Performance Analysis of Symmetric Encryption Algorithms for Enhancing Data Security

M.Pravallika[1] Dr. P. Bhargavi[2*]
[1]Research Scholar, Dept. of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati
[2]Assistant Professor, Dept. of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati*

Abstract: *With the increased adoption of cloud technology, securing sensitive data against cyberattacks through multi-layer encryption has become essential. In this multi-layer encryption, the first layer plays a crucial role because it balances both security and performance, and the second layer provides additional protection to safeguard data even if the first layer is compromised. This paper presents an Analysis of various symmetric encryption algorithms like AES-256, RC6, Blowfish, DES, and Chacha20 and a performance evaluation of these symmetric encryption algorithms is conducted based on the parameters of performance metrics and security strength. It explores secure session key exchange using asymmetric encryption algorithms and authenticating public keys within the framework of the public key infrastructure (PKI). This paper suggests that AES-256 is the optimal choice for the first layer of encryption, while RSA is employed for secure key exchange due to its support for both encryption and digital signatures. To validate these findings, real-world testing has been conducted with various sizes of financial datasets to determine the best symmetric encryption algorithm.*

Keywords: **Asymmetric Encryption, Encryption, Public Key Infrastructure, Symmetric Encryption, Secure Key Exchange.**

1. INTRODUCTION

In today's digital era, securing sensitive financial data during transmission and at rest has become a primary concern. As the increased adoption of digital platforms in the financial sector even from mobile payment apps to digital banking services because of their speed, scalability and efficiency that introduces a new security challenge such as man-in-the-middle attack, data breach, identity theft etc., which leads to compromised integrity of transactions, exposure of financial frauds and damages trust in the system.

To protect sensitive data from these risks, data security is necessary and it primarily focuses on the encryption mechanism that provides confidentiality and integrity to the data. Whether financial transactions are being processed, stored, or transmitted, encryption plays a protective role as a shield, making the data unreadable to attackers.

This article examines the performance of the various symmetric encryption algorithms like AES-256, DES, Blowfish, RC6, ChaCha20, based on the parameters of performance metrics, such as encryption time, decryption time, CPU Usage, Memory Usage, throughput and latency and security strength like Key size, Block size and security resistance, helps to identify the best algorithm that effectively balances the sensitive data protection and explores secure session key exchange for data encryption using asymmetric encryption algorithms and authenticating public keys within the framework of the public key infrastructure (PKI).

2. Related Works

This section describes the previous works on analysing the performance of the various symmetric encryption algorithms.

Kuntal Patel (2019) [9] has done research work on "Performance analysis of AES, DES and Blowfish cryptographic algorithms on small and large data files". Here, the author examines the performance of AES, DES and Blowfish algorithms on the basis of execution speed, memory usage, implementation efficiency and includes that encryption algorithms help in securing data against threats.

Ezeonyi, N. U., Okonkwo, O. R., & Enweka, O. A. (2023) [3] have done research work on "Applications of Information Cryptography in Its Various Stages of Evolution, from Antiquity to the Modern Era". Here, the author discussed the evolution of cryptography from the ancient to the modern era and included the concepts of encryption, decryption, key model, code-breaking techniques and advancements of the techniques.

Gupta, S., & Sharma, J. (2012, December) [30] have done research work on "A hybrid encryption algorithm based on RSA and Diffie-Hellman". Here, the author discusses the importance of these encryptions in securing the data, including the concept of key distribution for encryption, sharing of key for decryption by authorized users, confidentiality, integration, authentication and non-repudiation. These concepts are satisfied by the RSA and Diffie-Hellman algorithms.

Singh, S., Maakar, S. K., & Kumar, S. (2013) [27] have done research work on "A performance analysis of DES and RSA cryptography". Here, the author highlights the security of data during transit and performs an analysis of the algorithms AES, DES, and RSA based on encryption and decryption time.

Sood, R., & Kaur, H. (2023) [4] have done research work on "A literature review on RSA, DES and AES encryption algorithms". Here, the author highlights the security of the data during transmission and includes a comparative study of the RSA, DES and AES algorithms based on the encryption and decryption time.

Patil, P., Narayankar, P., Narayan, D. G., & Meena, S. M. (2016) [19] have done research work on "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish". Here, the author evaluated the performance of the algorithms AES, DES, 3DES, RSA and Blowfish based on encryption time, decryption time, entropy, number of bits encoded, avalanche effect and memory used.

Panda, M. (2016, October) [20] has done research work on "Performance analysis of encryption algorithms for security". Here, the author evaluated both symmetric algorithms AES, DES, Blowfish and asymmetric algorithm RSA using the files of Binary text and image files based on the parameters encryption time, decryption time and throughput.

3. Data Security

Data Security refers to a set of policies, technologies, and best practices used to protect data from unauthorized access. It preserves the confidentiality, integrity, and availability of data through mechanisms like encrypting data in transit and at rest. A good data security means having a strong, well-managed and adaptive system that protects against threats and can be accessed by authorized users, preventing misuse or loss of data. The main goal is to keep data secure and available. This will happen with encryption. In data security, encryption is the most important tool to provide data privacy and Security.

So, Cryptography plays a vital role in providing security to the data at rest and in transit by encrypting the data before sending it into the network and to the receiver (cloud, server).

Cryptography is an art of science that transforms readable text into unreadable text. It is a technique that encrypts the original text to cipher text and decrypts the cipher text to the original text with the help of encryption algorithms. Cryptography is classified as Symmetric Key and Asymmetric Key cryptography.

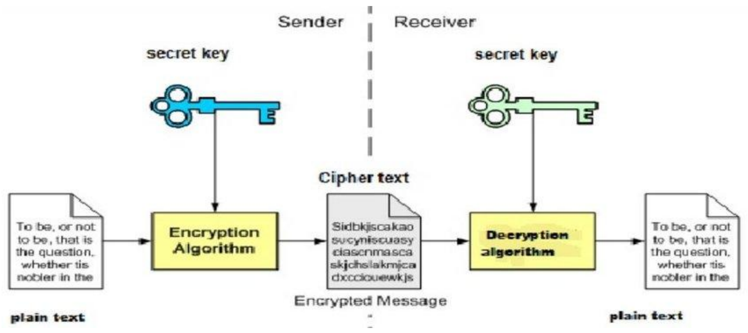

Sender
Receiver
secret key
secret key
Cipher text
To be, or not to be, that is the question, whether tis nobler in the
Encryption Algorithm
Sidbkjscakao sucynlscuasy dasenmasca skjchslakmjca dxccciouewkjs
Decryption algorithm
To be, or not to be, that is the question, whether tis nobler in the
plain text
Encrypted Message
plain text

Figure 1. Cryptography

Symmetric Key Cryptography is also known as Private Key Cryptography. Here, encryption and decryption happen with the same key. It is widely used due to its efficiency, speed and great choice for safeguarding a high volume of data in cloud environments.

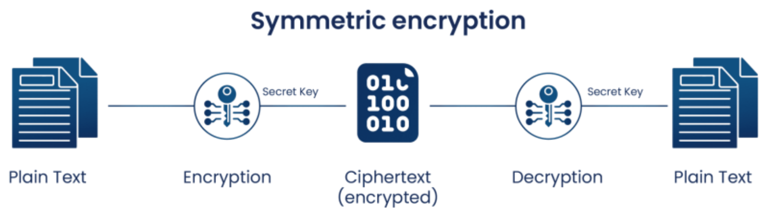


Figure 2. Symmetric Encryption

Asymmetric Cryptography is also known as Public Key Cryptography. Uses a pair of keys for encryption and decryption: one is a public key used for encrypting the data and a corresponding private key for decryption.

If the sender encrypts the message with the receiver's public key, when it reaches the receiver's side, the message will be decrypted with the help of the receiver's private key. It is widely used for secure key exchange, digital signatures and identity verification.

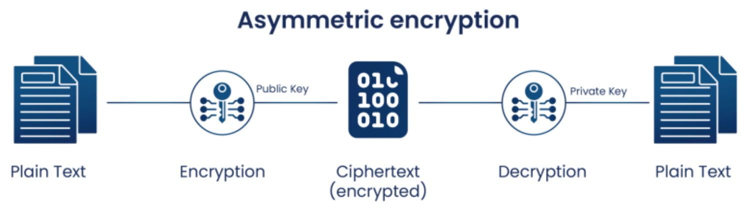


Figure 3. Asymmetric Encryption

3.1. Symmetric Algorithms for Data Encryption and Decryption

A.a Des (Data Encryption Standard)

Overview

Data Encryption Standard (DES) is a Symmetric encryption algorithm. It was designed by IBM and published by the National Institute of Standards of Technology (NIST) in 1977. DES is the first encryption algorithm, and initially, it was considered a strong encryption algorithm, due to its small key size, which limits its usage for large amounts of data encryption.

Architecture

1) The structure of DES is a Feistel Cipher Structure.
2) It encrypts 64-bit plaintext.
3) It performs the initial permutation and Final Permutation.
4) It consists of 16 rounds between the initial and final permutation.
5) Its key size is 64 bits, but only 56 bits are used for encryption, leaving 8 bits for every bite. 1 bit is used as a parity bit.
6) This 56-bit key is divided into 16 sub-keys, each consisting of 48 bits, and it is used for every round.
7) It consists of 8 S-boxes, and the same process is used in reverse for decryption.

