

QATAR UNIVERSITY

COLLEGE OF ENGINEERING

PERFORMANCE EVALUATION OF MULTIMEDIA TRANSMISSION OVER ERROR-

PRONE WIRELESS CHANNEL USING BLOCK AND STREAM CIPHERS

BY

SHAIKHA ABDULAZIZ AL-MUHANADI

A Project Submitted to
the Faculty of the College of Engineering
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computing

January 2021

© 2021. Shaikha Abdulaziz Al-Muhanadi. All Rights Reserved.

COMMITTEE PAGE

The members of the Committee approve the Project of
Shaikha Abdulaziz Al-Muhanadi defended on 24/11/2020.

Amr Mohamed
Project Supervisor

ABSTRACT

AL-MUHANADI,SHAIKHA,ABDULAZIZ., Masters : January : 2021,
Masters of Science in Computing

Title: Performance Evaluation of Multimedia Transmission over Error-Prone Wireless Channel Using Block and Stream Ciphers.

Supervisor of Project: Amr Mohamed.

Network security is one of the crucial topics discussed nowadays, as the world is emerging towards new systems and technologies such as Artificial Intelligence (AI), blockchain, and Internet of Things (IoT). Cryptography plays an important role in managing and providing security services to the information stored and exchanged over the digital network. Cryptographic algorithms are integrated in many of our daily life systems and applications such as: smart cards, electronic devices, mobile applications, and many social media platforms. Therefore, it is important to study the features of the existing cryptographic algorithms to find trends between stream ciphers and block ciphers. Since block ciphers operate at a fixed block size, it is very difficult to apply them in applications that require transmission of large amount of data over error-prone channels. In addition, the avalanche property in block ciphers cause error propagation from a single bit error, resulting in significant corruption to the whole data block. Therefore, cipher block modes of operation are used with the symmetric block ciphers to generate larger stream of input and providing security at the bit level to protect large data from error propagation.

In this project, two simulations are conducted to evaluate block and stream ciphers over an error-prone wireless channel in terms of image error rate and time complexity. The first simulation compares the performance of the Rivest (RC4) stream cipher with the following block ciphers: Data Encryption Standard (DES), 3DES and

Advanced Encryption Standard (AES). The second simulation examines how the following modes of operation: Cipher Block Chaining (CBC), Cipher Feed-Back (CFB) and Counter (CTR) applied to the AES would enhance the performance of AES compared to RC4. The results show a trade-off in the performance of the algorithms in terms of speed, security, and resistant to channel errors. Stream ciphers are faster and more efficient at localizing errors at a bit level, yet block ciphers are more secure. However, using the modes of operation with AES, the AES-CTR cipher was able to eliminate error propagation more than RC4. In terms of speed, the AES-CTR processed the data with less time compared to AES, but it required more time than RC4.

DEDICATION

For myself

&

My beloved parents

ACKNOWLEDGMENTS

This work was supported by Qatar University, Grant Number IRCC-2020-003. The statements made herein are solely the responsibility of the authors.

First and foremost, I praise and thank Allah the Almighty for his guidance and blessings to carry out my project.

I would like to express my sincere gratitude and appreciation to my parents for their unconditional love, support, and for always encouraging me to pursue this degree.

I would like to thank my supervisor professor Amr Mohamed for his time, generous guidance, and support.

I am particularly thankful for the help and advice of Dr. Mohsen Guizani and Dr. Uvais Ahmed Qidwai, without whom the learning curve would have been very much steeper.

Finally, I would like to take the opportunity to thank my sisters and friends for always being there for me in my difficult times.

TABLE OF CONTENTS

DEDICATION	iv
ACKNOWLEDGMENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER 1: INTRODUCTION	1
1.1 Project Goal and Objectives	4
1.2 Outline of the Project	4
CHAPTER 2: BACKGROUND & LITERATURE SURVEY	5
2.1 Symmetric Key Cryptography	5
2.2 Rivest Cipher 4 (RC4).....	7
2.3 Data Encryption Standard (DES)	8
2.4 Triple DES.....	8
2.5 Advanced Encryption Standard (AES)	9
2.5.1 Mathematical Preliminaries	10
2.5.2 Algorithm Specifications	11
2.5.3 Encryption and Decryption Process	13
2.6 Cipher Block Modes of Operation	16
2.6.1 CBC Mode of Operation.....	17
2.6.2 CFB Mode of Operation	18
2.6.3 CTR Mode of Operation.....	20

2.7 Related Work.....	21
CHAPTER 3: METHODOLOGY	23
3.1 System Model.....	23
CHAPTER 4: PERFORMANCE EVALUATION.....	26
4.1 Simulation Setup	26
4.2 Performance Metrics	28
4.3 Simulation Results.....	29
4.3.1 Block vs. Stream Ciphers	29
4.3.2 RC4 vs. AES with & without modes of operation	32
4.4 Discussion	35
CHAPTER 5: CONCLUSIONS & FUTURE WORK	38
REFERENCES	40

LIST OF TABLES

Table 1. Symmetric Key Cryptographic Algorithms Information [17] [18] [19].....	7
Table 2. AES: Key Size, Block Size, and Number of Rounds [26].....	13
Table 3. Algorithm Settings	28

LIST OF FIGURES

Figure 1. Classification of cryptography [8][9]	2
Figure 2. Symmetric encryption and decryption model [11].....	5
Figure 3. 3DES (a) encryption and (b) decryption process [11].....	9
Figure 4. AES algorithm: input and output paramters [22].	10
Figure 5. AES: input block [18].....	11
Figure 6. AES: key block [21]	12
Figure 7. Key expansion to generate round keys [26]	13
Figure 8. AES encryption process and key expansion [27]	14
Figure 9. AES round 1 transformation operations [27]	15
Figure 10. CBC mode (a) encryption and (b) decryption process [11]	18
Figure 11. CFB mode (a) encryption and (b) decryption process [11].....	20
Figure 12. CBC mode (a) encryption and (b) decryption process [11]	21
Figure 13. First system model: block and stream ciphers over error-prone channel...23	
Figure 14. Mona Lisa image	24
Figure 15. Second system model: RC4 and AES with and without modes of operation over error-prone channel.....	25
Figure 16. First system model: input and output parameters.....	27
Figure 17. Second system model: input and output parameters	27
Figure 18. First Model: Image error rate vs. Error probabilities	30
Figure 19. First Model: Recovered Images vs. Error Probabilities.	31
Figure 20. Second Model: Time duration vs. Error probabilities	32
Figure 21. First Model: Image error rate vs. Error probabilities	33
Figure 22. Second Model: Recovered Images vs. Error Probabilities.....	34
Figure 23. Second Model: Time duration vs. Error probabilities	35

CHAPTER 1: INTRODUCTION

Cryptography has been around for thousands of years, ensuring the confidentiality of the messages sent and received in the presence of intruders. It is a simple mechanism that scrambles or alters the message to ensure that while the message is on its way to the receiver, it will appear as gibberish to anyone who will read it except the receiver who will know the secret to read the message. Throughout history, several classical approaches have been practiced ensuring secure communication such as substitution and transposition [1]. Ancient Egyptians at around 2000 B.C. are believed to first use cryptography technique known as ‘hieroglyph’ to decorate the tombs of the kings and rulers [2]. Dramatic changes to the nature of cryptography appeared throughout the centuries, passing by the ancient Chinese ‘ideographic nature of their language’, to the ancient Greeks ‘Caesar Cipher’ and up to World War II ‘Enigma’ cipher for military purposes [3]. In today’s world, modern cryptography is the heart of our worldwide digital communication network, playing an important role in ensuring the confidentiality and security of the data and multimedia including images, audio, and videos transmitted and stored over the internet [4]. It is implemented in nearly all our daily basis transactions and networking such as smart cards, cell phones, mobile applications, bank cards, e-mails, and in all online transactions.

Modern cryptography is a more complex and secure digital system that uses algorithms and keys to ensure that the data is not altered, prove the origin of the data, authenticate the data, and ensure the confidentiality of the message [6]. The techniques of cryptography are derived from mathematical concepts such as algorithms, rule-based calculations, and probability theory [7]. The algorithm used by the cryptosystem is called a ‘cipher’, it converts readable ‘plaintext’ to gibberish ‘ciphertext’ under the process called encryption. While transmitting the data through the insecure channels,

the algorithm protects the confidentiality and privacy of the data by creating a cryptographic key that controls digital signing and the verification of the message [8]. The inverse operation is called decryption, taking the scrambled text, and reforming it to be clear and readable message.

The two main types of modern cryptography are Symmetric Key Cryptography (SKC) and Asymmetric Key Cryptography (AKC) as shown in Figure 1 [8][9]. The difference between the two types is mainly related to the numbers of keys used, SKC or sometimes referred as secret-key algorithm uses the same key during the encryption and decryption process. On the other hand, AKC uses different pairs of keys for encryption and decryption of data [7]. Technically, SKC are implemented on the internet for data protection and associated applications because it is more efficient and takes less time to encrypt data compared with the public-key algorithms and it could process larger amount of data [8].

