

Implementation and Performance Analysis of Different Symmetric Cryptographic Algorithms

A PROJECT

**Submitted in partial fulfilment of the requirements for the award of the
degree of**

**BACHELOR OF TECHNOLOGY
IN**

COMPUTER SCIENCE ENGINEERING

Under the supervision of

Dr. Amit Kumar Singh

By

Payal Sood (121261)

To



**JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY
WAKNAGHAT SOLAN – 173 234
HIMACHAL PRADESH INDIA
May, 2015**

CERTIFICATE

This is to certify that the work which is being presented in the project title “**Implementation and Performance Analysis of Different Symmetric Cryptographic Algorithms**” in partial fulfilment of the requirements for the award of the degree of Bachelor of technology and submitted in Computer Science Engineering Department, Jaypee University of Information Technology, Waknaghat is an authentic record of work carried out by Payal Sood during a period from August 2015 to May 2016 under the supervision of Dr. Amit Kumar Singh, Computer Science Engineering Department, Jaypee University of Information Technology, Waknaghat.

The above statement made is correct to the best of my knowledge.

Date: -

Dr. S P Grera
Professor and Head of Department
Computer Science Engineering
JUIT Waknaghat

Dr. Amit Kumar Singh
Assistant Professor
Computer Science Engineering
JUIT Waknaghat

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our project guide Dr. Amit Kumar Singh who helped us in conceptualizing the project and actual building of procedures used to complete the project. We would also like to thank our Head of department for providing us this golden opportunity to work on a project like this, which helped us in doing a lot of research and we came to know about so many things.

Secondly we would like to thank our family and friends who guided us throughout the project so as to complete our project on time.

Thanking you,

Payal Sood (121261)

TABLE OF CONTENTS

| | |
|--|-----|
| Certificate..... | i |
| Acknowledgement..... | ii |
| Table of Contents..... | iii |
| List of Figures..... | v |
| List of Tables..... | v |
| List of abbreviations and symbols..... | v |
| CHAPTER 1: Symmetric and Asymmetric Key Cryptography: An Introduction..... | 1 |
| 1.1 Cryptography..... | 1 |
| 1.2 Cryptanalysis..... | 1 |
| 1.3 Principles of Cryptography..... | 2 |
| 1.4 Types of Cryptography..... | 3 |
| 1.5 Type of Attacks..... | 5 |
| 1.6 Benefits of Cryptography..... | 6 |
| 1.7 Drawbacks of Cryptography..... | 7 |
| 1.8 Encryption Techniques..... | 7 |
| 1.8.1 DES..... | 7 |
| 1.8.2 TripleDES..... | 8 |
| 1.8.3 AES..... | 10 |
| 1.8.4 RC4..... | 11 |
| 1.8.5 Blowfish..... | 12 |
| 1.9 Comparison between various Encryption Techniques..... | 13 |
| CHAPTER 2: Literature Review..... | 14 |
| CHAPTER 3: Objectives..... | 25 |
| CHAPTER 4: Implementation and Performance Analysis..... | 26 |
| 4.1 Overview..... | 26 |
| 4.2 Technologies Used..... | 27 |
| 4.3 Task Performed..... | 28 |
| 4.4 Mathematical Formulas..... | 30 |
| 4.5 Performance Analysis of DES..... | 31 |
| 4.6 Performance Analysis of TripleDES..... | 34 |
| 4.7 Performance Analysis of RC4..... | 37 |
| 4.8 Performance Analysis of AES..... | 40 |

| | |
|---|----|
| 4.9 Performance Analysis of Blowfish..... | 43 |
| CHAPTER 5: System Screenshots and Result..... | 47 |
| 5.1 Screenshots of DES algorithm..... | 47 |
| 5.2 Screenshots of TripleDES algorithm..... | 49 |
| 5.3 Screenshots of RC4 algorithm..... | 51 |
| 5.4 Screenshots of AES algorithm..... | 53 |
| 5.5 Screenshots of Blowfish algorithm..... | 55 |
| 5.6 Result..... | 57 |
| CHAPTER 6: Conclusion and Future Direction..... | 58 |
| REFERENCE..... | 60 |

List of Figures

| Figure no | Description | Page no |
|-----------|---|---------|
| 1 | Secret Key Cryptography | 8 |
| 2 | Public Key Cryptography | 9 |
| 3 | Hash Function | 9 |
| 4 | General Structure of DES | 12 |
| 5 | Block Diagram of Triple DES | 14 |
| 6 | General Structure of AES | 15 |
| 7 | Encryption And Decryption in RC4 | 16 |
| 8 | Overall System Design for Triple Security of Data in Cloud Computing. | 24 |
| 9 | Flow chart for implementation of algorithm. | 30 |

List of Tables

| Table no | Description | Page no |
|----------|---|---------|
| 1 | Comparison between DES, Triple DES, AES , RC4 | 17 |

List of abbreviations and symbols

| S No. | Abbreviations & Symbols | Description |
|-------|-------------------------|--|
| 1 | DES | Data Encryption Standard |
| 2 | MAC | Message Authentication Code |
| 3 | NIST | National Institute of Standards and Technology |
| 4 | AES | Advanced Encryption Standard |
| 5 | RC4 | Rivest Cipher 4 |
| 6 | SSL | Secure Socket Layer |
| 7 | RTL | Register Transfer Level |
| 8 | FPGA | Field Programmable Gate Array |
| 9 | DSA | Data Signature Algorithm |
| 10 | JCE | Java Cryptographic Extension |

CHAPTER 1

Symmetric & Asymmetric Key Cryptography: An Introduction

1.1 Cryptography

Cryptography is a method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Cryptography is closely related to the disciplines of cryptology and cryptanalysis. Cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in storage or transit. However, in today's computer-centric world, cryptography is most often associated with scrambling plaintext (ordinary text, sometimes referred to as cleartext) into ciphertext (a process called encryption), then back again (known as decryption). Individuals who practice this field are known as cryptographers.

1.2 Cryptanalysis

Cryptanalysis refers to the study of ciphers, ciphertext, or cryptosystems (that is, to secret code systems) with a view to finding weaknesses in them that will permit retrieval of the plaintext from the ciphertext, without necessarily knowing the key or the algorithm. This is known as breaking the cipher, ciphertext, or cryptosystem. Breaking is sometimes used interchangeably with weakening. This refers to finding a property (fault) in the design or implementation of the cipher that reduces the number of keys required in a brute force attack (that is, simply trying every possible key until the correct one is found). For example, assume that a symmetric cipher implementation uses a key length of 2^{128} bits: this means that a brute force attack would need to try up to all 2^{128} possible combinations (rounds) to be certain of finding the correct key (or, on average, 2^{127} possible combinations) to convert the ciphertext into plaintext, which is not possible given present

and near future computing abilities. However, a cryptanalysis of the cipher reveals a technique that would allow the plaintext to be found in 2^{40} rounds. While not completely broken, the cipher is now much weaker and the plaintext can be found with moderate computing resources.

There are numerous techniques for performing cryptanalysis, depending on what access the cryptanalyst has to the plaintext, ciphertext, or other aspects of the cryptosystem.

1.3 Principles of Cryptography

1.3.1 Encryption

In a simplest form, encryption is to convert the data in some unreadable form. This helps in protecting the privacy while sending the data from sender to receiver. On the receiver side, the data can be decrypted and can be brought back to its original form. The reverse of encryption is called as decryption. The concept of encryption and decryption requires some extra information for encrypting and decrypting the data. This information is known as key. There may be cases when same key can be used for both encryption and decryption while in certain cases, encryption and decryption may require different keys.

1.3.2 Authentication

This is another important principle of cryptography. In a layman's term, authentication ensures that the message was originated from the originator claimed in the message. Now, one may think how to make it possible? Suppose, Alice sends a message to Bob and now Bob wants proof that the message has been indeed sent by Alice. This can be made possible if Alice performs some action on message that Bob knows only Alice can do. Well, this forms the basic fundamental of Authentication.

1.3.3 Integrity

Now, one problem that a communication system can face is the loss of integrity of messages being sent from sender to receiver. This means that Cryptography should ensure that the messages that are received by the receiver are not altered anywhere on the communication path. This can be achieved by using the concept of cryptographic hash.

1.3.4 Non Reputation

What happens if Alice sends a message to Bob but denies that she has actually sent the message? Cases like these may happen and cryptography should prevent the originator or sender to act this way. One popular way to achieve this is through the use of digital signatures.

1.4 Types of Cryptography

1.4.1 Secret Key Cryptography

This type of cryptography technique uses just a single key. The sender applies a key to encrypt a message while the receiver applies the same key to decrypt the message. Since only single key is used so we say that this is a symmetric encryption. Figure 1 shows the

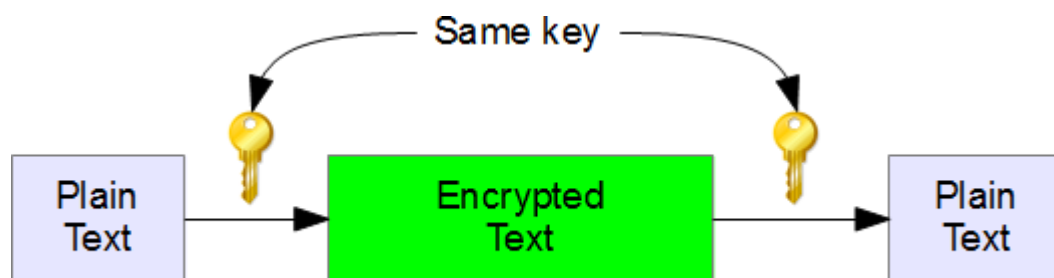


Figure 1: Concept of secret key cipher [1]

The biggest problem with this technique is the distribution of key as this algorithm makes use of single key for encryption or decryption.

1.4.2 Public Key Cryptography

This type of cryptography technique involves two key crypto system in which a secure communication can take place between receiver and sender over insecure communication channel. Since a pair of keys is applied here so this technique is also known as asymmetric encryption.

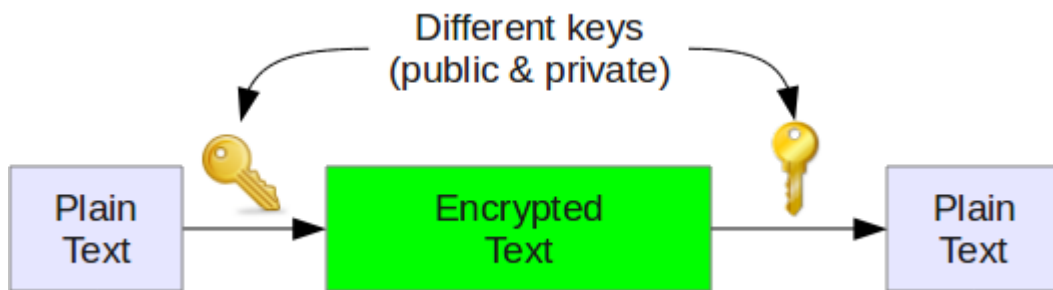


Figure 2: Concept of Public Key Cryptography [1]

In this method, each party has a private key and a public key. The private is secret and is not revealed while the public key is shared with all those whom you want to communicate with. If Alice wants to send a message to bob, then Alice will encrypt it with Bob's public key and Bob can decrypt the message with its private key.

1.4.3 Hash Functions

This technique does not involve any key. Rather it uses a fixed length hash value that is computed on the basis of the plain text message. Hash functions are used to check the integrity of the message to ensure that the message has not be altered, compromised or affected by virus.

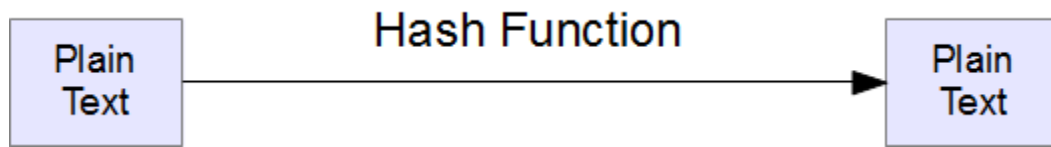


Figure 3: Working of Hash Function [1]

1.5 Types of Attacks

Known Plaintext and Ciphertext-Only Attacks

A known plaintext attack is an attack where a cryptanalyst has access to a plaintext and the corresponding ciphertext and seeks to discover a correlation between the two. A ciphertext-only attack is an attack where a cryptanalyst has access to a ciphertext but does not have access to corresponding plaintext. With simple ciphers, such as the Caesar Cipher, frequency analysis can be used to break the cipher.

Chosen Plaintext and Chosen Ciphertext Attacks

A chosen plaintext attack is an attack where a cryptanalyst can encrypt a plaintext of his choosing and study the resulting ciphertext. This is most common against asymmetric cryptography, where a cryptanalyst has access to a public key. A chosen ciphertext attack is an attack where a cryptanalyst chooses a ciphertext and attempts to find a matching plaintext. This can be done with a decryption oracle (a machine that decrypts without exposing the key). This is also often performed on attacks versus public key encryption; it begins with a ciphertext and searches for matching publicly-posted plaintext data.

Adaptive Chosen Plaintext and Adaptive Chosen Ciphertext Attacks

In both adaptive attacks, a cryptanalyst chooses further plaintexts or cipher texts (adapts the attack) based on prior results.

Brute Force Attacks

A brute force attack systematically attempts every possible key. It is most often used in a known plaintext or ciphertext-only attack.

