International Journal of Scientific Research and Engineering Development-– Volume 8 Issue 6, Nov- Dec 2025
Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Cryptography Algorithms Analysis

Tushar Galande, Yash Kamble,
Rupali Pawar, Trupti Tasgaokar
Master of Computer Application,
Zeal Institute of Business Administration, Computer Application and Research,
Pune-41104
Email:tushargalande40@gmail.com
Email: kyash5384@gmail.com
Email: rupali.pawar@zealeducation.com
Email: trupti.kapoor@zealeducation.com

Abstract:

Data security has become a critical issue in today's information technology landscape. The concern grows even more in cloud environments, where data is stored across multiple global locations. Encryption serves as an effective way to address this challenge, and various encryption algorithms play a key role in protecting cloud-based information. These algorithms ensure that stored data is accessible only to authorized users.

In this paper, we present the fundamental features—such as key length and block size—of different encryption categories: symmetric algorithms (AES, DES, 3DES, Blowfish [8], RC4), asymmetric algorithms (RSA, DSA, Diffie–Hellman [6], ElGamal, Paillier), and hashing techniques (MD5, MD6, SHA, SHA-256). Additionally, we implemented five commonly used encryption methods, including AES, DES, Blowfish [8], RC4, and RSA, and evaluated their performance by measuring the encryption and decryption time for files of varying sizes on a local system.

Keywords- **Cloud Security, Encryption Algorithms, AES, DES, Blowfish, RC4, RSA, Symmetric Cryptography, Asymmetric Cryptography, Hashing Techniques, Performance Evaluation, Data Protection.**

1. INTRODUCTION

Cloud computing provides a variety of IT services—such as storage, networking, hardware, software, and other computing resources—through the internet. These services offer many advantages like easy and flexible access to data from anywhere, cost savings, and high reliability. Because of these benefits, many organizations are moving their data to the cloud. However, this shift also increases the need to protect data from unauthorized access, modification, or misuse. Security becomes extremely important because cloud systems store sensitive and valuable information. One of the most effective ways to secure data is through encryption. Modern encryption and decryption techniques mainly fall into

three categories: Symmetric-key algorithms, where the same key is used to encrypt and decrypt the data. Asymmetric-key algorithms, where a public key is used to encrypt the data and a private key is used to decrypt it. Hashing algorithms, which help maintain the integrity of data by creating a fixed-length output that cannot be reversed. This paper examines five widely used encryption algorithms—AES, DES, RC4, Blowfish [8], and RSA. Their performance is tested using different file

CRYPTOGRAPHY ALGORITHMS

Cryptography means "secret writing" which is the science and art of transforming messages to make

them secure and immune to attacks by unauthorized user. The original data/message, before being transformed is called cipher text. An encryption is a process to transform the plaintext into cipher text and decryption transforms the cipher text back into plaintext. The sender uses an encryption algorithm and the receiver uses a decryption algorithm. Thus, encryption and decryption help to secure transmission of the message and protect the message from unauthorized users .

There are three types of cryptography algorithm that are given below:

- Symmetric key cryptography algorithm
- Asymmetric key cryptography algorithm
- Hashing cryptography

Symmetric (Secret) Key Cryptography: In this method, the same key is shared between the sender and the receiver. The sender uses this key along with an encryption algorithm to convert the original data into a secure form. The receiver then uses the same key with a matching decryption algorithm to turn the data back into its original form.

Below is a brief explanation of some commonly used symmetric key encryption algorithms.

AES (Advanced Encryption Standard) [4][5] is a symmetric block encryption method recommended for protecting data [3] [5]. It uses the same key for both encryption and decryption. AES supports key lengths of 128, 192, or 256 bits, and the default key size is 256 bits . It works on 128-bit data blocks and performs 10, 12, or 14 rounds depending on the chosen key size .

DES (Data Encryption Standard) is a symmetric block-cipher method recommended by NIST [3]. It became one of the most widely used encryption techniques. In DES, the same key and algorithm are used for both encryption and decryption, with only slight differences in how each process works. It takes a **64-bit plaintext input** and uses a **56-bit key** (plus 8 parity bits) to generate a **64-bit encrypted output** [13] [28].