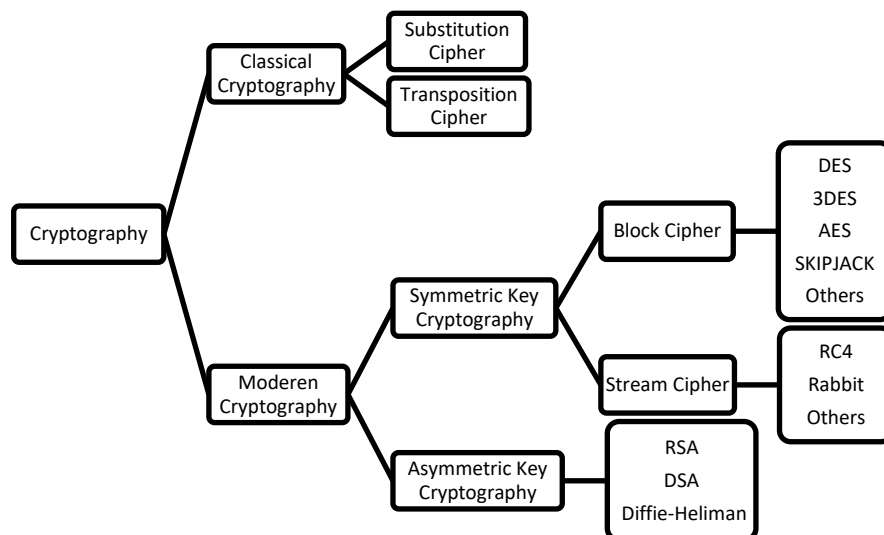


Figure 1. Classification of cryptography [8][9]

SKC algorithms are penetrating all the internet traffic in terms of preserving security and privacy of data over the internet. It plays a crucial role in protecting audio, images, and video streaming over the internet. SKC algorithms are classified as block ciphers and stream ciphers. Block ciphers process fixed size data at a time depending on the block size of each algorithm, while stream ciphers process smaller chunks of input data either a stream of a single bit or a byte at a time [10]. Therefore, security aspect and sensitivity of such technique to the channel impairments would be an important topic to study. If we do not know how sensitive these ciphers are to the channel impairments, we can use the wrong cipher according to a specific channel requirement. For example, if I have a non-ideal channel or an error channel (some possibility of error), the effect of using the wrong cipher on text is not the same as when is it applied to multimedia or online streaming. Also, multimedia applications, using diverse data such as images, videos, and graphics, interestingly do not perform well over channels with 100% reliability, due to latency caused by repeated retransmissions, imperfect flow control mechanisms. It is very difficult to apply block ciphers in applications that require the transmission of larger amount of data over error-prone channels. The avalanche property in the block ciphers cause error propagation from a single bit error, resulting in significant corruption to the whole data block. Accordingly, cipher block modes of operation are used with the symmetric block ciphers to generate larger stream of input, providing security at the bit level to protect large data from error propagation. Hence, the performance analysis of the encryption algorithms over error-prone channels will be helpful to determine the appropriate ciphers used for a specific application.

1.1 Project Goal and Objectives

The goal of this project is to study and analyze the performance of symmetric encryption algorithms, namely DES, 3DES, AES, and RC4 in transferring a multimedia source over an error-prone wireless channel. The performance of the algorithms and the sensitivity level in terms of image error rate and the time taken to recover the input source is analyzed. In addition, the same performance metrics are used to compare the performance of the stream cipher RC4 with the block cipher AES with and without the following block cipher modes of operations: Cipher Block Chaining (CBC), Cipher Feed-Back (CFB), and Counter (CTR).

The objectives of this project can be summarized as follows:

- 1) Develop a framework of an error-prone wireless channel for secure multimedia data transmission system using block ciphers and stream ciphers.
- 2) Conduct performance evaluation to compare common block ciphers and stream ciphers by studying the effect of channel impairments on the error propagation in multimedia.
- 3) Develop a framework using AES with the modes of operation and examine the effect in enhancing the security level and eliminating the error propagation in the multimedia data.

1.2 Outline of the Project

The report is organized as follows: chapter 2 provides a background and literature review related to symmetric encryption algorithms and block ciphers modes of operations. Chapter 3 discuss the methodology of the project and the implementation setup. Chapter 4 provides the simulation results and discussion. The conclusion and future work are discussed in Chapter 5.

CHAPTER 2: BACKGROUND & LITERATURE SURVEY

This introductory chapter describes the characteristics and the types of the symmetric cryptography ciphers and block cipher modes of operation. Followed by existing work conducted by others in the same field.

2.1 Symmetric Key Cryptography

Symmetric key cryptography is classified as either block ciphers or stream ciphers, which is related to how the ciphers operate on the plaintext to be encrypted. Generally, all symmetric encryption systems have five main components that make up the structure of the cryptosystem, these include: plaintext, encryption algorithm, secret key, ciphertext, and decryption algorithm as shown in Figure 2 [11].

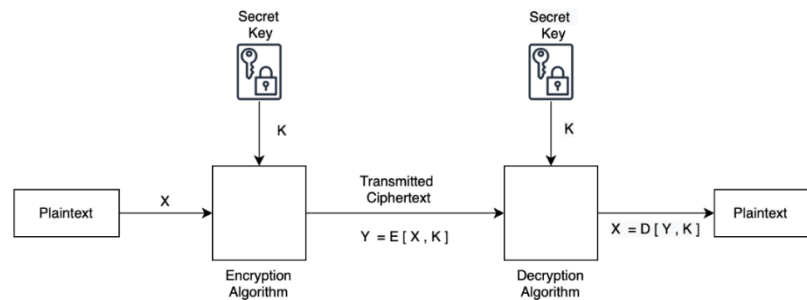


Figure 2. Symmetric encryption and decryption model [11]

The plaintext is the original message that is created and sent as the input to the encryption algorithm. The second input to the encryption algorithm is the secret key, it is a very crucial component that encrypt and decrypt the message. Without the secret key, the data cannot be retrieved. The size of the key is used to measure how secure the algorithm is, with a larger key size the system will be more resistant to brute force attack [11]. It is because the algorithms are publicly available, the key length plays important

role in defining the maximum potential strength of the algorithm. The encryption algorithm produces an unreadable format of the original message, a message that can only be read by who is intended to receive the message and have a copy of the key. It performs substitutions and permutations to scramble the data, and the key to ensure that the data is protected. The ciphertext is produced after scrambling the information in the encryption process, which is gibberish and cannot be read. The decryption algorithm performs the inverse operation used in the encryption process to recover the plaintext. The same key is used by both the sender and the receiver to obtain the original message from the receiver side [11].

Block ciphers operate at a fixed block size, if a block of b -bits is encrypted the output gives b -bit block. It is considered as a deterministic algorithm that encrypt data using a symmetric key. If the data being encrypted is not long enough to fill the block, padding is used to fill the extra space and ensure that the plaintext fits the blocks evenly. The drawback of deterministic algorithms is when the same input is encrypted several times the same output result is given, which makes the algorithm prone for attacks. Therefore, it requires additional security levels to make it more random and difficult to analyze [12].

On the other hand, stream ciphers operate on a stream of input and encrypts smaller chunks of data. A stream cipher is one that encrypts a stream of data, such as real-time video streaming or Telnet traffic [13]. The key size should be the same size as the input to avoid repetition and ensure that the encrypted message is unique [14]. The input data are processed one bit or a byte at a time, combining the input with a bit from the stream of pseudo random key to encrypt the data. The ciphered data is independent, hence there is a one-to-one relationship between the data input and encrypted data [15]. The underlying design of stream ciphers make the algorithm more

resistant to error propagation and in terms of processing data, it is very fast.

An important aspect to stream ciphers is that the initialization vector (IV) should never be reused, every time information is being encrypted, a new IV should be used otherwise the message could be easily accessed by intruders [16]. The information and properties of the symmetric key algorithms used to transfer the multimedia source over an error channel are shown in Table 1 [17] [18] [19].

Table 1. Symmetric Key Cryptographic Algorithms Information [17] [18] [19]

Algorithm	DES	3DES	AES	RC4
Publication Year	1977	1985	2001	1987
Network Structure	Feistel	Feistel	Substitution-Permutation	---
Key Size (Bits)	56	112 / 168	128 / 192 / 256	8 - 2048
Rounds	16	48	10/12/14	256
Cipher Type	Block	Block	Block	Stream
Block Size (Bits)	128	64	64	Variable length

2.2 Rivest Cipher 4 (RC4)

The RC4 is a stream cipher designed by Ron Rivest in 1987 with random permutations [11]. RC4 gained a widespread adoption because the algorithm speed is high and its low implementation complexity [31]. The algorithm supports variable key lengths from 8 to 2048 bits to perform byte-oriented operations to the stream of input [31]. In the encryption and decryption process, the input is processed one byte or larger number of bytes at a time. The simple underlying structure of the design eased the implementation of the algorithm on both hardware and software.

2.3 Data Encryption Standard (DES)

The Data Encryption Standard (DES) is one of the earliest and widely used encryption standards designed by IBM in the 1970s [17]. DES was adopted by the National Institute of Standards and Technology (NIST) as an official Federal Information Processing Standards (FIPS 46) for encrypting and securing governmental data based on a 16-round Feistel network [17]. DES is a linear symmetric block cipher algorithm that operates on a 64-bits input data using a 56-bit key. Practically, the key length is 64-bits, but 56-bits are used for the key and the other 8-bits are used for parity checking or in other words error checking [12]. This makes the practical key length of the DES only 56-bits, and hence the maximum possible number of keys to perform brute force attack are 2^{56} . Back in the 1970s it was considered long enough, but as technology advanced and computers got faster, 56-bits keys quickly proved to be too small [11]. The 56-bits key are used to generate a total of 16 subkeys used for each of the 16 rounds of the encryption process, mapping the input 64-bits block into a ciphered 64-bits block. The same key is used in reverse order for the decryption process [29].

2.4 Triple DES

The Triple Data Encryption Standard (3DES) is as an extension of the existing DES algorithm, has been introduced mainly to overcome the brute-force attack encountered in DES [17]. In 1999, the 3DES algorithm was publicly announced in FIPS PUB 46-3 to be part of the DES standard [17]. The algorithm uses three keys and three DES execution algorithms for both encryption and decryption shown in Figure 2, the DES encryption algorithm is denoted with E and the decryption algorithm with D [11].

