SEUSL

Sri Lankan Journal of Technology
Research

SLJoT
2025 Issue 01 34-52
ISSN-2773-6970

Foundations of Cryptographic Defense: Navigating Algorithmic Strengths, Key Dynamics, and Network Security Challenges

T.N.Ahamed[1], M.J.Sabani[2*], and M.S.Shafana[3]

[1,2,3]Department of Information and Communication Technology, South Eastern University of Sri Lanka

*Corresponding Author: mjasabani@seu.ac.lk || ORCID: 0000-0002-9583-9057

Received: 21-04-2024. * Accepted: 29-04-2025 * Published Online: 30-06-2025

Abstract - Although cryptography underpins secure communication, prevailing studies tend to emphasize performance metrics such as speed and throughput while overlooking broader evaluative dimensions.. In contrast, this study provides a comprehensive and methodologically rigorous review of cryptographic techniques, integrating considerations of security depth, usability, and key management. Using a validated systematic approach, we surveyed high-impact publications from the Scopus database and top-tier venues to ensure coverage of the most influential research. Our comparative evaluation of symmetric encryption algorithms highlights the superiority of AES across key sizes and platforms. However, unlike prior work, we also emphasize the often-overlooked practical dimensions of key handling. We analyze both symmetric and asymmetric schemes, exploring their trade-offs in security, key distribution, and operational complexity. The integration of digital signatures and public-private key pairs is shown to enhance confidentiality and non-repudiation. By focusing not just on algorithmic performance but on the real-world application, adaptability, and maintainability of cryptographic systems, this study offers novel insights for the future design of secure communication protocols. Our holistic approach fills critical gaps in existing reviews and serves as a practical guide for researchers and practitioners in cryptographic system design.

Keywords- Cryptography, encryption, decryption, Secret Key Cryptography, DES, AES, Blowfish, Public Key Cryptography

Recommended APA Citation
Ahamed, T. N., Sabani, M. J.A, & Shafana, M. S. (2025). Foundations of cryptographic defense: Navigating algorithmic strengths, key dynamics, and network security challenges. *Sri Lankan Journal of Technology, 6*(1), 34–52.

1. Introduction

The volume and complexity of data are rapidly increasing, with the digital universe projected to grow by 50 times from 2010 to 2020 [1]. In this era where the Internet serves as a vital tool for communication among millions of users, ensuring data security has become paramount. From safeguarding sensitive information in e-commerce transactions to protecting personal conversations and employing robust password mechanisms, security encompasses various elements and applications. Cryptography plays a pivotal role in ensuring secure communication channels. This paper aims to address two main objectives. Firstly, it seeks to identify prevalent security concerns within network infrastructure. Secondly, it delves into cryptography, including cryptographic algorithms, their implementations, and a comparative analysis of their efficacy.

Cryptography, with its origins tracing back to the Roman era, remains a fundamental technique for concealing information and securing messages. In contemporary settings, particularly in the realm of digital communication, cryptography assumes a pivotal role in protecting data integrity and confidentiality. Its application spans across various domains, including safeguarding sensitive information during storage and transmission over communication channels like the Internet. Encryption and decryption, the cornerstone operations of cryptography, entail the transformation of plaintext into ciphertext and its subsequent reversal, respectively. This cryptographic process ensures that information remains unintelligible to unauthorized entities, thereby upholding the confidentiality and integrity of data in digital communications.

The paper is structured as follows: Section II presents the methodology employed to fulfill the objectives of this study. Section III delves into the Discussion on security issues inherent in networks, which necessitate the use of cryptographic algorithms, various types of cryptographic algorithms currently in use are detailed, a literature review analysis focusing on symmetric and asymmetric algorithms, a comparison between early and modern cryptography approaches, and a comparative analysis of various encryption algorithms, followed by the conclusion of the research in the final section.

2. Methodology

A systematic approach was employed to achieve the objectives of this study, involving an in-depth exploration of various learning strategies within the field of cryptography. The methodology adopted is depicted in Fig. 1, illustrating the process of conducting a literature review to enhance understanding of cryptographic algorithms. This approach delineates fundamental procedures for locating, comprehending, and analyzing relevant research publications, facilitating the identification of supporting evidence.

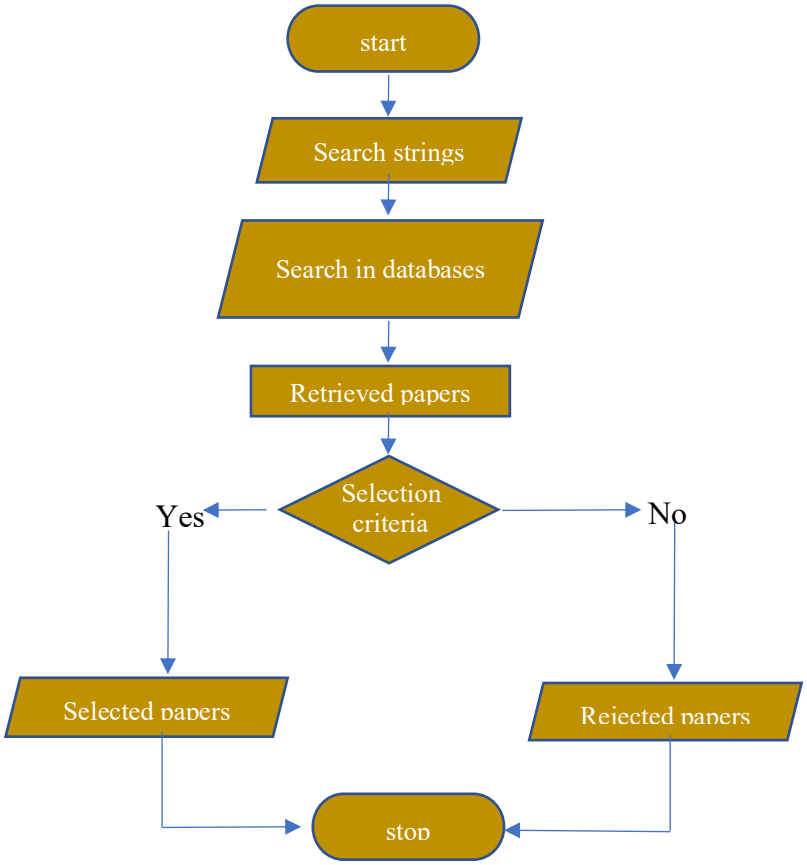

Figure 1. Methodology.