Triple Data Encryption Algorithm [7] (TDEA or Triple DEA) is a symmetric block-cipher method that works like DES but applies the encryption process three times, which increases its security level [6]. Because of this, it is slower compared to many other block-cipher algorithms. Triple DES uses a 64-bit block size and a 192-bit key.

Blowfish [8] is a symmetric-key encryption algorithm that works on 64-bit blocks and supports a variable key size ranging from 128 to 448 bits. It offers better performance than many other algorithms in terms of speed and power usage .

The RC4 (Rivest Cipher 4) [9] algorithm is a shared-key stream cipher that requires a secure way to exchange the key between users. RC4 is used in standards like IEEE 802.11, especially in WEP (Wireless Encryption Protocol), with key sizes of 40 bits and 128 bits.

To create the keystream, RC4 uses a **secret internal state** made up of two parts:

1. A permutation of all **256 possible bytes** (called "S").
2. Two **8-bit index pointers**, named "i" and "j".

This permutation is set up using a variable-length key (commonly **40 to 256 bits**) through the **key-scheduling algorithm (KSA)**.

Asymmetric key cryptography uses two different keys:

- a **public key** for encryption
- a **private key** for decryption

The sender uses the receiver's **public key** to encrypt the message. Only the receiver, who has the **private key**, can decrypt and read the encrypted data.

Below is a brief explanation of some commonly used asymmetric key encryption algorithms.

RSA (Rivest–Shamir–Adleman) [11] is a widely used asymmetric encryption and decryption method that works with a public key and a private key. The public key can be shared openly and is used to encrypt messages. Any message encrypted with the public key can be decrypted only with the private key. RSA helps protect user data by encrypting it before storage, securing user authentication processes, and creating safe communication channels for data transfer.
A 4096-bit key is used for the RSA algorithm in this work.
The RSA algorithm works in three main steps:

1. **Key Generation**
2. **Encryption**
3. **Decryption**

Diffie–Hellman [6]: The scheme was first revealed by Whitfield Diffie and Martin Hellman in 1976. Diffie–Hellman [6] key exchange is a specific method of exchanging cryptographic keys. With this method, two users who have never met or shared any information before can create a common secret key over an insecure channel. This shared key can later be used to encrypt their communication using a symmetric encryption method.
Paillier [6]: The Paillier [6] cryptosystem is an asymmetric encryption method. One of its special features is its homomorphic property, which means it can add encrypted values together and produce an encrypted result. Later, this encrypted sum can be decrypted to get the final answer—even though the individual values were never seen or known.
Hashing Cryptography: Hash functions are a fundamental elementary in the field of cryptography, used widely in a broad spectrum of important applications involving: message integrity and authentication, digital signatures, secure time stamping, and many other security processes.
A hash function **H** is a fast algorithm that takes an input message **M** of any length, and sometimes a fixed-length key **K** (in the case of a keyed hash function), and produces a **fixed-length output D**, known as the **message digest**.
The function works as:
H (K, M) = D
Where:

- **D** = Message Output
- **K** = Fixed Key Length
- **M** = Input Message Length

Below is a brief explanation of some widely used hashing algorithms.
MD5 (Message Digest5) [1] is a broadly used cryptographic hash function with a 128-bit hash value. It processes a variable-size message into a fixed-length output of 128 bits . The input message is split into 512-bit blocks, and padding is added to make the message length a multiple of 512 bits. In this method, the sender uses the receiver's public key to encrypt the message, and the receiver uses a private key to decrypt it.
The MD6 Message-Digest Algorithm is a cryptographic hash function. MD6 makes use of a substantially different tree-based mode of operation that allows for greater parallelism . MD6 can be seen as a tree-like design, where a **4-to-1 compression function** is applied at each level to reduce the message size step by step.
SHA (Secure Hashing Algorithm) [10] is a hashing algorithm. SHA-1 is most extensively used SHA hash function, but very quickly it is going to be replaced by the newer and stronger SHA-2 hash function. It is currently used in a wide variety of applications, including TLS, SSL, SSH and PGP.SHA1 outputs a 160-bit digest of any sized file or input. SHA-256 algorithm produces an almost-unique, fixed size 256-bit (32-byte) hash . SHA-256 is one of the improved versions that follows SHA-1 and is considered one of the strongest hashing functions in use today. It performs its operations using 32-bit words.