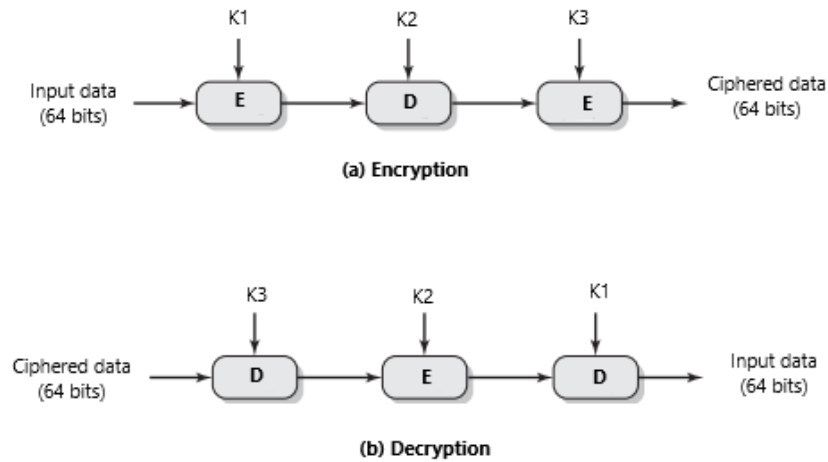


Figure 3. 3DES (a) encryption and (b) decryption process [11]

The decryption process as shown in Figure 2, reverses the key order as well as the algorithm used to obtain the original message. The standard specifies two scenarios when choosing the number of keys, either three distinct keys with a key size of 168-bits or two keys could be used by setting $K1 = K3$ to provide a 112 bits key size [11].

2.5 Advanced Encryption Standard (AES)

The Advanced Encryption Standard (AES) is a standard approved by the National Institute of Standard Technology (NIST) in 2001, based on the Rijndael algorithm developed by Dr. Daemen and Dr. Rijmen [20]. The final standard is published under the Federal Information Processing Standards Publications (FIPS PUB 197) [18]. Rijndael is a symmetric block cipher that processes data blocks and keys of size 128,192, and 256 bits for encryption and decryption [21]. However, the approved algorithm AES is an iterative structure that processes input data blocks of size 128-bits only with variable key lengths of 128, 192, and 256 bits shown in

Figure 4 [22]. The number of rounds required for each key size are 10, 12, and 14 rounds respectively [20].

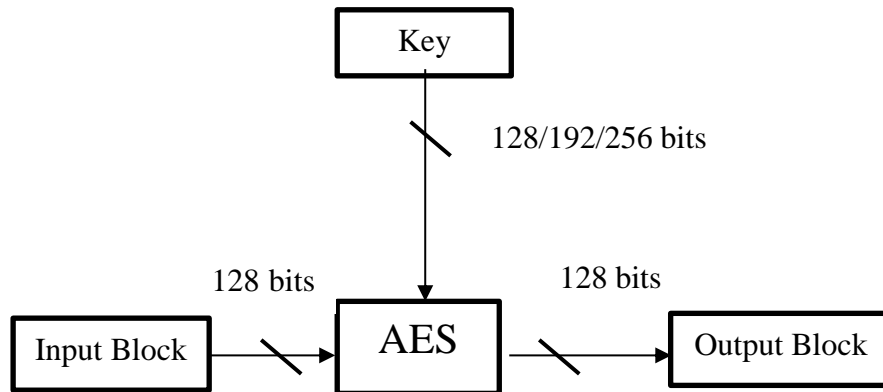


Figure 4. AES algorithm: input and output parameters [22].

2.5.1 Mathematical Preliminaries

In cryptography, fields with finite number of elements are required for all operations within the layers of the algorithm. Galois Field or finite field, named after Evariste Galois, is a field with finite number of elements denoted as $GF(p^n)$, where p is a prime number and n is any positive integer [23]. Galois theory is divided into two main fields which play a larger role in AES, $GF(2)$ and $GF(2^8)$. Considering the smallest prime field $GF(2)$ with only two elements in the field $\{0,1\}$, the modulo 2 addition and multiplication of coefficients of equal power [24]. The $GF(2)$ field is the building block of the arithmetic used in AES, the mathematical structure underlying the design of the AES is interpreted using Galois field $GF(2^8)$. There are 256 elements in the field, where each element is represented by a bit denoted by 0 or 1. The four main operations performed in the field include: addition, subtraction, multiplication, and the inverse operation which is basically division [23]. The operations are byte-oriented,

each polynomial in the field is stored as a byte vector containing elements b_i of 8 bits as shown in equation (2.1).

$$B = (b_7 + b_6 + b_5 + b_4 + b_3 + b_2 + b_1 + b_0) \quad (2.1)$$

and the maximum degree of the polynomial is given by $n - 1$. Transforming polynomials into bytes granted the manipulation and storage of digital data efficiently [24].

2.5.2 Algorithm Specifications

The AES algorithm is based on a network that uses substitutions and permutations transformations to shuffle the bytes in the input block columns creating diffusion and confusion [25]. In each round, certain byte-oriented round transformations are performed to the input block. The operations include a substitution function and three permutations functions as follows: Substitute Byte, Shift Rows, Mix Columns, and Add Round Key [11]. To process data into the algorithm and apply the operations, the algorithm arranges the 128 bits input data as a four-by-four byte-oriented block shown in Figure 5 [21]. The bytes are numbered to show how the bits are arranged in the block, where each byte consists of 8 bits.

b_0	b_4	b_8	b_{12}
b_1	b_5	b_9	b_{13}
b_2	b_6	b_{10}	b_{14}
b_3	b_7	b_{11}	b_{15}

Figure 5. AES: input block [18]

Similarly, the key is represented by a four-by-four block shown in Figure 6 [21].

k_0	k_4	k_8	k_{12}
k_1	k_5	k_9	k_{13}
k_2	k_6	k_{10}	k_{14}
k_3	k_7	k_{11}	k_{15}

Figure 6. AES: key block [21]

The AES algorithm expands the key to produce a key schedule that will be responsible in generating round keys. The operation used for the key expansion rotation, substitution using the S-Box, and a round constants [18]. It is important to note that the size of the cipher key and the round keys are of the same size. The key block of the AES-128 is illustrated in Figure 7 to show how the key is expanded to form the linear array key schedule representing each column of bytes as a word [26].

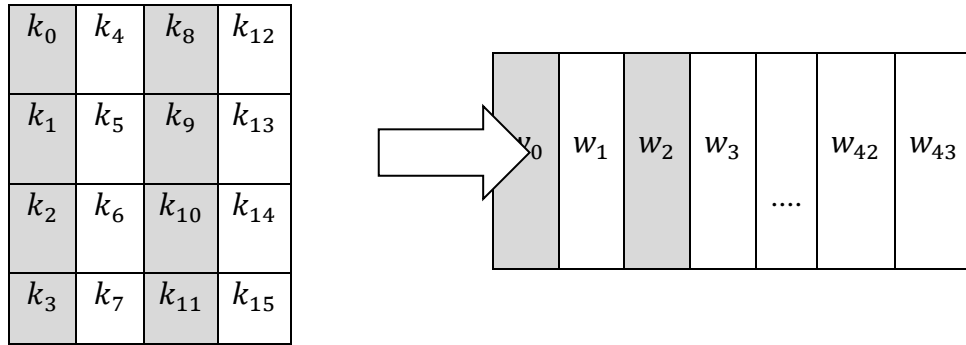


Figure 7. Key expansion to generate round keys [26]

To calculate the total number of words in the key, the following equation is used $N_b (N_r + 1)$. Where N_b is the key size (words) and N_r is the number of rounds, the values of each key size are highlighted in Table 2. For the AES using 128 bits key, the number of rounds is given by $4 (10 + 1) = 44$ words produced, each word consists of 32 bits with a total of 1408 bits [26].

Table 2. AES: Key Size, Block Size, and Number of Rounds [26]

Key size (bits)	Key size (words)	Block size (bits)	Block size (words)	No. of rounds
128	4	128	4	10
192	6	128	4	12
256	8	128	4	14

2.5.3 Encryption and Decryption Process

The input block and the key will be processed to the three main phases of the encryption and decryption processes, including the initial transformation, the main

transformation with $N_r - 1$ round, and the final transformations as shown in Figure 8 [27].