A.b Aes-256 (Advanced Encryption Standard)

Overview

Advanced Encryption Standard (AES) is a symmetric Encryption Algorithm. It was designed by Joan Daemen and Vincent Rijmen and published by the National Institute of Standards of Technology (NIST) in 2001. This algorithm is also called the Rijndael algorithm.

Architecture

1) It follows the Feistel Cipher Structure.
2) It is a Symmetric block cipher algorithm with a block size of 128 bits.
3) It takes 128 bits of plain text as input and produces 128 bits of cipher text as output.
4) Its key size is 256 bits and performs 14 rounds.
5) For every round, it uses a key with the size of 128 bits and 128 bits of plain text.
6) Each round consists of 4 transformations, i.e. Substitute bytes, shift row, mix column, Add round key.

A.c Blowfish Algorithm

Overview

Blowfish Algorithm is a symmetric key algorithm. It was designed by Bruce Schneier in 1993. It is an alternative to the DES encryption technique and the IDEA Algorithm.

Architecture

1) It follows the Feistel Cipher structure.
2) It is a symmetric block cipher algorithm with a block of 64 bits.
3) It has a variable key length, i.e., key size lies between 32 to 448 bits.
4) It consists of 18 subkeys and is stored in the P-array.
5) It performs 16 rounds.
6) It consists of 4 S-boxes having 256 entries each of 32 bits in size.

A.d RC6 (Rivest Cipher 6)

Overview

RC6 is a symmetric key algorithm. It was designed by Ron Rivest, Matt Robshaw, Ray Sidney and Yiqun Lisa Yin. They designed this algorithm to compete with the AES algorithm. It is derived from the RC5 algorithm.

Architecture

1) It is a symmetric block cipher algorithm.
2) Each block consists of 128 bits.
3) It uses Keys with the sizes of 128,192, and 256 up to 2040 bits.
4) It is very similar to RC5 in structure, using data-dependent rotations, modular addition, and XOR operations.
5) RC6 can be seen as intertwining two parallel RC5 encryption processes, with the key distinction being an additional multiplication operation that ensures the rotation depends on every bit in a word rather than just the least significant ones.

A.e ChaCha20

Overview

ChaCha20 is a symmetric key algorithm. It was designed by Daniel J. Bernstein. This algorithm is very well known for its speed and efficiency, but it works well on software applications.

Architecture

1) It is a Symmetric stream cipher algorithm.
2) It performs key generation using the key of a 256-bit key, along it uses 96-bit nonce.
3) With the help of the Key and nonce, it will set up a cipher's state.
4) It encrypts data with a block size of 512 bits using the current cipher state.
5) It performs XORing of the plaintext with the output of the data encryption step, and generates output as ciphertext.

4. Encryption Modes

Encryption algorithms act like changing the plaintext to the ciphertext, but they require the modes of operation that determine the process of converting plaintext to ciphertext. By default, encryption algorithms only ensure confidentiality, but with the help of these encryption modes, which provide security, efficiency, resistance to attacks and authentication.

After evaluating these modes based on the components like working process, security level and authentication, which are shown in Table I, AES is the most suitable choice for data encryption, as it operates in GCM mode, which is the most powerful and secure. It not only provides the mode for operating encryption, but also provides the operation of authentication. The remaining modes work only for encryption; there's no authentication. So, GCM is the best mode to run with the AES Encryption algorithm.

The remaining encryption algorithms DES, RC6, ChaCha20, and Blowfish are not supported with this GCM encryption mode. So, these algorithms are evaluated in CTR mode, which is the second most secure encryption mode.

Table 1. Various Encryption Modes

Mode	***Type***	***Working Process***	***Security Level***	***Authentication***
ECB (Electronic Codebook)	Block Cipher	Encryption of each block is done separately.	Weak	No
CBC (Cipher Block Chaining)	Block Cipher	Every plaintext block is XORed with the previous ciphertext block before encryption	Strong	No
CFB (Cipher Feedback Mode)	Block Cipher	Here, a block cipher is converted to a stream cipher, processes the data into small units instead of full blocks with the help of a shift register and an initialization vector. For the next block of encryption, it uses the previously generated ciphertext.	Strong	No
OFB (Output Feedback Mode)	Block Cipher	It is similar to CFB, but the difference is it sends encrypted output instead of sending XORed output to the next block encryption	Strong	No
CTR (Counter Mode)	Block Cipher	It is similar to OFB, but the difference is that it uses a counter value and supports parallel encryption because of independent block encryption	Very Strong	No
GCM (Galois/Counter Mode)	AHEAD	CTR + Authentication	Highly Secure	Yes

However, the main drawback of these symmetric encryption algorithms is that exchanging keys from the sender and receiver is necessary because they require a single key for both sides. If the key exchange occurs through an insecure channel, it leads to compromised communication between the sender and receiver. Simultaneously, the encrypted data over the network is also at risk. Therefore, we need a method that ensures secure key exchange. Secure key distribution takes place here with asymmetric key cryptography or asymmetric encryption algorithms.

5. Asymmetric Key Cryptography

Asymmetric Key Cryptography is also called with Public Key Cryptography. It consists of two keys: one is the Public Key and the other is the Private Key. Here, encryption and decryption are done with different keys, If the sender encrypts the data with the receiver's public key and sends it to the receiver, then the receiver will decrypt the message with his/her private key.

Here, the problem of key exchange is eliminated. It is widely used for authenticating the identities of users, devices, or servers, generating a digital certificate or signature and for secure key exchange. But not the right choice for encrypting a large amount of data.

Various asymmetric algorithms include RSA, Elliptic Curve Cryptography, Diffie-Hellman Key exchange, Digital Signature Algorithm, ECDH, and ECDSA. Among these, choose the algorithm that supports both encryption and a Digital signature.

Table 2. Comparison of Various Asymmetric Algorithms

Algorithm	***Purpose***	***Encryption***	***Signature***	***Key agreement***
RSA	Encryption and Digital Signature	Yes	Yes	Yes
DSA	Digital Signature only	No	Yes	No
Diffie-Hellman (DH)	Key exchange	No	No	Yes
ECC (Elliptic Curve Cryptography)	Cryptographic framework	No	No	No
ECC (ECDH)	Key exchange (via ECC)	No	No	Yes
ECC (ECDSA)	Digital Signature (via ECC)	No	Yes	No

By observing Table II, comparing various asymmetric algorithms, the RSA algorithm supports encryption and digital signatures. DSA only supports generating a digital signature. Diffie-Hellman only supports key exchange. ECC is a cryptographic framework; it is not suitable for encrypting and generating digital signatures. This framework can integrate with Diffie-Hellman and Digital signature algorithms, like ECC+DH supports for a key exchange. ECC+DSA supports generating a digital signature, not encryption.

RSA is a suitable algorithm for secure key exchange because this algorithm supports all the parameters, i.e., encrypting key, attaching a digital signature, and key agreement. This algorithm solves the problem of secure key exchange. It mostly supports secure protocols and certificate authorities.

A. Flow of Session Key Exchange

1) The sender encrypts the data with the help of the AES algorithm, which requires a session key to encrypt the data, which is generated at the sender's side.
2) This session key is shared with the receiver. So, for sharing the session key sender requires the receiver's public key.

3) Then the sender encrypts that session key with the receiver's public key using an asymmetric encryption algorithm (Via RSA).
4) Now, the sender encrypts the data with AES.
5) The sender sends the encrypted data and the Encrypted Session key to the receiver.
6) Now, the Receiver decrypts the session key with his/her private key.
7) The receiver decrypts the data with the help of the session key.

While RSA supports the mechanism of securely encrypting the session key, it still requires the sender's trust towards the receiver's public key. If the public key is intercepted, encryption becomes useless, and the communication will be compromised. This is where the PKI framework comes into play.

6. Public Key Infrastructure

Public Key Infrastructure is a mechanism that issues digital certificates and digital signatures to verified public keys. It builds assurance and trust in authenticated identities in digital communication.

If PKI is not used, then the sender trusts the receiver's public key without verifying it, uses it to encrypt a session key and sends it to the receiver. The attacker (posing as the receiver) decrypts the session key with their private key. Now, the attacker can access all the encrypted data and
future communications. This leads to compromised communication and vulnerabilities like man–in–the–middle attacks and unauthorized data access. PKI prevents this by ensuring that the public key is authenticated for use.

A. PKI Workflow

1) ***Key Generation***
   - Every entity (user/server) generates a pair of keys; one is a public key and the other is a private key
2) ***Certificate Signing Request (CSR)***
   - The entity sends a certificate signing request to the Certificate Authority (CA).
   - The CSR includes
     - The public key
     - Identity details
   - Then the server sends the CSR to the CA
3) ***Certificate Issuance by CA***
   - CA verifies the entity's public key and identity information.
   - After validation is successful.
   - Then CA signs the public key and identity information digitally.
   - Then the certificate is issued to the entity, containing:
     - The Public key
     - Identity information
     - The CA's digital signature.
4) ***Certificate Validation***
   - The receiver sends its public key to the sender
   - Then the sender validates the certificate by:
     - Decrypting the certificate with the trusted CA's public key.
     - Verifying the identity and integrity of the certificate.
   - If the certificate validation is successful, then that public key is authenticated as not a fake or malicious one.

