Review of Muldisciplinary Academic and Pratice Studies
ISSN: 3110-0635, Vol 3, No 2, 2026, 101-113 https://doi.org/10.61401/rmaps.v3i2.527

Modern Cryptography: A Comprehensive Review of Symmetric and Asymmetric Encryption Algorithms

Maha A. Sayal
University of Thi-Qar Iraq
maha.A.Sayal@utq.edu.iq

Article History
Received on 4 June 2026
1st Revision on 6 July 2026
2nd Revision on 15 July 2026
Accepted on 3 August 2026

Abstract

Purpose: This study aims to provide a comprehensive comparative review of modern cryptographic algorithms, specifically examining symmetric encryption techniques (AES and RC4) and asymmetric encryption techniques Rivest Shamir Adleman (RSA), to evaluate their performance, structure, and suitability for contemporary digital security applications.

Research Methodology: A systematic literature review methodology was employed to analyze and compare cryptographic primitives. Primary and secondary sources, including peer-reviewed journals and conference proceedings published predominantly within the last five years, were examined. Comparative parameters including key size, block size, structure, speed, security level, and computational cost were used to evaluate each algorithm.

Results: The findings demonstrate that symmetric algorithms, particularly Advanced Encryption Standard (AES), exhibit superior speed and computational efficiency compared to asymmetric algorithms such as RSA. AES with a 128-bit block size and variable key lengths of 128, 192, or 256 bits offers excellent security through its substitution-permutation network. RC4, as a stream cipher, proves effective for real-time and resource-constrained environments. RSA, while computationally expensive, remains indispensable for secure key exchange and digital authentication.

Conclusions: Modern cryptographic systems benefit from hybrid approaches integrating both symmetric and asymmetric algorithms to balance security and computational efficiency. AES remains the gold standard for symmetric encryption, while RSA continues to underpin public-key infrastructure.

Limitations: This study is limited to three primary algorithms (AES, RC4, RSA) and does not encompass all emerging lightweight cryptographic protocols or post quantum cryptographic schemes.

Contributions: This paper contributes a structured, comparative analysis of foundational cryptographic algorithms, bridging classical cryptographic theory with current implementation standards, providing researchers and practitioners with a consolidated reference framework.

Keywords*: Asymmetric Cryptography, Cryptographic Security, Digital Encryption, Public Key Infrastructure.*

How to Cite: Sayal, M. A. (2026). Modern Cryptography: A Comprehensive Review of Symmetric and Asymmetric Encryption Algorithms. *Review of Muldisciplinary Academic and Pratice Studies*, *3*(2), 101-113.

1. Introduction

Cryptography, derived from the Greek words kryptós and graphein meaning "hidden writing," represents one of the oldest and most critical disciplines in information security. For over two millennia, cryptographic techniques have evolved from simple substitution ciphers used in ancient civilizations to the complex mathematical constructs underpinning modern digital communications (Stamp & Low, 2007). In an era defined by exponential growth in internet connectivity, cloud computing, and digital

transactions, the importance of robust cryptographic mechanisms cannot be overstated (Kapoor & Thakur, 2022). The increasing sophistication of cyber threats has necessitated continuous advancements in encryption algorithms to safeguard sensitive information across diverse platforms and applications (Luo, Chen, Chen, & Cheng, 2024).

The fundamental objective of cryptography is to ensure information confidentiality, integrity, authentication, and non-repudiation in communications between parties (Barker & Roginsky, 2011). Modern cryptographic systems are broadly classified into two categories: symmetric-key cryptography, where the same key is used for both encryption and decryption, and asymmetric-key cryptography, which employs a pair of mathematically related public and private keys (Ghosh, Biradar, Shinde, Bhojaned, & Shirapure, 2015). Each paradigm carries distinct advantages and limitations that influence their suitability for specific use cases and deployment environments (Kamel Saadeldin, Abogabal, & Shahin, 2026).

Symmetric encryption algorithms, such as the Advanced Encryption Standard (AES) and the RC4 stream cipher, are widely adopted for their computational efficiency and suitability for bulk data encryption (Nurrohmah & Saputra, 2026). AES, standardized by the National Institute of Standards and Technology (NIST) in 2001, has become the de facto standard for government and commercial data protection worldwide (Zhang, 2021). RC4, despite its historical vulnerabilities in certain implementations, remains relevant in constrained and real-time processing environments due to its simplicity and speed (Jindal & Singh, 2017).

Asymmetric cryptography, further popularized through the RSA algorithm developed by Rivest, Shamir, and Adleman in 1977, revolutionized the field by enabling secure key exchange without the need for a pre-shared secret (Wijaya & Ainun, 2025). RSA and its variants form the backbone of public key infrastructure (PKI), underpinning protocols such as TLS/SSL, digital signatures, and certificate based authentication (Bos et al., 2018). However, the computational overhead associated with asymmetric algorithms limits their use in resource-constrained environments and high-throughput applications.

Despite the abundance of individual studies on cryptographic algorithms, there remains a need for integrative comparative analyses that synthesize findings across multiple algorithm types within the context of contemporary security requirements. Prior reviews have often focused narrowly on single algorithm families or specific application domains, leaving practitioners without comprehensive guidance for algorithm selection. Furthermore, rapid advances in computing power, including the emergence of quantum computing, have renewed interest in evaluating the long-term security of existing cryptographic standards (Alagic et al., 2022).