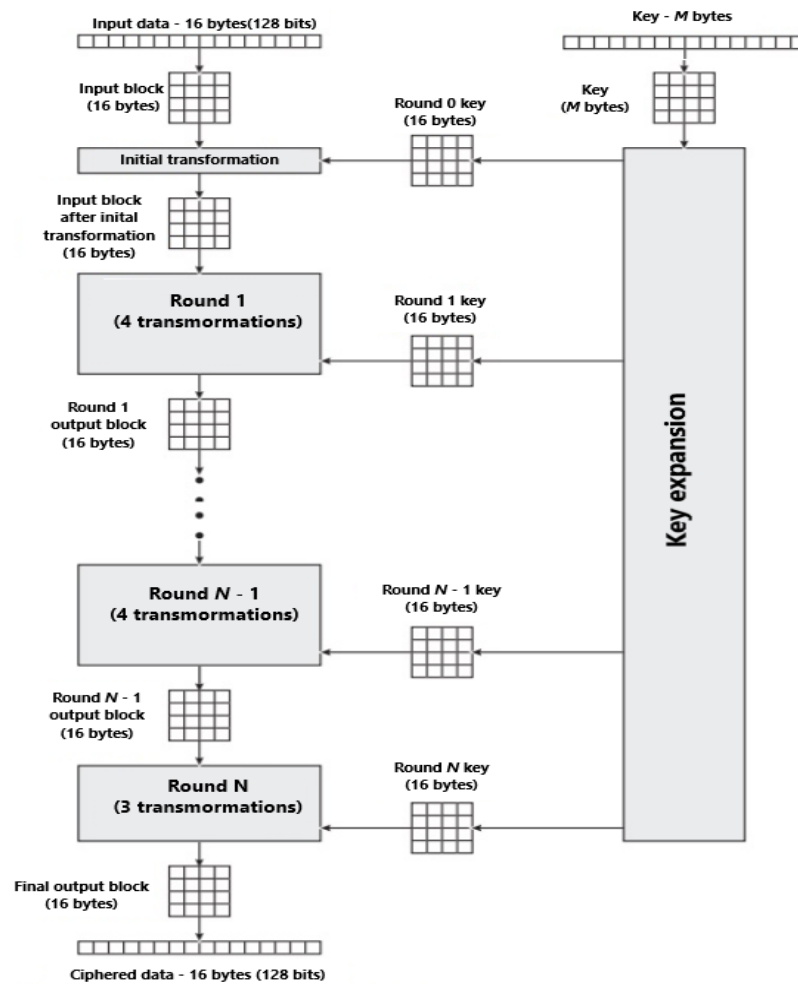


Figure 8. AES encryption process and key expansion [27]

In the initial round, the Add Round Key transformation is the only transformation used to take each column from the input block and perform an XOR addition operation with the words from the key expansion. The output block of the operation executed in the initial round is passed to the round 1, four different

transformations are performed to permutate and substitute the bytes shown in Figure 9 [27].

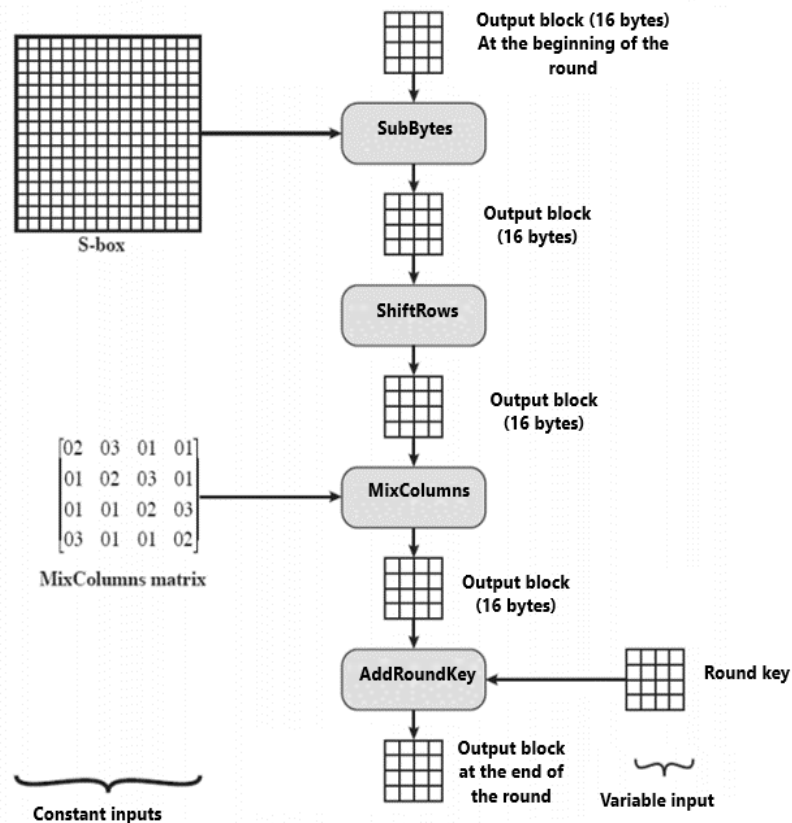


Figure 9. AES round 1 transformation operations [27]

The Substitute Bytes uses the invertible and non-linear S-Box to substitute each byte of the State with a corresponding value from the S-Box. The new output block is then passed to Shift Rows transformation to shift the rows to the left, the first row is left not shifted, the second, third, and the fourth rows gets shifted to the left by 1, 2, and 3 bytes respectively. After the shifting process, the new output block is processed to the

Mix Columns transformation. In the Mix Column transformation, each column of the output block is treated separately as a polynomial function over $GF(2^8)$ and multiplied by the pre-defined polynomial modulo $x^4 + 1$ matrix [24]. The output 16 bytes block will be fed again to the Add Round key transformation with the round key, it is a very crucial transformation that provides the security. Without the Add Round Key transformation, the other three transformations provide no security since the key is not involved in the operations [28]. The process will be repeated for $N - 1$ rounds. In the final round, the transformations used are the Substitute Byte, Shift Rows, and Mix Columns to give an output that is reversible [22]. The decryption algorithm processes the inverse transformation operations in the encryption algorithm to obtain the original message.

2.6 Cipher Block Modes of Operation

According to NIST (Special Publication 800-38A), the Electronic Codebook (ECB) mode, the Cipher Block Chaining (CBC) mode, the Cipher Feedback (CFB) mode, the Output Feedback (OFB) mode, and the Counter (CTR) mode are the five recommended modes of operation that can be used with the symmetric block ciphers approved by the Federal Information Processing Standards (FIPS) to process larger amount of data required by many applications [29]. The level of security provided by the modes is fully depended on the secrecy of the key and the underlying symmetric algorithm used [29]. The modes of operation discussed, implemented with underlying block ciphers, and analyzed in this report are the confidentiality modes CBC, CFB, and CTR. The following modes support a mechanism to implement a stream cipher from the symmetric block cipher.

2.6.1 CBC Mode of Operation

The Cipher Block Chaining (CBC) mode is a widely used to encrypt and decrypt a large stream of input by chaining the input block with the previous ciphered block. Since the first block has not previous ciphered block, a randomly generated initialization vector (IV) of the same size as the input block is used to add randomness and make the first output block unique. The modes divide the input stream into equal number of blocks, depending on the underlying symmetric key algorithm used. The total number of bits in the input stream must be a positive multiple of the symmetric algorithm block size chosen, otherwise padding is used to fill the block [30].

The CBC mode encryption and decryption processes are illustrated in Figure 10 (a) and 10 (b) respectively [11]. In the first block, the input to the encryption algorithm is an initialization vector (IV) of the same size as the input block XORed with the input block P_1 and a key K [11]. The IV is randomly generated to add randomness and make the output block unique. The output of the encryption algorithm is the cipher block C_1 that gets fed as the input to the next block instead of the IV which is used only in the first block [30]. The process is repeated until the stream of input is fully encrypted, the chaining property adds a feedback mechanism in the system [30].

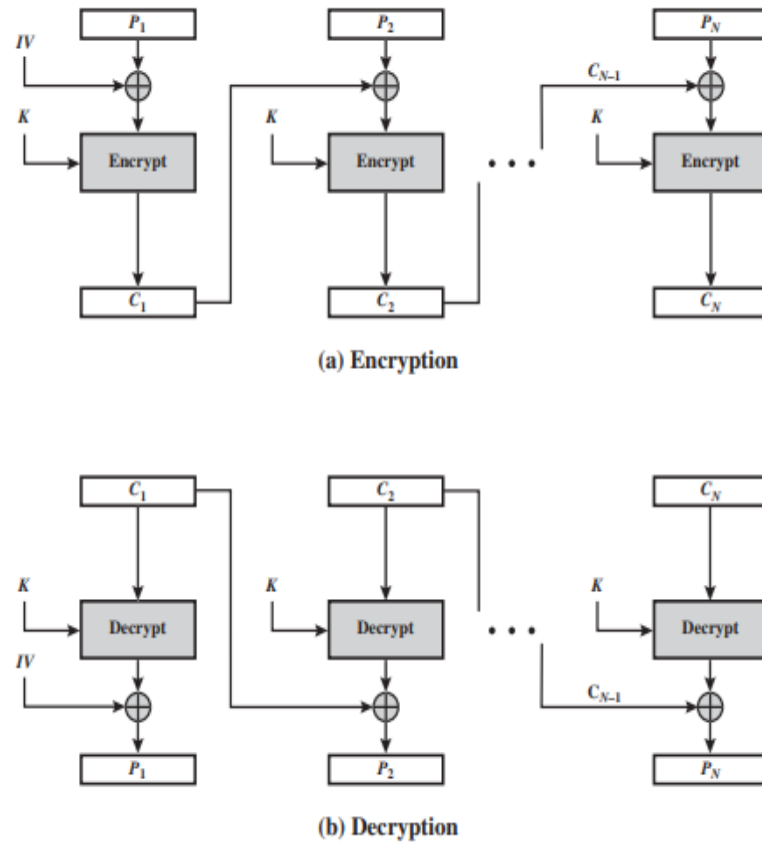


Figure 10. CBC mode (a) encryption and (b) decryption process [11]

In CBC decryption process, the inverse function is applied in parallel since the same key and IV used in the encryption process are used and the cipher block are already available [11]. It is important to note that the IV should be unpredictable and known for all parties communicating and sharing the information [29].

2.6.2 CFB Mode of Operation

The Cipher Feedback (CFB) mode supports a feedback mechanism that operates in smaller input segments denoted as s -bit parameter such that $1 \leq s \leq b$, where b is the block size of the underlying symmetric key algorithm. The most common values of the s -bit parameter used are 1-bit, 8-bits, 64-bits, and 128-bits [29]. The CFB mode uses an IV to add randomness and uniqueness to the output. The CFB mode encryption

and decryption process are illustrated in Figure 11 (a) and 11 (b) respectively [11].