This ensures that the public key being used truly belongs to the verified entity, not an attacker.

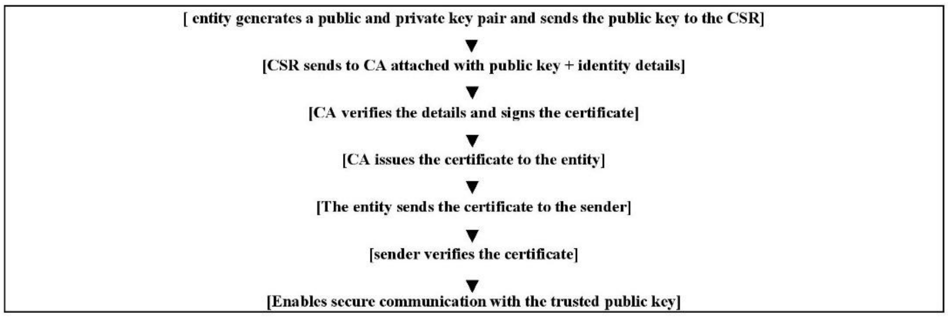

[entity generates a public and private key pair and sends the public key to the CSR]
[CSR sends to CA attached with public key + identity details]
[CA verifies the details and signs the certificate]
[CA issues the certificate to the entity]
[The entity sends the certificate to the sender]
[sender verifies the certificate]
[Enables secure communication with the trusted public key]

Figure 4. Workflow of Public Key Infrastructure

Table 3. Real World Usage of PKI

APPLICATION	PKI USAGE
HTTPS/TLS	In the authentication of the server, secure key exchange
EMAIL SECURITY (S/MIME)	For encrypting and signing the emails
CLOUD AUTHENTICATION	While accessing the cloud services and API's
VPNS	Client-server authentication
SOFTWARE SIGNING	For verification of trusted updates in software

7. Parameters Used to Evaluate the Best Algorithm

To choose the best symmetric algorithm for the first layer of data encryption, evaluating the suitable algorithm based on the parameters of performance metrics and security strength.

A. Performance Metrics

A.a Encryption and Decryption Time

Encryption time means the time taken to convert the readable text to unreadable text. It is measured in seconds.

Decryption time explains the time taken to convert the unreadable text to readable text. It is measured in seconds.

A.b CPU and Memory Utilization

It will evaluate how much CPU power an encryption algorithm requires, which will support parallel processing and reduce the CPU load.

With the help of Hardware acceleration, it speeds up the encryption process and reduces CPU Usage.

Encryption algorithms require RAM to store temporary data, Keys, and buffers to process the data present in the blocks.

While encryption algorithms perform encryption in certain modes, they require some storage for initializing vectors and authenticating tags.

A.c Latency and Throughput

How much quantity of data that are encrypted and decrypted per second, so the best algorithm gives high throughput.

Choosing the best algorithm that gives low latency because encryption and decryption will slightly delay while transmitting the data.

B. Security Strength

B.a Key Size

Selecting the algorithm that has a larger key size so that it provides stronger encryption.

B.b Block Size

Having a larger block size provides high security; a 128-bit block size is preferred for financial data.

B.c Security Resistance

Selected encryption algorithms should provide high security by offering strong cryptanalysis resistance, high brute-force resistance, effective side-channel attack resistance, and enhanced quantum attack resistance, which will be adaptable for both modern and quantum environments.

8. Methodology

A. Objective

The paper aims to evaluate and compare the performance of various symmetric encryption algorithms (AES256, DES, Blowfish, DES, ChaCha20) and exchange the session key using RSA within a PKI-based trust environment. The goal is to identify the most efficient encryption algorithm in terms of encryption, decryption, CPU Usage, Memory Usage, throughput, latency and secure key handling.

B. Environment Setup

Comparison of these algorithms, AES, DES, Blowfish, RC6, and ChaCha20 are performed using the Jupyter Notebook. These Experiments are implemented on a 12th Gen Intel(R) Core (TM) i5-1240p 1.70 GHz processor, 16 GB RAM, 64-bit operating system, and x64-based processor using the Python programming language.

C. Datasets

Generated Datasets

Plaintext Excel files are generated randomly using Python code with the dataset sizes 1MB,10MB,25MB,50MB,75MB, and 100 MB. These datasets consist of financial data with the field names of customer_id, account_number, customer_name, email, phone_number, address, date_of_birth, social_security_number, tax_identification_number, account_type, currency_type, account_balance, minimum_balance, interest_rate, overdraft_limit, transaction_id, transaction_type, transaction_amount, transaction_currency, transaction_timestamp, transaction_status, sender_account_number, receiver_account_number, merchant_id, transaction_location, card_number, card_expiry_date, cvv, card_type, billing_address, payment_gateway_id, authorized_limit, loan_id, loan_amount, loan_term, loan_interest_rate, monthly_installment, remaining_balance, collateral_value.

Real-Time Datasets

Real-Time Dataset with various dataset sizes 143MB, 470 MB, 759 MB and 1200 MB. These datasets consist the financial data with the field names
Id, date, client id, card id, amount, use chip, merchant id, merchant city, merchant state, zip, mcc, error, time, V1 to V28, amount, class, step, type, amount, name origin, old balance origin, new balance origin, nameDest, old balanceDest, newbalance Dest, is Fraud, isFlaggedFraud, Transaction id, timestamp, Sender account, receiver account, amount, transaction type, merchant category, location, device, isfraud, fraud type, time since last transaction, spending deviation score, velocity score, geo anomaly score, payment channel, ip address, device hash.

D. Workflow of Proposed Approach

1) Selecting an efficient symmetric encryption algorithm among AES256, DES, Blowfish, ChaCha20, and RC6 and a suitable encryption mode among GCM, CTR, CBC, ECB, CFB and OFB based on performance analysis.
2) The analysis of the algorithms is evaluated during the encryption and decryption phases, based on the performance metrics such as encryption time, decryption time, CPU Usage, Memory Usage, Throughput and Latency and security strength.
3) Performance analysis was done on the generated and Real-Time datasets and displayed their corresponding outputs.

4) While encrypting the data using a symmetric encryption algorithm, the secret key has to be shared securely, and it is shared secretly with an asymmetric encryption algorithm. A comparative analysis was done to choose the best asymmetric encryption.
5) Finally, evaluated the performance of generated datasets and Real-Time datasets for choosing the best algorithm among them.

E. Data Analysis

1) The performance results are exported and stored in Excel files.
2) These results are analysed by plotting the line graphs.

9. Results and Analysis

A. Performance Evaluation

The Performance Evaluation of Symmetric Encryption Algorithms was on

- The Generated Dataset, which was generated randomly using Python on various dataset sizes: 1MB, 10MB, 25MB, 50MB, 75MB, 100MB.
- The Real -Time Dataset with various dataset sizes: 143MB,470MB,759MB,1200MB.

Performance Evaluation on Generated Datasets

Evaluation of symmetric encryption algorithms AES256, DES, Blowfish, RC6, and ChaCha20, using dataset sizes of 1MB,10MB,25MB,50MB,75MB, and 100 MB. This evaluation considers the performance metrics such as encryption time, decryption time, CPU usage, memory usage, throughput and latency on the Jupyter Notebook environment. The results are visualised in the form graph and a table.

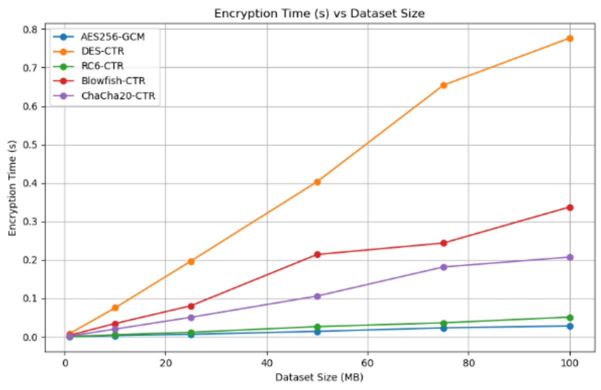


Figure 5.1 Encryption time on Generated Datasets

Figure 5.1 visualizes the analysis of all symmetric encryption algorithms while performing encryption on the datasets. These Visualizations help to choose the best algorithm that takes less encryption time. In Figure 5.1, the X-axis represents the dataset sizes and the Y–axis represents the encryption time in seconds. AES256 takes very little time as file size increases; for large files also it encrypts the data very fast. DES and Blowfish take more time according to the increased file size. RC6, ChaCha20 scales very low as file size increases, but is less efficient than AES256. Overall, AES-256 shows the best result.

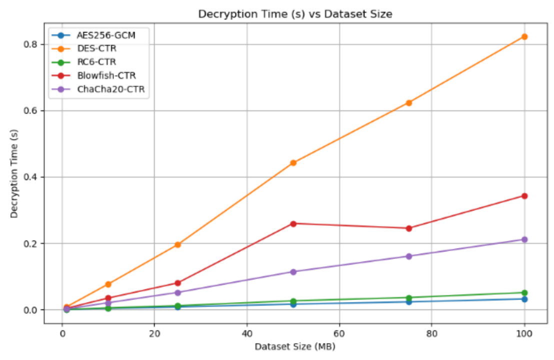


Figure 5.2 Decryption time on Generated Datasets

Figure 5.2 visualizes the analysis of all symmetric encryption algorithms while performing decryption on the datasets. These Visualizations help to choose the best algorithm that takes less decryption time. In Figure 4.2, the X-axis represents the dataset sizes and the Y–axis represents the decryption time in seconds. AES256 takes very little time as file size increases; for large files, it also decrypts the data very fast. DES and Blowfish take more time according to the increased file size. RC6, ChaCha20 scales very low as file size increases, but is less efficient than AES256. Overall, AES-256 shows the best result.