Linear Cryptanalysis

Linear cryptanalysis is a known plaintext attack that requires access to large amounts of plaintext and ciphertext pairs encrypted with an unknown key. It focuses on statistical analysis against one round of decryption on large amounts of ciphertext. The cryptanalyst decrypts each ciphertext using all possible subkeys for one round of encryption and studies the resulting intermediate ciphertext to seek the least random result. A subkey that produces the least random intermediate cipher for all cipher texts becomes a candidate key (the most likely subkey).

Differential Cryptanalysis

Differential cryptanalysis is a chosen plaintext attack that seeks to discover a relationship between cipher texts produced by two related plaintexts. It focuses on statistical analysis of two inputs and two outputs of a cryptographic algorithm. A plaintext pair is created by applying a Boolean exclusive or (XOR) operation to a plaintext. For example, XOR the repeating binary string 10000000 to the plaintext. This creates a small difference (hence the term differential cryptanalysis) between the two. The cryptanalyst then encrypts the plaintext and its XORed pair using all possible subkeys, and it seeks signs of non-randomness in each intermediate ciphertext pair. The subkey that creates the least random pattern becomes the candidate key

1.6 Benefits of Cryptography

- Encryption technique can guard the information and communication from unauthorized revelation and access of information.
- The cryptographic techniques such as MAC and digital signatures can protect information against spoofing and forgeries.
- The cryptographic hash functions are playing vital role in assuring the users about the data integrity.

- The digital signature provides the non-repudiation service to guard against the dispute that may arise due to denial of passing message by the sender

1.7 Drawbacks of Cryptography

- A strongly encrypted, authentic, and digitally signed information can be difficult to access even for a legitimate user at a crucial time of decision-making. The network or the computer system can be attacked and rendered non-functional by an intruder.
- High availability, one of the fundamental aspects of information security, cannot be ensured through the use of cryptography. Other methods are needed to guard against the threats such as denial of service or complete breakdown of information system.
- Another fundamental need of information security of selective access control also cannot be realized through the use of cryptography. Administrative controls and procedures are required to be exercised for the same.
- Cryptography does not guard against the vulnerabilities and threats that emerge from the poor design of systems, protocols, and procedures. These need to be fixed through proper design and setting up of a defensive infrastructure.
- The security of cryptographic technique is based on the computational difficulty of mathematical problems. Any breakthrough in solving such mathematical problems or increasing the computing power can render a cryptographic technique vulnerable

1.8 Encryption Techniques

1.8.1 DES

DES is a symmetric-key block cipher published by the NIST. DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration –

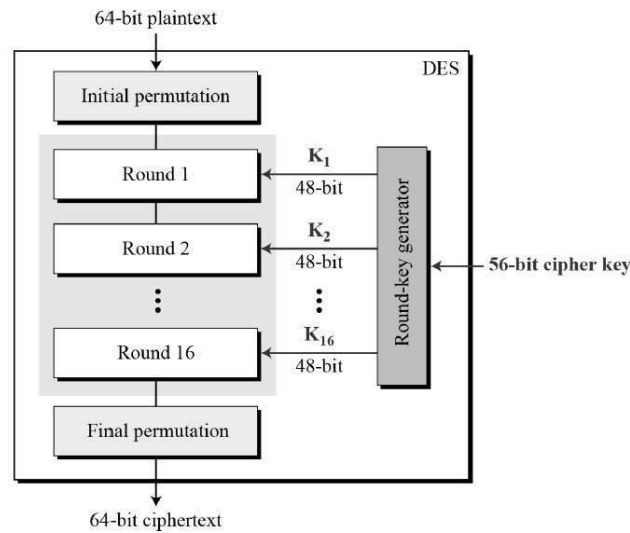


Figure 4: DES implementation stages [3]

Concerns about the Algorithm

- 56 bit key length (approx. 7.2×10^{16}) on initial consideration brute-force attack seems impractical. However with a massively parallel machine of about 5000 nodes with each node capable of achieving a key search rate of 50 million keys/sec, the time taken to do a brute-force search is approximately 100 hrs. Which is far from excessive.
- The nature of DES algorithm: of more concern is that cryptanalysis is possible by exploiting the characteristics of DES. The focus is the eight S-boxes used in each iteration. The design criteria for the complete algorithm has never been published and there has been speculation that the boxes were constructed in such a way that cryptanalysis is possible by an opponent who knows the weakness in the S-boxes.

1.8.2 Triple DES

Triple DES is based on the DES algorithm; it is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the attacks that can be used to reduce the amount of time it takes to break DES.

Triple DES is another mode of DES operation. It takes three 64-bit keys, for an overall key length of 192 bits. In Stealth, you simply type in the entire 192-bit (24 character) key rather than entering each of the three keys individually. The Triple DES DLL then breaks the user provided key into three subkeys, padding the keys if necessary so they are each 64 bits long. The procedure for encryption is exactly the same as regular DES, but it is repeated three times. Hence the name Triple DES. The data is encrypted with the first key, decrypted with the second key, and finally encrypted again with the third key.

Triple DES runs three times slower than DES, but is much more secure if used properly. The procedure for decrypting something is the same as the procedure for encryption, except it is executed in reverse. Like DES, data is encrypted and decrypted in 64-bit chunks. Although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. The least significant (rightmost) bit in each byte is a parity bit, and should be set so that there are always an odd number of 1s in every byte. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. This means that the effective key strength for Triple DES is actually 168 bits because each of the three keys contains 8 parity bits that are not used during the encryption process.

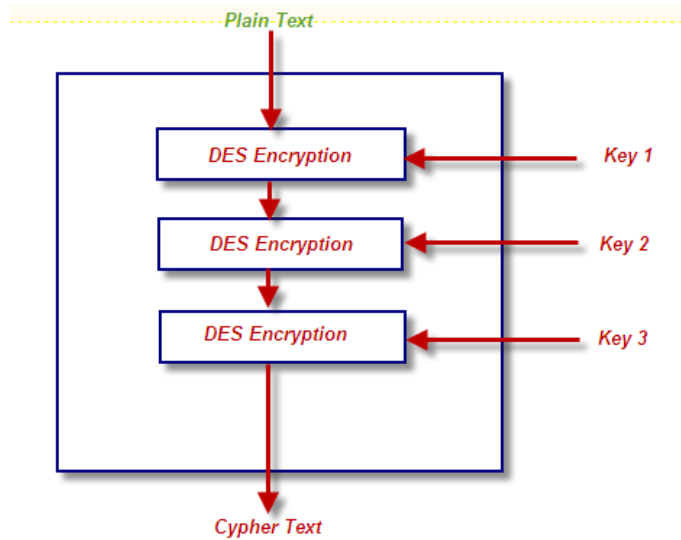


Figure 5: Triple DES stages [4]

1.8.3 AES

The Advanced Encryption Standard or AES is a symmetric block cipher used by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data. The origins of AES date back to 1997 when the NIST announced that it needed a successor to the aging DES which was becoming vulnerable to brute-force attacks.

AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively. (Rijndael was designed to handle additional block sizes and key lengths, but the functionality was not adopted in AES.) Symmetric or secret-key ciphers use the same key for encrypting and decrypting, so both the sender and the receiver must know and use the same secret key. All key lengths are deemed sufficient to protect classified information up to the "Secret" level with "Top Secret" information requiring either 192- or 256-bit key lengths. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys -- a round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of ciphertext.

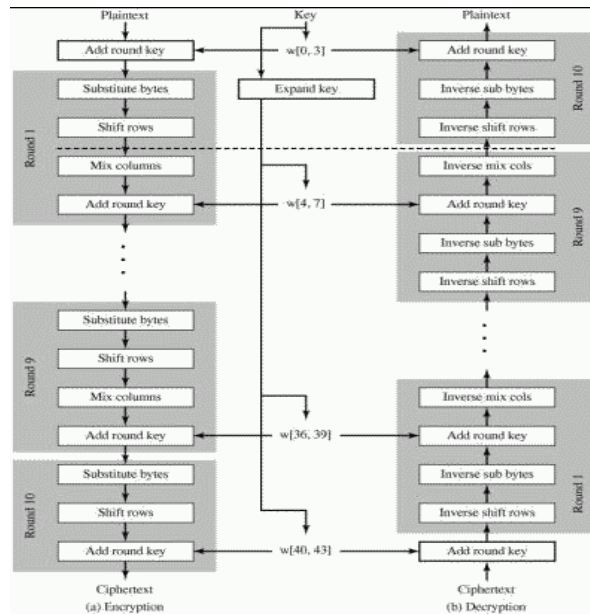


Figure 6 : AES implementation Stages [5]

1.8.4 RC4

RC4 is a stream cipher designed in 1987 by Ron Rivest for RSA Security. It is a variable keysize stream cipher with byte-oriented operations. The algorithm is based on the use of a random permutation. Analysis shows that the period of the cipher is overwhelmingly likely to be greater than 10100. Eight to sixteen machine operations are required per output byte, and the cipher can be expected to run very quickly in software. RC4 was kept as a trade secret by RSA Security. In September 1994, the RC4 algorithm was anonymously posted on the Internet on the Cypherpunks anonymous remailers list.

The RC4 algorithm is remarkably simply and quite easy to explain. A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S , with elements $S[0], S[1] \dots S[255]$. At all times, S contains a permutation of all 8-bit numbers from 0 through 255. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of k is generated, the entries in S are once again permuted.

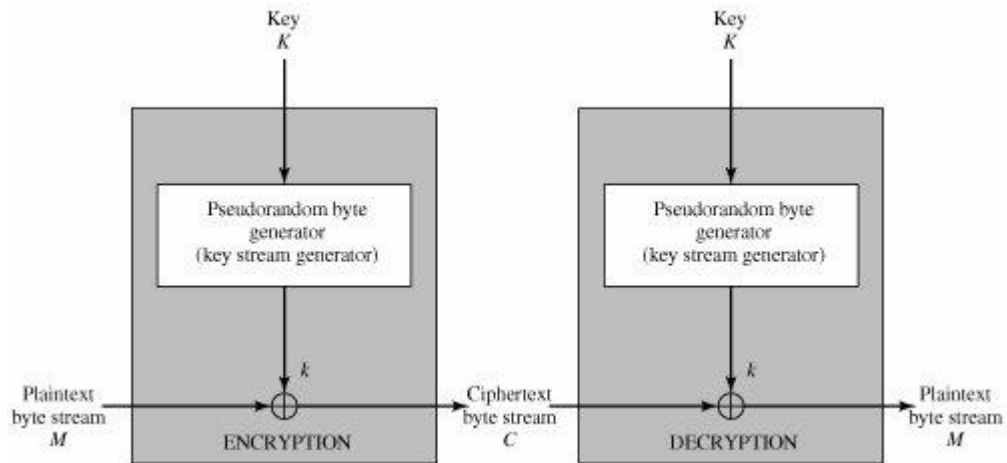


Figure 7: Encryption Decryption process in RC4 [6]

1.8.5 Blowfish

It is a Feistel network, with a secret-key block cipher. It has an iteration of 16 times with a block size of 64 bits and each key can be any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors.

The key size can be variable, with a 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several subkey arrays totalling 4168 bytes.

The encryption occurs with a 16-round Feistel network. A permutation of the key dependent is needed every round just as data dependent substitution. All the operations used are XORs and additions on 32-bit words. Also it use four indexed arrays data lookups.

1.9 Comparison between DES, Triple DES, AES, RC4 and Blowfish.

Following table shows the comparative analysis of different symmetric cryptographic algorithms

Table 1: Comparison between DES, Triple DES, AES and RC4.

| Algorithm | Key Size (bits) | Block Size (bits) | Type | Features |
|------------------|------------------------|--------------------------|---------------|--|
| DES | 64 | 64 | Block Cipher | Most Common , Not Strong Enough |
| Triple DES | 192 | 64 | Block Cipher | Modification of DES, adequate security |
| AES | Variable(128, 192,256) | 128 | Block Cipher | Replacement for DES , Excellent Security |
| RC4 | Variable(40 ,128) | Variable(32,64, 128) | Stream Cipher | Fast Stream Cipher In SSL |
| Blowfish | Variable(32 - 448) | 64 | Block Cipher | Fast, compact and Simple |

CHAPTER 2

A Brief Literature Review on Different Symmetric Algorithms

To study the different symmetric cryptographic algorithms, we have gone through various research paper and study the work done by them. Following is the summary of work done by them in the area of symmetric cryptographic algorithms like DES, TripleDES, AES, RC4 and Blowfish.

1) Rani et al. [7] proposed a method for analysing various symmetric and Asymmetric key encryption algorithms (DES, Triple DES, AES, and RC4) based on different parameters such as Avalanche effect, encryption and decryption time etc. With the fast evolution of digital data exchange, security information is very important in data storage and transmission. Data encryption is widely used to ensure security in open networks such as the internet. Security is a very important factor in every field such as Government Agencies (CBI, FBI), Research Organization, E-commerce and etc. where internet is being used. Each type of data has its own features, therefore, different techniques should be used to protect confidential image data from unauthorized access. Cryptography is a technique to secure data on the network from unauthorized user. There are different types of a cryptography algorithm (a) symmetric and (b) asymmetric has been designed. To secure data it is necessary to know which algorithm provides better security, efficiency, accuracy and effectiveness. This paper presents the complete analysis of various symmetric key encryption algorithms (DES, Triple DES, AES, and RC4) based on different parameters. Analysis of the sensitivity to secret key, differential analysis, an analysis of the key space and the encryption speed, etc. This Paper compares the various algorithms based on

- Architecture
- Scalability
- Avalanche effect
- Flexibility
- Security level

2) Kumar et al. [8] proposed VLSI Implementation of DES & TDES Algorithm with Cipher Block Concept. This paper presents Field Programmable Gate Array (FPGA) implementation of the DES and Triple-DES with improved security against power analysis attacks. This is programmed in Verilog. DES & TDES is basically used in various cryptographic applications and wireless protocol security layers. The proposed designs use Boolean masking, a previously introduced technique to protect smart card implementations from these attacks. Triple DES was the answer to many of the shortcomings of DES. Since it is based on the DES algorithm, it is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES. However, even this more powerful version of DES may not be strong enough to protect data for very much longer. The DES algorithm itself has become obsolete and is in need of replacement. DES encrypts data in 64-bit and it is a symmetric algorithm. The key length is 56-bits.