A. Literature Retrieval Process

With its vast collection of scholarly publications and important conference proceedings, Scopus was the principal database used. We used additional resources including IEEE Xplore, SpringerLink, and MDPI (such as Sensors) to improve the paper compilation and decrease selection bias. In addition, we ran extra searches on Google Scholar to find important publications that were mentioned in technical conversations but were missed by the more conventional databases.

Example search terms used included:
"Cryptography", "Encryption", "Decryption", "Symmetric Key Cryptography", "Asymmetric Key Cryptography", "DES", "AES", "RSA", "Blowfish", "Key Management", and "Digital Signatures". Search strings were iteratively refined to optimize recall and relevance across multiple databases.

B. Inclusion and Exclusion Criteria

To ensure quality and consistency, the following criteria were applied during paper selection:

- ***Inclusion Criteria:***
  - Peer-reviewed articles or conference papers indexed in Scopus, IEEE Xplore, or SpringerLink.
  - Publications from 2015 onwards, unless foundational (e.g., DES, ElGamal).
  - Clear technical discussion of cryptographic algorithm performance or security.
  - Papers that benchmark algorithms using measurable criteria (e.g., execution time, key length, cryptanalytic resistance).

- **Exclusion Criteria**:
  - Non-peer-reviewed or editorial/opinion-based publications.
  - Papers lacking technical depth or focused solely on hardware without cryptographic analysis.
  - Journals delisted from Scopus (e.g., *World Applied Sciences Journal* after 2016) or those not indexed but included only when contextually significant.

While most references were drawn from indexed sources, a few exceptions were retained for their conceptual relevance or technical illustrations of older cryptographic methods.

C. Paper Screening and Classification

A total of 68 papers were initially identified. After applying the inclusion/exclusion criteria, 18 key publications were selected for detailed review. These were classified as follows:

- Symmetric Cryptography: AES, DES, Blowfish, etc.
- Asymmetric Cryptography: RSA, ElGamal, ECC, etc.
- Hybrid and Signature-based Approaches: Digital signatures, key exchange protocols.

Each paper was analyzed for its contribution to:

- Efficiency benchmarks (e.g., time complexity, throughput).
- Security metrics (e.g., resistance to attacks, key strength).
- Implementation environments (hardware/software adaptability).
- Key management practices and usability trade-offs.

This methodical approach ensures a balanced evaluation of cryptographic systems in terms of performance, applicability, and resilience in securing communication infrastructures.

3. Analysis and Discussion

A. Security issues in network

Security in network operations is paramount for ensuring the integrity, confidentiality, and overall protection of data during transmission [1]. Several security issues can arise within networks, posing threats to the integrity and confidentiality of transmitted data.

1) Authenticity issue in WSN

In Wireless Sensor Networks (WSNs), ensuring authenticity is paramount due to the potential risk of message injection by adversaries. Receiver nodes play a crucial role in verifying that data used for decision-making originates from a trusted and reliable source. The primary objective of maintaining data authenticity is to safeguard the identities of communication nodes, which is essential for fulfilling various administrative responsibilities [2]

2) Eavesdropping Attack in WSN

Eavesdropping, also referred to as confidentiality breach, involves the interception of communication content as it traverses the Wireless Sensor Network (WSN) channel. This attack poses significant risks such as network wormhole or blackhole attacks, leading to the collection of sensitive WSN information and compromising node privacy and confidentiality. Eavesdropping typically operates within external and passive threat models, falling within the intersection model of network security threats. Protecting confidentiality is paramount in mitigating the impact of this attack on the network [2].

3) DOS (Denial of Services) Attack

Denial of Service (DOS) attacks can inflict damage on various layers of the Wireless Sensor Network (WSN), including the data link layer (Layer 2), network layer (Layer 3), and transport layer (Layer 4). By flooding the network with a barrage of broadcast packets, attackers can compel sensor nodes to engage in resource-intensive signature verification processes, thereby disrupting normal network operations. The pervasive nature of DOS attacks affects the functionality of multiple network layers, undermining the WSN's overall performance and reliability [2].

4) Modification

Modification attacks represent an attempt to compromise the integrity of the system by unauthorized entities. In this type of assault, not only does the attacker gain access to a resource, but they also possess the capability to alter it. This poses a significant threat to the integrity and reliability of the system, as unauthorized modifications can lead to data corruption or unauthorized access to sensitive information [1].

5) Secrecy

Confidentiality, also referred to as secrecy, is a fundamental aspect of network security aimed at preventing unauthorized access to information. It is often the primary concern when considering network security measures. The goal of confidentiality is to restrict access to sensitive information to only authorized individuals or entities. Before divulging sensitive information or engaging in transactions, it is essential to first authenticate the identity of the party with whom you are communicating. This ensures that sensitive data remains protected and secure from unauthorized access or interception [3].

6) Message Integrity

Message integrity is a crucial aspect of network security that ensures the recipient of a message that it has not been altered in any way during transmission. While the sender and receiver may verify each other's identities, it is equally important to ensure that the content of their communication remains unchanged throughout its journey across the network. This assurance is essential to prevent any unauthorized modifications to the message, whether intentional or accidental. By maintaining message integrity, the integrity and trustworthiness of the communication are preserved, enhancing overall security [3].

7) Non repudiation

Non-repudiation ensures the verifiability of a transmitted message's source, inhibiting the sender from rejecting its delivery. This concept is essential for maintaining accountability and validating authenticity in digital communication inside network security. The core principle of non-repudiation is on the establishment of irrefutable evidence linking a communication to its originator. It is often executed using digital signatures, which provide cryptographic verification of both the message's origin and its integrity. This method not only enhances confidence between communication parties but also precludes the sender from contesting the transfer, hence fostering safe and dependable digital interactions [3].