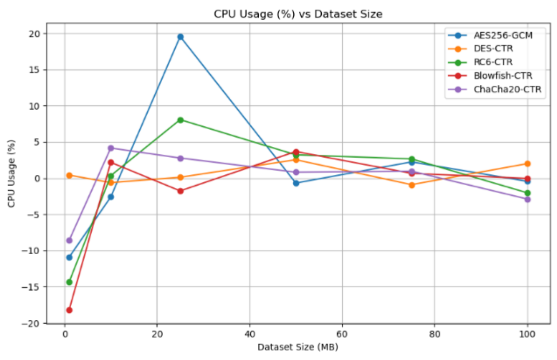


Figure 5.3 CPU Usage on Generated datasets

Figure 5.3 illustrates the amount of processor power consumed by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that consumes less CPU Usage. In Figure 4.3, the X-axis represents the dataset sizes and the Y–axis represents the CPU usage in Percentage. AES256 and RC6 have more fluctuations in CPU usage for the small file sizes; overall, they have stable CPU consistency. DES has a small variation in CPU consumption, and it maintains stable behaviour. Blowfish and ChaCha20 have slight fluctuations; they have good CPU consistency compared to other algorithms. Overall, DES shows the best result.

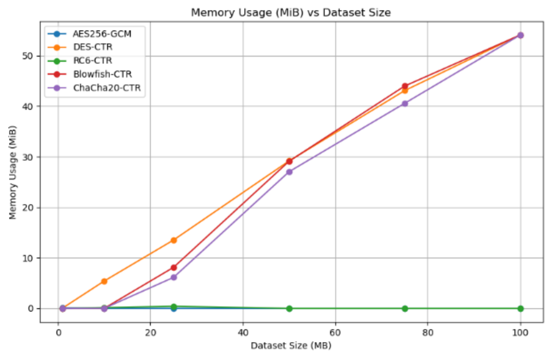


Figure 5.4 Memory Usage on Generated datasets

Figure 5.4 illustrates the amount of memory consumed by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that consumes less memory. In Figure 4.4, the X-axis represents the dataset sizes and the Y–axis represents the Memory usage in MiB. AES256 consumes less Memory for all dataset sizes, it is very efficient. RC6, Blowfish, and ChaCha20 have moderate growth in CPU usage, growth rate increases when the file size increases. DES has a high growth rate in CPU usage; it has bad scaling. Overall, AES-256 shows the best result.

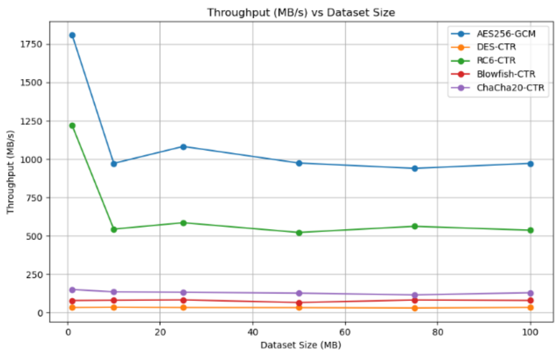


Figure 5.5 Throughput Usage on Generated Datasets

Figure 5.5 illustrates the amount of data processed per second by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that generates higher throughput. In Figure 4.5, the X-axis represents the dataset sizes and the Y–axis represents the throughput in MB/s. AES256 has a high throughput rate, it maintains an excellent speed. RC6 and ChaCha20 have a moderate throughput rate, but still maintain good speed. Blowfish has a moderate throughput with increased file size. DES has less throughput with increased file size, and has a bad scaling. Overall AES-256 shows the best result.

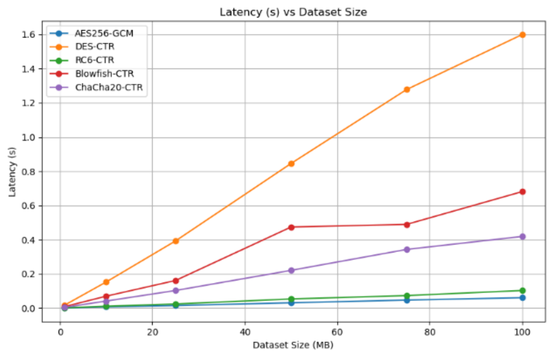
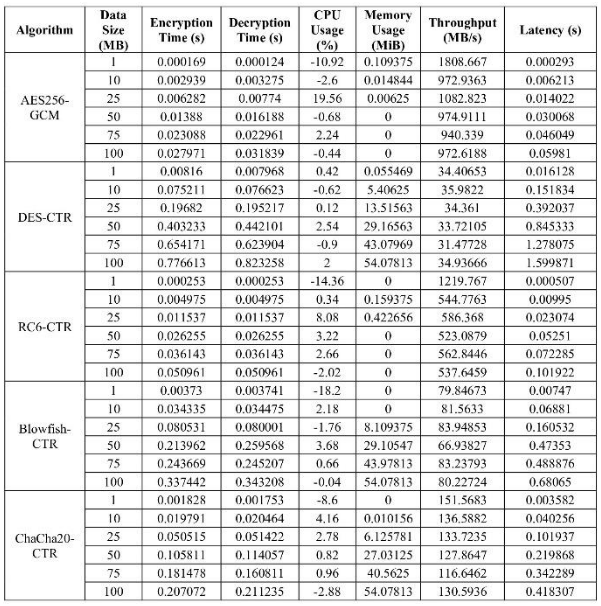


Figure 5.6 Latency on Generated datasets

Figure 5.6 illustrates the amount of processor power consumed by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that consumes less CPU Usage. In Figure 5.6, the X-axis represents the dataset sizes and the Y–axis represents the latency in seconds. AES256 maintains low and consistent growth as the file size increases, and it is still very stable. RC6, Blowfish, and ChaCha20 maintain low latency, but not as efficiently as AES256. DES is producing high-latency results. Overall, AES-256 shows the best result.



Algorithm	Data Size (MB)	Encryption Time (s)	Decryption Time (s)	CPU Usage (%)	Memory Usage (MiB)	Throughput (MB/s)	Latency (s)
AES256-GCM	1	0.000169	0.000124	-10.92	0.109375	1808.667	0.000293
	10	0.002939	0.003275	-2.6	0.014844	972.9363	0.006213
	25	0.006282	0.00774	19.56	0.00625	1082.823	0.014022
	50	0.01388	0.016188	-0.68	0	974.9111	0.030068
	75	0.023088	0.022961	2.24	0	940.339	0.046049
	100	0.027971	0.031839	-0.44	0	972.6188	0.05981
DES-CTR	1	0.00816	0.007968	0.42	0.055469	34.40653	0.016128
	10	0.075211	0.076623	-0.62	5.40625	35.9822	0.151834
	25	0.19682	0.195217	0.12	13.51563	34.361	0.392037
	50	0.403233	0.442101	2.54	29.16563	33.72105	0.845333
	75	0.654171	0.623904	-0.9	43.07969	31.47728	1.278075
	100	0.776613	0.823258	2	54.07813	34.93666	1.599871
RC6-CTR	1	0.000253	0.000253	-14.36	0	1219.767	0.000507
	10	0.004975	0.004975	0.34	0.159375	544.7763	0.00995
	25	0.011537	0.011537	8.08	0.422656	586.368	0.023074
	50	0.026255	0.026255	3.22	0	523.0879	0.05251
	75	0.036143	0.036143	2.66	0	562.8446	0.072285
	100	0.050961	0.050961	-2.02	0	537.6459	0.101922
Blowfish-CTR	1	0.00373	0.003741	-18.2	0	79.84673	0.00747
	10	0.034335	0.034475	2.18	0	81.5633	0.06881
	25	0.080531	0.080001	-1.76	8.109375	83.94853	0.160532
	50	0.213962	0.259568	3.68	29.10547	66.93827	0.47353
	75	0.243669	0.245207	0.66	43.97813	83.23793	0.488876
	100	0.337442	0.343208	-0.04	54.07813	80.22724	0.68065
ChaCha20-CTR	1	0.001828	0.001753	-8.6	0	151.5683	0.003582
	10	0.019791	0.020464	4.16	0.010156	136.5882	0.040256
	25	0.050515	0.051422	2.78	6.125781	133.7235	0.101937
	50	0.105811	0.114057	0.82	27.03125	127.8647	0.219868
	75	0.181478	0.160811	0.96	40.5625	116.6462	0.342289
	100	0.207072	0.211235	-2.88	54.07813	130.5936	0.418307

Figure 5.7. Performance Evaluation of Symmetric Encryption Algorithms on Generated Datasets

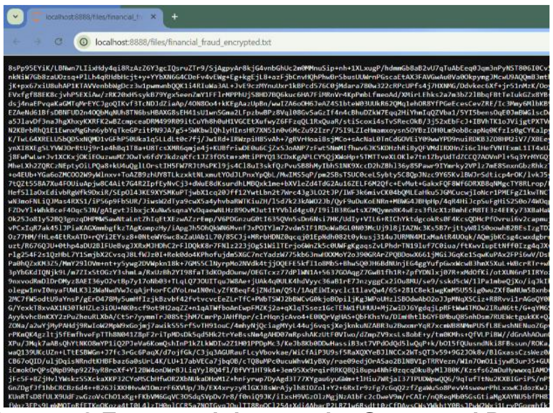
By observing Figure 5.7, it evaluates the performance of 5 symmetric algorithms: AES256-GCM, DES-CTR, RC6-CTR, Blowfish-CTR, ChaCha20-CTR, using datasets with the sizes of 1MB, 10MB, 25MB, 50MB, 75MB, 100MB based on the parameters of encryption time, decryption time, CPU Usage, Memory Usage, Throughput and Latency. Averaging across multiple dataset sizes ensures a balanced and fair comparison, it can capture the general performance of each algorithm and minimizes the impact of occasional fluctuations.