In addition, this paper also covers DES and Triple DES algorithm with Cipher Block concept, simulation results, basic FPGA technology and the implementation details of the proposed DES and Triple DES architecture. Register transfer level (RTL) of DES and Triple DES algorithm is designed, simulated and implemented separately using Verilog in different FPGA devices including Cyclone II, Spartan 3, Vertex 5 and Vertex E series FPGAs. The results from the comparison with existing implementations show that the proposed design was efficient in all aspects. In this work, a compact hardware implementation of DES and Triple DES was presented. The design was implemented in real hardware with Cyclone II FPGA. The proposed architecture was also implemented with Spartan 3E, Vertex 5 and Vertex E FPGA devices and compared with the existing results. Here Cipher Block Chaining modes have been used by combining the previous cipher text block with the current message block before encrypting. DES and Triple DES algorithm are used significantly in satellite communications and electronic financial transactions, cryptographic key encryption for automated key management applications, file encryption, mail encryption, and other applications. In fact, it is extremely difficult, if not impossible, to find a cryptographic application where the DES cannot be applied.

Technologies are becoming smarter and compact day by day, so we hope this work will add new dimension in that trend. This design will play a remarkable role with its significant speed and efficiency.

3) Narula et al. [9] proposed Implementation of Triple Data Encryption using Verilog. Encryption is an essential tool for protecting the confidentiality of data. Network security protocols such as SSL or IPsec use encryption to protect Internet traffic from eavesdropping. Encryption is also used to protect sensitive data before it is stored on non-secure disks or tapes. Encryption, however, is computationally expensive. A computer server that must encrypt data for thousands of clients before sending it over the network can easily become crypto-bound. The capacity of the server is then determined by the speed at which it can perform encryption. This is especially the case when slow encryption protocols such as the Digital Encryption Standard or Triple-DES are employed. Since DES and Triple-DES are very widely used, it is important to optimize the performance of these algorithms. Triple-DES is basically used in various cryptographic applications and wireless protocol security layers. This paper presents the design and the implementation of the Triple Data Encryption Standard algorithm. The main objective is to provide with a deep insight of the theory and design of a digital cryptographic circuit with the use of Verilog.

The proposed implementation of DES and TDES provide high-speed performance with very compact hardware implementation. This paper examines the full procedure of implementing a DES and Triple DES algorithm using a high-level hardware description language Verilog. It is a flexible solution for any cryptographic system and security layers of wireless protocol. The electronic payment industry uses Triple DES and continues to develop and promulgate standards based upon it to protect user content and system data.

4) Agrawal [10] proposed Implementation and analysis of various symmetric cryptosystems. This paper implements some of the widely used symmetric encryption techniques i.e. data encryption standard, triple data encryption standard, advanced encryption standard, BLOWFISH and RC4 in MATLAB software. After the

implementation, these techniques are compared on some points. These points are avalanche effect due to one bit variation in plaintext keeping the key constant, avalanche effect due to one bit variation in key keeping the plaintext constant, memory required for implementation and simulation time required for different message lengths.

All the above mentioned techniques i.e. DES, Triple DES, AES, Blowfish and RC4 have been implemented in MATLAB 7.0 software. DES is the most widely used encryption scheme, especially in financial applications. In Triple DES memory required for implementation is the highest means it is the slowest algorithm. This is the main drawback of Triple DES. It is having a sufficient value of avalanche effect. Several internet-based applications have adopted triple DES. But because of various drawbacks it is not a reasonable candidate for long term use. In AES the avalanche effect is highest. AES is being considered by the US government as a replacement for DES. AES is ideal for encrypting messages sent between objects via chat-channels, and is useful for objects that are part of a game, or anything involving monetary transactions. Blowfish is a very strong algorithm because of key dependent S-box design. In DES the design of Sboxes is fixed but in Blowfish the S-box design is key dependent. This feature especially with larger Sboxes (e.g. 8 x 32) yields highly non-linear results and therefore very difficult to cryptanalyze. In Blowfish each round consists of a key-dependent permutation, and a key- and data-dependent substitution therefore it is only suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. Blowfish is used in Secure Shell (SSH) technologies. It is used in a number of implementations, including the encryption of communications and HDTV transmissions. RC4 is the shortest algorithm means it requires minimum memory space for implementation. RC4 is used in many commercial software packages such as Lotus notes and Oracle secure SQL. It is used in popular protocols such as Secure sockets layer (to protect Internet traffic) .It requires the minimum simulation time. Each cryptosystems is having its own advantages and disadvantages.

5) Mudaliar et al. [11] implemented Effectiveness of DES, Triple DES and AES on MPLS network.MPLS is Multi-Protocol Label Switching Network .Internet Engineering Task

force developed this technology specially to speed up the forwarding characteristics of routers. It uses the protocols of both layer 2 and layer 3. It employs label switching technique hence making the technology fast, efficient and secure. Various types of encryption algorithm are used to secure MPLS network. Some of them are advanced Encryption Standard, Data Encryption Standards and Triple DES to secure the network against brute force attack. In this paper we have encrypted a string and analysed the DES, Triple DES and AES algorithm on MPLS network against brute force attack and plotted a graph to show the effectiveness in MATLAB environment.

Classes available in JAVA package javax.crypto is used to implement AES, DES and Triple DES. Separate functions for encryption and decryption have been implemented in MATLAB using JAVA cryptography API. Brute force attack is implemented in MATLAB environment and is thoroughly optimized to give the maximum performance for the algorithm.

Started the attack with 8 bit of key length extended up to 48 bit on a string. Label is generated and the encrypted data is transported across MPLS network. After the successful completion of brute force attack the key and the label applied are cracked after some number of iterations. It can be further extended up to 256 bits which is supported key block for AES so we can use parallel computers with high computational powers to decrease the time required to find the key for above algorithms. The AES has a better security against brute force attack than DES and Triple DES as observed in the results. Hence we can say AES proves to be a better security algorithm than DES and Triple DES. AES takes much more time to break by the brute force attack for a given key length. This time rises in exponential manner with increase in key length.

6) Sachin et al. [12] proposed Implementation and Analysis of AES, DES and Triple DES on GSM Network. Global System for Mobile Communications (GSM) is the most widely used cellular technology in the world. Main objective in mobile communication systems is security of data exchanged. GSM uses several cryptographic algorithms for security like A5/1, A5/2 and A5/3. But it has been found that these algorithms are cracked by various

practical attacks so these algorithms do not provide sufficient levels of security for protecting the confidentiality of GSM therefore it is desirable to secure data by additional encryption. In this paper, they have done additional encryption by implementing DES, TripleDES, and AES algorithms on GSM Network. This paper also analyses the effectiveness of these algorithms against brute force attack implemented in MATLAB environment. This paper outlines the provision of encrypted information over GSM. For security of data in GSM networks such encryption and mechanisms to provide it are required. In this paper a new approach to encryption has been proposed which includes extra encryption with AES, DES and Triple DES algorithm. GSM uses stream ciphers for encryption which requires the data to be in binary form. Our technique does encryption directly on symbols without going on to the bit level. Also, this technique does not require any hardware; it is totally based on software. This technique is much simpler than existing techniques thus a more robust and efficient system is achieved. Following Figure 5 show the data encryption in GSM.

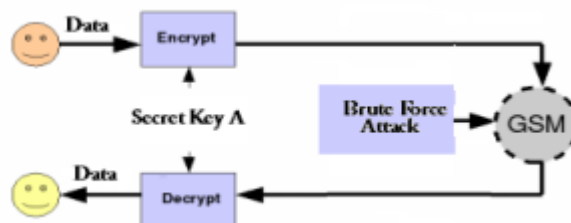


Figure 8: data encryption in GSM[12]

The implementation of DES, TripleDES and AES uses classes available in JAVA package javax.crypto. Separate functions for encryption and decryption have been implemented in MATLAB using JAVA cryptography API. Brute force program is implemented in MATLAB environment. This implementation is thoroughly tested and is optimized to give the maximum performance for the algorithm. The presented results showed that AES has a better security against the brute force attack than other common encryption algorithms used; therefore it is an excellent candidate to be considered as a standard encryption algorithm for GSM Network.

7) Rani et al. [13] proposed Suspicious Email Detection System via Triple DES Algorithm.

The need for Suspicious email detection System is increasing due to the rapid usage of Email communication in the Internet world. The proposed “Suspicious email detection System” provides a way to identify the criminal activities. It detects the suspicious Emails indicating Keywords by applying the Cryptography algorithm called “Triple Data Encryption Standard”. In this paper, we have applied Cryptography techniques to detect suspicious emails, i.e., an email that alerts of upcoming terrorist events. We have applied Triple DES algorithms, emphasizing initially on given a plaintext message, the first key is used to DES- encrypt the message. The second key is used to DES-decrypt the encrypted message. (Since the second key is not the right key, this decryption just scrambles the data further.) The twice scrambled message is then encrypted again with the first key to yield the final cipher text. This three-step procedure is called triple-Triple-DES is just DES done three times with two keys used in a particular order. (Triple-DES can also be done with three separate keys instead of only two. In either case the resultant key space is about 2^{112} .)

Detecting Suspicious and criminal activities prior to the attacks and providing security to the people is the challenging task for the investigators or administrator Email Is a technology that includes passing and sending information from one place to another, using computer and the Internet. It is beneficial in both our personal and professional life. As Electronic mail is largely used by the terrorists for their communication, there is a need for suspicious email detection system that classifies emails to detect suspicious activities and make the administrator alert. In this paper work, we will detect the suspicious mails sent from the users who are already registered on this System. Firstly new users sign up themselves on the site to send the mails to those users who already registered and then view the messages from the registered users. Triple DES Algorithm used by admin to encrypt the messages sent to the users or sent some warnings about the other users’ suspicious activity. In this work, suspicious words dictionary is used to detect the suspicious words which are not actually used in the normal messaging or communication.

The proposed System is solved the problem definition by detecting the suspicious mails. Admin is created the data dictionary of suspicious words and this data dictionary makes

help to detect the suspicious activity of the users. Admin further will be added the suspicious words into the existing Suspicious Words data dictionary.

8) Debnath et al. [14] proposed DES, AES and triple DES: symmetric key cryptography algorithm. Security is one of the most challenging aspects in the internet and networks. Cryptography is the one of the main categories that converts information into an unreadable form. Cryptography allows people to carry their confidential data over the network without worries and insecurity. This paper provide comparison between three symmetric key cryptosystem i.e. DES, AES and triple DES.

In this paper a new comparative study between DES, Triple DES and AES were presented into various factor, which are key length, cipher type, block size, developed, cryptanalysis resistance, security, and possibility key. The time required to check all possible key at 50 billion second are proved that the AES is better than DES and Triple DES.

9) Saini et al. [15] proposed Triple Security of Data in Cloud Computing. Cloud computing is the biggest buzz in computer world now a days. It is providing excellent facilities by flexible infrastructure. Cloud computing is based on clientserver architecture. Cloud computing is a hub of various server and many database to store data. Cloud computing provide many services to user which is reliable, efficient and low cost. As it is internet based technology security, data security becomes a big issue to the cloud data. Many issues like data authenticity, integrity, data hiding and availability. In this paper we introduce a mechanism to provide secure data. We combine three algorithm DSA, DES and Steganography to provide security of data in cloud computing.

‘Triple Security in Cloud Computing’ by using three different security algorithms such as-

- 1) DSA
- 2) DES
- 3) Steganography – hiding data behind an audio file.

In their proposed work they provide security by implementing three algorithm DSA, DES and steganography together to cloud network. To implement these three algorithm they use Asp.net as a platform.

In their proposed system for encryption first apply DSA for authentication of data. Then apply AES algorithm for encryption and then hiding data within audio file for provide maximum security to the data. Receiver can get original plain text by reversing the steganography, AES and DSA.