Given the diversity and severity of these network threats—including eavesdropping, integrity violations, and denial-of-service attacks robust cryptographic mechanisms are indispensable for securing digital communications. In response to these challenges, this study explores the landscape of cryptographic algorithms, classified based on their key structures and operational principles. The subsequent section elaborates on symmetric and asymmetric cryptographic approaches and analyzes their efficacy against the discussed threats.

B. Types of cryptographic algorithms

Cryptographic algorithms can be classified in a variety of ways. The three primary classification methods to be described are as follows in Fig 2. Here it has been classified for the purposes of this research depending on the number of keys used for encrypting and decrypting, as well as their application and usage.

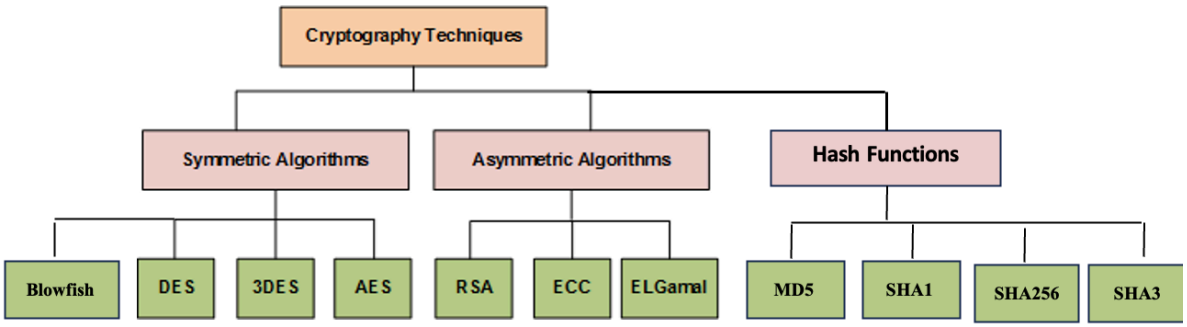

Figure. 2. describes the taxonomy of cryptography techniques [6]

1. *Symmetric Algorithm / Secret Key Cryptography (SKC)*

Symmetric key cryptography, also referred to as synchronized key cryptography, is applied when two parties need to exchange information securely. In this approach, a single secret key is shared between them and used for both encrypting and decrypting the message. The fact that the same key is used on both ends of the communication process is what gives this method its symmetric nature. Especially in trusted environments where speed and simplicity are prioritized, this approach proves highly effective [13]. It is through this shared key mechanism that the confidentiality of exchanged data is preserved.

This technique is known as self-certification, where both parties certify each other's identity through the shared secret key. It is essential that the key be transmitted through a secure channel or medium to prevent unauthorized access by attackers, who could otherwise intercept the key and decrypt the message. Despite this requirement for secure key exchange, symmetric key cryptography is preferred f or its efficiency and resource-saving capabilities. Various encryption algorithms have been developed to implement symmetric key cryptography, including AES, DES, 3DES, and Blowfish, each offering different levels of security and performance [4].

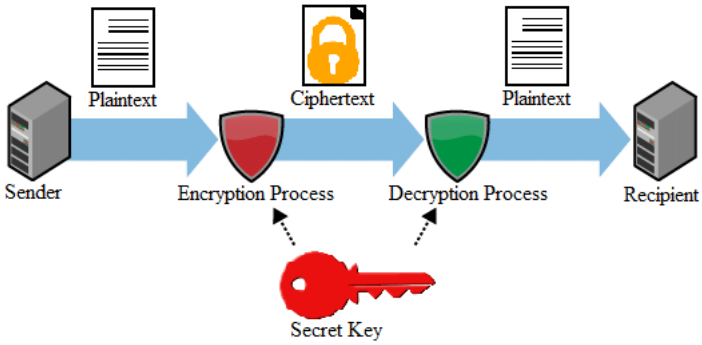

Figure. 3. Components of symmetric block cipher. [11]

1.1 Data Encryption Standard (DES)

The Data Encryption Standard (DES) was initially developed by IBM but later adopted as a National Standard by the US Government [5]. DES operates as a Feistel Cipher, employing the Feistel structure with 16 rounds and a block size of 64 bits. However, the key length for DES is 64 bits, of which only 56 bits are utilized for encryption and decryption. The remaining 8 bits serve as check bits and are not involved in the cryptographic processes [8, 9].

Despite its widespread application in commercial, military, and other sectors, the Data Encryption Standard (DES) has notable vulnerabilities, including susceptibility to attacks such as brute force and Man-in-the-Middle attacks, leading to low overall efficiency [4]. In 1998, DES was successfully cracked within 22 hours using a device constructed by the Electronic Frontier Foundation, primarily due to its short key length and vulnerability to linear cryptanalysis attacks [11].

1.2 Triple Data Encryption Standard (3DES)

The Triple Data Encryption Standard (3DES) is an enhanced version of the original Data Encryption Standard (DES), which was developed by IBM and later adopted as a National Standard by the US Government. Like DES, 3DES operates as a Feistel Cipher, utilizing the Feistel structure with 16 rounds and a block size of 64 bits. However, 3DES significantly improves upon DES's security by employing three rounds of encryption, hence the "triple" designation. This results in a much longer effective key length, enhancing resistance to brute force and other attacks. 3DES uses a key length of 168 bits, derived from three 56-bit keys used sequentially in three encryption stages. While the use of three keys increases security, it also introduces computational overhead compared to DES. Despite this, 3DES remains widely utilized in various sectors due to its compatibility with existing DES implementations and its enhanced security features. However, it's important to note that while 3DES addresses many of the vulnerabilities of DES, it is not immune to all attacks. Its susceptibility to certain cryptographic attacks, combined with its computational complexity, has led to the exploration of more advanced encryption standards in recent years. Nonetheless, 3DES continues to be a reliable encryption option for applications requiring robust security and backward compatibility with legacy systems [18].

1.3 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a symmetric key encryption algorithm designed to replace DES in commercial applications. Developed by Belgian cryptographers Joan Daemen and Vincent Rijmen, AES operates on 128-bit blocks and follows a Substitution-Permutation Network structure [5]. The algorithm is referred to as AES-128, AES-192, or AES-256 depending on the key size, with block size fixed at 128 bits. AES performs encryption by executing 9 rounds of calculations to produce the encrypted message.