This paper presents a structured comparative review of three foundational cryptographic algorithms AES, RC4, and RSA examining their operational mechanisms, performance characteristics, and security properties. The novelty of this study lies in its unified comparative framework that evaluates these algorithms across multiple parameters relevant to modern deployment scenarios, including computational efficiency, key management complexity, security robustness, and applicability to contemporary network security protocols. The primary objective is to provide researchers and security practitioners with a consolidated reference that facilitates informed algorithm selection and hybrid cryptographic system design. Specifically, this review seeks to: (1) analyze the internal mechanisms of AES, RC4, and RSA; (2) compare their performance across critical parameters; and (3) discuss their applicability in current information security contexts, thereby bridging foundational cryptographic theory with practical implementation guidance.

2. Literature Review and Hypothesis/es Development

2.1 Evolution of Cryptographic Algorithms

The evolution of cryptographic algorithms reflects the ongoing arms race between cryptanalysts and cybersecurity practitioners. Early symmetric ciphers such as the Data Encryption Standard (DES), introduced in 1977, laid the groundwork for modern block cipher design; however, DES's 56-bit key

length proved inadequate against brute-force attacks as computational power increased throughout the 1990s (Zacharias & Lao, 2024) . The subsequent development of Triple-DES (3DES) extended the security of DES by applying the algorithm three times with different keys, but at the cost of significant computational overhead. The selection of AES in 2001 marked a turning point in symmetric cryptography, offering a configurable key length and a more sophisticated mathematical structure based on the Rijndael cipher developed by Joan Daemen and Vincent Rijmen (Syam & Niswaty, 2025).

In parallel, the development of stream ciphers provided alternatives optimized for continuous data streams and hardware-constrained environments. RC4, developed by Ron Rivest in 1987 and published in 1994, became ubiquitous in wireless security protocols including WEP and early implementations of WPA and SSL/TLS (Hammood, Yoshigoe, & Sagheer, 2015). However, subsequent cryptanalysis revealed statistical biases in RC4's output, particularly in its initial keystream bytes, which led to its deprecation in TLS 1.3 and modern wireless security standards. More recent stream ciphers such as ChaCha20 have been proposed and adopted as secure alternatives, though RC4 remains in use in legacy and specialized systems.

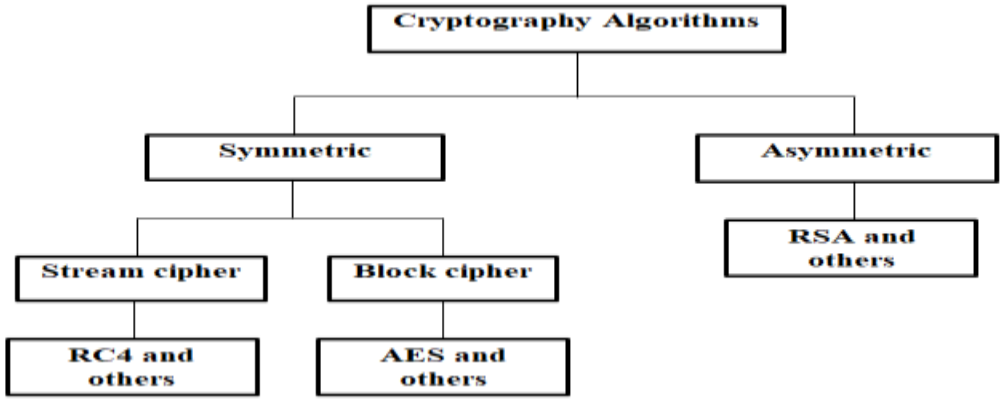

Figure 1. Cryptography algorithms

The figure 1 presents a classification of cryptographic algorithms into two main categories, namely symmetric cryptography and asymmetric cryptography. Symmetric cryptography uses a single shared key for both encryption and decryption processes and is further divided into stream ciphers and block ciphers. Stream ciphers encrypt data one bit or byte at a time, with RC4 (Rivest Cipher 4) being a well-known example. In contrast, block ciphers encrypt data in fixed size blocks, providing stronger security and wider adoption in modern systems. AES (Advanced Encryption Standard) is one of the most widely used block cipher algorithms. Asymmetric cryptography, on the other hand, employs a pair of keys consisting of a public key and a private key. This category includes RSA (Rivest Shamir Adleman) and other public key algorithms that are commonly used for secure key exchange, digital signatures, and authentication. The diagram illustrates the hierarchical organization of cryptographic techniques and demonstrates how different algorithms are classified according to their encryption methods and key management mechanisms.

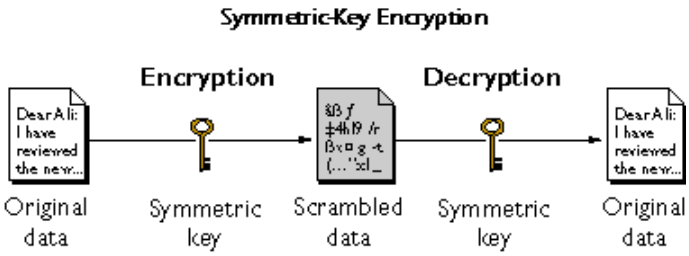

Figure 2. A technique for symmetric-key

The figure 2 illustrates the process of symmetric key encryption, in which the same secret key is used for both encryption and decryption. Initially, the original data, known as *plaintext*, is processed through an encryption algorithm together with a symmetric key. This process transforms the readable information into an unreadable form called *ciphertext* or scrambled data, preventing unauthorized users from accessing its contents. When the encrypted data reaches the intended recipient, the same symmetric key is applied in the decryption process to convert the ciphertext back into its original plaintext form. The security of this method depends on the confidentiality of the shared key, as anyone possessing the key can both encrypt and decrypt the data. Due to its high speed and computational efficiency, symmetric encryption is widely used for securing large volumes of data in modern communication systems.