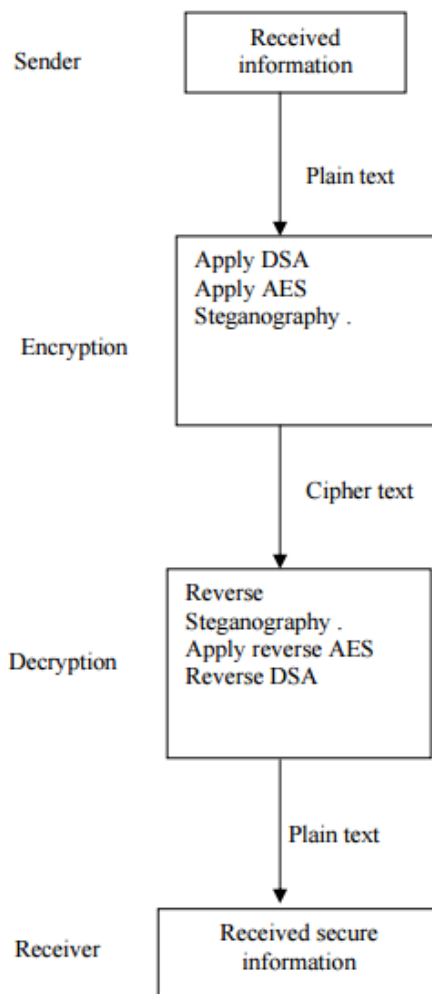


Figure 9: cryptography in cloud network [4]

10) Jeon et al. [16] proposed Optical Implementation of Triple DES Algorithm Based on Dual XOR Logic Operations. In this paper, we propose a novel optical implementation of a Triple DES algorithm based on dual XOR logic operations for a cryptographic system. In the schematic architecture, the optical Triple DES system consists of dual XOR logic operations, where XOR logic operation is implemented by using a free-space interconnected optical logic gate method. The main point in the proposed Triple DES method is to make a higher secure cryptosystem, which is acquired by encrypting an individual private key separately, and this encrypted private key is used to decrypt the plain text from the cipher text. Schematically, the proposed optical configuration of this cryptosystem can be used for the decryption process as well. The major advantage of this optical method is that vast 2-D data can be processed in parallel very quickly regardless of data size. The proposed scheme can be applied to watermark authentication and can also be applied to the OTP encryption if every different private key is created and used for encryption only once. When a security key has data of 512×256 pixels in size, our proposed method performs 2,048 DES blocks or 1,024 Triple DES blocks cipher in this paper. Besides, because the key length is equal to 512×256 bits, 2512×256 attempts are required to find the correct key. Numerical simulations show the results to be carried out encryption and decryption successfully with the proposed Triple DES algorithm.

They propose a novel optical implementation method of a triple DES algorithm based on dual XOR logic operations for a cryptographic system. The optical Triple DES system is realized by dual XOR logic operations, where XOR logic operation is implemented by using a free-space interconnected optical logic gate method. The optical schematic of the proposed method has two Mach-Zehnder type interferometers Simultaneously. The inner interferometer is used for encrypting a private key with a public key, while the outer interferometer is used for encrypting a plain text with the Same private key. The suggested optical setup of the cryptosystem can also be used for the decryption process. The major merit of this optical implemented cryptography is that a vast amount of data can be processed in parallel very quickly and the encryption/decryption processing time is much faster than electronic methods. Also, the encryption/ decryption processing time does not diminish in spite of data expansion owing to the parallel processing property. The proposed system is

convenient for exchanging the different private keys in the form of cipher and for decrypting the plain text only by the corresponding private key. This fact implies that our scheme can be applied to the OTP encryption if users create different private keys each time at their own discretion. Of course, the proposed dual XOR optical encryption method provides higher security strength to use double key encryption, and has an advantage of simple optical setup configuration. Their proposed method seems to perform 2,048 DES blocks or 1,024 Triple DES blocks cipher. Besides, because the key length is equal to 512×256 bits, $2^{512 \times 256}$ attempts are required to find the correct key. Computer experiments verified that the proposed method is perfect and suitable for cryptographic applications and secure communication system.

CHAPTER 3

Objectives of the Proposed Work

Security is one of the most challenging aspects in the internet and networks. Cryptography is the one of the main categories that converts information into an unreadable form. Cryptography allows people to carry their confidential data over the network without worries and insecurity.

Increase the security level provided by the strong cryptographic algorithm(s)

Our main aim is to implement various cryptographic algorithms like DES, Triple DES, AES, RC4 and Blowfish.

Input a file of variable size and implement the algorithms and examine the results.

Performance Analysis of these algorithms.

Choose the appropriate performance metric to evaluate the algorithm.

CHAPTER 4

Implementation and Performance Analysis

4.1 Overview

Designing this project system ideally consists of a continuous spectrum of activity, ranging from theoretically analysing the problems to be solved to evaluating the technology to be used for the components. When dealing with an electronic digital computer of more than modest size that is intended to be used for fairly complex applications, one is forced to split the planning spectrum into arbitrary segments, each segment being developed with due regard for its neighbours.

We start with going through research papers and make our own analysis of how thing should work and carry on steps regarding it. We need to have Sun's Java Cryptography Extension (SunJCE) package installed on your computer and SunJCE installed as a security provider in your java. Security.file in order to run the program. This is because the program uses symmetric ciphers.

All the important work done in implementation of this project is divided into three milestone completed in one semester with overview of following sequence of steps-

- Employ our understanding from previous 10 years research papers.
- All the theoretical attacks to be understood.
- We use NetBeans IDE to write code and compile on.
- Java cryptography library functions to be selected.
- Design and analysis of code.
- Thorough debugging while coding
- Output results are compared
- Different parameters are chosen to emphasize the performance analysis

- Necessary graphs to be drawn
- Throughput, time taken are to be measured

4.2 Technologies Used

Software Support

Operating System: Windows or GNU/Linux Programming

Language: Java Eclipse – 4.5.0 IDE

System Requirements: JDK 1.4 or Higher.

Java being a platform independent language, the projects runs on any platform.

SUN JCE

The Java Cryptography Extension from Sun Microsystems is an optional package to the Java 2 platform. It is a framework for implementing encryption, key generation and key agreement, and Message Authentication Code algorithms.

Understanding JCE Package classes-

- javax.crypto.spec.SecretKeySpec;
- javax.crypto.Cipher;
- javax.crypto.CipherInputStream;
- javax.crypto.CipherOutputStream;

The javax.crypto package defines classes and interfaces for various cryptographic operations. The central class is Cipher, which is used to encrypt and decrypt data. CipherInputStream and CipherOutputStream are utility classes that use a Cipher object to encrypt or decrypt streaming data. SealedObject is another important utility class that uses a Cipher object to encrypt an arbitrary serializable Java object.

This class performs encryption and decryption of byte arrays. Cipher is provider-based, so to obtain a Cipher object, you must call the static getInstance () factory method. The arguments to this method are a string that describes the type of encryption desired and, optionally, the name of the provider whose implementation should be used. To specify

the desired type of encryption, you can simply specify the name of an encryption algorithm, such as "DES". Or you can specify a three-part name that includes the encryption algorithm, the algorithm operating mode, and the padding scheme. These three parts are separated by slash characters, as in "DES/CBC/PKCS5Padding". Finally, if you are requesting a block cipher algorithm in a stream mode, you can specify the number of bits to be processed at a time by following the name of the feedback mode with a number of bits. For example: "DES/CFB8/NoPadding".

This class is an input stream that uses a Cipher object to encrypt or decrypt the bytes it reads from another stream. You must initialize the Cipher object before passing it to the CipherInputStream () constructor.

This class is an output stream that uses a Cipher object to encrypt or decrypt bytes before passing them to another output stream. You must initialize the Cipher object before passing it to the CipherOutputStream () constructor. If you are using a Cipher with any kind of padding, you must not call flush () until you are done writing all data to the stream; otherwise decryption fails.

4.4 Steps

Milestone #1

- ✓ Introduction with cryptography
- ✓ Advantage of cryptography
- ✓ Computation limitation of different algorithms
- ✓ What are the different attacks
- ✓ Differential and Linear cryptanalysis
- ✓ Five research papers each on it
- ✓ Performance metric

Milestone #2

- ✓ Performance comparison of DES, TripleDES, RC4, AES and Blowfish.
- ✓ Parameters taken –

- Encryption time
 - Decryption time
 - Throughput in kb/sec
 - Memory usage
- ✓ Plot graph for above parameters
 - ✓ File size taken into consideration
 - 1000kb
 - 1500kb
 - 2000kb
 - 2500kb
 - 3000kb
 - ✓ Calculate encryption throughput and decryption throughput
 - ✓ Calculate encryption and decryption memory usage
 - ✓ Effect of change on key size on time.

Implementation

Following are the steps to generate Secret keys and encryption:

1. Create an interface of the SecretKey interface. This interface contains no methods. Its only purpose is to group secret keys.
2. KeyGenerator Class: Class provides functionality of symmetric key generator. They are constructed using one of the getInstance class methods.
3. To generate key of the DES algorithm we use: `KeyGenerator getInstance(String algorithm)` method with `KeyGenerator key=KeyGenerator.getInstance("DES").getInstance();`
4. The getInstance method generates a KeyGenerator object for the specified key algorithm from the specified provider. Some 15 of the standard algorithms available with Java Cryptography Architecture API Specification are: AES, DES, Blowfish, DESede, RSA.

5. The implementation uses managed wrappers for DES and Triple DES available in System. Security.Cryptography that wraps unmanaged implementations available in CryptoAPI. These are DESCryptoServiceProvider and TripleDESCryptoServiceProvider respectively.

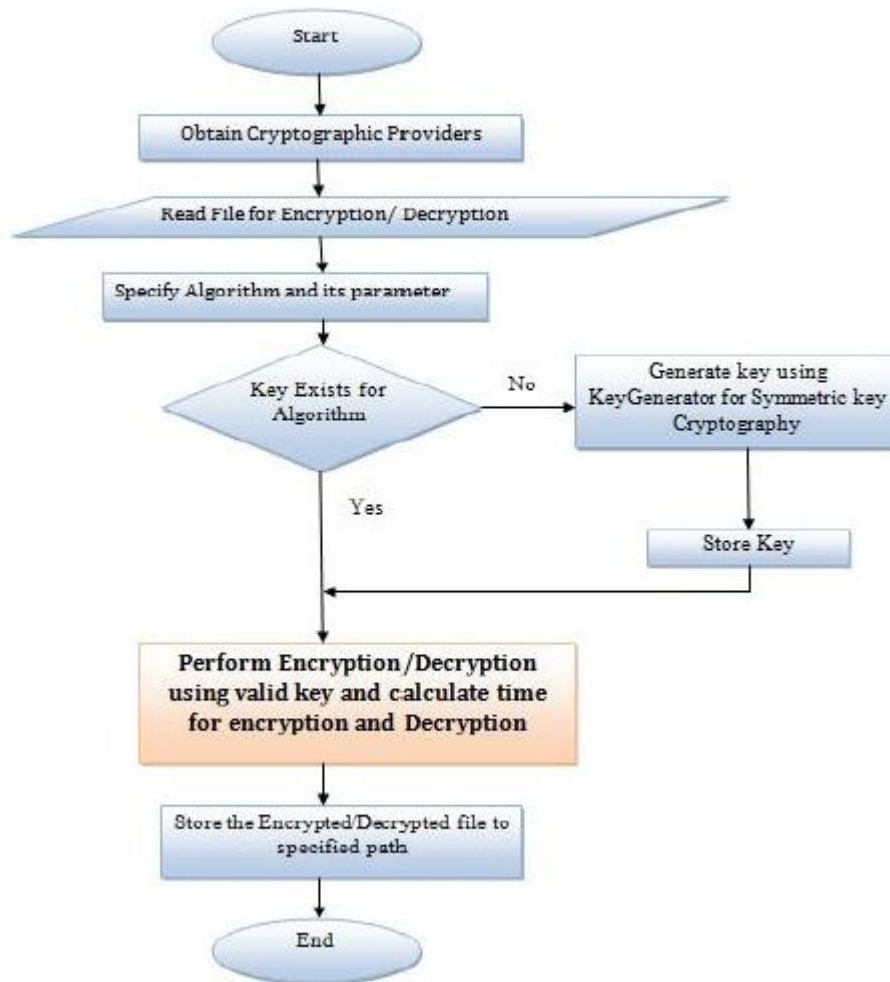


Figure 10: Encryption decryption flowchart [17]

4.5 Performance Metrics

- Encryption Time:-The time required to encrypt data and convert it into unreadable form is called encryption time.
- Decryption Time: - The time require to get back data after encryption is called as decryption.

- Time Efficiency: - A measure of time for an algorithm to execute.
- Space Efficiency: - A measure of the amount of memory needed for an algorithm to execute.
- Encryption Throughput: - File Size/ Total Encryption Time in mbps
- Decryption Throughput: - File Size / Total Decryption Time in mbps
- 1 bit = $1.25 * 10^{-7}$ mb

4.6 Performance Analysis of DES

Following figure shows a graph between encryption and decryption time .With increasing file Size encryption Time increases.



Figure 4.6.1: Graph of File Size against Encryption Time

Following figure shows a graph between File Size and Decryption time. With increasing file Size, decryption time increases.

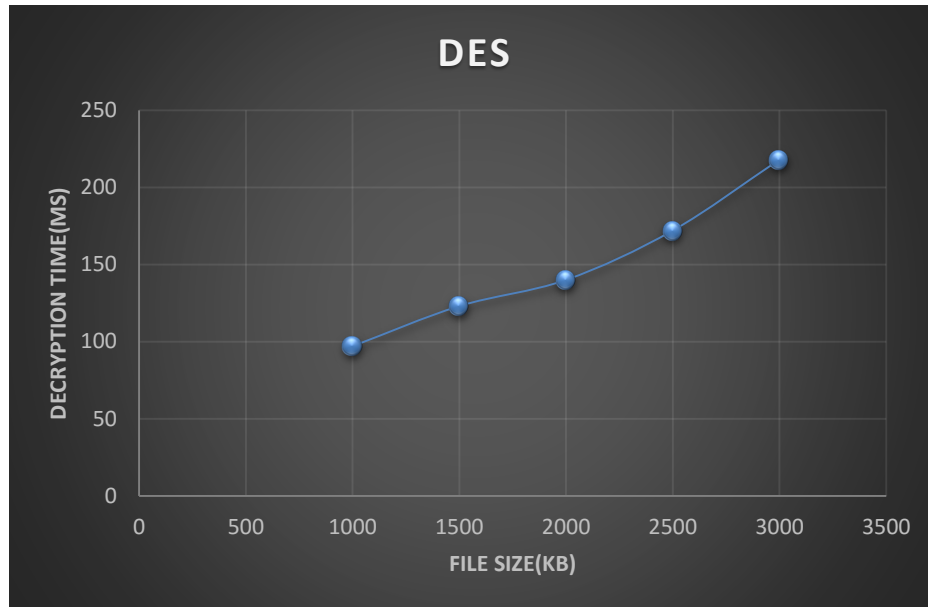


Figure 4.6.2: Graph of File Size against Decryption Time

Following figure shows a graph between file size and encryption throughput.