In the CFB encryption process, the input to the encryption algorithm is the key K and the IV . The input stream is not restricted by the block size of the underlying symmetric key algorithm, but rather divided in s -bit blocks. A shift register is applied to the output b -bit block to obtain the leftmost significant s -bits, which will be XORed with the input s -bit segment P_1 , and the rightmost $b - s$ -bits gets discarded. The output s -bits segment C_1 is then fed to the rightmost s -bits of the IV shift register [29]. The process is repeated until the entire stream of s -bit input segments are encrypted to obtain the ciphered s -bit segments.

The CFB decryption process is very much like the encryption process, the only difference is that the ciphered segment C_1 is XORed with the leftmost significant s -bits of the output b -bit block of the encryption algorithm. During the decryption process, the mode uses the encryption algorithm instead of the decryption algorithm as shown in Figure 11 [11].

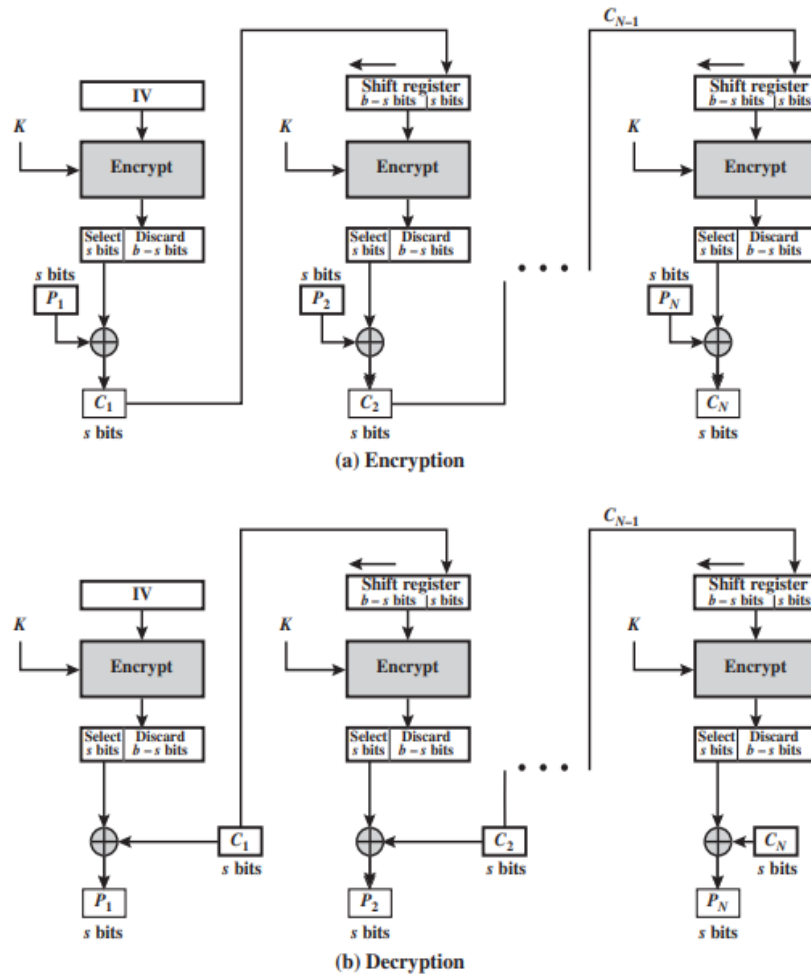


Figure 11. CFB mode (a) encryption and (b) decryption process [11]

2.6.3 CTR Mode of Operation

The Counter (CTR) mode is widely used in many applications such as the asynchronous transfer mode in the ATM and IPsec [30]. A unique counter equal to the input block size is required for each input block. If the same counter is used twice, the confidentiality of the blocks will be compromised [33]. The modes divide the input stream into equal number of blocks, depending on the underlying symmetric key algorithm used. To generate a sequence of unique counters for all the input blocks, an incrementing function is used. In addition, the first counter is carefully chosen to ensure that the sequence of counters will not happen equal counters [33]. The CTR mode

encryption and decryption process are illustrated in Figure 12 (a) and 12 (b) respectively [11].

The CTR encryption process is simply taken the counter and the key K as the input to the encryption algorithm. The output is XORed with the input block P_1 to produce the cipher block C_1 . The chaining property is not implemented in the CTR mode, therefore parallel encryption operations can be performed. The CTR decryption process uses the same sequence of counters to retrieve the original input stream.

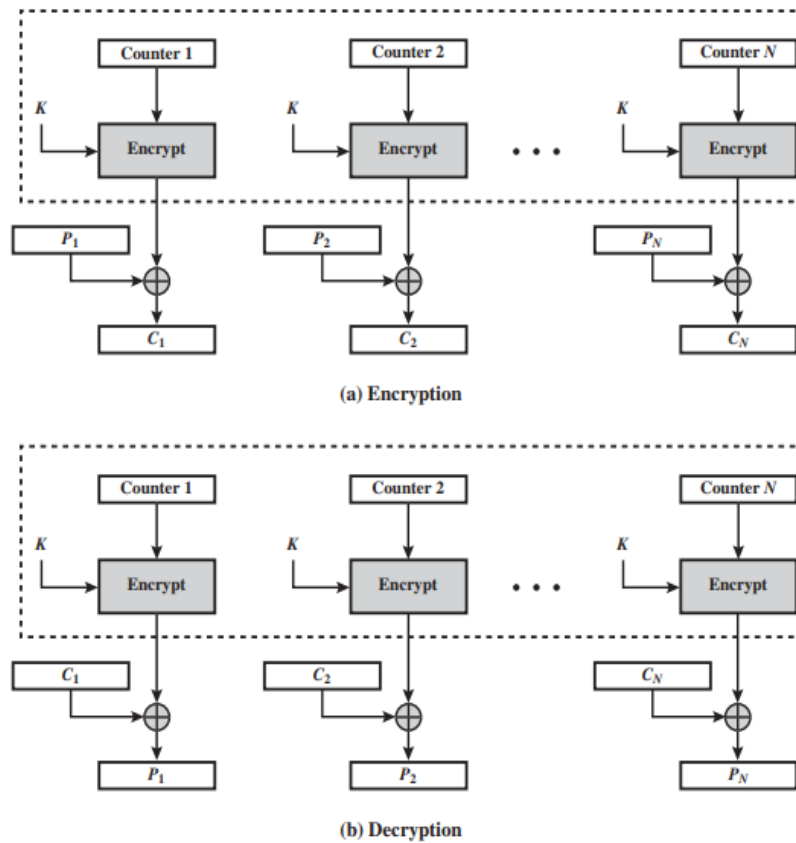


Figure 12. CBC mode (a) encryption and (b) decryption process [11]

2.7 Related Work

Extensive research in the field of cryptography is conducted by researchers on

both hardware and software to implement the block and stream ciphers in various devices and applications. The author in [31] used the modes of operation ECB, CBC, and CFB to compare the symmetric key algorithms AES with RC4 on hardware and software. The performance metrics used to evaluate the performance are as follows: throughput, CPU process time, encryption time, decryption time, and memory utilization. In addition, the author different key size with the AES algorithm, 128, 192, and 256 bits, respectively. The results show that RC4 performed better in in terms of speed on both hardware and software, and memory utilization compared to AES with the modes of operation. The increase in the size of the key resulted in more time required for both encryption and decryption process.

In addition, the author [32] performed a detailed analysis of the modes of operation: ECB, CBC, CFB, OFB, and CTR on AES. The evaluation was based on encryption time, decryption time, and throughput. The results show that the difference in the performance was unnoticeable with smaller file size, yet, with larger file size the difference was clear. ECB required less time for the encryption and decryption process compared to the other modes. Different symmetric key cryptographic algorithms are discussed in the following published work [33][34][35][36][37].

A study conducted by the author in [38] analyzed and studied the security-throughput tradeoff of AES with and without the modes of operations ECB and CBC with link adaptive encryption schemes over a secure channel. Results show that the AES-CBC is more reliable for a channel with no errors. Similarly, the author in [39] designed and analyzed the AES-CBC algorithm in terms of error propagation and used the correction hamming code as the detection and correction model.

CHAPTER 3: METHODOLOGY

This chapter describes the approach taken in this project to evaluate the performance of the block and stream ciphers in real-world scenario.

3.1 System Model

In the proposed methodology, the symmetric key algorithms are used to transmit a multimedia source through an insecure channel as shown in Figure 13. This is used to demonstrate thousands of confidential data being transmitted through the internet. The insecure wireless channel as shown in the figure below, demonstrates the communication channels over the internet. Therefore, it is very important to secure the content being transmitted with the appropriate cipher. The reason behind not choosing secure channels to demonstrate the transfer of the multimedia is mainly because the latency caused by packet queuing. Applications that require real-time video streaming or diverse multimedia sharing require the content to be delivered fast and with less errors, therefore, insecure channels are used.

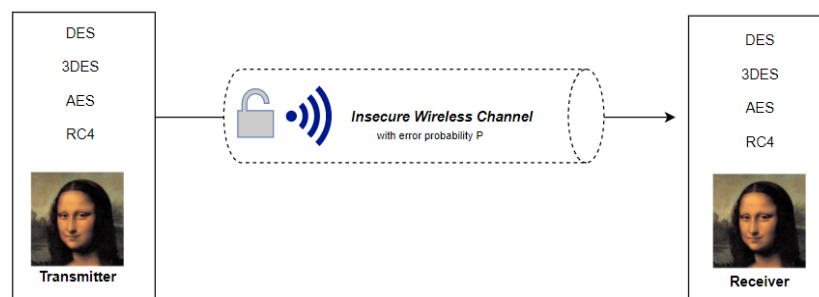


Figure 13. First system model: block and stream ciphers over error-prone channel.