Despite its strengths, AES is vulnerable to certain attacks such as side-channel attacks. However, its efficiency in both software and hardware is noteworthy. Due to its longer keys, AES is difficult to break, but the trade-off is that it requires more time for calculations [4].

1.4 Blowfish

Blowfish, a symmetric block cipher, was created by Bruce Schneier in 1993 as a faster alternative to the DES encryption algorithm. Unlike some other encryption methods, Blowfish is available without a license and is free for all usage [15]. The algorithm employs a Feistel Network structure with two main components: a data encryption component and a key expansion component [5]. It utilizes a variable key length, with a maximum key length of 448 bits and a block size of 64 bits. Blowfish performs 16 rounds of calculations to encrypt a message. Known for its efficiency and resilience against attacks, Blowfish remains a popular choice for encryption in various applications [4].

Table 1 provides a comparative analysis of symmetric encryption algorithms, including DES, AES, and Blowfish, highlighting their key parameters and characteristics.

Table 1

Comparative Analysis of Symmetric Encryption Algorithms

Attribute	**DES**	**3DES**	**AES**	**Blowfish**
Published	1977	1998 (based on DES)	2001	1993
Algorithm Structure	Feistel Network	Triple DES (3x DES)	Substitution-Permutation Network	Feistel Network
Block Cipher	Yes	Yes	Yes	Yes
Key Length	56 bits	168 bits (3x 56 bits)	128, 192, 256 bits	32 - 448 bits (variable)
Flexibility/Modification	No	Limited (based on DES)	High (supports different key sizes)	High (variable key size)
Number of Rounds	16	48 (3x 16)	10, 12, or 14	16
Block Size	64 bits	64 bits	128 bits	64 bits
Throughput	Low	Lower than AES	High	High
Level of Security	Adequate (once considered secure)	Improved over DES, but not as secure as modern algorithms	Excellent	Excellent
Encryption Speed	Slow	Slower than most modern algorithms	Fast	Fast

Effectiveness	Primarily for legacy systems	Primarily for legacy systems	Widely used and efficient	Efficient in both software and hardware
Analysis	Vulnerable to brute-force attacks due to short key size	More secure than DES, but still vulnerable to certain attacks	Highly secure and resistant to known attacks	Potential for key-related attacks
Attacks	Brute-force, differential cryptanalysis	Similar to DES, plus meet-in-the-middle attacks	None known (highly secure)	Dictionary attacks, potential for related-key attacks
Real World Applications	Legacy banking systems, smart card systems	Financial services (SWIFT), ATMs, EMV chip-based cards (being phased out)	TLS/SSL, VPNs, disk encryption (e.g., BitLocker, VeraCrypt), Wi-Fi security (WPA2/WPA3)	Password hashing (e.g., bcrypt), file encryption tools (e.g., CryptoForge)

Blowfish and AES emerge as prominent symmetric encryption algorithms, as illustrated in Table 1. Blowfish delivers impressive throughput, benefiting from a lightweight key schedule and variable-length key design. This performance benefit, however, is offset by its known vulnerability to key-related attacks. AES, by comparison, offers greater resistance to cryptographic threats through its standardized key lengths, which has led to its adoption in both governmental and commercial systems. Despite the advantages of symmetric cryptography, its dependence on shared secret keys limits its effectiveness in distributed network environments. In response, asymmetric cryptography provides a dual-key model that enables secure exchanges without the prior distribution of secrets. The next section examines core asymmetric encryption techniques and evaluates their roles within modern cryptographic frameworks.

2. Public Key Cryptography (PKC):

Public key techniques enable secure communication without the need for pre-shared secret keys. In this method, the recipient generates a key pair (pk, sk), where pk is the public key and sk is the private key. The sender encrypts the message using the recipient's public key (pk), and the recipient decrypts it using their private key (sk). This process ensures confidentiality and integrity of the communication. As depicted in Fig 4, public key encryption facilitates secure interactions without prior key agreement.

Public key cryptography involves the use of two distinct keys: a public key and a private key. The public key is accessible to everyone, while the private key is kept confidential. This asymmetric approach ensures secure communication. Fig 4 illustrates the operation of public key encryption. Notable asymmetric algorithms include Rivest-Shamir-Adleman (RSA), Diffie-Hellman, and the Digital Signature Algorithm (DSA) [1].

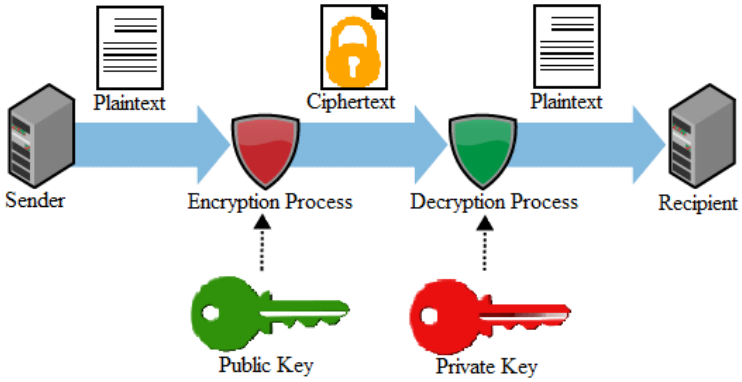

Figure. 4. Components of asymmetric block cipher [11]

2.1 ElGamal:

ElGamal, introduced by Taher Elgamal in 1984, presents a novel approach to public key encryption, leveraging discrete logarithms [12]. Serving as an alternative to RSA, this asymmetric encryption method draws inspiration from the Diffie-Hellman key exchange. Additionally, ElGamal finds application in the ElGamal signature technique, a process for generating digital signatures. To bolster its security, the paillier homomorphic method is employed [6], ensuring semantic security whereby the encryption of the same plaintext consistently yields distinct ciphertexts. However, a notable drawback of ElGamal lies in its ciphertext being twice as long as the plaintext. Despite this limitation, the ElGamal cryptosystem finds utility in key exchange, enabling two parties to exchange keys securely, as well as in authentication and the encryption and decryption of short messages [16]. One distinctive characteristic of the ElGamal method is the ciphertext being twice the length of the plaintext during the encryption process [17].