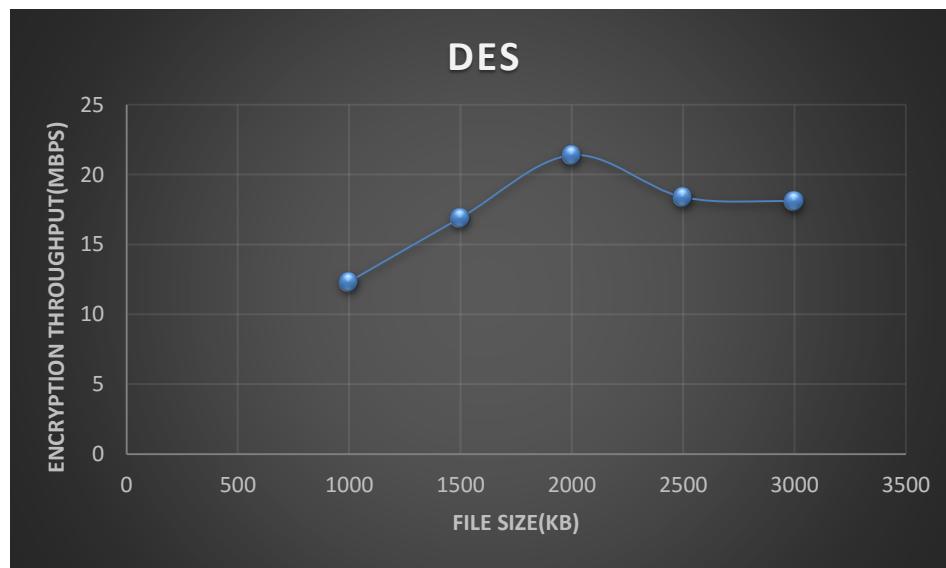


Figure 4.6.3: Graph of File Size against Encryption Throughput

Following Figure shows a graph between file Size and decryption throughput. With increasing file size, decryption throughput increases

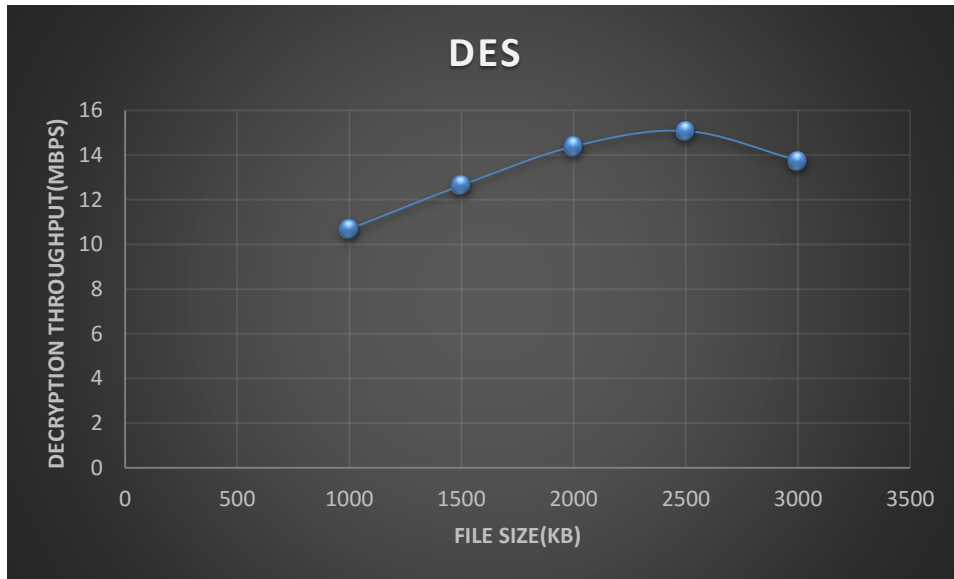


Figure 4.6.4: Graph of File Size against decryption throughput

Following figure shows a graph between file size and encryption memory usage.

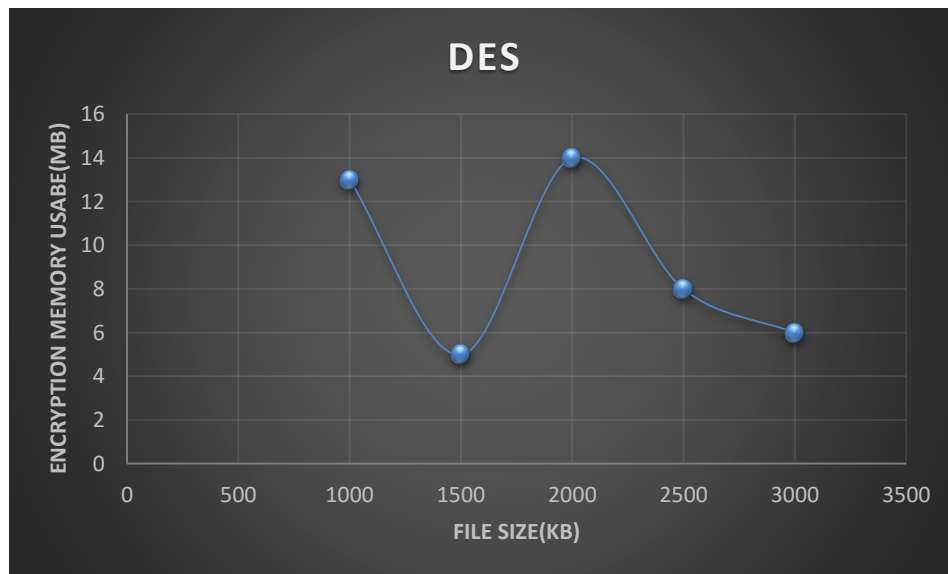


Figure 4.6.5: Graph of File Size against Encryption Memory Usage.

Graph between file size and decryption memory usage.

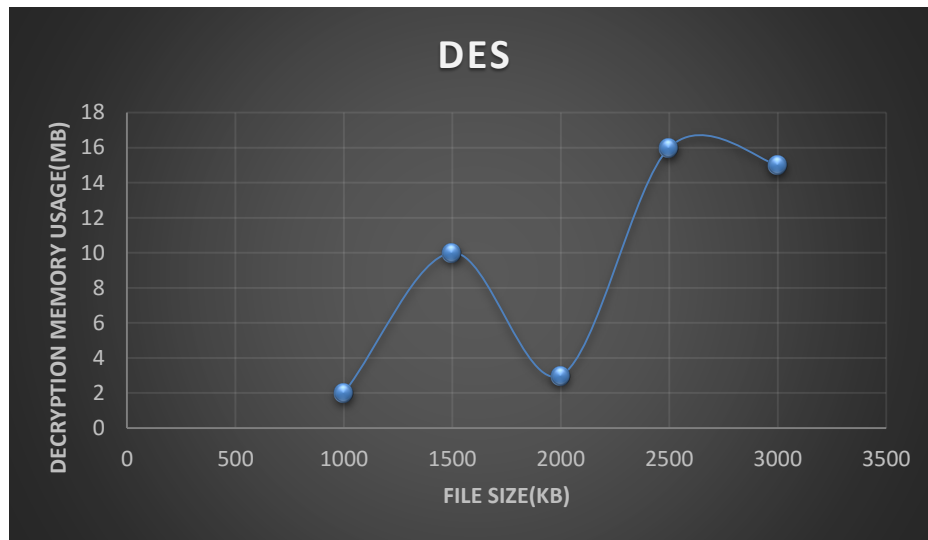


Figure 4.6.6: Graph of File Size against Decryption Memory Usage.

4.7 Performance Analysis of Triple DES

Shows the graph between encryption and decryption time .With increasing file Size encryption Time increases

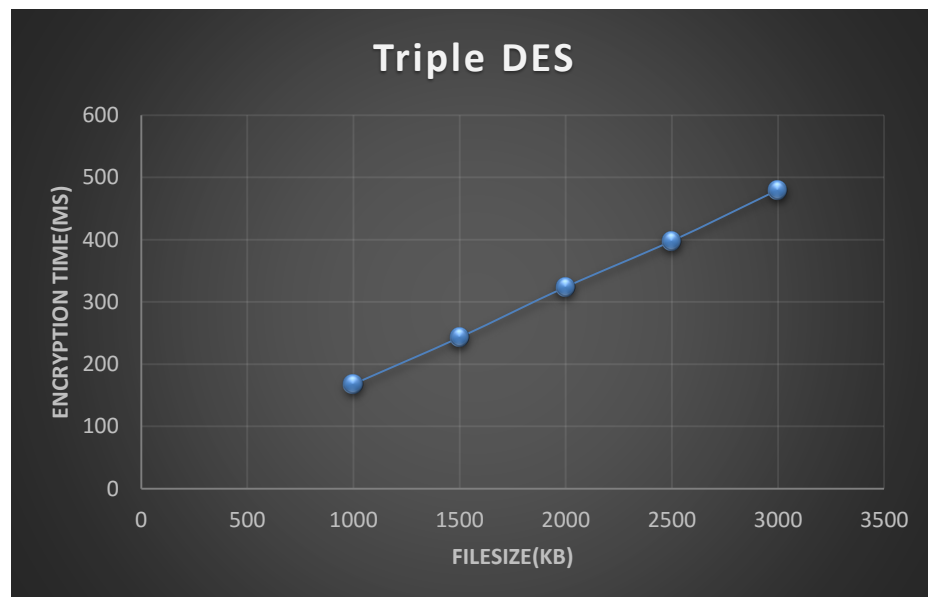


Figure 4.7.1: Graph of File Size against Encryption Time

Following Figure shows the Graph between File Size and Decryption time. With increasing file Size, decryption time increases.

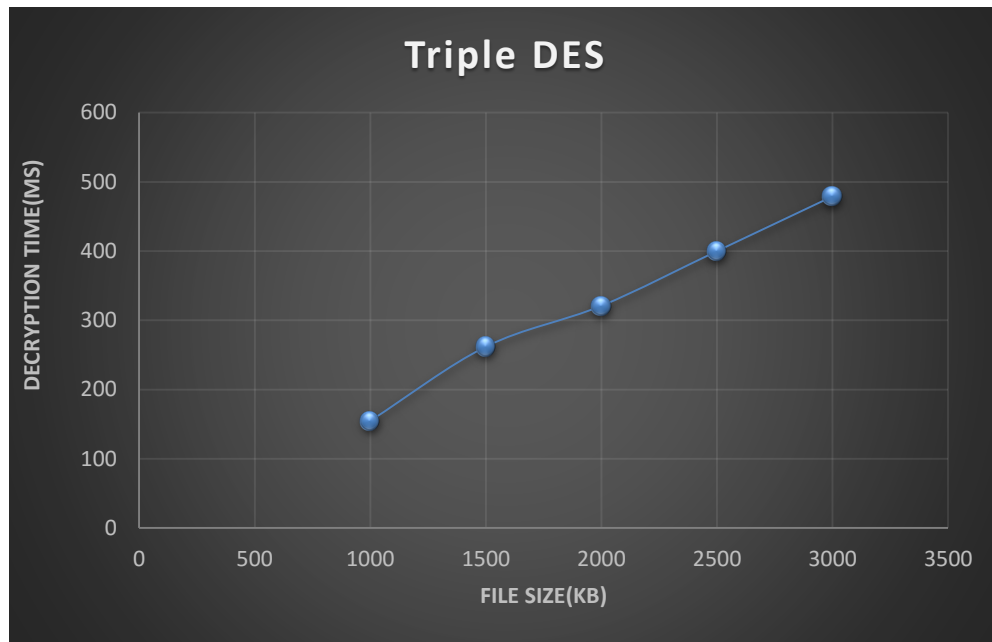


Figure 4.7.2: Graph of File Size against Decryption Time

Graph between file size and encryption throughput.

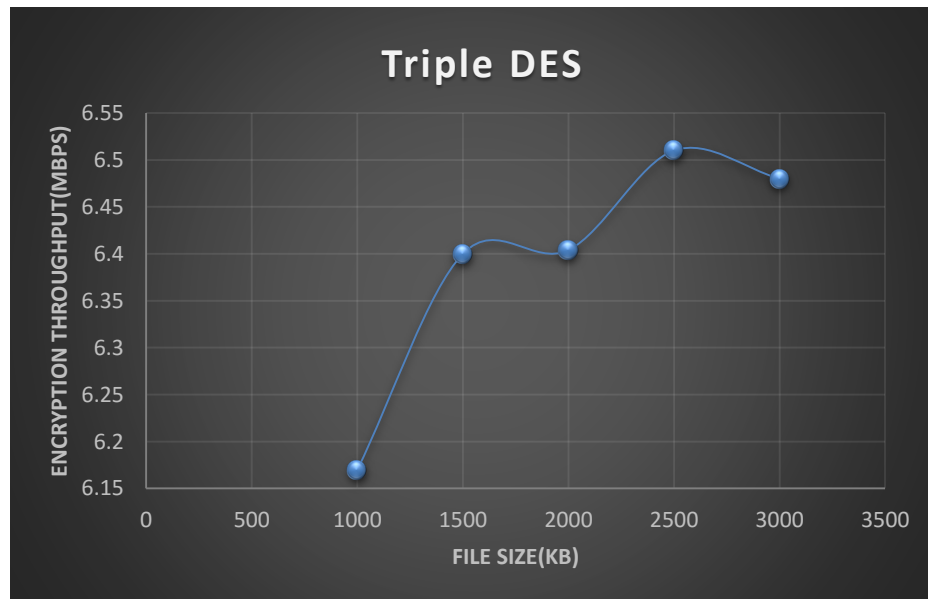


Figure 4.7.3: Graph of File Size against Encryption Throughput

Following figure shows a Graph between file Size and decryption throughput. With increasing file size, decryption throughput is moreover constant.