2.2 Cipher Enclose

When a block of information is encrypted using the same key each time, the block cipher is named. Specifically, when using a block cipher, the same code key will encrypt the same chipper-text every time in a different stream encryption. A block cipher is referred to as an iterated cipher if the chip is determined by repeatedly applying an iterative applying the round function on the plaintext. With every round, the entered document is coupled with a round key. The Advanced Encryption Standard (AES) was introduced in 2001 for the encryption of electronic data was created by NIST, the National Institute of Standards and Technology. Even the government of the United States recognized the standards for data encryption (DES) continued value in the 1990s. NIST started searching for a DES substitute in 1997 and eventually decided to use the AES algorithm in its place. The AES must be a block cipher capable of handling blocks with 128 bits. making use of the estimated 128, 192, and 256-bit keys.

AES has the ability to both encrypt and decrypt data for the symmetric block cipher. "The Encryption" transforms the data into an unintelligible format known as a text cipher. The process of returning the data to its original format is known as decryption. Unlike DES, AES has a variable number of rounds since it is dependent on the key's length. In this case, AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. A distinct 128-bit round key is used for each of these rounds. The AES standard the round is calculated using the key of the 128-bit key. AES is separated into four fundamental parts of operations that handle data at the byte or bit level. The assortment of bytes is organized as a 4x4 matrix. using the "State" function. The four crucial steps in the AES algorithm are as follows:

Step 1: Sub Bytes & InvSubBytes Transformation: a substation table (S-Box) is used to transform a non-linear substitution byte. This stage is referred to as Sub Bytes for the forward process' byte-by-byte substitution. InvSubBytes is the name of InvSubBytes is the name of the comparison substitution process utilized in the decryption stage. This procedure uses a 16 x 16 query table to replace a specified byte in the input-state. The AES-based byte matrix is called S-box, Table 1. The parts in the query table are created by destroying the bit-level relationships within each byte using GF's multiplicative inverses and bit mix ideas. The substation byte procedure is depicted in figure 3.

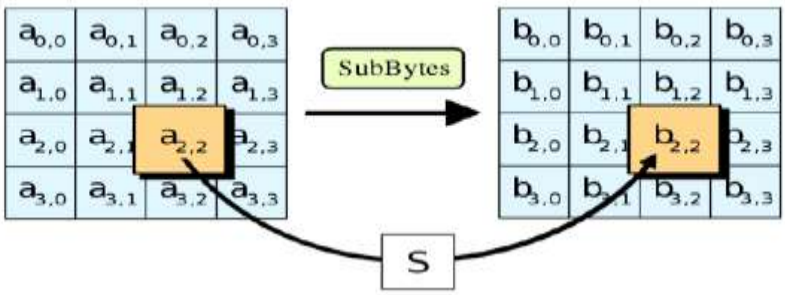

Figure 3. The sub byte's function

The figure 3 illustrates the SubBytes transformation in the Advanced Encryption Standard (AES) algorithm. During this step, each byte in the AES state matrix is replaced with a corresponding byte using a predefined Substitution Box (S-Box). In the example, the element (Luo et al., 2024) from the

input state matrix is selected and passed through the S-Box, represented by the symbol S, producing a new value (in the output state matrix. This substitution process is applied independently to all bytes in the 4 × 4 state matrix.

The primary purpose of the SubBytes operation is to introduce nonlinearity into the encryption process, making the relationship between the plaintext and ciphertext more complex and difficult for attackers to analyze. By replacing each byte according to the carefully designed S-Box lookup table, AES enhances its resistance against cryptanalytic attacks, particularly linear cryptanalysis and differential cryptanalysis. As a result, SubBytes serves as a critical component in ensuring the overall security and robustness of the AES encryption algorithm (Montazeri, Kebriaei, & Araabi, 2020).

Table 1. The AES's inverse S-box

X\Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

The same byte substitution technique is used for decoding in the inversion byte replacement operation. Table (2) displays the inverse S-box.

Table X. AES inverse s box values used in the in sub-bytes transformation

X\Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Step 2: The Transformation of Shift Row and InvShiftRow, the goal of this update is to jumble the byte by rearranging the state array's rows inside each 128-bit block during the forward operation. The associated alteration during the backward operation is called InvShift Row. The algorithm is linear thanks to the two-featured Shift Rows transformation scheme. a straightforward byte transposition in which the One to three bytes make up the left shift offset., and the last three state rows have bytes that are episodically shifted. To complete the decryption step, the Inverse Shift Rows transformation has to use the data to flip the state matrix rows' encrypted message in the other manner (Budler, Župič, & Trkman, 2021).