Table-1: Characteristics of Cryptography Algorithm

Scheme	Algorithm Type	Contributor	Key Length	Rounds	Block Size
AES	Symmetric Block Cipher	Rijndael	150, 208, 272 bits	11, 13, 15	136 bits
DES	Symmetric Block Cipher	IBM 75	60 bits	18	68 bits
3DES	Symmetric Block Cipher	IBM 78	120, 240 bits	54	68 bits
Blowfish	Symmetric Block Cipher	Bruce Schneier 93	140–480 bits	22	72 bits
RC4	Symmetric Stream Cipher	Ronald Rivest 87	56–300 bits	—	Stream cipher (no block)
RSA	Asymmetric Public-Key	Rivest, Shamir, Adleman 77	1536, 4096 bits	1	Variable (≥ 640 bits)
DSA	Asymmetric Signature	NIST 91	1536, 3072 bits	—	—
Diffie–Hellman	Asymmetric Key Exchange	Diffie, Hellman 76	2048, 4096 bits	—	—
El-Gamal	Asymmetric Public-Key	Elgamal 84	1536–3584 bits	—	—
Paillier	Asymmetric Homomorphic	Paillier 99	3072 bits	—	—
MD5	Hashing Algorithm	Rivest 91	128-bit output (fixed)	—	528-bit blocks
MD6	Hashing Algorithm	Prof. Rivest 08	Up to 600-bit output (variable)	—	Tree-structured (no fixed)
SHA-1	Hashing Algorithm	NIST 95	160-bit output (fixed)	—	544-bit blocks
SHA-256	Hashing Algorithm	NIST	256-bit output (fixed)	—	544-bit blocks

Table-1 compares different cryptographic algorithms such as AES, DES, 3DES, Blowfish [8], RC4, RSA, DSA, Diffie–Hellman [6], El-Gamal, Paillier, MD5, MD6, SHA, and SHA-256. The comparison is based on five factors: the algorithm name, who created it, key length, number of rounds, and block size.

Each algorithm uses a different key size, which affects its security level. For example, in the updated data, DES uses a 60-bit key, while AES supports 150, 208, and 272-bit keys. Blowfish [8] has a flexible key range from 140 to 480 bits, giving it more variety. RSA, which is an asymmetric algorithm, uses very large keys such as 1536 bits, making it suitable for secure communication.

EXPERIMENTAL METHODOLOGY & ENVIRONMENT

In this experiment, we tested different encryption algorithms to see how well they perform. The tests were done on a normal computer using different sizes of input data. The results depend on three main parts:

a) Evaluation Parameters (What we measured)

These are the things we checked for each algorithm:

1. Encryption Time

- This is the amount of time an algorithm takes to convert plain text → cipher text (normal readable data → encrypted unreadable data).

2. Decryption Time

- This is the amount of time an algorithm takes to convert cipher text → plain text (encrypted data → original data).

b) Evaluation Platform (Where the testing was done)

The performance depends on the computer and software used. The testing setup was:

1. Software Used

- Eclipse (for Java programming)
- JDK 8 (Java Development Kit)
- MATLAB 2014
- Windows 8.1 Pro (64-bit)

2. Hardware Used (Computer Specifications)

- Processor: Intel Core i5 (2.40 GHz), 4th generation
- RAM: 4 GB
- Storage: 1 TB HDD

c) Key Management (Key sizes used)

Key management means how the secret keys used for encryption are handled. A stronger and larger key means better security.

Here are the key sizes used for each algorithm:

Table-2: Algorithm and it's Key Size Used

Algorithm	Key Size Used
AES	256 bits
Blowfish	128 bits
DES	56 bits
RC4	64 bits
RSA	1024 bits

EXPERIMENTAL RESULTS AND ANALYSIS

Experimental results for the AES, Blowfish [8], DES, RC4, and RSA encryption algorithms are shown in the updated Table-2. In this study, the algorithms were tested using the new input file sizes of 350 bytes, 850 bytes, and 2200 bytes (for AES), and similar updated file sizes for the other algorithms as shown in the table. The key sizes used for each algorithm were also updated, such as AES (256-bit), Blowfish [8] (128-bit), DES (56-bit), RC4 (64-bit), and RSA (1024-bit).