AES256-GCM shows faster encryption and decryption time across the entire dataset sizes, it just recorded 0.027971 seconds for encryption and 0.031839 seconds for decryption, even for a 100MB dataset.

Blowfish shows the least CPU Consumption among all the algorithms across the entire dataset sizes. Here, AES consumes high CPU usage, showing the highest spike at the dataset of 25 MB.

AES256 uses the least memory among all the algorithms across the entire dataset sizes, and it also shows the highest throughput and low latency.

Output of the Generated Dataset



localhost:8888/files/financial_fraud_encrypted.txt

[illegible]

Figure 5.8 Encrypted data on the Generated Dataset

Figure 5.8 Visualizes the encrypted data performed on the Generated Dataset. Here, the original plain text of the Excel file is converted into unreadable text using symmetric encryption algorithms. Encrypting original data into ciphertext will provide security to

financial or sensitive data from cryptographic, brute-force and modern attacks. The result shown in Figure 5.8 was generated for every dataset.

Performance Evaluation on Real-time Datasets

Evaluation of symmetric encryption algorithms AES256, DES, Blowfish, RC6, and ChaCha20, using dataset sizes of 143MB,470MB,759MB,1200MB. This evaluation considers the performance metrics such as encryption time, decryption time, CPU usage, memory usage, throughput and latency on Real-Time Datasets. The results are visualized in the graphs and the table.

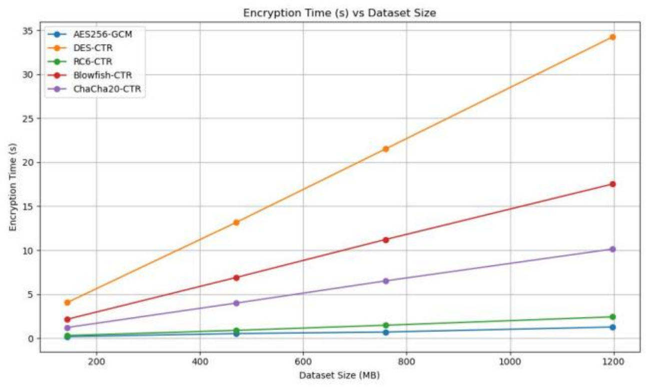


Figure 6.1 Encryption Time on Real-Time Datasets

Figure 6.1 visualizes the analysis of all symmetric encryption algorithms while performing encryption on the datasets. These Visualizations help to choose the best algorithm that takes less encryption time. In Figure 6.1, the X-axis represents the dataset sizes and the Y–axis represents the encryption time in seconds. AES256 scales efficiently as file size increases; for large files, it also encrypts the data very fast. DES and Blowfish take more time according to the increased file size. RC6 consumes less time, which closely matches AES-256. ChaCha20 scales linearly as file size increases. Overall, AES-256 shows the best results.

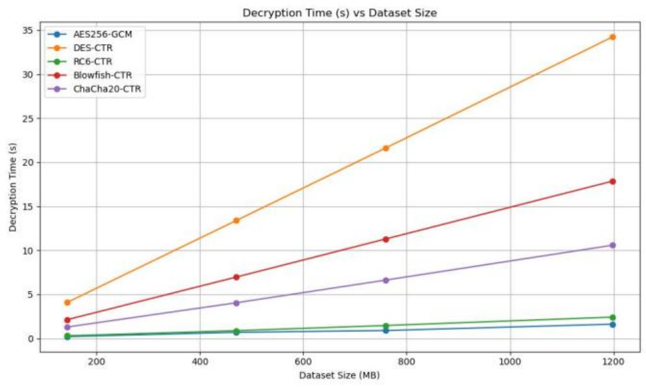


Figure 6.2 Decryption Time on Real-Time Datasets

Figure 6.2 visualizes the analysis of all symmetric encryption algorithms while performing decryption on the datasets. These Visualizations help to choose the best algorithm that takes less decryption time. In Figure 6.2, the X-axis represents the dataset sizes and the Y–axis represents the decryption time in seconds. AES256 scales optimised performance as file size increases; for large files, it also decrypts the data very fast. DES and Blowfish take more time according to the increased file size. RC6 consumes less decryption time, which closely matches AES-256. ChaCha20 shows moderate decryption performance. Overall, AES256 shows the best results.

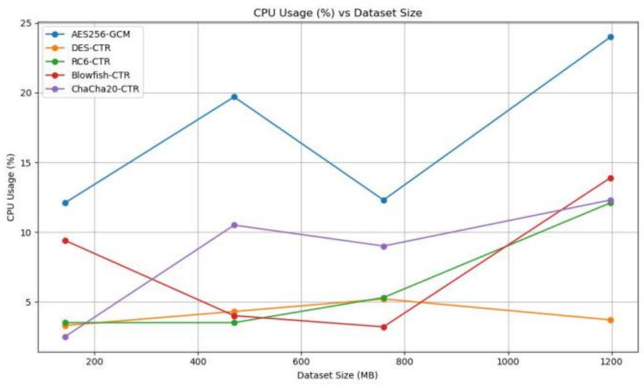


Figure 6.3 CPU Usage on Real-Time Datasets

Figure 6.3 illustrates the amount of processor power consumed by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that consumes less CPU Usage. In Figure 6.3, the X-axis represents the dataset sizes and the Y–axis represents the CPU usage in Percentage. AES256 consumes high CPU power even for small datasets. DES and RC6 consume low CPU power. Blowfish has moderate CPU consumption and ChaCha20 has very low CPU consumption compared to all algorithms. Overall, ChaCha20 shows the best results.

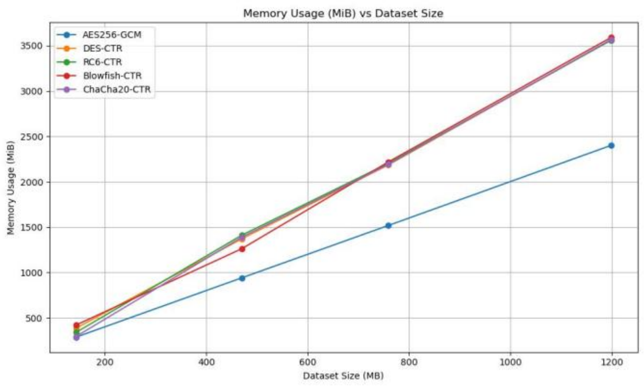


Figure 6.4 Memory Usage on Real-Time Datasets

Figure 6.4 illustrates the amount of memory consumed by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that consumes less memory. In Figure 6.4, the X-axis represents the dataset sizes and the Y–axis represents the Memory usage in MiB. AES256 consumes less Memory over all dataset sizes. RC6, DES, and ChaCha20 have moderate growth in memory usage, and the growth rate increases when the file size increases. Blowfish has a high growth rate in memory usage, it has bad scaling. Overall, AES-256 shows the best result.

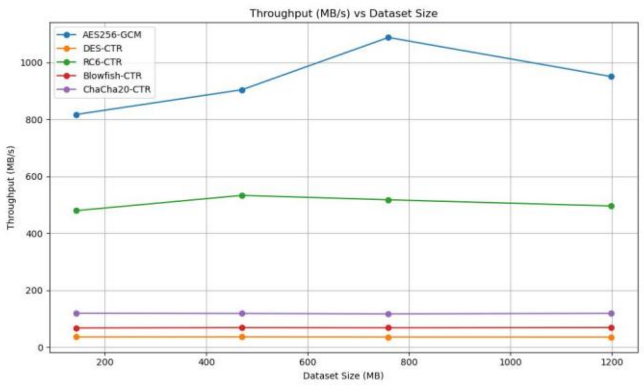


Figure 6.5 Throughput on Real-Time Datasets

Figure 6.5 illustrates the amount of data processed per second by all symmetric encryption algorithms during encryption and decryption of the datasets. These

Visualizations help to choose the best algorithm that generates higher throughput. In Figure 6.5, the X-axis represents the dataset sizes and the Y–axis represents the throughput in MB/s. AES256 has a high throughput rate, it maintains an excellent speed. RC6 and ChaCha20 have a moderate throughput with increased file size, but still maintain good speed. DES and Blowfish have less throughput with increased file size, and have a bad scaling. Overall, AES shows the best results.