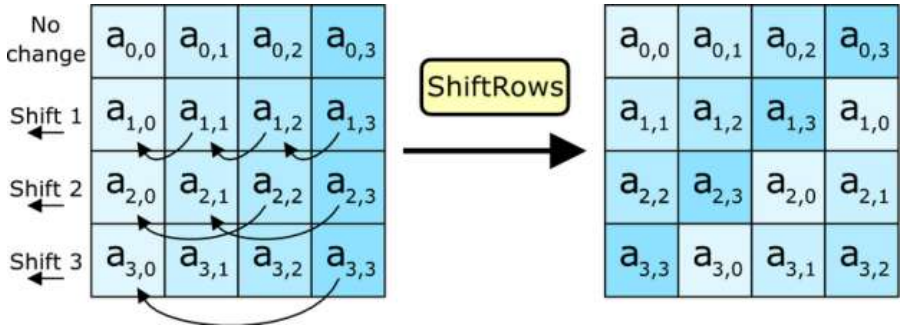

Figure 4. The shift row transformation

Step 3: (Inverse Mix Columns & Mix Columns) Transformation: Use Mix Columns to independently stir up the data in each column throughout the encryption process. The analogous transformation during the decryption process is called InvMix Columns. Additionally, mixing up the 128-bit input block is the goal. After ten rounds of preparation, the ciphertext's bits are dependent on the plaintext's bits. the MixColumn process in conjunction with the ShifRows method. Refer to figure 5. The Mix Column's phase is a linear conversion that combines each state matrix column. The Mix Column operation is the primary diffusion component in the AES. Every four-byte column is treated as a vector and multiplied by a predetermined four-byte matrix. The matrix has consistent entries. Depending on the inverse matrix, the decryption methodology uses the same conversion mechanism as the inverse transformation of Mix Columns. The GF (2^8) is used for coefficient addition and multiplication (Wangu, Mangnus, & van Westen, 2020).

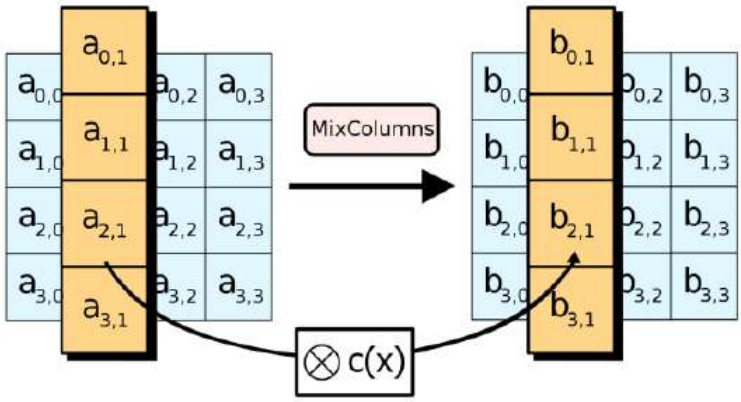

Figure 5. The mix columns operation

Step 4: Add-round-key Conversion, one technique for creating round keys is key scheduling, often known as round key expansion. This is a simple, direct XOR operation between the working state and the round-key. Refever to figure 6.

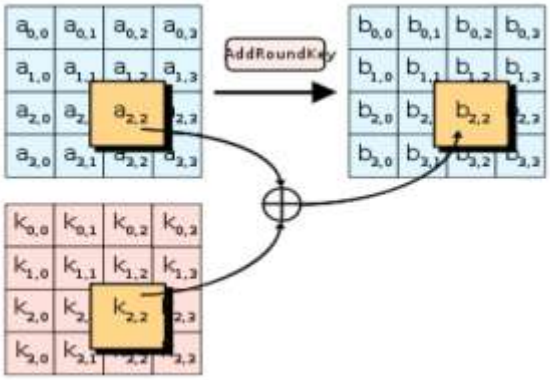

Figure 6. The Add Round Key operation

Figure 7. Flowchart representation of the conventional AES system, showing the main stages of the encryption procedure.

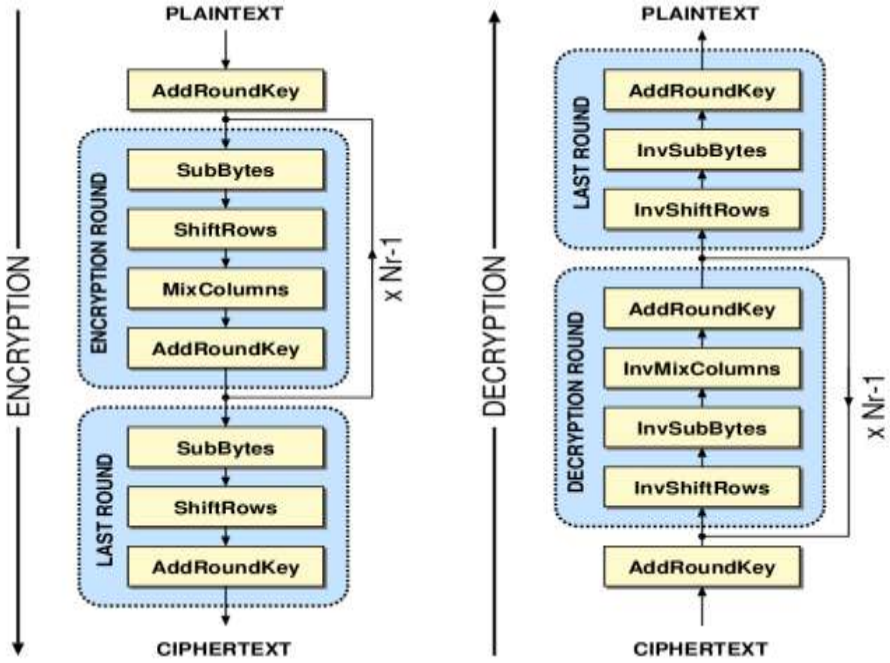

Figure 7. The AES algorithm flowchart illustration

The figures illustrate two important components of the Advanced Encryption Standard (AES) algorithm: the AddRoundKey transformation and the overall encryption–decryption process. In the AddRoundKey stage, each byte of the state matrix is combined with the corresponding byte of the round key using a bitwise Exclusive OR (XOR) operation. For example, the state element a2,2 is XORed with the round key element k2,2, producing the output value b2,2. This operation is applied to every byte in the 4 × 4 state matrix and serves to incorporate the secret key into the encryption process, ensuring that the resulting ciphertext depends directly on the encryption key (Roggeveen & Sethuraman, 2021).