Each of the following ciphers: DES, 3DES, AES, and RC4 are integrated in the transmitter, with certain specifications in terms of key size and block size discussed in

previous chapters. The insecure wireless channels are set to error probabilities 0.1%, 1%, and 10%. The purpose of this method is to examine the strength of the ciphers with different error probabilities in transferring the multimedia source from the transmitter to the receiver and find a tradeoff between the properties of the ciphers to make good decision when choosing the appropriate cipher for a certain application. Each time the transmitter initializes the proper cipher to be used, a new randomly generated key is created to ensure no-repetitive keys are used. The multimedia source used to be encrypted and transferred through the error channel is an image of Mona Lisa shown in Figure 14. The original image size is 300x300 RGB pixels, which gets reshaped into a stream of 270000 bits.



Figure 14. Mona Lisa image

To process the stream of bits using the algorithms, padding is used to ensure that the total number of bits in the input stream is integer multiple of the block size of the underlying symmetric key algorithm. After that step, the input stream gets divided into blocks equivalent to the block size of each symmetric key algorithm.

The ciphers DES and 3DES are simply not designed for modern hardware due to their complex design, when compared to AES they are less secure because of the inherent weakness of the small key size of the DES. As a result, in the second

implementation the same methodology is used to compare the performance of RC4 with the block cipher AES with and without the modes of operation CBC, CFB, and CTR as shown in Figure 15.



Figure 15. Second system model: RC4 and AES with and without modes of operation over error-prone channel

CHAPTER 4: PERFORMANCE EVALUATION

This chapter describes the simulation setup of the two proposed models with error probabilities, the performance metrics used to evaluate the block and stream ciphers, the simulation results obtained from both models, and finally a discussion that evaluates and compare the performance of the block and stream ciphers.

4.1 Simulation Setup

The experimental setup is divided into two main simulations conducted using MATLAB 9.9. In the first experimental setup, the stream cipher RC4 and the following block ciphers DES, 3DES, and AES are implemented to compare and analyze the performance of the ciphers in transmitting an image from the transmitter to the receiver through an error channel. In the second implementation, the same setup is used to compare the performance of RC4 with the block cipher AES with and without modes of operation: CBC, CFB, and CTR. In both experimental setups, the error channel is adjusted with error probabilities 0.1%, 1%, and 10%.

The wireless error channel is setup to generate a random noise N of size equivalent to the input using the error probabilities. The noise is then added to the input to simulate an error according to the adjusted probabilities. To demonstrate the real-world insecure channels, an error channel is built to measure the performance of the algorithms in resisting and handling the noise or errors in the channel while transmitting the multimedia data.

We have implemented the two systems models using native Matlab functions. The functions will run all the ciphers in consecutive order starting from the first in the list as shown in both system models and generate two graphs that compare the performance of the ciphers in term of image error rate and time duration. Figures 16 and 17, represent a detailed structure of both models showing the input and output

parameters.

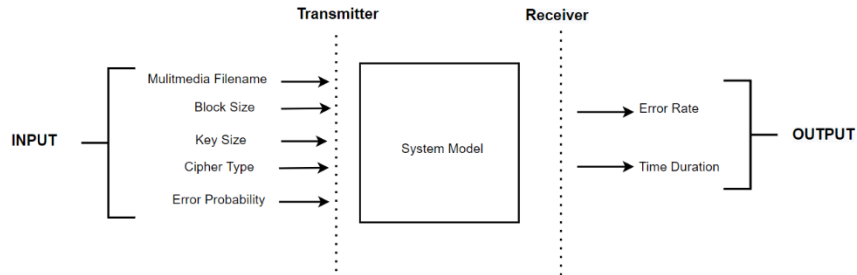


Figure 16. First system model: input and output parameters

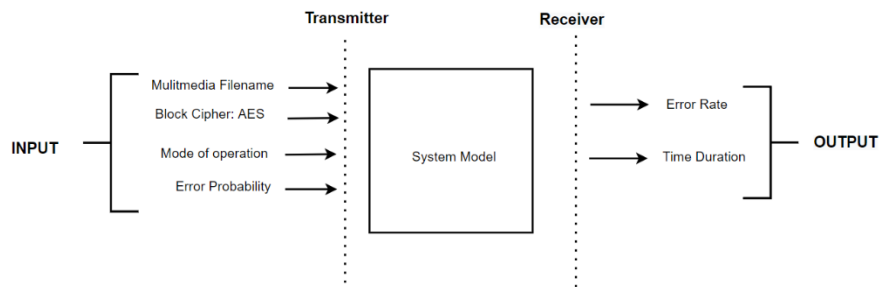


Figure 17. Second system model: input and output parameters

The symmetric ciphers are pre-defined in MATLAB individually, and tested before designing the models. The encryption and decryption codes for the block ciphers are implemented with the main function that will run the code and the function that will generate the sub keys according to each design specification. On the other hand, the design specification of the stream cipher RC4 is less complex. A main function is used for both encryption and decryption, taking the input from the pseudorandom generator (PRGA), which takes as the input stream of keys initially generated. In Table 3, the key size and block size of each cipher is shown. For the AES algorithm in both models, a

key size of 128-bits is used. For the DES and 3DES algorithm a key size of 56 bits is used, 3DES uses three 56-bit keys.

Table 3. Algorithm Settings

Algorithm	Key Size (Bits)	Block Size (Bits)
AES	128	128
DES	56	64
3DES	168	64
RC4	128	64

For the block cipher modes of operation requirements, the CBC and CFB modes require a nonce *IV* that is equal to the input block to ensure the uniqueness of the output. It is only used once for the first block, the second block will take the input of the previous block. On the other hand, the CTR require a counter of the same size as the input that can only be used once. The counter is a value that can only be used once since the blocks are independently processed. Therefore, to ensure that the value of the counter is not repeated, an incrementing function is used on the first counter to generate the other counters. For the CFB mode, the *s*-bit parameter is the value used to divide the input block into a smaller segment. The *s*-bit parameter chosen is 8-bits, the 128 input bits will be divided into 16 blocks of 8 bits each.

4.2 Performance Metrics

The parameters used to compare and analyze the performance of the symmetric key algorithms with and without the modes of operations to transfer the multimedia data over error-prone channel are as follows: 1) image error rate (IER) and 2) time

duration (TD). The IER given by equation (3.4.1), measures the number of pixels with errors from the image retrieved at the receiver side with respect to the total number of pixels in the image. Such metric focuses in examining the strength of the ciphers during the propagation of error in the multimedia data transferred. The time duration is calculated as the total time taken for each cipher to transfer the image from the transmitter to the receiver, indicating the complexity of each cipher.

$$\text{IER} = \frac{\text{Number of pixels in error}}{\text{Total number of pixels in the image}} \quad (3.4.1)$$

4.3 Simulation Results

The simulation results for the first and second model is explained in detail in sections 4.3.1 and 4.3.2, respectively. For each model, there is a table that represent all the images retrieved from all the ciphers at the receiver side with error probabilities 0.1%, 1%, and 10%. In addition, two graphs that illustrate the image error rate and the time duration with respect to the error probabilities for all the ciphers to easily compare the ciphers.

4.3.1 Block vs. Stream Ciphers

In terms of image error rate as shown in Figure 18 and Table 4, the recovered images from the block ciphers show more image error rate compared to the stream cipher RC4 as the error probability increased. For the images retrieved from the RC4, as the error probability increase the image is still clear with minor errors that are very hard to distinguish as shown in Table 4. The block ciphers DES and 3DES showed less error rate when compared with AES as the error probability increased.

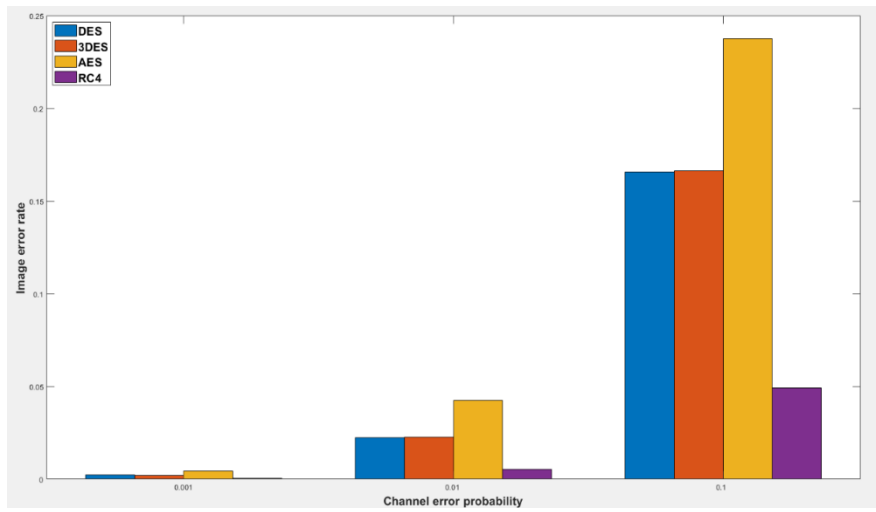


Figure 18. First Model: Image error rate vs. Error probabilities



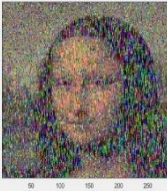

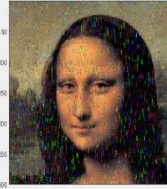



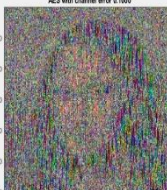



Error Probability		0.1%	1%	10%
Block Cipher	DES	DES with channel error 0.0010 	DES with channel error 0.0100 	DES with channel error 0.1000 
	3DES	3DES with channel error 0.0010 	3DES with channel error 0.0100 	3DES with channel error 0.1000 
	AES	AES with channel error 0.0010 	AES with channel error 0.0100 	AES with channel error 0.1000 
Stream Cipher	RC4	RC4 with channel error 0.0010 	RC4 with channel error 0.0100 	RC4 with channel error 0.1000 

Figure 19. First Model: Recovered Images vs. Error Probabilities.

