie & Hellman, 1976). The RSA algorithm, named after its inventors Rivest, Shamir, and Adleman, became the first practical public-key system and remains widely deployed in digital certificates, secure email, and authentication protocols. More recently, Elliptic Curve Cryptography (ECC) has gained prominence by offering equivalent security to RSA with significantly smaller key sizes, addressing the growing demand for efficient encryption in mobile devices and Internet of Things (IoT) applications (Hankerson et al., 2004).

The research motivation stems from the critical need to understand the trade-offs between different cryptographic approaches in modern security architectures. While symmetric algorithms excel in performance, they face key management challenges in large-scale distributed systems. Asymmetric algorithms solve key distribution but introduce computational overhead. The emergence of quantum computing threatens both RSA and current elliptic curve implementations, making it essential to understand their fundamental vulnerabilities and prepare for post-quantum cryptography transition (Bernstein & Lange, 2017). Furthermore, practical implementations often combine multiple algorithms in hybrid systems, requiring security architects to understand each technology's strengths and limitations.

This research aims to provide a comprehensive analysis of fundamental cryptographic principles, examining DES, AES, RSA, and Elliptic Curve algorithms from both theoretical and practical perspectives. The objectives include: (1) analyzing the mathematical foundations and operational mechanisms of each algorithm; (2) evaluating security strengths, vulnerabilities, and attack vectors; (3) comparing performance characteristics and resource requirements; (4) identifying optimal use cases and implementation scenarios; and (5) providing guidelines for selecting appropriate cryptographic solutions based on specific security requirements and operational constraints.

2. Literature Review

The evolution of cryptography spans millennia, from ancient substitution ciphers to modern quantum-resistant algorithms. Shannon's (1949) information theory laid the mathematical groundwork for analyzing cipher security, introducing concepts of confusion and diffusion that remain fundamental to

modern encryption design. The development of DES by IBM in the 1970s, standardized by NIST, marked the beginning of systematic cryptographic standardization (National Bureau of Standards, 1977). Despite initial controversy over its 56-bit key length, DES dominated commercial encryption for decades and influenced subsequent cipher designs.

The vulnerability of DES became apparent as computational power increased exponentially. The Electronic Frontier Foundation demonstrated a brute-force attack in 1998, cracking DES in 56 hours (Electronic Frontier Foundation, 1998). This prompted the development and standardization of AES through an open international competition won by the Rijndael algorithm (Daemen & Rijmen, 2002). AES provides flexible key lengths of 128, 192, and 256 bits with corresponding security levels. The algorithm's efficiency in both software and hardware implementations has made it the de facto standard for symmetric encryption across industries.

Public-key cryptography emerged from the groundbreaking work of Diffie and Hellman (1976), who proposed the revolutionary concept of asymmetric encryption and introduced the first practical key exchange protocol. RSA, developed by Rivest, Shamir, and Adleman (1978), provided the first complete public-key system based on the computational difficulty of factoring large composite numbers. The algorithm's security relies on the RSA problem: given n=pq where p and q are large primes, and c=m^e mod n, computing m without knowing the private key d is computationally infeasible. Decades of cryptanalytic research have validated RSA's security when implemented with proper key lengths and padding schemes (Boneh, 1999).

Elliptic Curve Cryptography was proposed independently by Koblitz (1987) and Miller (1986) as an alternative to traditional public-key systems. ECC achieves security equivalent to RSA with substantially smaller key sizes: a 256-bit ECC key provides security comparable to a 3072-bit RSA key (Barker, 2020). This efficiency advantage becomes critical in resource-constrained environments such as mobile devices, smart cards, and IoT sensors where computational power, memory, and battery life are limited. The mathematics of elliptic curves over finite fields provides the foundation for the elliptic curve discrete logarithm problem (ECDLP), which is believed to be harder than integer factorization (Hankerson et al., 2004).

Recent research has focused on practical implementation vulnerabilities and side-channel attacks. Kocher et al. (1999) demonstrated that implementation details such as timing variations and power consumption can leak cryptographic keys, leading to the development of constant-time algorithms and countermeasures. The emergence of quantum computing poses a fundamental threat to current asymmetric systems. Shor's algorithm (1997) can efficiently factor large integers and solve the discrete logarithm problem on quantum computers, potentially breaking RSA and ECC. This has catalyzed research into post-quantum cryptography, with NIST currently standardizing quantum-resistant algorithms (National Institute of Standards and Technology, 2022). Hybrid cryptosystems

that combine symmetric and asymmetric algorithms have become standard practice, leveraging the efficiency of symmetric encryption with the key management advantages of public-key systems (Menezes et al., 1996).