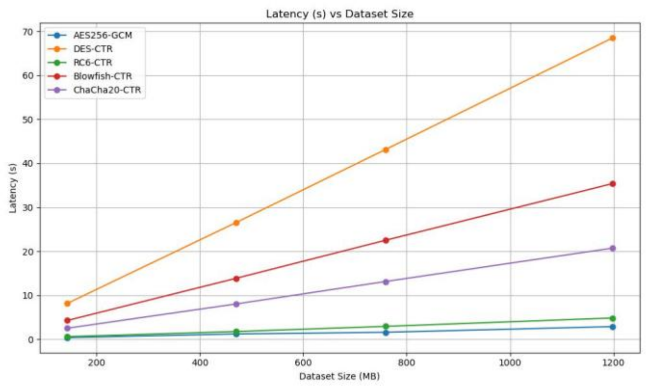
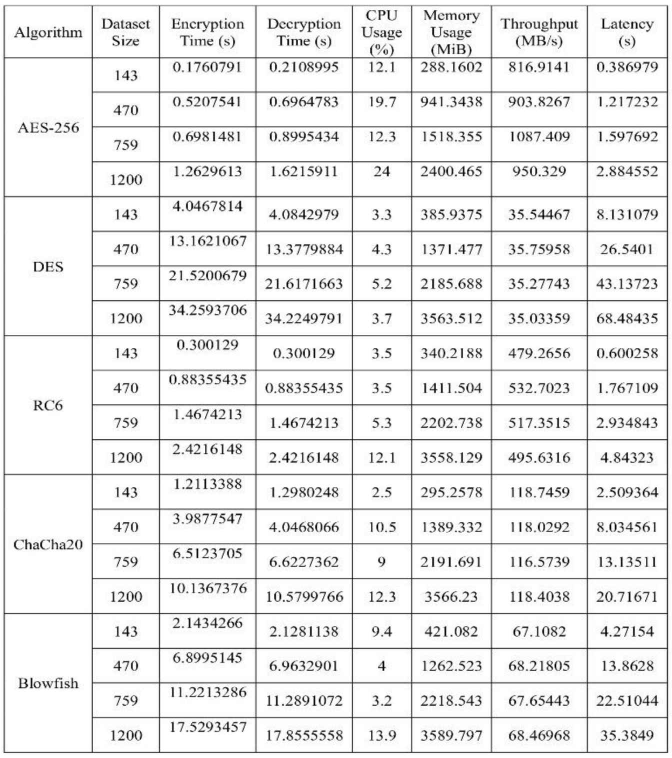


Figure 6.6 Latency on Real-Time Datasets

Figure 6.6 illustrates the amount of processor power consumed by all symmetric encryption algorithms during encryption and decryption of the datasets. These Visualizations help to choose the best algorithm that consumes less CPU Usage. In Figure 6.6, the X-axis represents the dataset sizes and the Y–axis represents the latency in seconds. AES256 has a very low latency rate and maintains consistent growth as the file size increases. RC6 has a low latency rate and closes matches AES-256. ChaCha20 has a moderate latency rate. Blowfish and DES are producing high-latency results. Overall, AES-256 shows the best result.



Algorithm	Dataset Size	Encryption Time (s)	Decryption Time (s)	CPU Usage (%)	Memory Usage (MiB)	Throughput (MB/s)	Latency (s)
AES-256	143	0.1760791	0.2108995	12.1	288.1602	816.9141	0.386979
	470	0.5207541	0.6964783	19.7	941.3438	903.8267	1.217232
	759	0.6981481	0.8995434	12.3	1518.355	1087.409	1.597692
	1200	1.2629613	1.6215911	24	2400.465	950.329	2.884552
DES	143	4.0467814	4.0842979	3.3	385.9375	35.54467	8.131079
	470	13.1621067	13.3779884	4.3	1371.477	35.75958	26.5401
	759	21.5200679	21.6171663	5.2	2185.688	35.27743	43.13723
	1200	34.2593706	34.2249791	3.7	3563.512	35.03359	68.48435
RC6	143	0.300129	0.300129	3.5	340.2188	479.2656	0.600258
	470	0.88355435	0.88355435	3.5	1411.504	532.7023	1.767109
	759	1.4674213	1.4674213	5.3	2202.738	517.3515	2.934843
	1200	2.4216148	2.4216148	12.1	3558.129	495.6316	4.84323
ChaCha20	143	1.2113388	1.2980248	2.5	295.2578	118.7459	2.509364
	470	3.9877547	4.0468066	10.5	1389.332	118.0292	8.034561
	759	6.5123705	6.6227362	9	2191.691	116.5739	13.13511
	1200	10.1367376	10.5799766	12.3	3566.23	118.4038	20.71671
Blowfish	143	2.1434266	2.1281138	9.4	421.082	67.1082	4.27154
	470	6.8995145	6.9632901	4	1262.523	68.21805	13.8628
	759	11.2213286	11.2891072	3.2	2218.543	67.65443	22.51044
	1200	17.5293457	17.8555558	13.9	3589.797	68.46968	35.3849

Figure 6.7 Performance Evaluation of Symmetric Encryption Algorithms on Real-Time Datasets

By observing Figure 6.7, it evaluates the performance of 5 symmetric algorithms: AES256-GCM, DES-CTR, RC6-CTR, Blowfish-CTR, ChaCha20-CTR, using datasets with the sizes of 143 MB, 470 MB, 759 MB, 1200 MB based on the parameters of encryption time, decryption time, CPU Usage, Memory Usage, Throughput and Latency.

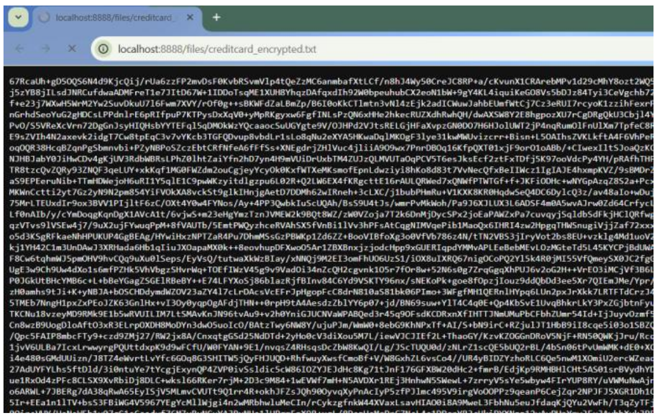
Averaging across multiple dataset sizes ensures a balanced and fair comparison, it can capture the general performance of each algorithm and minimises the impact of occasional fluctuations.

AES256-GCM shows faster encryption and decryption time across the entire dataset sizes, it just recorded 1.26 seconds for encryption and 1.62 seconds for decryption, even for a 1200MB dataset.

ChaCha20 shows the least CPU Consumption among all the algorithms across the entire dataset sizes. Here, AES consumes high CPU usage, showing the highest spike at the dataset of 1200 MB,

AES256 uses the least amount of memory among all the algorithms across the entire dataset sizes, and it also shows the highest throughput and low latency.