2.2 Rivest-Shamir-Adleman (RSA)

RSA encryption operates with a key pair (d, e), where d is the private key and e is the public key. It employs the same function for both encryption and decryption. The security of RSA lies in the difficulty of deriving the private key from the public key, making it highly secure. Computing the inverse of e, especially for attackers, is extremely challenging. Key generation in RSA is complex and time-consuming, which can be a drawback. Despite its complexity, it is not necessarily similar to factorization, as factoring large numbers remains difficult. To enhance security, RSA keys should be greater than 1024 bits in length [4].

Table 2 provides a comparative analysis of Asymmetric encryption algorithms, including RSA, ECC, ElGamal highlighting their key parameters and characteristics.

2.3 Elliptic Curve Cryptography (ECC)

Public key cryptography has advanced to include Elliptic Curve Cryptography (ECC), which utilises the intricate mathematics of elliptic curves over finite fields. ECC distinguishes itself by sustaining strong security with much reduced key sizes compared to conventional techniques such as RSA. This efficiency renders ECC especially appealing for systems with constrained processing capabilities, and its implementation has become common in apps like WhatsApp, iOS Secure Enclave, and HTTPS encryption using TLS.

Table 2
Comparative Analysis of Asymmetric Encryption Algorithms

Attribute	**RSA**	**ECC**	**ElGamal**
Published	1978	1985	1984
Algorithm Structure	Public-key (asymmetric)	Elliptic Curve Discrete Logarithm Problem	Public-key (asymmetric)
Block Cipher	No	No	No
Key Length	Large (variable, e.g., 2048 bits)	Smaller than RSA (e.g., 160 - 571 bits)	Large (variable, similar to RSA)
Flexibility/Modification	High (variable key size)	High (variable curve parameters)	High (variable key size)
Throughput	N/A	High	N/A
Level of Security	High (for large key sizes)	High (for appropriate curve parameters)	High (for large key sizes)
Encryption Speed	Slower than symmetric algorithms	Faster than RSA	Slower than RSA
Effectiveness	Suitable for digital signatures and key exchange	Efficient for digital signatures and key exchange	Suitable for digital signatures and encryption
Analysis	Vulnerable to chosen-ciphertext attacks for small key sizes	Considered secure for appropriate curve parameters	Vulnerable to chosen-ciphertext attacks for small key sizes
Attacks	Factoring attacks, chosen-ciphertext attacks	Side-channel attacks	Chosen-ciphertext attacks
Real World Applications	Digital certificates (SSL/TLS), email encryption (PGP), blockchain wallets, VPN authentication	Mobile device encryption, WhatsApp, Signal, iOS Secure Enclave, Bitcoin, TLS in HTTPS	Digital signatures, secure key exchange in open networks, privacy-preserving applications

Table 2 highlights the relative strengths and limitations of prominent asymmetric algorithms. RSA, although secure, is computationally expensive and slower in decryption compared to ECC. ECC offers a compelling alternative by providing equivalent security at shorter key lengths, making it suitable for resource-constrained devices. ElGamal, while secure, incurs a higher ciphertext size, limiting its use in bandwidth-sensitive scenarios.

3. Hash Functions

Hash functions are cryptographic algorithms designed to generate a unique fixed-size output, known as a hash value or digest, from input data of arbitrary size. These functions are widely used in various applications, including data integrity verification, password storage, and digital signatures. Similar to the Data Encryption Standard (DES), which was initially developed by IBM and later adopted as a National Standard by the US Government, hash functions operate on principles of cryptographic security. However, while DES operates as a Feistel Cipher with a fixed block size and specific key length, hash functions typically do not require a key and can process data of any size. Despite their utility, hash functions may also exhibit vulnerabilities, such as collision attacks, where different inputs produce the same hash value. Therefore, careful selection

and implementation of hash functions are crucial to ensure the integrity and security of data in cryptographic system [3].

Table 3 provides a comparative analysis of Asymmetric encryption algorithms, including MD5: Message-Digest Algorithm 5 (MD5), Secure Hash Algorithm 1(SHA-1), Secure Hash Algorithm 256 (SHA-256), Secure Hash Algorithm 3 (SHA-3), highlighting their key parameters and characteristics.

Table 3

Comparative Analysis of Asymmetric Encryption Algorithms

Attribute	**MD5 (1991)**	**SHA-1 (1995)**	**SHA-256 (2002)**	**SHA-3 (2015)**
Published	1991	1995	2002	2015
Algorithm Structure	Merkle-Damgård	Merkle-Damgård	Merkle-Damgård	Sponge construction
Block Cipher	No	No	No	No
Block Size	512 bits	512 bits	512 bits	Variable
Throughput	Fast	Fast	Slower than SHA-1	Slowest
Level of Security	Considered broken, not recommended	Currently considered secure, but potential vulnerabilities	Generally considered secure	Considered secure
Encryption Speed	Fast	Fast	Moderate	Slow
Effectiveness	Legacy systems (not recommended)	Digital signatures, message verification	Digital signatures, message verification, secure hashing	Secure hashing, message authentication codes (MACs)
Analysis	Collisions found, pre-image attacks possible	Collision attacks possible, theoretical weaknesses	No known attacks	No known attacks
Attacks	Collisions, pre-image attacks	Collisions possible	No known attacks	No known attacks
Real World Applications	Legacy checksum validation (no longer secure, only used for integrity in non-critical systems)	Code signing, certificate validation (deprecated due to collision attacks)	Blockchain (Bitcoin, Ethereum), TLS certificates, file integrity verification	Post-quantum cryptography research, secure hash applications in IoT and secure tokens

From Table 3, it is clear that modern hash functions such as SHA-256 and SHA-3 provide stronger resistance to collision and pre-image attacks compared to legacy functions like MD5 and SHA-1. Despite their increased computational cost, their enhanced security makes them essential for digital signature validation, password hashing, and secure message authentication.

C. Analysis of symmetric and asymmetric algorithms

According to the analysis conducted by F. Maqsood et al. [7], the efficiency of symmetric and asymmetric cryptographic methods was examined in terms of key generation and encryption/decryption times.