3. Research Methodology

This research employs a comprehensive analytical approach combining theoretical analysis, algorithm comparison, and security evaluation methodologies. The study framework encompasses four primary components: mathematical foundation analysis, security assessment, performance evaluation, and practical implementation analysis. This multi-dimensional approach enables thorough understanding of each cryptographic algorithm's characteristics, strengths, and limitations.

The mathematical foundation analysis examines the algorithmic structure and cryptographic primitives of DES, AES, RSA, and Elliptic Curve systems. For symmetric algorithms (DES and AES), this includes analysis of substitution-permutation network design, key scheduling mechanisms, and round functions. The examination covers the Feistel structure in DES, including its 16-round transformation process, S-box design criteria, and permutation operations. For AES, the analysis focuses on the substitution-permutation network architecture, including SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations, along with key expansion procedures for 128, 192, and 256-bit variants.

For asymmetric algorithms, the mathematical analysis examines number theory foundations and computational complexity assumptions. RSA analysis includes modular arithmetic operations, Euler's totient function, the Chinese Remainder Theorem optimization, and padding schemes such as OAEP (Optimal Asymmetric Encryption Padding). Elliptic Curve analysis covers point addition and doubling operations over finite fields, scalar multiplication algorithms, and the mathematical properties that make the elliptic curve discrete logarithm problem computationally hard.

Security assessment methodology evaluates each algorithm against established cryptanalytic techniques and attack models. The evaluation framework considers known attack vectors including brute-force attacks, differential cryptanalysis, linear cryptanalysis, algebraic attacks, and side-channel attacks. For each algorithm, the research analyzes: (1) theoretical security based on key length and mathematical hardness assumptions; (2) resistance to specific cryptanalytic techniques; (3) historical vulnerabilities and security incidents; (4) current threat landscape including quantum computing implications; and (5) recommended security parameters and implementation best practices.

Performance evaluation examines computational efficiency, resource requirements, and scalability characteristics. The analysis framework includes: encryption and decryption throughput measured in megabytes per second; computational complexity in terms of operation counts; memory footprint for key storage and operational buffers; energy consumption particularly relevant for mobile and IoT deployments; and implementation considerations for hardware acceleration support. The evaluation

considers both software implementations on general-purpose processors and specialized hardware implementations.

Comparative analysis methodology structures the evaluation across multiple dimensions to enable systematic comparison. The framework establishes evaluation criteria including: security strength measured by equivalent security level in bits; key management complexity and distribution requirements; computational overhead and performance characteristics; implementation complexity and resource requirements; and suitability for specific use cases and deployment scenarios. This structured approach enables identification of optimal algorithm selection for specific requirements and constraints.

4. Results and Discussion

4.1 Data Encryption Standard (DES) Analysis

DES represents a milestone in cryptographic history as the first publicly available encryption standard. The algorithm operates on 64-bit blocks using a 56-bit key (formally 64 bits with 8 parity bits). The encryption process employs a Feistel network structure with 16 rounds of substitution and permutation operations. Each round applies an initial permutation (IP), followed by 16 iterations of the round function combining expansion, key mixing, substitution through S-boxes, and permutation, concluding with a final permutation (IP^{-1}).

The Feistel structure divides the 64-bit input block into two 32-bit halves (L_0, R_0). Each round i transforms these halves using the function: $L_i = R_{i-1}$ and $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$, where f is the round function and K_i is the round key derived from the master key. This structure ensures that encryption and decryption use the same algorithm with reversed key schedule. The round function incorporates four operations: expansion (E) that expands 32 bits to 48 bits; XOR with the round key; substitution through eight S-boxes each mapping 6 bits to 4 bits; and permutation (P) that rearranges the 32-bit output.

The security analysis reveals fundamental vulnerabilities stemming from the 56-bit key space. With $2^{56} \approx 7.2 \times 10^{16}$ possible keys, modern computing power enables brute-force attacks within feasible timeframes. The Electronic Frontier Foundation's Deep Crack machine demonstrated this in 1998, breaking DES in 56 hours. Subsequent improvements reduced attack time to mere hours using distributed computing. Differential and linear cryptanalysis techniques provide theoretical attacks requiring fewer operations than brute force, though still impractical with 2^{47} and 2^{43} complexity respectively.

Despite known vulnerabilities, DES influenced modern cipher design significantly. The S-box design criteria, kept secret by NSA initially but later revealed, demonstrated sophisticated understanding of differential cryptanalysis years before its public discovery. Triple DES (3DES) emerged as a transitional solution, applying DES three times with two or three independent keys, achieving

effective security of 112 or 168 bits. While 3DES remains approved for some applications, its performance overhead and 64-bit block size make it suboptimal for modern systems, leading to AES migration recommendations.