The time duration of all the ciphers is illustrated in Figure 19. The RC4 stream cipher was the fastest to retrieve the original image, and from the block ciphers, AES was the fastest then DES and 3DES, respectively. The error probability had no impact

on the time taken by each cipher to retrieve the original image, the time complexity was consistent as the error probability increased.

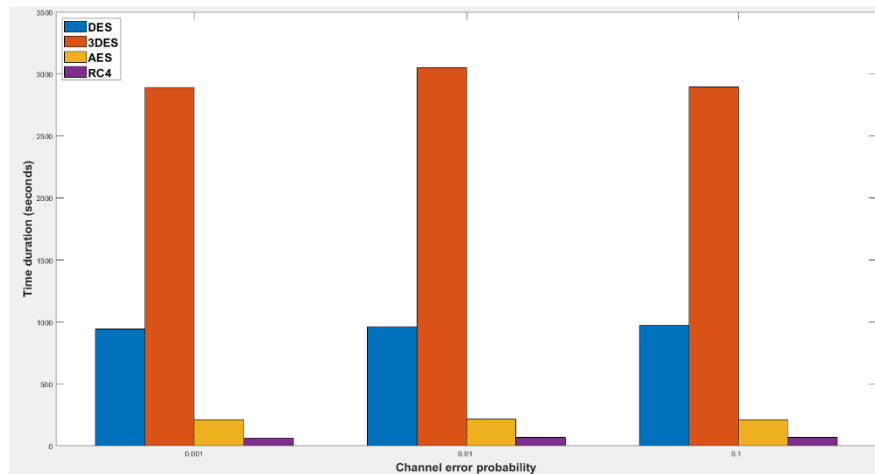


Figure 20. Second Model: Time duration vs. Error probabilities

4.3.2 RC4 vs. AES with & without modes of operation

The simulations obtained from the second model show significant results in terms of image error rate, as shown in Figure 20 and Table 5. The most compatible mode of operation compared to the stream cipher RC4 in terms of image error rate, is the CTR mode. The images recovered from the AES-CTR cipher had less errors when compared with the stream cipher RC4 as the error probability increased. The AES-CTR cipher was able to minimize the effect of the noise in the channel, converting AES to an efficient stream cipher. On the other hand, the performance of AES-CBC and AES-CFB was slightly identical to the performance of the AES algorithm. In fact, Figure 20 shows that the AES-CBC and AES-CFB modes had somewhat more image error rate compared to AES as the error probability increased.

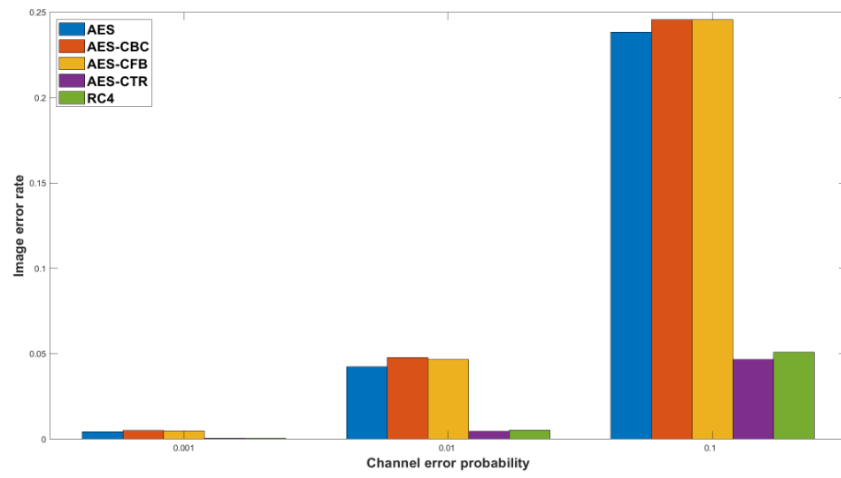


Figure 21. First Model: Image error rate vs. Error probabilities



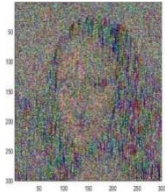












Error Probability		0.1%	1%	10%
Block Cipher with and without Modes of operations	AES			
	AES-CBC			
	AES-CFB			
	AES-CTR			
Stream Cipher	RC4			

Figure 22. Second Model: Recovered Images vs. Error Probabilities

The time taken for the algorithms to restore the original image at the receiver side is shown in Figure 21. The AES-CTR mode was the fastest when compared with AES and the other modes implemented with AES. However, the AES-CTR required slightly more time when compared to the stream cipher RC4. The error channel had no significant impact on the algorithms in terms of speed, as shown in Figure 21. As the error probability increased from 0.1% to 1%, the required time for retrieving the

image decreased slightly for all the ciphers except RC4. However, as the error probability increased from 1% to 10%, the time required to retrieve the image slightly increased for AES-CFB, decreased for AES and AES-CBC, and for AES-CTR and RC4 the change is unnoticeable.

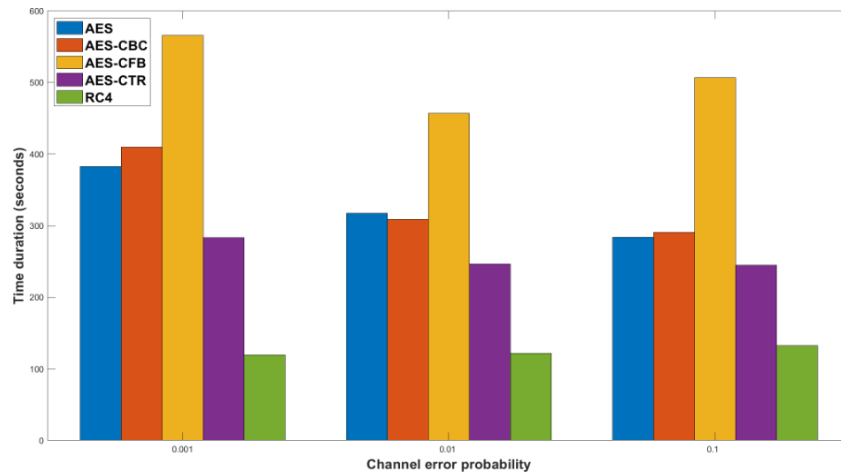


Figure 23. Second Model: Time duration vs. Error probabilities

4.4 Discussion

In many applications, choosing the appropriate cryptographic technique that can process the input stream through the symmetric cipher multiple times with minimal impact to the system and not compromising the network is mandatory. Therefore, this project aims to analyze the performance of the block and stream ciphers using the following evaluation criteria: strength of the cipher to resist security attacks, time complexity, and the ability to eliminate error propagation in the channel.

Block ciphers are more robust to security threats compared to stream ciphers, it is due to the complex structure of the cipher and the avalanche effect that provide additional security. Therefore, block cipher modes of operation are used to convert the secure block ciphers into stream ciphers. From the evaluated block ciphers, AES is more secure than DES and 3DES. The key size indicates how secure the cipher is, the

longer the key, the more time is required to break the key using brute force attack. AES with 128-bits key compared to a key of 56-bits used by DES and 3DES. Brute force attack is simply guessing the key with 2^k possibility, where k is the size of the key. However, in terms of image error rate as shown in the results obtained from the first model, block ciphers are weak. As the error probability increased, AES was the weakest amongst the block cipher in resisting errors in the channel. This is mainly because of the AES 128-bits block size compared to DES and 3DES with 56-bits.

Block ciphers introduce challenges in term handling errors due to the avalanche effect, 1-bit error could cause noise at the receiver side resulting in the corruption of the whole image. In addition, the structure underlying block ciphers are more complex compared to the stream cipher RC4. If an error occurred in the stream cipher, it is localized to a bit level. In terms of time complexity, stream ciphers took less time to process data compared to block ciphers. This is mainly due the simple underlying structure of RC4. From the block ciphers, AES performed better than DES and 3DES. AES took less time because the permutation-substitution network operates at a bit level, while DES and 3DES operate at a byte level through the Feistel network. Among the block ciphers, 3DES took the longest to retrieve the image since the algorithm repeats DES algorithm three times and the process is time consuming as shown in Figure 19.

Although AES was the weakest amongst the block ciphers to eliminate error from propagating, it is more secure than DES and 3DES. In the second model, AES was used as the underlying algorithm for demonstrating the effect of the modes of operation CBC, CFB, and CTR in shifting AES to a stream cipher. The performance of AES with and without modes of operation is compared with the stream cipher RC4, results show a trade-off in terms of security, time complexity, and resistant to errors. Results show that the AES-CTR cipher tackled the above issues with block ciphers in

terms of localizing the error at a bit level. The image error rate plotted in Figure 20, clearly shows that there are less error pixels retrieved from the original image compared to RC4. The error rate of the AES-CBC and AES-CFB was very high because of the chaining property found in both modes. Worth noting here that the effect of such chaining on the security of the algorithm should be better and calls for a separate study to prove the algorithm robustness and resistance to security attacks.

In terms of speed, the AES-CTR algorithm compared to the other modes used with AES, improved the speed of the algorithm as shown in Figure 21. Yet, it is considered slightly slower than the RC4. The design of the CFB mode, and the use of shift register and the s -bit parameter resulted in a more complex structure that requires more time to process the data through several operations.