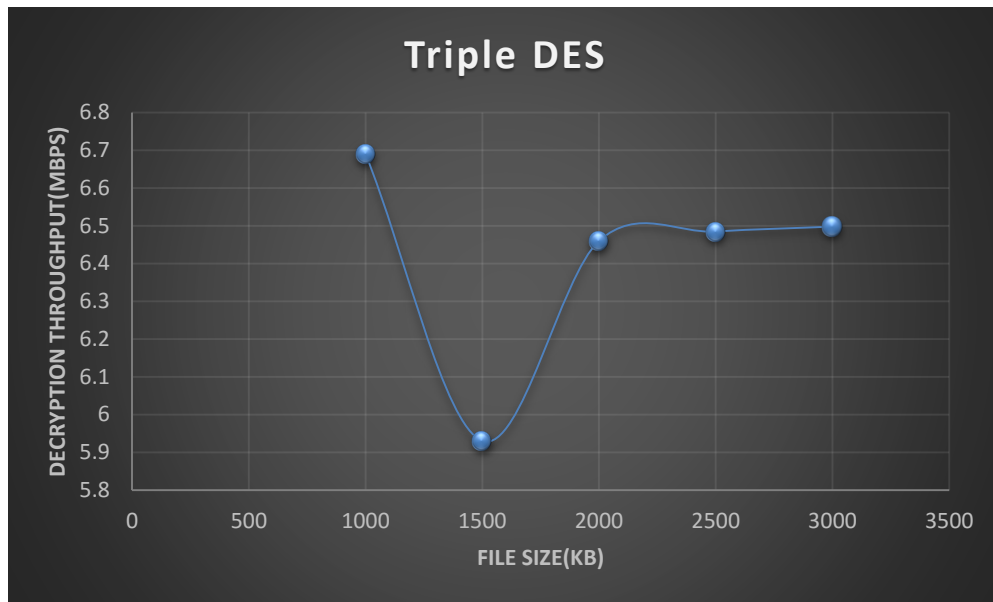


Figure 4.7.4: Graph of File Size against decryption throughput

Graph between file size and encryption memory usage

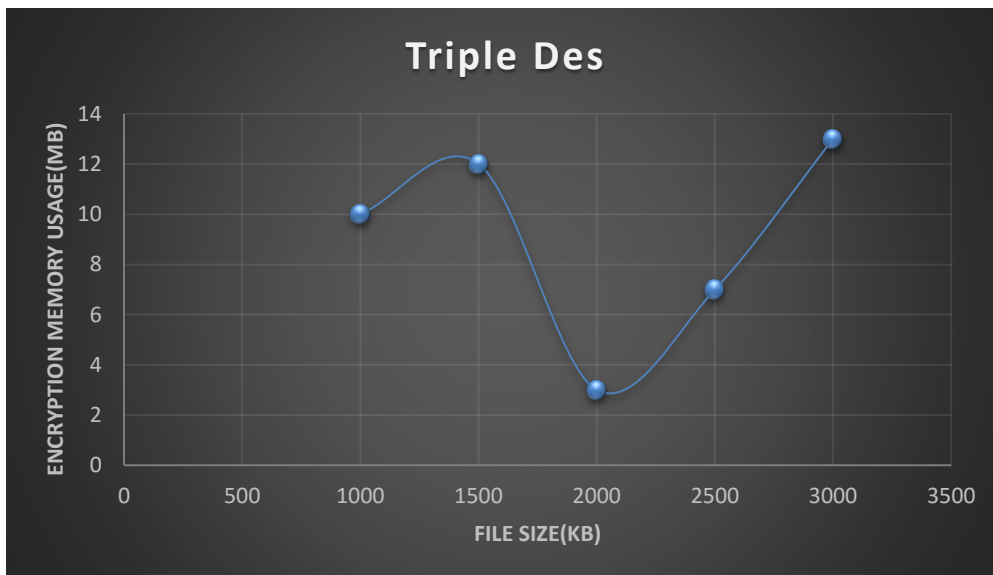


Figure 4.7.5: Graph of File Size against Encryption Memory Usage

Graph between file size and decryption memory usage

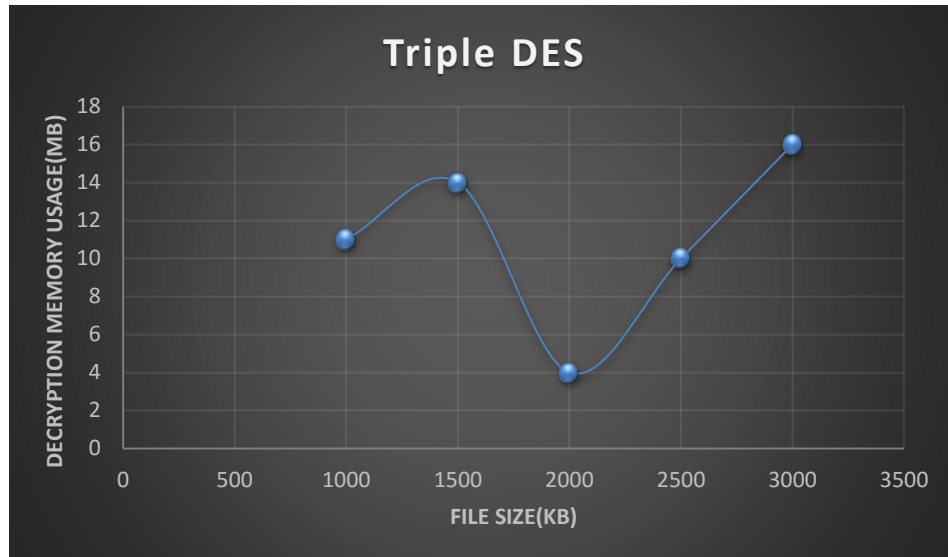


Figure 4.7.6: Graph of File Size against Decryption Memory Usage.

4.8 Performance Analysis of RC4

Following figure shows a graph between File Size and Encryption time .With increasing file Size encryption Time increases.

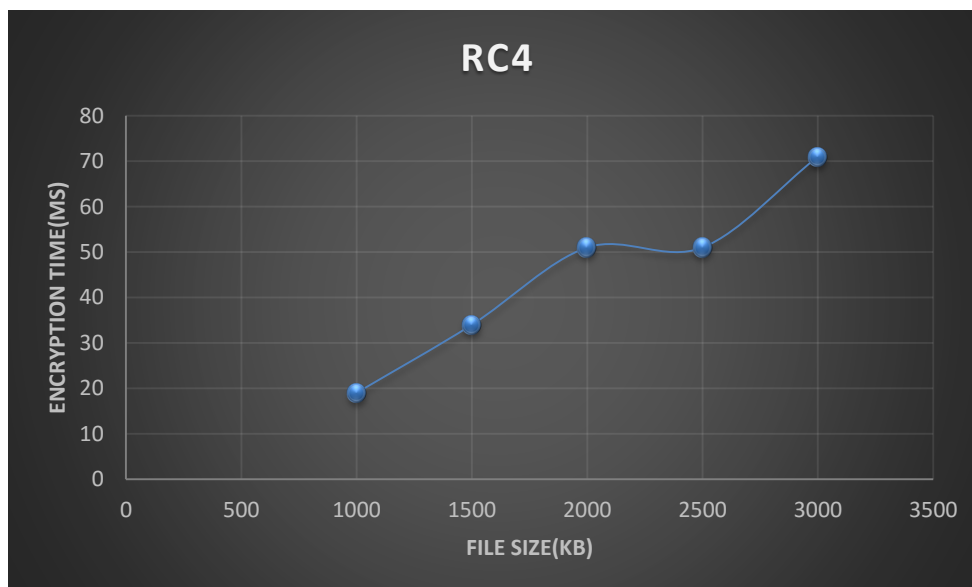


Figure 4.8.1: Graph of File Size against Encryption Time

Shows the Graph between File Size and Decryption time. With increasing file Size, decryption time increases.

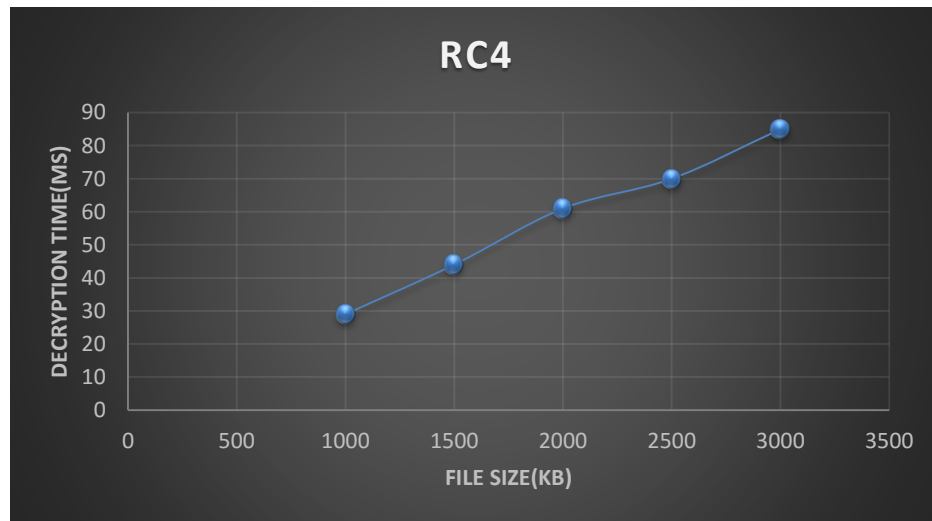


Figure 4.8.2: Graph of File Size against Decryption Time

Graph between file size and encryption throughput.

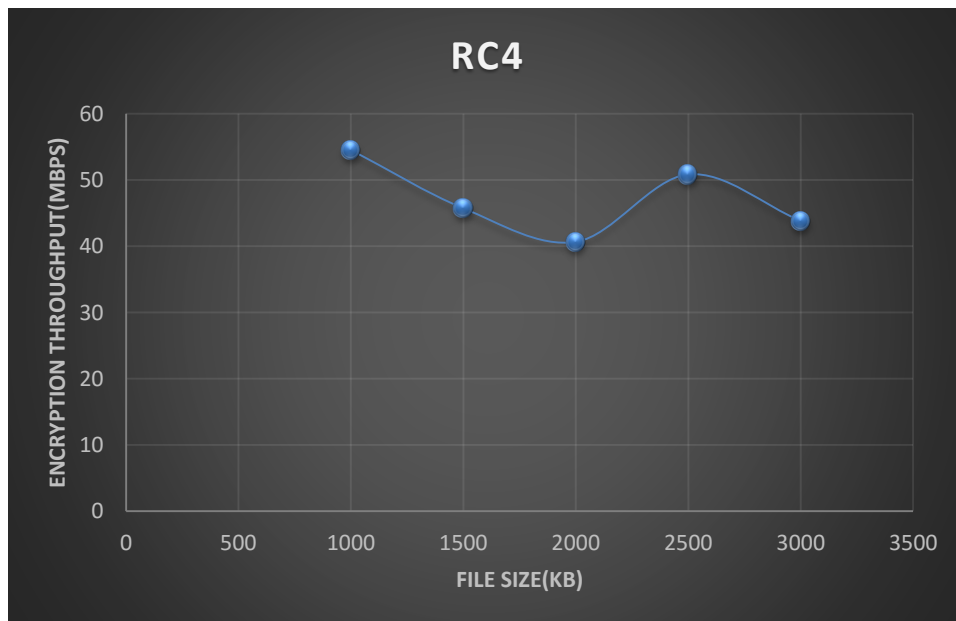


Figure 4.8.3: Graph of File Size against Encryption Throughput

Following figure shows a graph between file Size and decryption throughput. With increasing file size, decryption throughput is moreover constant.

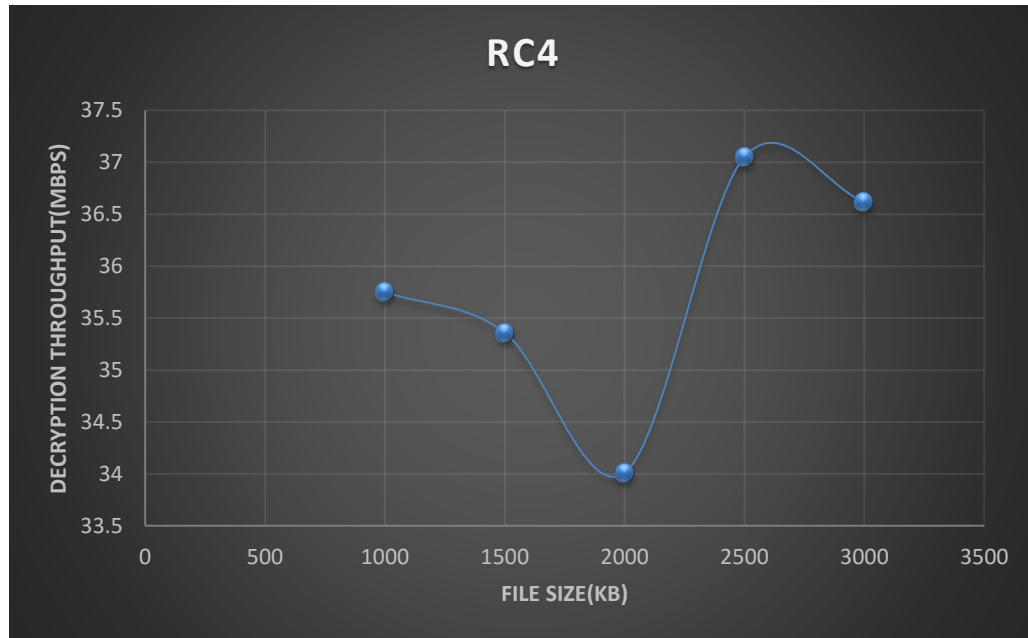


Figure 4.8.4: Graph of File Size against decryption throughput

Graph between file size and encryption memory usage.