The second figure presents the complete AES workflow for both encryption and decryption. During encryption, plaintext first undergoes an initial AddRoundKey operation, followed by multiple rounds consisting of SubBytes, ShiftRows, MixColumns, and AddRoundKey transformations. These rounds are repeated Nr−1 times, where Nr represents the total number of rounds determined by the key length. In the final round, the MixColumns operation is omitted, and the output becomes the ciphertext. During decryption, the process is reversed using the inverse transformations InvShiftRows, InvSubBytes,

AddRoundKey, and InvMixColumns. The decryption rounds are also repeated Nr−1 times, with the final round excluding InvMixColumns to recover the original plaintext. Together, these transformations provide AES with strong confusion and diffusion properties, making it one of the most secure and widely used symmetric encryption algorithms (Li et al., 2020).

2.3 Symmetric Cryptography: Mechanisms and Applications

Symmetric cryptography constitutes the foundation of efficient data encryption in contemporary systems. The AES algorithm operates on a 4×4 byte state matrix, applying successive rounds of four transformations: SubBytes (non-linear byte substitution using an S-Box), ShiftRows (cyclic row transposition), MixColumns (linear column mixing), and AddRoundKey (XOR with a derived round key) (Broadbent & De Barba, 2023). The number of rounds varies with key length: 10 rounds for 128-bit, 12 rounds for 192-bit, and 14 rounds for 256-bit keys. This design ensures both confusion and diffusion, the two fundamental properties of a secure cipher as defined by Shannon in 1949, referenced in (Sharma, Koohang, Rana, Abed, & Dwivedi, 2023).

Recent research has extensively examined AES hardware and software implementations across diverse platforms. demonstrated that hardware-accelerated AES implementations on FPGA platforms achieve throughputs exceeding 10 Gbps, making AES suitable for high-speed network encryption. Software implementations leveraging AES-NI CPU instructions achieve near-zero latency overhead in modern desktop and server environments (Verma & Arora, 2025). AES has been extensively adopted in cloud storage encryption, VPN protocols, full-disk encryption systems, and mobile communication standards (Luo et al., 2024). Its security against differential and linear cryptanalysis remains robust under standard computational assumptions, with no practical attacks known against AES-128 or higher (Hasan & Talukder, 2023).

RC4's operational simplicity consisting of a Key Scheduling Algorithm (KSA) and a Pseudo-Random Generation Algorithm (PRGA) enables rapid software implementation with minimal memory requirements (Maity, Sinha, & Sinha, 2017). The KSA initializes a 256-byte permutation array using the secret key, while the PRGA generates a pseudorandom keystream that is XORed with plaintext to produce ciphertex. Despite known vulnerabilities in specific usage contexts, particularly when used with static or related keys, RC4 remains deployed in environments where computational resources are severely constrained or where backward compatibility is required. Recent studies have proposed enhanced variants, such as Modified RC4 (MRC4), which address known biases through additional key mixing phases and extended initialization sequences (Hodges & Moore, 2025).

2.4 Asymmetric Cryptography: Public Key Infrastructure and RSA

The introduction of public key cryptography fundamentally transformed information security by enabling secure key exchange over insecure channels without prior secret sharing. The RSA algorithm, proposed operationalizes this concept through the mathematical difficulty of factoring the product of two large prime numbers. RSA's security guarantees are therefore contingent on the infeasibility of integer factorization, a problem for which no polynomial-time classical algorithm is known. RSA key sizes have grown over time in response to advances in factoring algorithms and computational power; contemporary recommendations specify minimum key sizes of 2048 bits, with 3072 or 4096 bits for long-term security applications (Huang, 2021).

RSA remains central to modern public key infrastructure (PKI), underpinning X.509 digital certificates, the TLS handshake protocol, secure email (S/MIME and PGP), and code signing systems. Hybrid cryptographic systems, which use RSA for secure key exchange and AES for bulk data encryption, represent the dominant paradigm in contemporary secure communications protocols. The computational cost of RSA operations particularly modular exponentiation scales with key size, imposing performance constraints in latency-sensitive applications and resource-constrained devices such as IoT sensors (Huang, 2021). Alternative asymmetric algorithms, including Elliptic Curve Cryptography (ECC), have gained traction as more computationally efficient alternatives offering equivalent security with shorter key lengths (Agarwal & Mannil, 2023).

2.5 Comparative Studies and Current Implementation Challenges

Comparative analyses of AES, RC4, and RSA have been conducted across multiple dimensions including throughput, latency, memory footprint, and energy consumption. Provided an early systematic comparison of cryptographic algorithms, finding AES superior in security-performance trade-off for general-purpose applications. More recent comparative studies have extended this analysis to mobile, embedded, and cloud environments. A significant trend in contemporary research is the evaluation of cryptographic performance in constrained IoT environments, where energy efficiency and memory usage are critical constraints alongside security (Zacharias & Lao, 2024).

The emergence of post-quantum cryptography represents a critical frontier in the field. Shor's algorithm, if implemented on a sufficiently powerful quantum computer, could factor the large integers upon which RSA security is based in polynomial time, effectively rendering RSA and other number-theoretic cryptosystems insecure. NIST has been conducting a post-quantum cryptographic standardization process since 2016, with several candidate algorithm including CRYSTALS-Kyber and CRYSTALS Dilithium recently selected as standards. Symmetric algorithms such as AES are considered more resilient to quantum attacks, with the effective security of AES-256 estimated to remain at approximately 128 bits against Grover's quantum search algorithm. This divergence in quantum resilience further underscores the importance of understanding the respective strengths and limitations of symmetric versus asymmetric cryptographic approaches in the context of long-term security planning (Agarwal & Mannil, 2023).