When comparing AES-CTR and RC4 with respect of security robustness, AES-CTR is more secure because of the underlying AES algorithm. It is important to note that the modes of operation do not provide the security to the system, they fully depend on the underlying symmetric algorithm used. The AES-CTR algorithm compared to the other modes used with AES is very simple to implement since the encryption algorithm is used for both encryption and decryption process. However, compared with RC4 it is more complex.

CHAPTER 5: CONCLUSIONS & FUTURE WORK

In conclusion, all symmetric key cryptographic algorithms in one way or another enhance the security of any network system. The appropriate cipher can be used depending on the requirement of the application. The evaluation criteria of the algorithms conducted from the resulted obtained include a tradeoff between speed, security, and resistance to channel errors. The results from the simulation of the first model clearly showed that the RC4 stream cipher performed better than the block ciphers DES, 3DES, and AES in terms of speed and localizing errors at a bit level. This is mainly because of the simple structure of the RC4 algorithm, processing smaller size data will be much easier and faster. On the other hand, one of the properties that exists in block ciphers and not stream ciphers is the avalanche effect. The effect enhances error propagation due to the underlying structure of block ciphers. In terms of security, however, block ciphers are known to be more secure compared to the stream cipher RC4.

To fulfil the requirements of many applications that need a secure algorithm to handle large amount of data being transmitted over an insecure wireless channel. The cipher block mode of operations with block ciphers are used to transmit a large stream of input, instead of the fixed size blocks. Since AES is more secure when compared with the other two block ciphers DES and 3DES, it is used in the second model as the underlying symmetric algorithm with the modes of operation: CBC, CFB, and CTR. The simulation results of the second model show that the performance of the AES-CTR algorithm is better than RC4 in terms of managing the errors and preventing them from propagating. In addition, the AES-CTR enhanced the speed of the AES algorithm. Yet, the RC4 is still faster. Therefore, choosing the appropriate algorithm is fully dependent on the requirement of the application.

In future work, the performance of other multimedia sources such as videos and online streaming applications will be conducted. For the CFB mode, different values of the s -bit parameter will be used to study the effect of the segment on the speed of the algorithm and the ability to mitigate error propagation.

REFERENCES

- [1] Kester, Q. A. (2013). A Hybrid Cryptosystem based on Vigenere cipher and Columnar Transposition cipher. arXiv preprint arXiv:1307.7786
- [2] S. Singh. The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography. Anchor Books, 2000.
- [3] G. Zorpette, "Breaking the enemy's code: British intelligence deciphered Germany's top-secret military communications with colossus, an early vacuum-tube computer," in IEEE Spectrum, vol. 24, no. 9, pp. 47-51, Sept. 1987, doi: 10.1109/MSPEC.1987.6448935.
- [4] Paar C, Pelzl J. Key establishment. In: Understanding Cryptography. Berlin Heidelberg: Springer, 2010
- [5] Behrouz A. Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, 2nd Edition, Tata McGraw Hill, 2012.
- [6] NIST, 2006. SP 800-100: National Institute of Standards and Technology Special Publication 800-100, Information Security Handbook: A Guide for Managers. Recommendations of the National Institute of Standards and Technology, [on-line], October 2006. Available from: <http://csrc.nist.gov/publications/nistpubs/800-100/SP800-100-Mar07-2007.pdf> [Accessed November, 12 2011].
- [7] N. Smart Cryptography: An Introduction. (Third Edition), McGraw-Hill, 2004
- [8] E. Barker, "NIST Special Publication 800-175B Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms," August 2016.
- [9] Khan, M., & Shah, T. (2014). A literature review on image encryption techniques. 3D Research, 5(4), 29.
- [10] Schneier, Bruce; "Applied Cryptography", John Wiley & Sons, Inc 1996.
- [11] William Stallings. Network security essentials: Applications and Standards

Fourth edition. Prentice Hall, USA, 2011. – P.417.

[12] Data encryption and decryption by using triple DES and performance analysis of crypto system, Karthik .S ,Muruganandam .A, ISSN (Online): 2347-3878 Volume 2 Issue 11, November 2014.

[13] Xiao Y, Guizani M. Optimal stream-based cipher feedback mode in error channel. Proceedings of the IEEE Global Telecommunications Conference (Globecom '05), November 2005, 1660–1664.

[14] Xiao Y, Chen H, Du X, Guizani M. Stream-based Cipher feedback mode in wireless error channels. IEEE Transactions on Wireless Communications 2009; 8(2):622–626.

[15] X Liang, Y Xiao, S Ozdemir, AV Vasilakos, H Deng, Cipher feedback mode under go-back-N and selective-reject protocols in error channels. Secur.Commun. Netw. 6, 942–954 (2013).

[16] Monika Agrawal, Pradeep Mishra, ‘A Comparative Survey on Symmetric Key Encryption Techniques’, International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 4, No. 05, pp.877, May 2012.

[17] FIPS Publication 46-3, “Data Encryption Standard (DES),” U.S. DoC/NIST, Oct. 25, 1999.

[18] National Institute of Standards and Technology: Specification for the Advanced Encryption Standard (AES). <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>, (2001).

[19] Ranjeet Masram, Vivek Shahare, Jibi Abraham, Rajni Moona,” Analysis and Comparison of Symmetric Key Cryptographic Algorithms based on various file features”, International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.4, July 2014, DOI : 10.5121/ijnsa.2014.6404.

- [20] J. Daemen, V. Rijmen: AES proposal: Rijndael. <http://www.esat.kuleuven.ac.be/~rijmen/rijndael/rijndaeldocV2.zip>, (1999).
- [21] V. Rijmen: The block cipher Rijndael. <http://www.esat.kuleuven.ac.be/~rijmen/rijndael/>, (2001).
- [22] E. Barker, “NIST Special Publication 800-175B Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms,” August 2016.
- [23] The Mathworks: Galois Field Computations. <http://www.mathworks.com/access/helpdesk/help/toolbox/comm/tutor3.shtml>, Communications Toolbox, (2001).
- [24] J. Daemen, L. R. Knudsen, and V. Rijmen: The Galois Field $GF(2^8)$. <http://www.ddj.com/documents/s=936/ddj9710e/9710es1.htm>, Dr. Dobb’s Journal, (October 1997).
- [25] Harris Nover, Algebraic Cryptanalysis of AES, 1-6
- [26] Buchholz, P. D.-I. (2001, December 19). Matlab Implementation of the Advanced Encryption Standard. Retrieved June 23, 2014, from Prof. Dr.-Ing. Jörg J. Buchholz: <http://buchholz.hs-bremen.de/aes/AES.pdf>
- [27] Rihan, Shaza & Salih, Ahmed & Eldin, Saife & Osman, Faten. (2015). A Performance Comparison of Encryption Algorithms AES and DES.
- [28] V .Kumara Swamy, Dr Prabhu G Benakop, “Performance Analysis of Secure Integrated Circuits using Blowfish Algorithm”, Global Journal of Computer Science and Technology Network, Web & Security, Volume 13, Issue 17, Version 1.0,Page no.10-15, December 2013, Global Journals Inc (USA), Online ISSN. 0975-4172, Print ISSN.0975-4350.
- [29] FIPS Publication 800-38A, “Recommendation for Block Cipher Modes of

Operation: Methods and Techniques," U.S. DoC/NIST, 2001.

[30] D. Blazhevski, A. Bozhinovski, B. Stojchevska, and V. Pachovski, "Modes of Operation of the AES Algorithm," 2013.

[31] Nidhi Singhal and J.P.S.Raina, "Comparative Analysis of AES and RC4 Algorithms for Better Utilization", International Journal of Computer Trends and Technology, Vol 2, Issue 6, July-Aug 2011, pp.177-181.

[32] S. Almuhammadi and I. Al-Hejri, "A comparative analysis of AES common modes of operation," 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), Windsor, ON, 2017, pp. 1-4, doi: 10.1109/CCECE.2017.7946655.

[33] V .Kumara Swamy, Dr Prabhu G Benakop, "Performance Analysis of Secure Integrated Circuits using Blowfish Algorithm", Global Journal of Computer Science and Technology Network, Web & Security, Volume 13, Issue 17, Version 1.0,Page no.10-15, December 2013, Global Journals Inc (USA), Online ISSN. 0975-4172, Print ISSN.0975-4350.

[34] Diaa Salama, Hatem Abdual Kader and Mohiy Hadhoud (2011), "Studying the Effects of Most Common Encryption Algorithms", International Arab Journal of e-Technology, Vol. 2, No. 1, January 2011, pp 1-10.

[35] Gurjeevan Singh, Ashwani Kumar Singla, K. S. Sandha - Through Put Analysis of Various Encryption Algorithms, IJCST Vol.2, Issue3, September 2011.

[36] Monika Agrawal, Pradeep Mishra, 'A Comparative Survey on Symmetric Key Encryption Techniques', International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 4, No. 05, pp.877, May 2012.

[37] Patil, Priyadarshini & Narayankar, Prashant & Narayan, DG & S M, Meena. (2016). A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES,

RSA and Blowfish. *Procedia Computer Science*. 78. 617-624.

10.1016/j.procs.2016.02.108.

[38] Su, Xiaochun & Liu, Yanheng & Wan, Jian. (2019). Joint Optimization QoS and Security of Wireless Communication Networks. 310-316.

[39] M. Vaidehi and B. J. Rabi, "Design and analysis of AES-CBC mode for high security applications," in 2nd International Conference on Current Trends in Engineering and Technology (ICCTET), 2014. IEEE, 2014, pp. 499–502.