To ensure accurate results, each operation was executed 100 times, and the average value was calculated for better comparison. The encryption and decryption times were recorded in milliseconds, and the input file sizes were measured in bytes. These average results were then used to compare the performance of the algorithms and to prepare the graphs. All observations and measurements were taken on the same system to maintain consistency.

Table-3: Performance Comparison of Different Algorithms

S. No	Algorithm	Key Size (bit)	File Size (bytes)	Avg Encryption Time (ms)	Avg Decryption Time (ms)
1	AES	256	350	275	285
			850	315	330
			2200	365	380
2	Blowfish	128	360	255	265
			900	290	305
			2300	330	345
3	DES	56	340	310	325
			820	350	365
			2100	400	420
4	RC4	64	345	220	235
			780	250	265
			2000	300	315
5	RSA	1024	400	540	560
			950	600	625
			2500	660	690

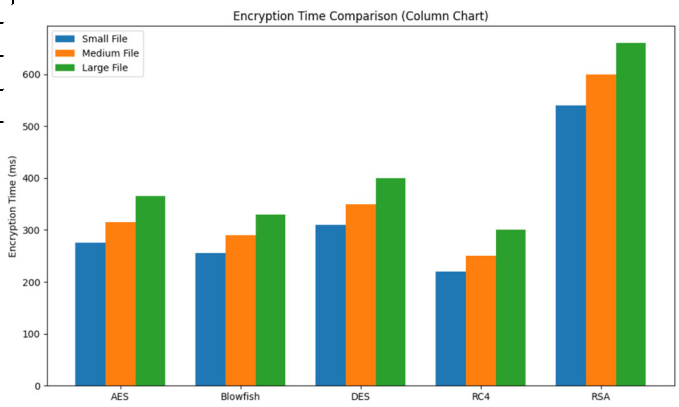

Fig. 1 Encrytion Time Comparison(Column Chart)

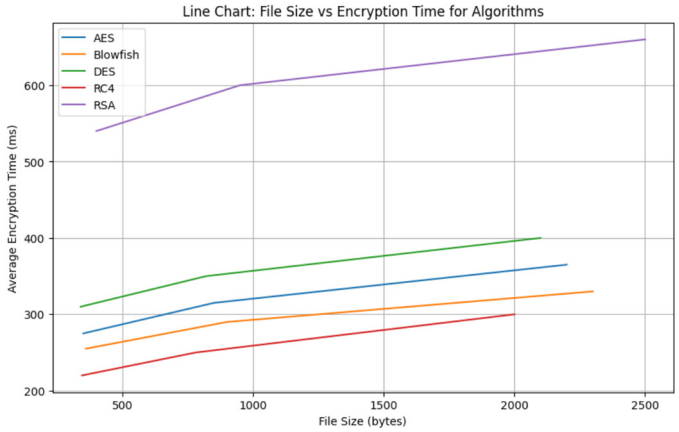

Fig. 2 Line Chart: File Size vs Encryption Time for Algorithms

The graphs in figures 1 and 2 show the encryption times, while figures 3 and 4 show the decryption times for the symmetric algorithms (AES, Blowfish [8], DES, and RC4) and the asymmetric algorithm (RSA).

From these graphs, we can see that symmetric algorithms work much faster than the asymmetric RSA algorithm for both encryption and decryption.

In most cases, when the file size increases, the time taken by the algorithm also increases. This trend is clear for almost all algorithms, except DES and RSA, which do not show a strong increase in time as the file size grows.

If we look at Table-2, it is clear that RSA takes the highest time for both encryption and decryption when compared to AES, Blowfish [8], DES, and RC4. The symmetric algorithms are much quicker and more efficient in comparison.