Output of the Real-Time Dataset


localhost:8888/files/creditcard_
localhost:8888/files/creditcard_encrypted.txt

```
67RcaUh+gD5OQS6N4d9KjcQij/rUa6zzFP2mvDsF0KvbRSvmVlp4tQeZzMC6anmbafXtLCf/n8hJ4Wy50CreJC8RP+a/cKvunX1CRArebMPv1d29cMhY8ozt2WQ5
j5zYB8jILsdJNRCufdwaADMFreT1e7JItD67W+1IDDoTsqME1XUH8YhqzDAfqxdIh92W0bpeuhubCX2eoN1bW+9gY4KL4iquiKeGO8Vs5bDJz84Tyi3CeVgchb72
f+e23j7WXwH5WrM2Yw2SuvDkuU7l6Fwm7XVY/rOf0g++sBKWFdZaLBmZp/B6I0oKkCTlmtn3vNl4zEjk2adICWuwJahbEUmfWtCj7Cz3eRUI7rcyoK1zzihFexrF
nGrhdSeoYuG2gHDCsLPPdnlrE6pRIfpuP7KTPysDxXqV0+yMpRKgyxw6FgfINLsPzQN6xHHe2hkecRUZXdhRwhQH/dwAXSW8Y2E8hgpozXU7rCgDRgQkU3Cbjl4Y
PvO/S5VReXcVrn72DgGnJsyHIQHsbYYTEFql5qDMOkWzYQcaaocSuUGYgte9V/OJHPd2VJtsRELGjHFaXvpzGN0DO7H6HJolUWT2jP4nqRumOlFnUlXm7TpfeC8R
E9sZVIh4N2axevk2idgT7Cw8tpEqC3v7vYKcb3TGFQDvup8vbdLr1sLo8qNu2eXYA5HKwaDqlMKOgF3lye31kwMWUvizcrr+Bisn+L5OAIhsZVKLkftA4F6VhPeR
oqOQR38HcqBZqnPgSbmnvbi+PZyNBPoSZczEbtCRfNfeA6fFfSs+XNEgdrjZHlVuc4jliiA9O9wx7PnrDBOq16KfpQXT01xjF9orO1oABb/+CIwexIltSJoaQzKC
NJHBJabY0JiHwCDv4gKjUV3RdbWBRsLPhZ0lhtZaiYfn2hD7yn4H9mVUiDrUxbTM4ZUJzQLMVUTaOqPCV5T6esJksEcf2ztFxTDfj5K97ooVdcPy4YH/pRAfhTHF
TR8tzcQvZQRy93ZNQF3qeLUY+xkKqf1MG0FWZdm2ouCgjeyYcyOk0KxfWTXeMKsmofEpnLdwziyi8hKo8d83t7VvNecQfxBeIIWcz1IgIAJE4hxmpKVZ/9sBMDrZ
aS9EPEeruNib+TTmHDWejoH6uRI1Y5qlE1C9pwWKzyitdlgzpu6L02R+Q2LW6EX4fKRgcttE16rAULQRWed7xQNWfPTWTGf+f+JKFiODHc+wNYGpAzqZ8S2a+Pcx
MKWnCctti2yt7Gz2yN9N2pm854YiFVOkXA8vckSt9glkIHnjgAetD7DDMh62wIRneh+3cLXC/j1bubPHmRu+V1KXK8KR0HqdwSeQ4DC6DylcQ3z/av48aIo+wDuj
75MrLTEUxdIr9ox3BVV1PIjltF6zC/OXt4Y0w4FYNos/Ay+4PP3QwbkIuScUQAh/BsS9U4tJs/wmrPvMkWoh/Pa9J6XJLUX3L6ADSF4m0A5wvAJrw0Zd64CrfycL
Lf0nAIb/y/cYmDoqgKqnDgX1AVcA1t/6vjwS+m23eHgYmzTznJVMEW2k9BQt8WZ/zW0VZoja7T2k6DnMjDycSPx2joEaPAWZxPa7cuvqyjSqldbSdFkjHClQRfwg
qzVTvs9lVSEw4j7/9uX2ujFYwuqPpM+8fVAUTb/5EmtPWQyzhceRVAhSX5fVnBi1lVv3hPFsAtCqgNIMVqePib1MaoQx6IHRI4zw2HpgqTHWSnugiVjjZaf72xxx
o5d3KSgRFkaeNhHPUKUP4GgBEAg/HYwiHxzNPTZaR4Pu7DhmMSsGzPBWKp1Zd6ZZ+BooVIBfoXg3o0VfVb786z4N/tTN2VBS3jIryVot2bs8EU+vzklg4Md1uoVZ
kj1YH42C1m3UnDAwJ3XRHada6Hb1qIiuJXOapaMX0k++8eovhupDFXwoO5Ar1ZBXBnxjzjodcHpp9xGUERIqpdYMMvAPLEeBebMEvLOzMGteTd5L45KYCPjBdUWA
F8Cw6tqhmWJ5pmOHV9hvCQq9uXu0lSeps/EyVsQ/tutwaXkWzBIay/xNNQj9M2EI3omFhU06UzS1/iOX8uIXRQ67nigOCoPQ2Yl5k4R0jMI55VfQmeySX0JC2fgC
UgE3w9Ch9Uw4dXo1s6mfPZHk5VhVbgzSHvrWq+TOEfIWzV45g9v9VadOi34nZcQH2cgvnk105r7fOr8w+52N6s0g7ZrqGgqXhPUJ6v2oG2H++VrEO3iMCjVf3B6L
P0JGkUtBHcYMB6c+L+bBeYGagZSGElRBeBY++E74LFYXoSj86blazRjfBInv84C6Yd9VSKTY96nx/sNEKoPk+goe8fOpzjIouz9ddQbDd3ee5Xr7QIEmJMe/Ypr/
zH0amhs9tJi+K+yNBJA+bOSCHDdymdWZOV23aZY4l7cLrDAcsVcEFrJpHgopFcC8drN810aS81bk06PImo+3WFgfMH1QERnlHYpq6LUn2pxJrXkk7LRTFTdCrzJ4
5TMEb7NngH1pxZxPEoJZK63GnlHx+vI3Oy0yqpOgAFdjTHN++0rpH9tA4AesdzZblYY6p07+jd/BN69suw+YlT4C4q0E+Qp4KbSvE1UvqBhkrLkY3PxZGjbtnFyu
TKCNu18vzeyMD9RMk9E1b5wRVUILIM7LtSMAvKnJN96tvAu9+v2h0YniGJUCNVaWPABQed3r45q9OFsdKCDRxnXfIHTTJNmUMuPbCFbhZUmr54Id+IjJuyvOzmf5
Cn8wzB9UogDloAftO3xR3ELrpOXDH8MoDYn3dwO5uoIcO/BAtzTwy6NW8Y/ujuPJm/WmW0+8ebG9KhNPxTf+AI/S+bN9irC+RZjulJT1HbB9iI8cqe5i03o1SBZQ
/Qpc5FAIP8mbcFTy9+czd9ZMj27/RW2jx8A/CnxqtgGSd25NdDTd+2yHo0cV3diXou5M7L/iewVJCJIEf2L+ThaoGY/KzvKZOGGnDRoV5NjF+RN50QWKjJru/Rcc
1jvV6ULBa7IcxLrwwyrgPQUttdxpK9d9wFCfU/W0FYAN+9E1/nvqsZ4RHsqsDcZbW8KwQI/Lg/JScTUQU0d/zNLrZ1scQE5bUQ2rBL/4bSn06tPvUmWMK+dE0+XC
i4e480sGMdUUizn/J8TZ4eWvrtLvYfc6GOq8G3SHITW5jQyFHJUQD+RhfwuyXwsfCmoBf+V/W8GxhZL6vsCo4//UR4yBIDZYzhoRLC6Qe5nwM1XOmiU2ercWZeac
27AdUYFYLhs5ftDld/3i0ntuYe7tYcgjExynQP4ZVP0ivSsldic5cW86IOZYJEJdHc8Kg71tJnF176GFXBW20dHc2+fmrB/EdjKp9RMHBHlCHt5AS01srBVydhYD
ue1RxOd4zPFc8CLSX9XvRbiDj8DLC+wksl66RKer7rjM+2D3c9M84+1wEVWf7mH+N5AVDXr1REj3HnhwN5SWewL+7zrryV5sYe5wbyw4FIrYUP8RY/uVWMuNwAjr
o6ARWL+7JBERg7dA38qRwA65EyISjV5MLmvCVUTt9Q1rr4R+okhJFZsJQh90OyvqXyPnAcIyP5zfPJlmc495V9irgVoOOPPz9qeanP6CejZqr2NPJFJ5XGR1Dh1C
5Ir+EEa1n1lTV+bsS3FBiWG4V5967YEgYcMllWjgi4n2wMRbhwluMeCIn/rCykzgfnkW44XVaxLsavHIAO0i8A9MweL3FbhNu5euJfdaqKjQYu2VwFh/T3qZyTFj
```

Figure 6.8 Encrypted Data on Real-Time Dataset

Figure 6.8 Visualizes the encrypted data performed on the Generated Dataset. Here, the original plain text of the Excel file is converted into unreadable text using symmetric encryption algorithms. Encrypting original data into ciphertext will provide security to financial or sensitive data from cryptographic, brute-force and modern attacks. The result shown in Figure 6.8 was generated for every dataset.

Comparing the Analysis Results based on Generated and Real-Time Datasets

In this evaluation, a comparison of various symmetric encryption algorithms was conducted based on two types of datasets

1) Generated Random financial datasets by using Python with sizes of 1MB, 25MB, 50MB, 75MB, 100MB.
2) Real-Time datasets with various sizes: 143MB, 470MB, 759MB, 1200 MB

These analyses were done based on the performance metrics: Encryption time, Decryption time, CPU Usage, Memory Usage, Throughput, and Latency. This analysis helps in choosing the best encryption algorithm for the first encryption layer that secures the financial data at rest and in transit.

Here, the comparison is performed on the result of the generated dataset and the result of the real-time dataset.

Table 4. Evaluating the performance results of generated and real-time datasets

Metric	***Generated dataset***	***Real-Time Dataset***
Encryption Time	Very Fast	Very Fast
Decryption Time	Very Fast	Very Fast
CPU Usage	Higher CPU consumption	Higher CPU consumption
Memory Usage	Efficient Usage of memory	Efficient usage of memory
Throughput	Higher	Higher
Latency	Low	Low

Table IV visualizes the performance of the AES-256 algorithm on two datasets. AES-256 maintains stable behaviour and efficiently processes data on both small and large datasets. It takes less time for encryption and decryption of data, even for large datasets, uses the resources efficiently, generates high throughput, and has low latency.

AES-256 is a secure, efficient and scalable encryption standard. It is an optimal choice for securing financial or sensitive data.

Security Strength

Table 5, Evaluating the Strength of Symmetric Encryption Algorithms

Algorithm	***Key Size***	***Block Size***	***Security Resistance***
AES	256 bits	128 bits	Resistant to known cryptanalysis techniques and brute-force attacks due to large key size.
DES	56 bits	64 bits	It is vulnerable to brute-force attacks due to a small key size.
Blowfish	32-448 bits	64 bits	Less resistant to modern attacks due to its small block size
RC6	128,192,256 bits	128 bits	Limited adoption, less cryptanalysis, and less resistance to modern attacks
ChaCha20	256 bits	64 bits	High resistance to cryptographic attacks, mostly adaptable to software, not for web-based applications.