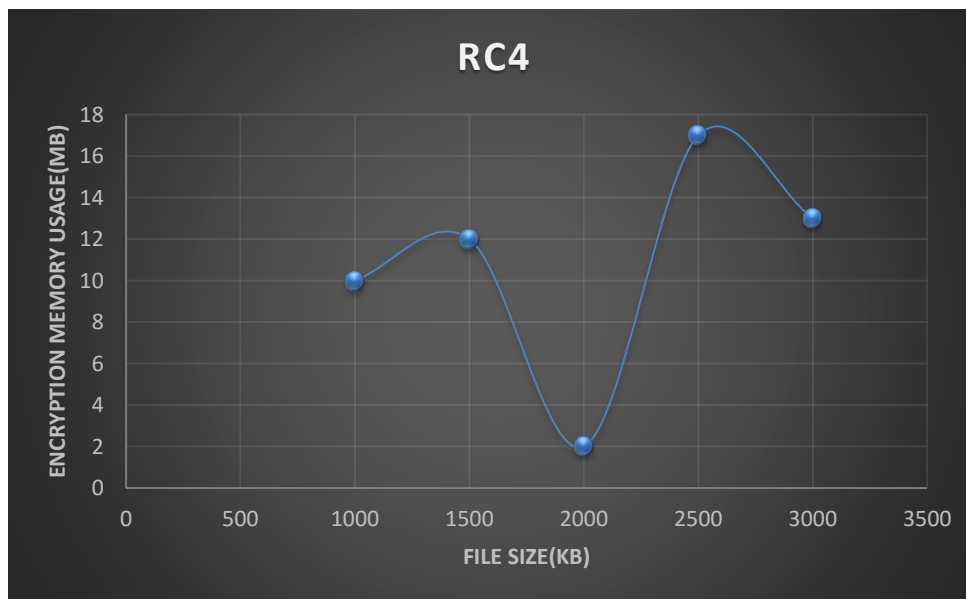


Figure 4.8.5: Graph of File Size against Encryption Memory Usage.

Graph between file size and decryption memory usage.

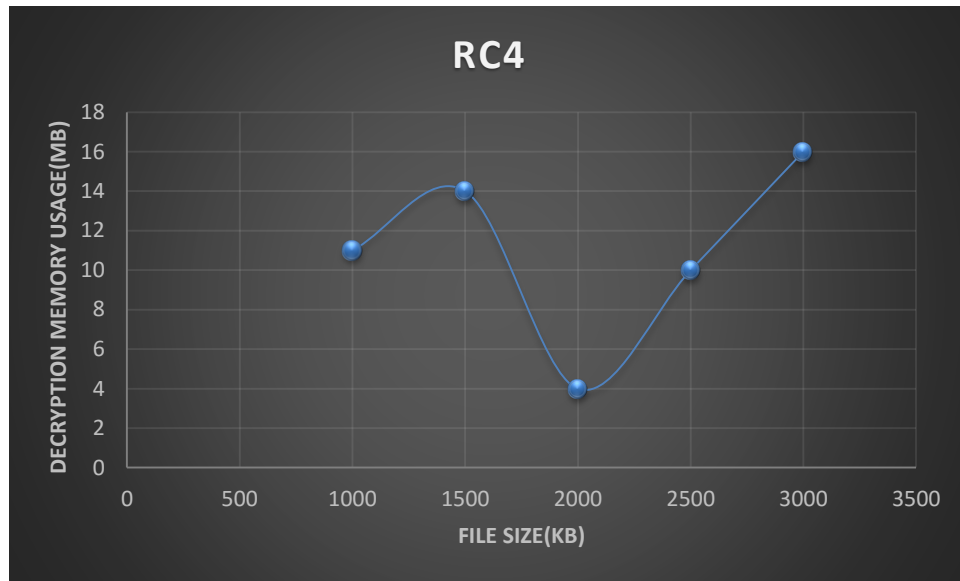


Figure 4.8.6: Graph of File Size against Decryption Memory Usage.

4.9 Performance Analysis of AES

Following figure shows a graph between File Size and Encryption time .With increasing file Size encryption Time increases

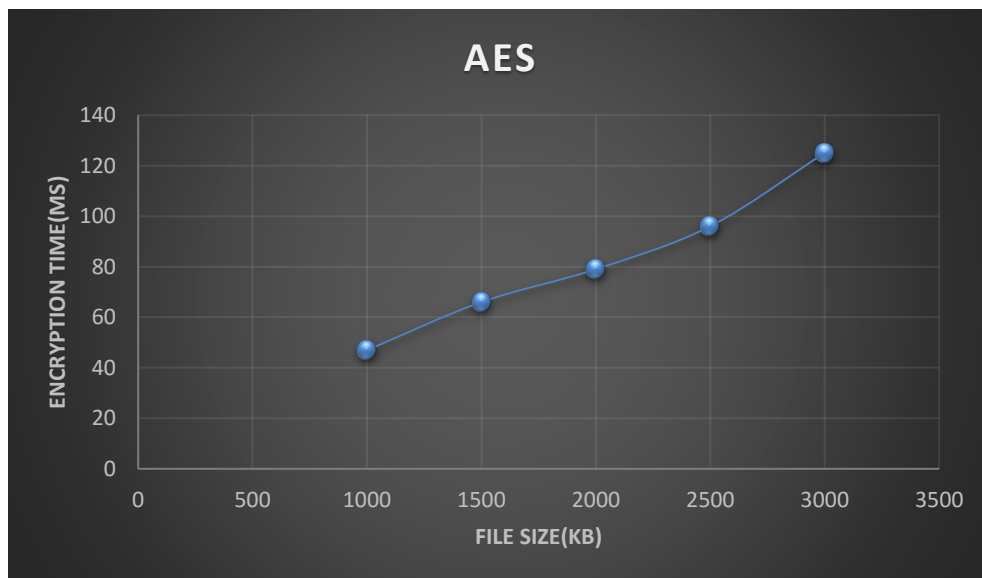


Figure 4.9.1: Graph of File Size against Encryption Time

Following figure shows a Graph between File Size and Decryption time. With increasing file Size, decryption time increases.

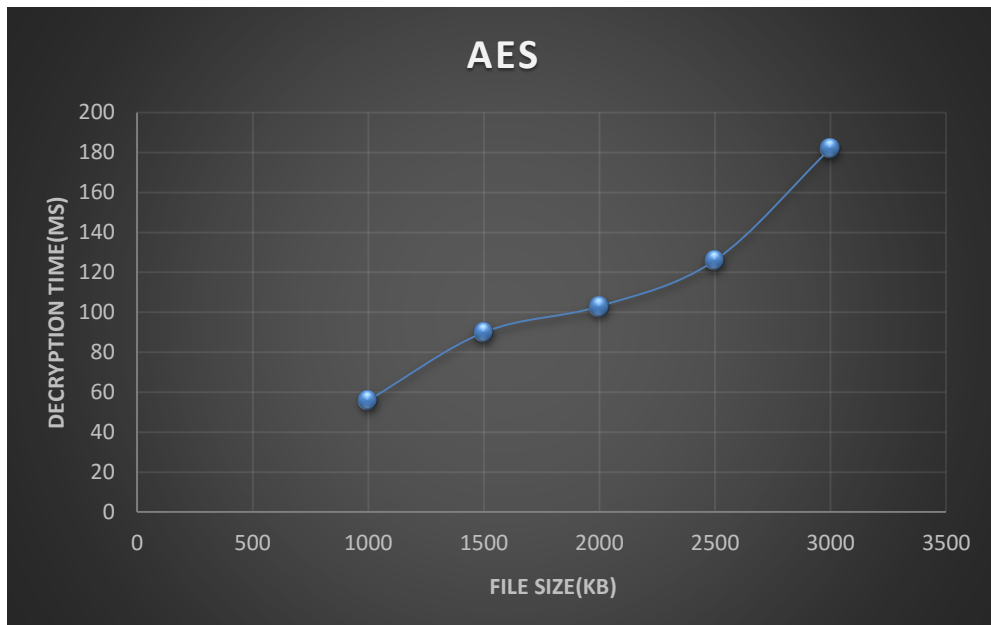


Figure 4.9.2: Graph of File Size against Decryption Time

Graph between file size and encryption throughput.

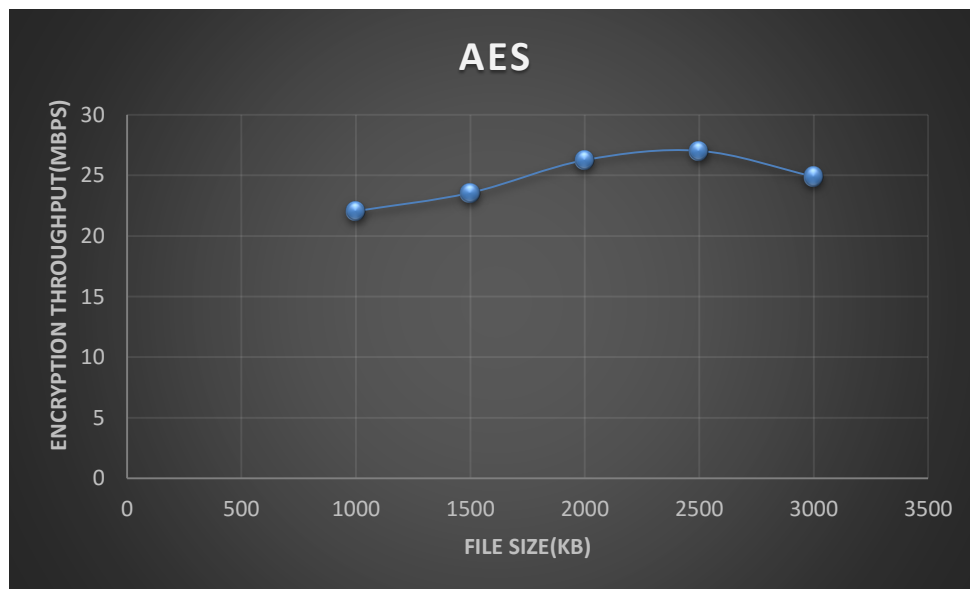


Figure 4.9.3: Graph of File Size against Encryption Throughput

Following figure shows a graph between file Size and decryption throughput. With increasing file size, decryption throughput is moreover constant.

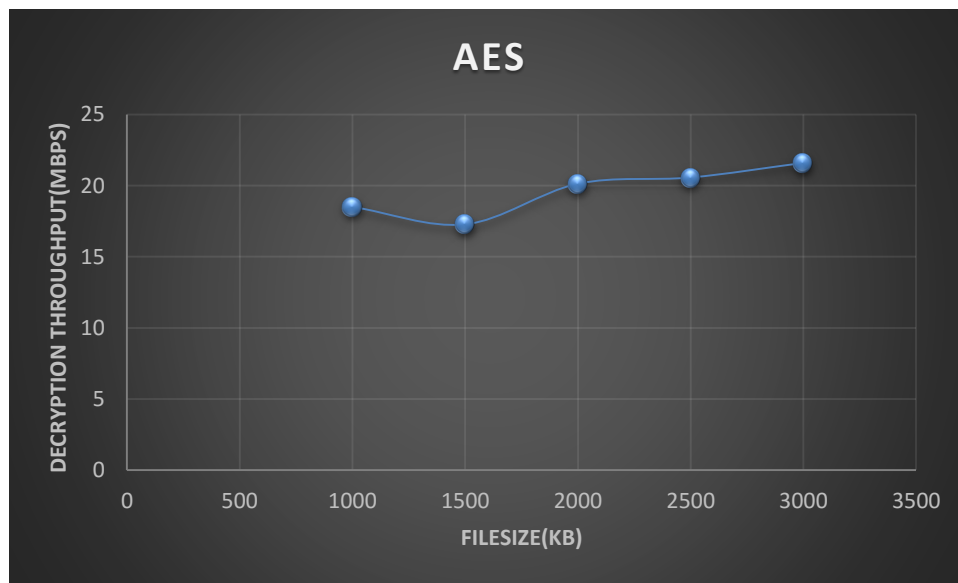


Figure 4.9.4 : Graph of File Size against decryption throughput

Graph between file size and encryption memory usage.