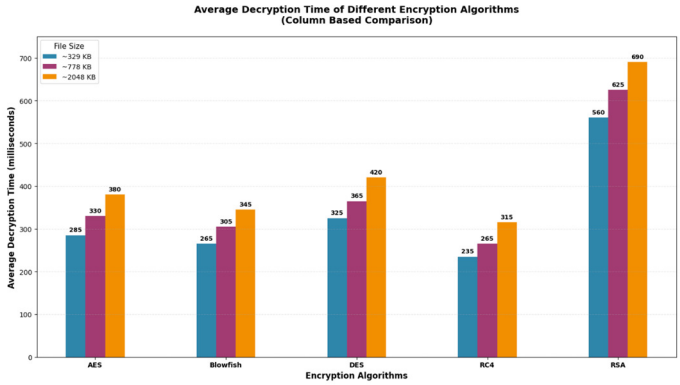

Fig. 3 Average Decryption Time of Different Encryption Algorithms (Column Based Comparison)

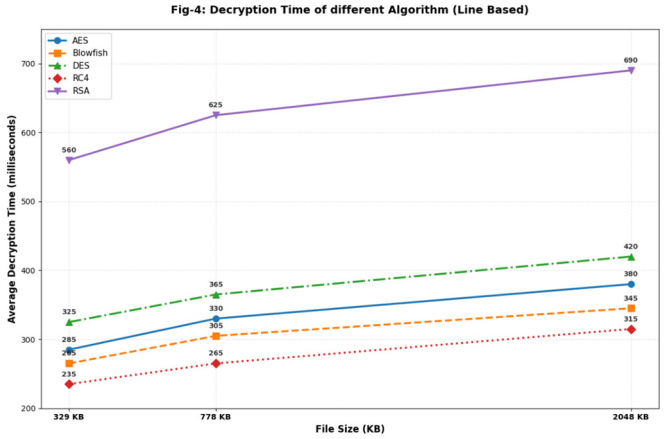

Fig. 4:Decryption Time of different Algorithm

CONCLUSION

In this study, a detailed examination of symmetric, asymmetric, and hashing-based cryptographic algorithms was carried out to understand their structural differences, security strength, and performance behavior. The experimental analysis demonstrated that symmetric algorithms—particularly AES, Blowfish, and RC4—provide significantly faster encryption and decryption times compared to asymmetric algorithms like RSA. As file size increases, the execution time generally rises for all algorithms, with RSA showing the slowest performance due to its computational complexity.

The comparison of algorithm characteristics highlights how key size, block size, and operational rounds influence overall security and efficiency. Symmetric-key algorithms are more suitable for large data encryption because of their speed, whereas asymmetric algorithms are ideal for secure key exchange and authentication. Hashing algorithms serve as essential tools for ensuring data integrity.

Overall, the study confirms that no single algorithm fits all security requirements; instead, a combination of cryptographic techniques is often necessary to achieve strong, reliable, and efficient data protection in cloud environments.

ACKNOWLEDGMENT

The authors would like to express their gratitude to the faculty members and mentors for their valuable guidance and continuous support throughout the research work. We also acknowledge the use of laboratory facilities and computational resources that made the experimentation and performance analysis possible. Special thanks to all contributors who provided constructive feedback that helped in improving the quality of this study.

REFERENCES

[1] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson, 2017.

[2] B. Schneier, *Applied Cryptography*, 2nd ed. Wiley, 1996.

[3] National Institute of Standards and Technology, “Data Encryption Standard (DES),” FIPS PUB 46-3, 1999.

[4] National Institute of Standards and Technology, “Advanced Encryption Standard (AES),” FIPS PUB 197, 2001.

[5] J. Daemen and V. Rijmen, *The Design of Rijndael: AES — The Advanced Encryption Standard*. Springer, 2002.
[6] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.
[7] National Institute of Standards and Technology, “Recommendation for the Triple Data Encryption Algorithm [7] (TDEA),” SP 800-67, 2017.
[8] B. Schneier, “Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish [8]),” *Fast Software Encryption*, Springer, 1994.
[9] R. Rivest, “RC4 Stream Cipher,” RSA Security, 1994.
[10] National Institute of Standards and Technology, “SHA-256 Standard,” FIPS PUB 180-4, 2015.
[11] R. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” *Communications of the ACM*, 1978.