Table 1. DES Algorithm Characteristics

Parameter	Value
Block Size	64 bits
Key Length	56 bits (64 bits with parity)
Number of Rounds	16
Structure	Feistel Network
Security Level	Broken (vulnerable to brute force)
Performance	Moderate (hardware optimized)

Source: Stallings (2017); National Bureau of Standards (1977)

4.2 Advanced Encryption Standard (AES) Analysis

AES emerged from the NIST competition to replace DES, with the Rijndael algorithm selected in 2001 as the new standard. Unlike DES's Feistel structure, AES employs a substitution-permutation network operating on a 4×4 array of bytes called the state matrix. The algorithm supports key lengths of 128, 192, and 256 bits with corresponding 10, 12, and 14 rounds. Each round applies four transformations: SubBytes performs byte-level substitution using an S-box derived from multiplicative inverse in GF(2^8); ShiftRows cyclically shifts rows by different offsets; MixColumns applies a linear transformation mixing columns through matrix multiplication; and AddRoundKey XORs the state with round key.

The mathematical foundation of AES provides provable security properties. The S-box design ensures resistance to linear and differential cryptanalysis through specific criteria: no fixed points ($S(x) \neq x$) except $x=0$, minimized differential uniformity, and balanced Boolean functions. MixColumns provides diffusion by ensuring each output byte depends on all four input bytes of a column. The combination of substitution (confusion) and permutation (diffusion) operations ensures that small changes in plaintext or key propagate extensively through subsequent rounds, achieving Shannon's principles of strong encryption.

Security analysis demonstrates AES's robust resistance to known attacks. The best published attacks against full AES-128 require approximately $2^{126.1}$ operations, only marginally better than brute force 2^{128}. For AES-256, related-key attacks theoretically reduce security to $2^{99.5}$ operations, still computationally infeasible and irrelevant for proper key management. Side-channel attacks pose practical implementation risks, particularly cache-timing attacks exploiting table lookups in software implementations. Constant-time implementations and AES-NI hardware instructions mitigate these vulnerabilities effectively.

Performance analysis reveals AES's efficiency advantages. Modern processors include AES-NI instruction set extensions enabling hardware acceleration with throughput exceeding 10 GB/s on contemporary CPUs. Software implementations achieve 100-500 MB/s depending on optimization level and processor architecture. The algorithm's regular structure enables efficient parallel implementation on GPUs and hardware accelerators. AES's 128-bit block size provides adequate security for most applications, though birthday bound considerations limit data volume per key to approximately 2^{64} blocks (256 petabytes).

Table 2. AES and DES Comparison

Characteristic	DES	AES
Development Year	1977	2001
Key Length	56 bits	128, 192, 256 bits
Block Size	64 bits	128 bits
Rounds	16	10, 12, 14
Security Status	Deprecated	Secure
Performance	Moderate	High (HW accelerated)

Source: Daemen & Rijmen (2002); Stallings (2017)

4.3 RSA Cryptosystem Analysis

RSA revolutionized cryptography by providing the first practical public-key system. The algorithm's security relies on the computational difficulty of factoring large composite numbers. Key generation selects two large prime numbers p and q, computes n = pq as the modulus, and φ(n) = (p-1)(q-1) as Euler's totient. The public exponent e is chosen coprime to φ(n), typically 65537 for efficiency, and the private exponent d is computed as e^{-1} mod φ(n). The public key (n, e) enables encryption c = m^e mod n, while the private key (n, d) enables decryption m = c^d mod n.

The mathematical correctness follows from Euler's theorem: if gcd(m, n) = 1, then m^φ(n) ≡ 1 (mod n). Since ed ≡ 1 (mod φ(n)), we have ed = 1 + kφ(n) for some integer k. Therefore, c^d = (m^e)^d = m^(ed) = m^(1+kφ(n)) = m · (m^φ(n))^k ≡ m · 1^k = m (mod n). This elegant mathematical property enables asymmetric encryption where the decryption key cannot be efficiently derived from the public key.

Security analysis centers on the RSA problem's computational hardness. The best known factoring algorithms are the General Number Field Sieve (GNFS) for general integers with sub-exponential complexity O(exp((64/9)^(1/3) · (ln n)^(1/3) · (ln ln n)^(2/3))), and the Special Number Field Sieve for numbers with special forms. Current recommendations specify 2048-bit keys for short-term security and 3072 or 4096 bits for long-term protection. The RSA-2048 challenge remains unsolved, with estimated security equivalent to 112-bit symmetric keys.