- In reference to encryption time, depicted in Fig 5, evaluations were made concerning the encryption durations of DES, AES, RSA, and ElGamal across various file sizes. The findings indicate that the DES algorithm exhibits notably longer encryption durations compared to alternative methods, including AES, RSA, and ElGamal. Conversely, all other encryption methods, notably RSA, demonstrate shorter encryption durations in comparison to DES. Thus, it is evident from the analysis that asymmetric algorithms, such as RSA, entail lesser time for data encryption in contrast to symmetric methods.

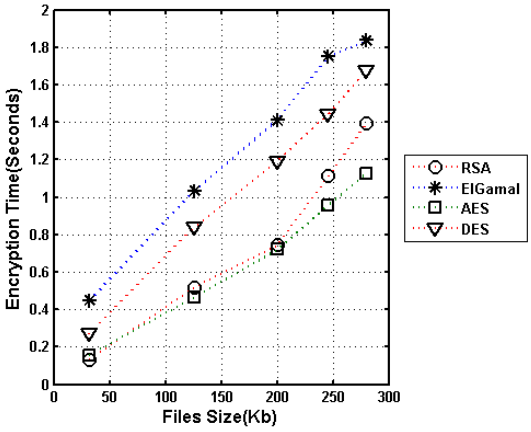

Figure. 5. Encryption Time (DES, AES, ElGmal and RSA). [6]

- In consideration of decryption time, as illustrated in Fig 6, an examination was conducted on the decryption durations of DES, AES, RSA, and ElGamal across diverse file sizes. It is observed that in comparison to alternative encryption algorithms such as DES, AES, and ElGamal, the RSA technique necessitates significantly more time for decoding data.

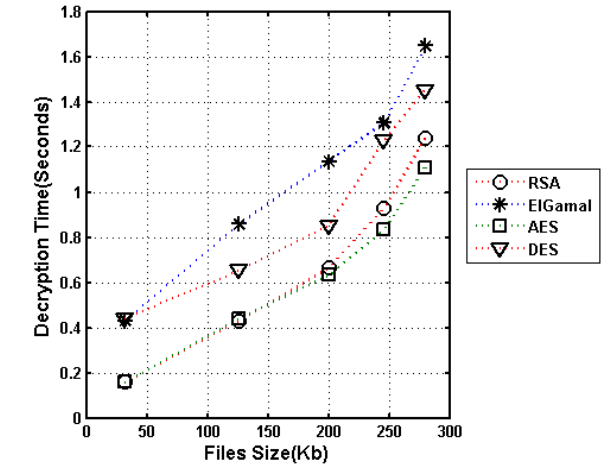

Figure 6. Decryption Time (DES, AES, ElGmal and RSA).

- In examining key generation time, as depicted in Fig 7, the duration of both symmetric and asymmetric algorithms' key generation processes was scrutinized. Notably, the time required for key generation is contingent upon the bit length of the key, whereby an

extension in bit length correlates with a proportional increase in the duration of the process. Specifically, the RSA method, characterized by a 1024-bit key length, incurs a notably lengthier duration for key generation compared to the DES algorithm, which employs a 56-bit key length, thereby expediting the key generation process.

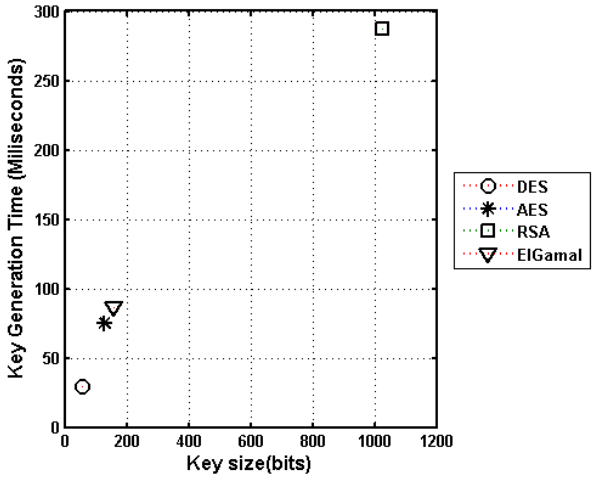

Figure 7. Key Generation Key (DES, AES, ElGmal and RSA).

Throughput analysis, achieved by dividing the total volume of plaintext, measured in megabytes, by the cumulative encryption time, provides a comprehensive measure of the encryption system's efficiency. This metric offers valuable insights into the system's ability to process encryption tasks in relation to the quantity of data being encrypted. Additionally, the relationship between power consumption and CPU processing is of paramount importance. Notably, as throughput increases, there is a discernible reduction in both power utilization and CPU processing load.

Moreover, the metric of CPU Process Time serves as a crucial indicator of the CPU's workload during encryption operations. The duration of CPU time required for executing encryption tasks directly influences the strain imposed on the CPU. Consequently, an elongated encryption process exacerbates the burden on the CPU, potentially leading to diminished performance.

Figure 8 provides compelling evidence that, with the exception of the Blowfish algorithm, all other techniques exhibit markedly low throughput. This finding underscores the unsuitability of these techniques for deployment in low-computing platforms, where efficient utilization of computational resources is imperative [10].

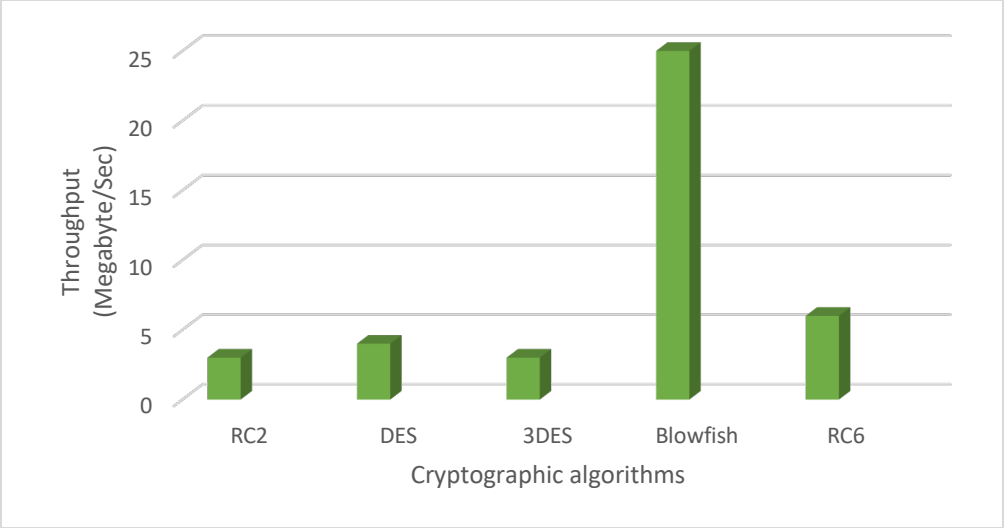

Figure. 8. Throughput of each encryption algorithm (Megabyte/Sec)