Recent implementations of cryptographic algorithms have increasingly incorporated Hardware Security Modules (HSMs), Trusted Platform Modules (TPMs), and secure enclaves to protect key material and cryptographic operations from software-level attacks. Side-channel attack mitigations, including timing attack countermeasures and cache-resistant implementations, have become standard components of production-grade cryptographic libraries such as OpenSSL and libsodium. The integration of cryptographic algorithms into zero-trust security architectures and end-to-end encrypted communication platforms represents a dominant application trend, driving demand for performant, standards-compliant cryptographic implementations (Bui, Nguyen, Le, Nguyen, & Nguyen, 2026).

3. Methodology

This study adopts a Systematic Literature Review (SLR) methodology to examine, evaluate, and synthesize existing research on symmetric and asymmetric cryptographic algorithms, with a specific focus on AES, RC4, and RSA. The review process follows a structured protocol inspired by PRISMA principles, which are widely used in scientific literature reviews to ensure transparency, rigor, and reproducibility in the selection and analysis of studies. The literature was collected from major academic databases, including IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, and Google Scholar, using carefully selected keywords related to encryption algorithms, cryptographic performance, and security evaluation. The search scope was primarily limited to peer-reviewed journal articles and conference papers published within a recent timeframe, while also incorporating foundational studies to provide theoretical and historical context.

The selection of studies was guided by clear inclusion and exclusion criteria to ensure relevance and quality. Included studies were required to focus directly on the operational mechanisms, performance characteristics, security properties, or comparative evaluations of AES, RC4, and RSA. Studies that addressed unrelated cryptographic areas or lacked empirical or analytical depth were excluded from the review. After filtering, the remaining studies were systematically organized and prepared for further analysis to ensure consistency and comparability across different research sources.

To conduct the comparative analysis, a structured framework was developed based on key evaluation parameters, including algorithm standardization, structural design, key size configuration, processing rounds, computational efficiency, and security strength against known attacks. Data extracted from the selected studies were compiled using a standardized template to ensure uniformity in comparison. Due to variations in experimental environments and measurement methods across studies, the analysis was synthesized in a qualitative manner rather than strictly quantitative. The results were further validated

by cross-referencing authoritative cryptographic standards and foundational references to ensure accuracy and reliability, thereby providing a solid basis for evaluating and comparing the three cryptographic algorithms (Nurrohmah & Saputra, 2026).

4. Results and Discussion

4.1 Results

The systematic review identified 35 primary studies and reference documents that met the inclusion criteria, covering algorithm specifications, performance benchmarking, security evaluations, and comparative analyses. The findings are organized based on the three main algorithm families: AES (symmetric block cipher), RC4 (symmetric stream cipher), and RSA (asymmetric public-key cryptosystem). AES consistently demonstrates superior performance in high-throughput encryption across both hardware and software implementations. Hardware-accelerated AES using instruction set extensions achieves multi-gigabit encryption speeds, while FPGA-based implementations further enhance throughput through parallel processing and pipelining techniques. In software environments without hardware acceleration, AES still delivers strong performance with moderate computational overhead. The algorithm's structure, consisting of multiple transformation rounds that vary depending on key size, ensures both efficiency and strong security, making it one of the most widely adopted encryption standards.

RC4 shows the highest raw encryption speed among the three algorithms due to its simple stream-based design and minimal computational requirements. It performs efficiently on low-resource and embedded systems, making it suitable for legacy applications and real-time processing environments. However, despite its speed advantage, RC4 suffers from significant security weaknesses, particularly when improper key usage or insufficient keystream initialization occurs. These vulnerabilities have led to its deprecation in many modern security protocols, as attackers can exploit statistical biases in its output to recover plaintext information. Although modified versions of RC4 have been proposed to mitigate these issues, they have not achieved widespread adoption due to persistent concerns regarding cryptographic robustness.

In contrast, RSA exhibits the highest computational complexity among the three algorithms, primarily due to the use of large integer modular exponentiation. Its encryption and decryption processes are significantly slower than symmetric algorithms, with decryption being particularly resource-intensive. Key generation also requires substantial computational effort, involving large prime number selection and complex mathematical operations. Despite these limitations, RSA remains essential in modern cryptographic systems because of its unique ability to support secure key exchange and digital signatures without requiring pre-shared secrets. As a result, RSA continues to play a critical role in public key infrastructure, secure communication protocols, and authentication systems, despite its lower efficiency compared to symmetric encryption methods.

4.2 Disscusion

The comparative analysis shows that no single cryptographic algorithm is universally optimal across all dimensions, so the selection of an appropriate method must depend on specific application needs, balancing security strength, computational performance, and key management complexity. AES emerges as the most suitable option for high-performance and general-purpose encryption, particularly in modern secure communication systems and data storage environments. Its efficiency, standardized design, and strong resistance to known attacks make it the preferred symmetric encryption method across enterprise, governmental, and consumer applications.

RC4, although historically important and still present in some legacy systems, is no longer considered secure for modern cryptographic use due to well-documented vulnerabilities in its keystream generation process. These weaknesses have led to its removal from widely used security protocols and its replacement with more secure alternatives in most applications. However, in extremely resource-constrained environments where computational limitations are critical, modified versions of RC4 may still be used as a trade-off between performance and security, although with caution due to residual risks.

RSA, on the other hand, remains essential for secure key exchange and digital authentication but is significantly more computationally intensive than symmetric encryption methods. Because of this limitation, it is typically not used for direct data encryption but instead operates within hybrid cryptographic systems where it securely exchanges symmetric session keys, while algorithms like AES handle the actual data encryption. Looking forward, the emergence of quantum computing introduces new security challenges, particularly for RSA and other asymmetric schemes, making the transition toward quantum-resistant cryptographic approaches an important consideration for future secure system design.