Implementation vulnerabilities pose significant risks beyond mathematical security. Textbook RSA (unpadded encryption) is vulnerable to chosen-ciphertext attacks and mathematical attacks exploiting message structure. Modern implementations use OAEP (Optimal Asymmetric Encryption Padding) incorporating randomness and redundancy. Timing attacks can extract private keys by measuring decryption times, requiring constant-time implementations. The Chinese Remainder Theorem optimization, while improving performance by 4x, introduces vulnerability to fault attacks where induced computation errors leak the private key.

Performance characteristics reveal RSA's computational intensity. Encryption with small public exponent e=65537 requires minimal operations, enabling fast signature verification. Decryption and signing, using the large private exponent, are 100-1000 times slower than encryption. Hardware accelerators and algorithmic optimizations using CRT reduce this overhead. The performance gap motivates hybrid cryptosystems where RSA encrypts symmetric keys for bulk data encryption with AES or other fast symmetric ciphers.

4.4 Elliptic Curve Cryptography Analysis

Elliptic Curve Cryptography provides public-key functionality with dramatically smaller key sizes compared to RSA. The system operates on elliptic curves defined over finite fields, typically in the Weierstrass form $y^2 = x^3 + ax + b \pmod p$ for prime fields. The security foundation is the Elliptic Curve Discrete Logarithm Problem (ECDLP): given points P and Q = kP on an elliptic curve, finding the scalar k is computationally infeasible. This problem appears harder than integer factorization, enabling shorter keys with equivalent security.

The mathematical operations on elliptic curves define a group structure. Point addition combines two points to produce a third point on the curve through geometric and algebraic operations. Point doubling represents adding a point to itself. Scalar multiplication kP, computed through repeated point additions and doublings using efficient algorithms like double-and-add or windowed methods, forms the cryptographic primitive. The key generation selects a random scalar d as private key and computes Q = dG as public key, where G is a standardized base point.

Security analysis demonstrates ECC's efficiency advantage. A 256-bit ECC key provides security equivalent to a 3072-bit RSA key, approximately 128-bit symmetric security level. The best known attacks against ECDLP, such as Pollard's rho algorithm, have fully exponential complexity $O(\sqrt{n})$ where n is the curve order, compared to sub-exponential factoring algorithms. This fundamental difference enables much shorter keys. The NIST P-256, P-384, and P-521 curves are widely deployed, though concerns about potential backdoors in their random parameters have led to increased use of alternative curves like Curve25519 with verifiable generation methods.

Implementation considerations are critical for ECC security. Side-channel attacks exploit implementation details such as conditional branches in point addition formulas that leak information

about the secret scalar. Constant-time implementations using complete addition formulas prevent timing and power analysis attacks. Invalid curve attacks attempt to trick implementations into computing on weak curves by providing malicious public keys. Point validation ensures received points actually lie on the intended curve and belong to the correct subgroup.

Performance analysis reveals ECC's advantages for resource-constrained environments. The smaller key sizes reduce bandwidth requirements for key exchange and digital signatures. Computational costs for scalar multiplication are manageable even on embedded processors, making ECC practical for IoT devices, smart cards, and mobile applications. Modern implementations achieve ECDH key exchanges in milliseconds and ECDSA signatures in similar timeframes, substantially faster than equivalent-security RSA operations. Hardware acceleration support in modern processors further improves performance.

Table 3. RSA and Elliptic Curve Cryptography Comparison

Parameter	RSA	ECC
Key Size (128-bit security)	3072 bits	256 bits
Mathematical Basis	Integer factorization	Discrete logarithm on EC
Encryption Speed	Fast (small e)	Moderate
Decryption Speed	Slow	Moderate
Bandwidth Efficiency	Low (large keys)	High (small keys)
Best Use Case	Signature verification	Mobile & IoT devices

Source: Hankerson et al. (2004); Barker (2020)

4.5 Hybrid Cryptosystems and Practical Applications

Modern secure systems predominantly employ hybrid cryptosystems that combine symmetric and asymmetric algorithms to leverage each technology's strengths. The standard approach uses asymmetric cryptography (RSA or ECC) to establish a secure channel for symmetric key exchange, then performs bulk data encryption with efficient symmetric algorithms (typically AES). This architecture solves the key distribution problem while maintaining performance for large data volumes. The TLS protocol exemplifies this approach: during the handshake, RSA or ECDHE establishes session keys, then AES-GCM encrypts application data.