In the comparative study conducted[1], the efficacy of several encryption algorithms, namely DES, AES, and Blowfish, was meticulously assessed. Their findings revealed that Blowfish exhibited superior performance across all block cipher modes, boasting the fewest vulnerabilities compared to its counterparts. Conversely, AES demonstrated comparatively inferior performance outcomes.

The research also underscored a fundamental distinction between symmetric and asymmetric encryption methodologies. Symmetric encryption techniques typically demonstrate swifter processing speeds than their asymmetric counterparts. However, a notable drawback of symmetric encryption lies in the shared key mechanism, which entails the dissemination of the encryption key among participating entities. Conversely, asymmetric encryption employs distinct public and private key pairs, thereby enhancing security. Nevertheless, the adoption of asymmetric encryption entails a trade-off, as it necessitates greater processing time compared to symmetric encryption techniques [15].

D. Early Vs modern cryptography

Transitioning to a discussion on the evolution of cryptography, it is imperative to explore its historical origins and contemporary advancements. A comparative analysis between early and current cryptographic methodologies underscores the significant complexity inherent in the latter. In antiquity, cryptography primarily served the purpose of concealing military intelligence from adversaries. However, in the modern era, its applications have diversified, facilitating global communication and safeguarding electronic data. Cryptography now plays a pivotal role in preserving the confidentiality of sensitive personal, financial, E-commerce, and medical information, shielding it from unauthorized access.

In the realm of contemporary cryptography, notable advancements have been achieved. A pivotal moment occurred in January 1977 when the National Bureau of Standards (NBS) sanctioned the Data Encryption Standard (DES), marking a transformative milestone in cryptographic research and development amidst the burgeoning era of computing technology. Subsequently, cryptography made significant inroads into the realm of commerce when the

American National Standards Institute (ANSI) endorsed DES in December 1980. This validation underscored the method's reliability and suitability for commercial applications.

The central focus of this paper is on cryptographic techniques that are widely adopted in modern digital networks. While classical ciphers such as Caesar and Rail Fence hold foundational value, their role today is primarily educational, with limited relevance to current practical applications. These early techniques were favored for their simplicity and ease of implementation, which made them suitable for historical uses involving basic secrecy. However, their security was minimal and easily compromised by contemporary analytical methods. In contrast, modern algorithms like AES, RSA, and ECC offer significantly stronger protection, supporting secure communication, data privacy, and e-commerce activities. Though computationally intensive, these advanced systems demonstrate high resilience against modern cyber threats and are designed to adapt across diverse operational contexts. What follows, having traced the historical development of cryptographic systems, is a critical analysis of their real-world deployment. Not only does this comparison highlight differences in algorithmic strength, but it also reveals how each method balances security, efficiency, and usability in modern applications.

E. Comparison of various encryption algorithms

Table 4 presents a comprehensive comparison of various encryption algorithms, delineating their efficacy in ensuring data security and resilience against potential attacks, alongside an assessment of their encryption and decryption speeds [7].

Table 4

Encryption Algorithm Key Sizes

Symmetric Key Cryptography	**Size of Key**	**In Steps of**	**Advantages and Challenges**
DES	40 - 56 Bits	8 bits	Simple and fast for hardware; Weak security, vulnerable to brute-force attacks
3-DES (Two key)	64 - 112 Bits	8 bits	Better than DES; Still vulnerable, slow and deprecated
3-DES (three key)	120 - 168 Bits	8 bits	Improved security over 2-key; High computational cost
Public key Encryption			
Diffie-Hellman	512 - 2048 Bits	64 bits	Enables secure key exchange; Vulnerable to MITM if not authenticated
RSA	512 - 2048 Bits	64 bits	Widely adopted, good for signatures; Slower, requires large keys
Digital Signatures			
DSA	512 - 2048 Bits	64 bits	Strong for digital signatures; Requires secure randomness, slower verification
RSA	512 - 2048 Bits	64 bits	Standard for digital certificates; Heavy key management burden

Cryptography stands as the quintessential solution for facilitating efficient data transmission, encompassing a plethora of methods rooted in both symmetric and asymmetric key cryptography.

These cryptographic techniques serve as bulwarks, ensuring data privacy, integrity, authenticity, and non-repudiation through meticulously crafted algorithms. However, certain avenues for enhancement remain unexplored. Notably, quantum cryptography emerges as a promising alternative to traditional methodologies like the Diffie-Hellman method, offering unparalleled data security. Nevertheless, it remains susceptible to conventional bucket brigade attacks, prompting the exploration of novel strategies to mitigate such vulnerabilities. Furthermore, the efficacy of cipher algorithms is contingent upon the block cipher modes employed. Various applications necessitate distinct security levels, categorized based on operational requirements:

1. High Security: Applications prioritizing robust security over encryption speed, such as national defense communications networks and healthcare systems.
2. Medium Security: Balancing encryption speed and security, crucial for applications like social media platforms and voice communication.
3. Low Security: In contexts where swift encryption takes precedence over security, such as encrypting data on personal computing devices or intra-enterprise data transmission.