5. Conclusions

5.1 Conclusion

This paper presents a comprehensive comparative analysis of three fundamental cryptographic algorithms, namely AES, RC4, and RSA, by examining their operational mechanisms, performance efficiency, security strengths, and applicability in modern information security systems. The findings indicate that symmetric encryption, particularly AES, provides significantly higher computational efficiency compared to asymmetric encryption methods such as RSA, making it more suitable for securing large volumes of data in high-performance environments. AES demonstrates strong resistance against known cryptographic attacks and benefits from widespread standardization and hardware-level acceleration, which further enhances its reliability and adoption in various security applications. In contrast, RC4, despite its historical importance and simplicity in implementation, has become largely obsolete in modern security systems due to well-documented vulnerabilities in its keystream generation process, which compromise its ability to ensure data confidentiality in sensitive environments. On the other hand, RSA continues to play a crucial role in public key infrastructure, particularly in secure key exchange, authentication, and digital signature systems, although its relatively high computational overhead limits its use for large-scale data encryption. The comparative evaluation of these algorithms highlights the importance of leveraging their complementary strengths, leading to the widespread adoption of hybrid cryptographic architectures that combine the efficiency of symmetric encryption with the security advantages of asymmetric encryption. Overall, the study underscores that no single cryptographic algorithm is universally optimal, and the selection of appropriate encryption techniques must be guided by specific security requirements, performance constraints, and application contexts in order to achieve a balanced and robust information security framework.

5.2 Research Limitations

This study is subject to several limitations that should be considered when interpreting the findings. First, the comparative analysis is restricted to three algorithms (AES, RC4, and RSA) and does not encompass the full landscape of contemporary cryptographic standards, including ECC, ChaCha20, or post-quantum algorithms. Second, the performance comparisons are based on synthesized findings from heterogeneous experimental environments, making direct quantitative comparison difficult. Third, as a literature review, this study is inherently constrained by the scope and quality of the primary studies reviewed, and may not capture the most recent implementation advances or vulnerability disclosures. Fourth, the analysis does not address side-channel attacks including timing, power, and cache-based attacks in depth, which represent an increasingly important attack surface in practical implementations. Future research should address these limitations through empirical benchmarking studies across standardized environments and expanded algorithm coverage including lightweight and post-quantum cryptographic candidates.

5.3 Suggestions and Directions for Future Research

Several productive directions for future research can be identified from this review. First, empirical benchmarking of AES, RC4 variants, and RSA on resource-constrained IoT environments, such as ARM Cortex-M microcontrollers and RISC-V embedded systems, is needed to provide more accurate performance insights for real-world deployment in smart devices and industrial applications. Second, future studies should incorporate post-quantum cryptographic algorithms such as CRYSTALS-Kyber and CRYSTALS-Dilithium alongside classical encryption methods to evaluate their comparative efficiency and security readiness for the post-quantum era. Third, the development and evaluation of hybrid cryptographic systems that combine classical and post-quantum approaches should be further

explored, particularly in the context of secure communication protocols and key encapsulation mechanisms suitable for large-scale deployment. Fourth, energy efficiency and computational cost analysis of cryptographic algorithms in battery-powered and edge computing devices represent an increasingly important research direction, especially with the rapid expansion of mobile and IoT ecosystems. Finally, there is a growing need for intelligent and automated cryptographic selection frameworks capable of recommending optimal encryption schemes based on application requirements, security threat levels, and hardware limitations, thereby supporting more adaptive and efficient cybersecurity system design.

Acknowledgement

The author wishes to thank the Computer Science Department, College of Computer Sciences and Mathematics, University of Thi-Qar, Iraq, for providing the institutional support and academic resources necessary for the completion of this research. The author also acknowledges the contributions of the researchers whose foundational and contemporary work in cryptography forms the basis of this review.

Author Contributions

MAS contributed to conceptualization, methodology, writing original draft, writing review and editing, validation. All authors have read and agreed to the published version of the manuscript.

References

Agarwal, S., & Mannil, N. (2023). Household financial decision making *Handbook of Financial Decision Making* (pp. 375-410): Edward Elgar Publishing.

Alagic, G., Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., . . . Miller, C. (2022). Status report on the third round of the NIST post-quantum cryptography standardization process. doi:https://doi.org/10.6028/NIST.IR.8413

Barker, E., & Roginsky, A. (2011). Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. *NIST Special Publication, 800*(1), 131-145.

Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J. M., . . . Stehlé, D. (2018). *CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM.* Paper presented at the 2018 IEEE European symposium on security and privacy (EuroS&P).

Broadbent, J., & De Barba, P. (2023). Self-regulated learning and the student experience in online higher education *Research handbook on the student experience in higher education* (pp. 179-190): Edward Elgar Publishing.

Budler, M., Župič, I., & Trkman, P. (2021). The development of business model research: A bibliometric review. *Journal of Business Research, 135*, 480-495. doi:https://doi.org/10.1016/j.jbusres.2021.06.045

Bui, T. T. M., Nguyen, N. T., Le, D. A., Nguyen, T. N. B., & Nguyen, N. N. K. (2026). Consumers' ethical beliefs, ethical sensitivity and multi-dimensional ethical consumption behavior: a multi-method investigation using PLS-SEM, NCA, and fsQCA. *International Journal of Ethics and Systems*, 1-41. doi:https://doi.org/10.1108/IJOES-11-2025-0693

Ghosh, S., Biradar, D. T., Shinde, G., Bhojaned, S. D., & Shirapure, M. R. (2015). Performance analysis of AES, DES, RSA and AES-DES-RSA hybrid algorithm for data security. *International Journal of Innovative and Emerging Research in Engineering, 2*(5), 83-88.