Table V illustrates the strength of all the symmetric encryption algorithms. These visualizations help in choosing the symmetric encryption algorithms that have strong resistance against attacks. Table V shows that AES and ChaCha20 show good resistance against cryptographic attacks and brute-force attacks. Remaining DES, Blowfish and RC6 show less resistance against the attacks.

After evaluating the performance of a five symmetric algorithms AES256, DES, RC6, Blowfish and ChaCha20 across the Generated and Real-Time datasets based on the parameters of the performance metrics and security strength, it concludes that AES-256 is most suitable and secure for data encryption due to its large key size, i.e. 256 bits and block size, i.e., 128 bits as well as it provides strong security and high resistance against modern attacks. It hits a good balance on performance metrics such as fast encryption and decryption speed, minimal CPU and memory usage, very high throughput and predictable latency, which makes it an optimal choice for the first layer of encryption.

Remaining algorithms, i.e., RC6, closely compete with AES256, having fast performance and excellent consistency across all datasets. ChaCha20 also provided strong performance, it is mostly suitable for lightweight environments where fast cryptography is preferred. On the other hand, DES and Blowfish are not a good choice because of their limited adoption, resistance against modern attacks, weak security, as well as showing moderate and low performance, having higher encryption and decryption time, higher latency, unstable CPU and Memory consumption. AES256 is an Optimal Choice for the First layer of encryption.

10.Conclusion

This paper presented a performance evaluation of various symmetric encryption algorithms on both generate and real-time datasets based on the parameters of encryption time, decryption time, CPU Usage, Memory Consumption, Throughput, Latency and Security Strength.

Among all the evaluated algorithms, AES-256 performance was very stable and consistent on both datasets. AES-256 consumes extremely low encryption and decryption time, generates higher throughput, acceptable resource usage and consists low latency which makes to be suitable for real-time applications.

This comparative analysis confirms that AES-256 obtained balanced performance and security among all the datasets. Therefore, it is a recommended encryption standard for securing sensitive or financial data. So, AES-256 is an optimal choice for data encryption, using RSA, maintains a secure key exchange and the receiver's public key is authenticated with the PKI Infrastructure.

Acknowledgments

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Funding

No funding received.

REFERENCES

[1] *Olaiya, O. P., Adesoga, T. O., Adebayo, A. A., Sotomi, F. M., Adigun, O. A., & Ezeliora, P. M. "Encryption techniques for financial data security in fintech applications". International Journal of Science and Research Archive, 12(1),* ***(2024)*** *2942-9.*

[2] *Sarkar, B., Saha, A., Dutta, D., De Sarkar, G., & Karmakar, K." A Survey on the Advanced Encryption Standard (AES): A Pillar of Modern Cryptography".* ***(2024)***

[3] *Ezeonyi, N. U., Okonkwo, O. R., & Enweka, O. A. "Applications of Information Cryptography in Its Various Stages of Evolution, from Antiquity to the Modern Era". Journal of Medicine, Engineering, Environmental and Physical Sciences (JOMEEPS),* ***(2023)*** *1(1).*

[4] *Sood, R., & Kaur, H. "A literature review on rsa, des and aes encryption algorithms". Emerging Trends in Engineering and Management,* ***(2023)*** *57-63.*

[5] *Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. "Symmetric encryption algorithms: Review and evaluation study". International Journal of Communication Networks and Information Security, 12(2),* ***(2020)*** *256-272.*

[6] *Abd Zaid, M., & Hassan, S. "Survey on modern cryptography". Journal of Kufa for Mathematics and Computer, 7(1),* ***(2020)*** *1-8.*

[7] *Hesse, J. (2020, September). "Separating symmetric and asymmetric password-authenticated key exchange". In International Conference on Security and Cryptography for Networks (pp. 579-599). Cham: Springer International Publishing.* ***(2024).***

[8] *Al Busafi, S., & Kumar, B." Review and analysis of cryptography techniques". In 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), IEEE* ***(2020)*** *pp. 323-327.*

[9] *Patel, K." Performance analysis of AES, DES and Blowfish cryptographic algorithms on small and large data files". International Journal of Information Technology, 11(4),* ***(2019)*** *813-819.*

[10] *Abood, O. G., & Guirguis, S. K. "A survey on cryptography algorithms". International Journal of Scientific and Research Publications, 8(7),* ***(2018)*** *495-516.*

[11] *Thant, M., & Zaw, T. M. "Authentication Protocols and Authentication on the Base of PKI and ID-Based". In 2018 Wave Electronics and Its Application in Information and Telecommunication Systems (WECONF), IEEE* ***(2018)*** *pp. 1-8.*

[12] *Lozupone, V." Analyze encryption and public key infrastructure (PKI)". International Journal of Information Management, 38(1),* ***(2018)*** *42-44.*

[13] *Maqsood, F., Ahmed, M., Ali, M. M., & Shah, M. A. "Cryptography: a comparative analysis for modern techniques". International Journal of Advanced Computer Science and Applications,* ***(2017)*** *8(6).*

[14] *Abdullah, A. M. "Advanced encryption standard (AES) algorithm to encrypt and decrypt data". Cryptography and Network Security, 16(1),* ***(2017)*** *11.*

[15] *Habib, B., Cambou, B., Booher, D., & Philabaum, C. "Public key exchange scheme that is addressable (PKA)". In 2017 IEEE Conference on Communications and Network Security (CNS), IEEE* ***(2017)*** *(pp. 392-393.*

[16] *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), IEEE* ***(2017)*** *pp. 1-7.*

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

[18] *Kumar, P., & Rana, S. B. "Development of modified AES algorithm for data security". Optik, 127(4),* ***(2016)*** *2341-2345.*

[19] *Patil, P., Narayankar, P., Narayan, D. G., & Meena, S. M. "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish". Procedia Computer Science, 78,* ***(2016)*** *617-624.*

[20] *Panda, M. "Performance analysis of encryption algorithms for security". In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), IEEE* ***(2016)*** *pp. 278-284.*

[21] *Bisht, N., & Singh, S. "A comparative study of some symmetric and asymmetric key cryptography algorithms". International Journal of Innovative Research in Science, Engineering and Technology, 4(3),* ***(2015)*** *1028-1031.*

[22] *Ebrahim, M., Khan, S., & Khalid, U. B. "Symmetric algorithm survey: a comparative analysis". arXiv preprint arXiv:1405.0398.* ***(2014).***

[23] *Yin, X. C., Liu, Z. G., & Lee, H. J. "An efficient and secured data storage scheme in cloud computing using ECC-based PKI". In 16th international conference on advanced communication technology, IEEE* ***(2014)*** *pp. 523-527.*

[24] *Mitali, V. K., & Sharma, A. "A survey on various cryptography techniques". International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 3(4),* ***(2014)*** *307-312.*

[25] *Al-Hazaimeh, O. M. A." A new approach for complex encrypting and decrypting data". International Journal of Computer Networks & Communications, 5(2),* ***(2013)*** *95.*

[26] *Mandal, B. K., Bhattacharyya, D., & Bandyopadhyay, S. K. "Designing and performance analysis of a proposed symmetric cryptography algorithm". In 2013 International Conference on Communication Systems and Network Technologies, IEEE* ***(2013)*** *pp. 453-461.*

[27] *Singh, S., Maakar, S. K., & Kumar, S. "A performance analysis of DES and RSA cryptography". International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), 2(3),* ***(2013)*** *418-423.*

[28] *Marwaha, M., Bedi, R., Singh, A., & Singh, T. "Comparative analysis of cryptographic algorithms". Int J Adv Engg Tech/IV/III/July-Sept, 16,* ***(2013).*** *18*

[29] *Mohammad, O. K. J., Abbas, S., El-Horbaty, E. S. M., & Salem, A. B. M. "A comparative study between modern encryption algorithms based on cloud computing environment". In 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013),IEEE* ***(2013)*** *pp. 531-535.*

[30] *Gupta, S., & Sharma, J. "A hybrid encryption algorithm based on RSA and Diffie-Hellman". In 2012 IEEE International Conference on Computational Intelligence and Computing Research, IEEE.* ***(2012)*** *pp. 1-4.*

[31] *Pavithra, S. "Performance evaluation of symmetric algorithms". Journal of Global Research in Computer Science, 3(8),* ***(2012).*** *43-45.*

[32] *Singh, G., Kumar, A., & Sandha, K. S.. "A study of new trends in Blowfish algorithm". Int. J. Eng. Res. Appl, 1(2),* ***(2011)*** *321-326.*

[33] *Komninos, N." PKI systems. Network Security: Current Status and Future Directions",* ***(2007)*** *409-418.*

[34] *Zaidan, B. B., Zaidan, A. A., & Mwafak, H. "New Comprehensive Study to Assess Comparatively the QKD, XKMS, KDM in the PKI encryption algorithms". Int. J. Comput. Sci. Eng, 1(3),* ***(2009)*** *263-268.*

[35] *Cryptography retrieved from https://www.researchgate.net/figure/General-Model-of-Cryptographic-System_fig1_316507085.*

[36] *Symmetric Key Cryptography retrieved from https://www.encryptionconsulting.com/education-center/what-is-cryptography/#*

[37] *Asymmetric Key Cryptography retrieved from https://www.encryptionconsulting.com/education-center/what-is-cryptography/#*