Selection criteria for cryptographic algorithms depend on specific requirements and constraints. For bulk data encryption at rest, AES-256 provides strong security with excellent performance on modern hardware. Network communications benefit from TLS 1.3 with ECDHE key exchange and AES-GCM for authenticated encryption. Digital signatures commonly use RSA-2048 or ECDSA P-256 based on performance and compatibility requirements. Resource-constrained IoT devices prefer ECC due to smaller key sizes and lower computational requirements. Government and high-security applications often mandate specific algorithms and key lengths based on classified data sensitivity levels.

The quantum computing threat necessitates planning for post-quantum cryptography transition. Shor's algorithm can break RSA and ECC on sufficiently powerful quantum computers, though practical large-scale quantum computers remain years or decades away. NIST's post-quantum cryptography standardization process has selected algorithms based on lattice problems, hash functions, and code-based cryptography. Organizations handling long-term sensitive data should begin implementing crypto-agility: the ability to rapidly transition between cryptographic algorithms as threats evolve and new standards emerge.

5. Conclusion

This comprehensive analysis of symmetric and asymmetric cryptography demonstrates the evolution and current state of encryption technologies fundamental to information security. DES, despite its historical significance and influence on modern cipher design, has been superseded by AES due to inadequate key length. AES represents the current standard for symmetric encryption, providing robust security, excellent performance, and hardware acceleration support across diverse platforms. The algorithm's proven resistance to cryptanalytic attacks and flexible key length options ensure its continued relevance for the foreseeable future.

In the asymmetric domain, RSA remains widely deployed despite computational intensity, particularly for signature verification where small public exponents enable efficiency. However, the trend favors Elliptic Curve Cryptography for new implementations due to superior key size efficiency and performance characteristics. ECC's advantages become increasingly important as cryptographic operations migrate to resource-constrained devices in mobile and IoT environments. The mathematical hardness of ECDLP provides security equivalent to RSA with dramatically smaller keys, reducing bandwidth, storage, and computational requirements.

The practical reality of modern cryptographic systems is the hybrid approach combining symmetric and asymmetric algorithms. This architecture leverages asymmetric cryptography's key management advantages for secure key exchange while utilizing symmetric encryption's performance for bulk data protection. Understanding the trade-offs between algorithms enables security architects to make informed decisions based on specific requirements including performance needs, security level, computational resources, and operational constraints.

Looking forward, the quantum computing threat requires proactive preparation for cryptographic transition. While current algorithms remain secure against classical computing attacks, organizations should implement crypto-agility to enable rapid algorithm updates as the threat landscape evolves. The ongoing development and standardization of post-quantum algorithms will shape the next generation of cryptographic systems, potentially requiring hybrid approaches combining classical and quantum-resistant algorithms during transition periods.

Future research directions include optimizing post-quantum algorithms for practical deployment, developing side-channel resistant implementations, and creating efficient hybrid schemes balancing security with performance. As computing paradigms evolve with quantum computing, homomorphic encryption, and secure multi-party computation, the fundamental principles examined in this research will continue guiding the development of next-generation cryptographic systems protecting digital information in an increasingly connected world.

References

Barker, E. (2020). Recommendation for key management: Part 1 – General. NIST Special Publication 800-57 Part 1, Revision 5. National Institute of Standards and Technology.

Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. Nature, 549(7671), 188-194.

Boneh, D. (1999). Twenty years of attacks on the RSA cryptosystem. Notices of the American Mathematical Society, 46(2), 203-213.

Daemen, J., & Rijmen, V. (2002). The design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag.

Diffie, W., & Hellman, M. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.

Electronic Frontier Foundation. (1998). Cracking DES: Secrets of encryption research, wiretap politics & chip design. O'Reilly Media.

Hankerson, D., Menezes, A. J., & Vanstone, S. (2004). Guide to elliptic curve cryptography. Springer-Verlag.

Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of Computation, 48(177), 203-209.

Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. In Advances in Cryptology - CRYPTO '99 (pp. 388-397). Springer.

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.

Miller, V. S. (1986). Use of elliptic curves in cryptography. In Advances in Cryptology - CRYPTO '85 (pp. 417-426). Springer.

National Bureau of Standards. (1977). Data Encryption Standard. FIPS Publication 46. U.S. Department of Commerce.

National Institute of Standards and Technology. (2022). Post-quantum cryptography standardization. Retrieved from https://csrc.nist.gov/projects/post-quantum-cryptography

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.

Shannon, C. E. (1949). Communication theory of secrecy systems. Bell System Technical Journal, 28(4), 656-715.

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.

Stallings, W. (2017). Cryptography and network security: Principles and practice (7th ed.). Pearson Education.