Hammood, M. M., Yoshigoe, K., & Sagheer, A. M. (2015). Enhancing security and speed of RC4. *International Journal of Computing and Network Technology, 3*(02). doi:https://doi.org/10.12785/ijcnt/030101

Hasan, R., & Talukder, K. A. (2023). A Systematic Review of Predictive Analytics In Marketing Decision-Making Exploring AI-Driven Consumer Segmentation And AB Testing. *International Journal of Business and Economics Insights, 3*(1), 68-96. doi:https://doi.org/10.63125/2hvfh110

Hodges, C. B., & Moore, S. L. (2025). Building a policy research agenda in educational technology: current disconnects and future recommendations. *Journal of Computing in Higher Education, 37*(2), 679-694. doi:https://doi.org/10.1007/s12528-025-09441-9

Huang, L. (2021). *Consumer Financial Decision Making.* The University of Arizona.
Jindal, P., & Singh, B. (2017). Optimization of the security-performance tradeoff in RC4 encryption algorithm. *Wireless Personal Communications, 92*(3), 1221-1250. doi:https://doi.org/10.1007/s11277-016-3603-3
Kamel Saadeldin, M. M., Abogabal, M., & Shahin, M. M. (2026). Beyond the green marketing mix: exploring Gen Z motives towards sustainable food products in an emerging market. *Management & Sustainability: An Arab Review*, 1-30. doi:https://doi.org/10.1108/MSAR-06-2025-0201
Kapoor, J., & Thakur, D. (2022). Analysis of symmetric and asymmetric key algorithms *ICT analysis and applications* (pp. 133-143): Springer.
Li, Z., Sha, Y., Song, X., Yang, K., ZHao, K., Jiang, Z., & Zhang, Q. (2020). Impact of risk perception on customer purchase behavior: a meta-analysis. *Journal of Business & Industrial Marketing, 35*(1), 76-96. doi:https://doi.org/10.1108/JBIM-12-2018-0381
Luo, X., Chen, X., Chen, X., & Cheng, Q. (2024). A survey on the application of blockchain in cryptographic protocols: X. Luo et al. *Cybersecurity, 7*(1), 79. doi:https://doi.org/10.1186/s42400-024-00324-7
Maity, S., Sinha, K., & Sinha, B. P. (2017). *An efficient lightweight stream cipher algorithm for wireless networks.* Paper presented at the 2017 IEEE Wireless Communications and Networking Conference (WCNC).
Montazeri, M., Kebriaei, H., & Araabi, B. N. (2020). Learning pareto optimal solution of a multi-attribute bilateral negotiation using deep reinforcement. *Electronic Commerce Research and Applications, 43*, 100987. doi:https://doi.org/10.1016/j.elerap.2020.100987
Nurrohmah, D., & Saputra, H. A. (2026). Factor analysis of green marketing mix in purchasing decisions of subang district le minerale customer. *Proceeding of SINERGY, 1*(1), 1013-1025. doi:https://doi.org/10.XXXX/UNITA.SINERGY.735
Roggeveen, A. L., & Sethuraman, R. (2021). The Year That Was! *Journal of Retailing, 97*(1), 1. doi:https://doi.org/10.1016/j.jretai.2021.02.003
Sharma, A., Koohang, A., Rana, N. P., Abed, S. S., & Dwivedi, Y. K. (2023). Journal of computer information systems: intellectual and conceptual structure. *Journal of Computer Information Systems, 63*(1), 37-67. doi:https://doi.org/10.1080/08874417.2021.2021114
Stamp, M., & Low, R. M. (2007). *Applied cryptanalysis: breaking ciphers in the real world*: John Wiley & Sons.
Syam, H., & Niswaty, R. (2025). Marketing Mix Increases Customer Interest in Taking Credit at BNI Bank Senayan Branch. *Jurnal Office: Jurnal Pemikiran Ilmiah dan Pendidikan Administrasi Perkantoran*, 83-89. doi:https://orcid.org/0000-0003-1180-420X
Verma, T., & Arora, D. S. (2025). Influence Without Intrusion: Decoding the Role of Digital Nudging in Shaping Online Decisions. *Electronic Commerce Research*, 1-40. doi:https://doi.org/10.1007/s10660-025-10059-3
Wangu, J., Mangnus, E., & van Westen, A. (2020). Limitations of inclusive agribusiness in contributing to food and nutrition security in a smallholder community. A case of mango initiative in Makueni county, Kenya. *Sustainability, 12*(14), 5521. doi:https://doi.org/10.3390/su12145521
Wijaya, N. Q., & Ainun, M. B. (2025). The influence of marketing mix strategy on sales increase with purchasing decisions as a mediating variable at the keris center, aeng tong-tong village, Sumenep. *EKOMBIS REVIEW: Jurnal Ilmiah Ekonomi dan Bisnis, 13*(4), 3269–3284-3269–3284. doi:https://doi.org/10.37676/ekombis.v13i4
Zacharias, Y., & Lao, H. (2024). The effects of discounts and bonus packs on impulse buying for consumers of Ramayana Flobamora Mall Kupang. *Journal of Practical Management Studies, 2*(1), 15-25. doi:https://doi.org/10.61106/jpms.v2i1.23
Zhang, S. (2021). *Characterizing and Optimizing Asynchronous Event-Driven Architecture for Modern Cloud Systems*: Louisiana State University and Agricultural & Mechanical College.