Additionally, the security of encryption is influenced by factors such as block size, key size, and the number of rounds in the encryption process. Larger block sizes bolster security but may impede encryption and decryption speed, whereas larger key sizes enhance security at the expense of processing speed. Moreover, increasing the number of encryptions rounds augments data complexity and fortifies security measures. These nuanced considerations underscore the intricate interplay between security requirements and encryption performance in contemporary cryptographic applications. Homomorphic encryption is not the primary focus of this comparative study; however, it is acknowledged as a rapidly emerging paradigm in modern cryptography. For the sake of scope completeness, a concise overview of this method will be included. Not only is homomorphic encryption conceptually innovative, but it also represents a significant step toward secure computation on encrypted data.

In the realm of algorithmic security assessment, the absence of a definitive tool to ascertain the robustness of an algorithm, regardless of its complexity, is a prevailing challenge. Consequently, the onus falls upon us to construct a bespoke suite of code tailored to our specific requirements, facilitating the generation of pertinent outputs for evaluation. However, the reliability of these outputs may be compromised by the inherent susceptibility to human error present within the coding process. Thus, navigating this terrain poses a formidable challenge, particularly when striving for accuracy and precision in our testing endeavors.

4. Conclusion

Beyond the core discussions, this study also examines the foundational principles of cryptography along with how these principles function in real-world systems. While such systems are built for security, it must be acknowledged that vulnerabilities may still exist—an issue this research addresses directly. Cryptography operates by preserving confidentiality through structured processes that have been refined over time. The most trusted algorithms are those supported by documentation, tested under rigorous conditions, and shown to remain reliable through extended use. When an algorithm consistently resists failure, it earns a reputation for dependability.

Yet the overall security of a cryptographic system is not determined by the algorithm alone; how keys are managed plays an equally vital role. In symmetric encryption, the sharing of secret keys can introduce serious risk. Asymmetric encryption helps resolve this by using a key pair, one for public distribution and another kept private, to secure communications. Releasing only the public key while protecting the private one enables systems to maintain confidentiality and limit

unauthorized access. Digital signatures enhance this approach by supporting message integrity and enabling users to verify authenticity.

Although asymmetric encryption provides strong protection, symmetric methods remain important due to their simplicity, speed, and adaptability across many practical scenarios. This study also explores the various types of attacks that threaten cryptographic systems. What matters most when responding to such threats is not only recognizing their nature but also choosing encryption strategies that match specific operational needs. Ultimately, secure system design depends on selecting methods capable of resisting evolving digital threats and maintaining long-term resilience.

REFERENCES

Kumar, M.G.V. and Ragupathy, U.S., 2016, March. A survey on current key issues and status in cryptography. In 2016 International Confer- ence on Wireless Communications, Signal Processing and Networking (WiSPNET) (pp. 205-210). IEEE.

Anwar, R.W., Bakhtiari, M., Zainal, A., Abdullah, A.H. and Qureshi, K.N., 2014. Security issues and attacks in wireless sensor network. World Applied Sciences Journal, 30(10), pp.1224-1227.

Devi, T.R., 2013, April. Importance of cryptography in network secu- rity. In 2013 International conference on communication systems and network technologies (pp. 462-467). IEEE.

Chandra, S., Paira, S., Alam, S.S. and Sanyal, G., 2014, November. A comparative survey of symmetric and asymmetric key cryptography. In 2014 international conference on electronics, communication and computational engineering (ICECCE) (pp. 83-93). IEEE.

Yegireddi, R. and Kumar, R.K., 2016, November. A survey on con- ventional encryption algorithms of Cryptography. In 2016 International Conference on ICT in Business Industry Government (ICTBIG) (pp. 1-4). IEEE.

Maqsood, F., Ahmed, M., Ali, M.M. and Shah, M.A., 2017. Cryptogra- phy: a comparative analysis for modern techniques. International Journal of Advanced Computer Science and Applications, 8(6), pp.442-448.

Souvik, B., Bratati, R., Debrupa P., 2021. Network Securityand Cryptography: A Review. International Journal of Advances in Engineering and Management (IJAEM), Volume 3, Issue 7 , pp: 4172-4176.

Khalifa, O.O., Islam, M.R., Khan, S. and Shebani, M.S., 2004, October. Communications cryptography. In 2004 RF and Microwave Conference (IEEE Cat. No. 04EX924) (pp. 220-223). IEEE.

Ali, K., Akhtar, F., Memon, S. A., Shakeel, A., Ali, A., & Raheem, A. (2020, January). Performance of cryptographic algorithms based on time complexity. In 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. 1-5). IEEE.

Surendran, S., Nassef, A., & Beheshti, B. D. (2018, May). A survey of cryptographic algorithms for IoT devices. In 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT) (pp. 1-8). IEEE.

Mushtaq, M. F., Jamel, S., Disina, A. H., Pindar, Z. A., Shakir, N. S. A., & Deris, M. M. (2017). A survey on the cryptographic encryption algorithms. International Journal of Advanced Computer Science and Applications, 8(11).

Elkabbany, G. F., Rasslan, M., & Aslan, H. K. (2017, November). Enhanement of elgamal signature scheme using multi-processing system. In 2017 Intl Conf on Advanced Control

Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT) (pp. 179-186). IEEE.

Salavi, R. R., Math, M. M., & Kulkarni, U. P. (2019). A survey of various cryptographic techniques: From traditional cryptography to fully homomorphic encryption. In Innovations in Computer Science and Engineering (pp. 295-305). Springer, Singapore.

Yang, X., Hou, Y., Ma, J., & He, H. (2019). CDSP: A solution for privacy and security of multimedia information processing in industrial big data and internet of things. Sensors, 19(3), 556.

Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017, August). Comprehensive study of symmetric key and asymmetric key encryption algorithms. In 2017 international conference on engineering and technology (ICET) (pp. 1-7). IEEE.

Elkabbany, G. F., Rasslan, M., & Aslan, H. K. (2017, November). Enhanement of elgamal signature scheme using multi-processing system. In 2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT) (pp. 179-186). IEEE.

Wu, Z., Su, D., & Ding, G. (2014, July). ElGamal algorithm for encryption of data transmission. In 2014 International Conference on Mechatronics and Control (ICMC) (pp. 1464-1467). IEEE.

Antonios, F., Nikolaos, P.A., Panagiotis, M., & Emmanouel, A.V. (2006). Hardware Implementation of Triple-DES Encryption / Decryption Algorithm.