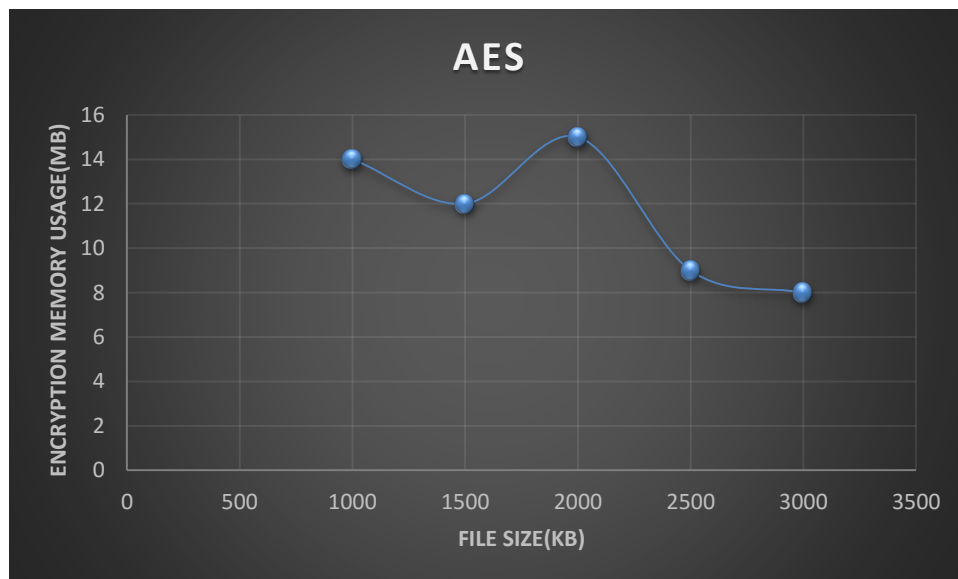


Figure 4.9.5: Graph of File Size against Encryption Memory Usage.

Graph between file size and decryption memory usage

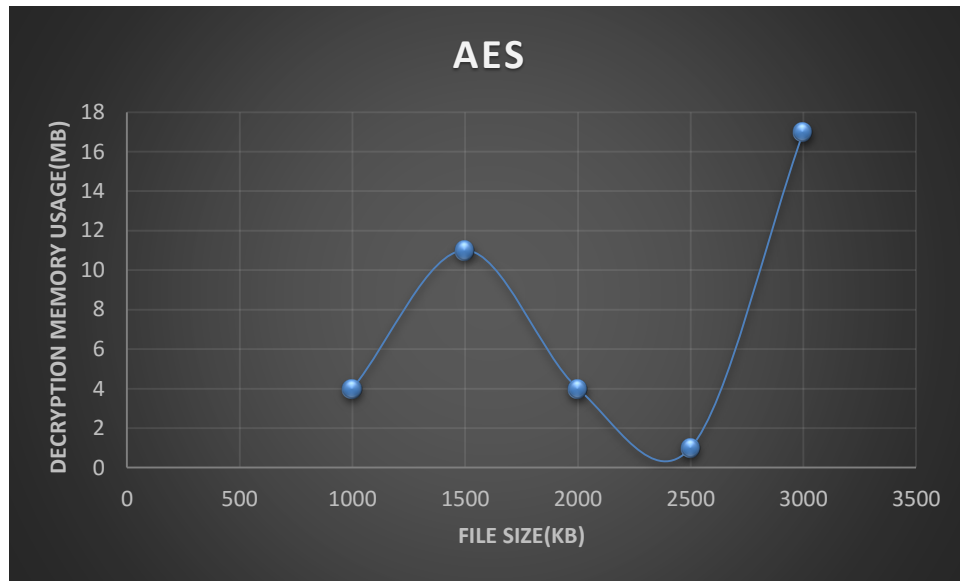


Figure 4.9.6: Graph of File Size against Decryption Memory Usage.

4.10 Performance Analysis of Blowfish

Following figure shows a graph between File Size and Encryption time .With increasing file Size encryption Time increases.

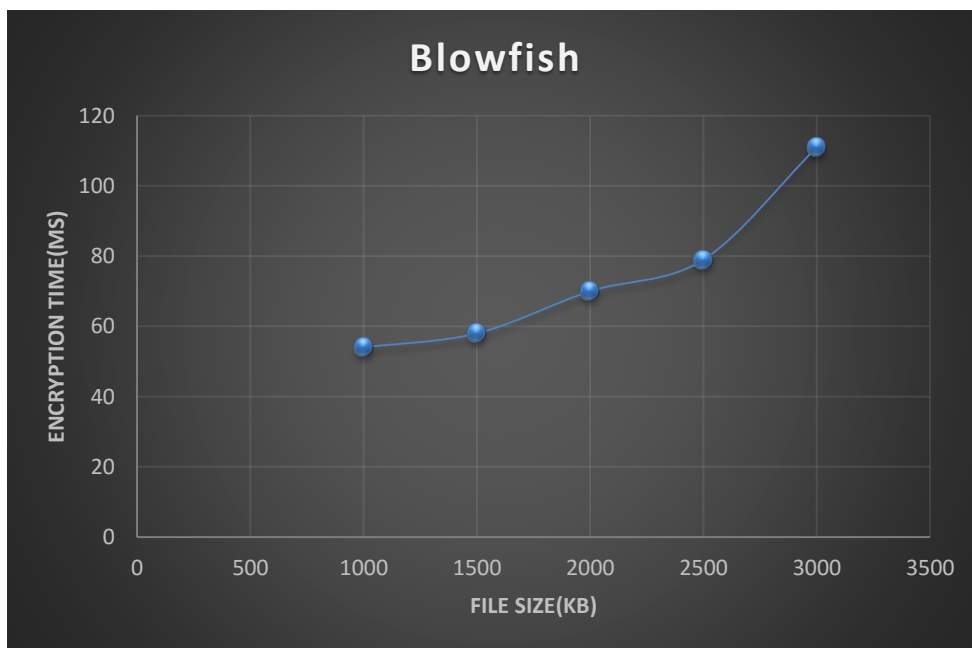


Figure 4.10.1: Graph of File Size against Encryption Time

Following figure shows a Graph between File Size and Decryption time. With increasing file Size, decryption time increases.

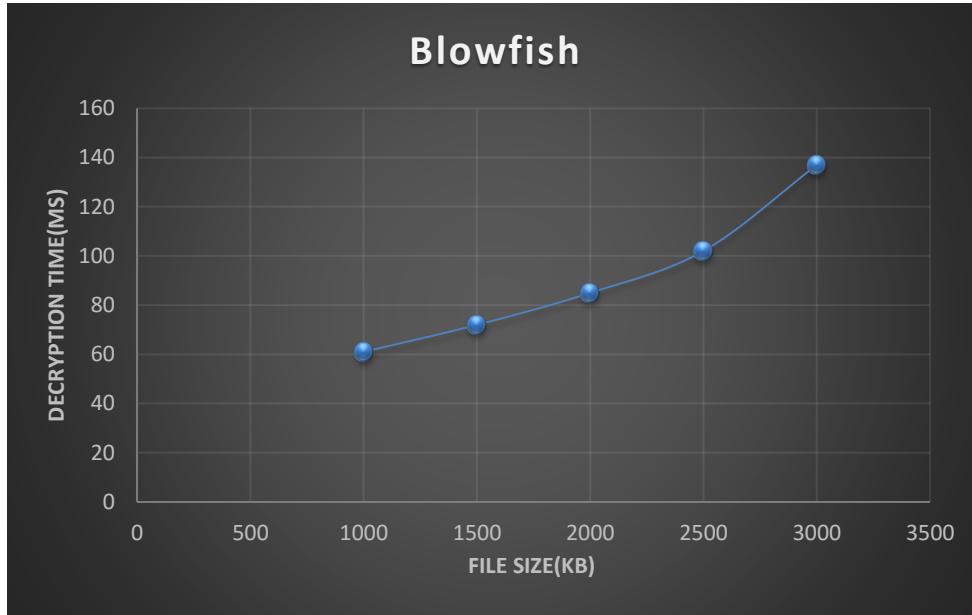


Figure 4.10.2: Graph of File Size against Decryption Time

Graph between file size and encryption throughput.

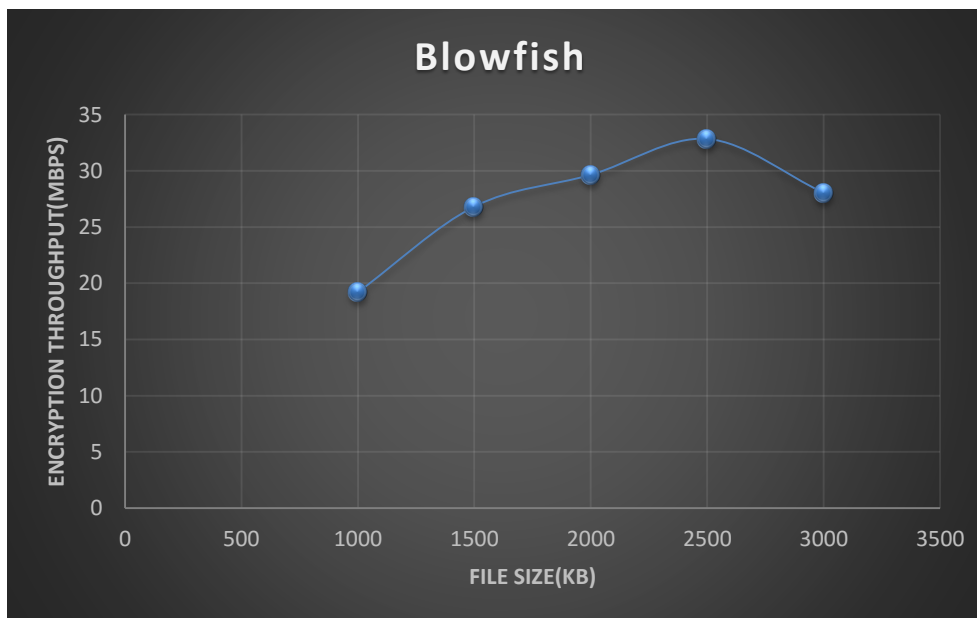


Figure 4.10.3: Graph of File Size against Encryption Throughput

Following figure shows a graph between file Size and decryption throughput. With increasing file size, decryption throughput increases.

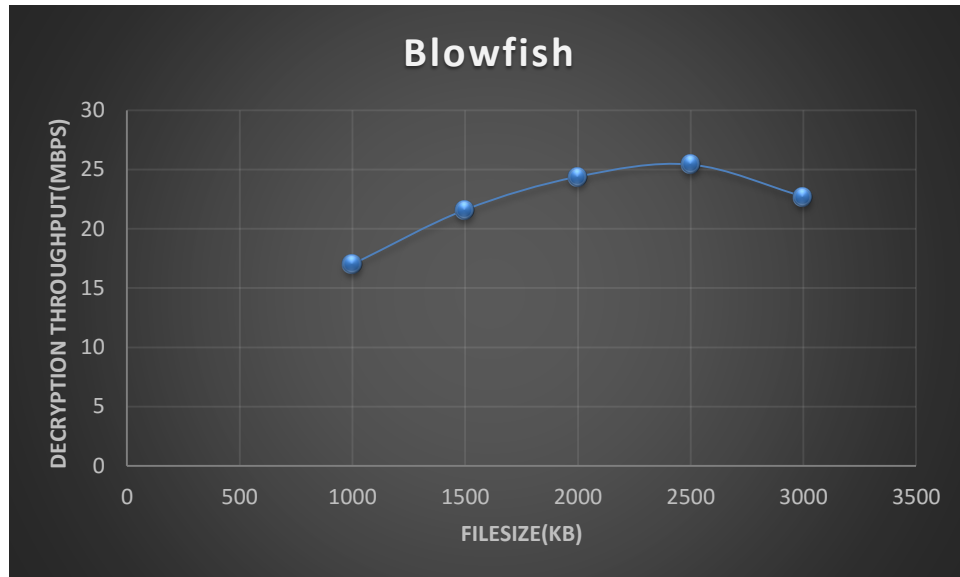


Figure 4.10.4: Graph of File Size against decryption throughput

Graph between file size and encryption memory usage.

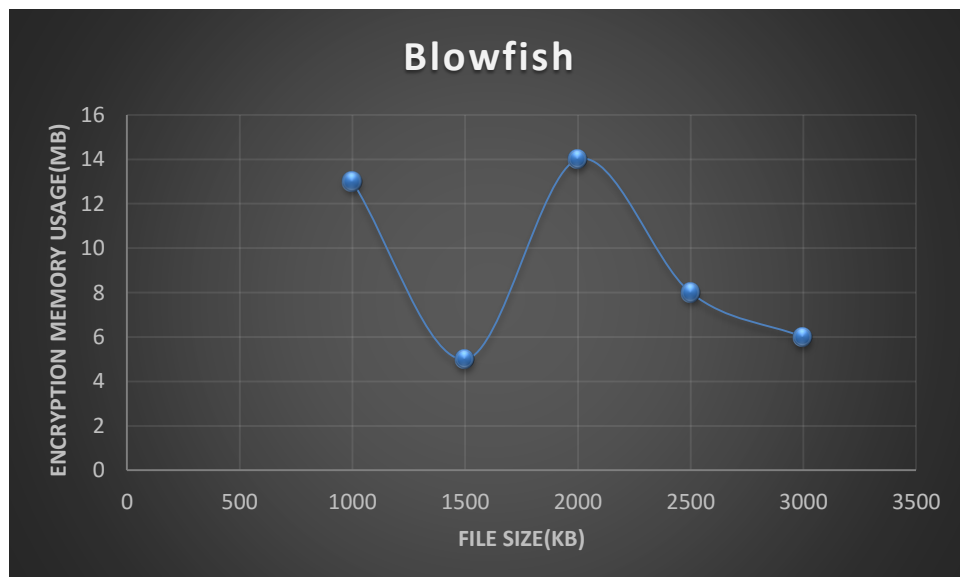


Figure 4.10.5: Graph of File Size against Encryption Memory Usage.

Graph between file size and decryption memory usage

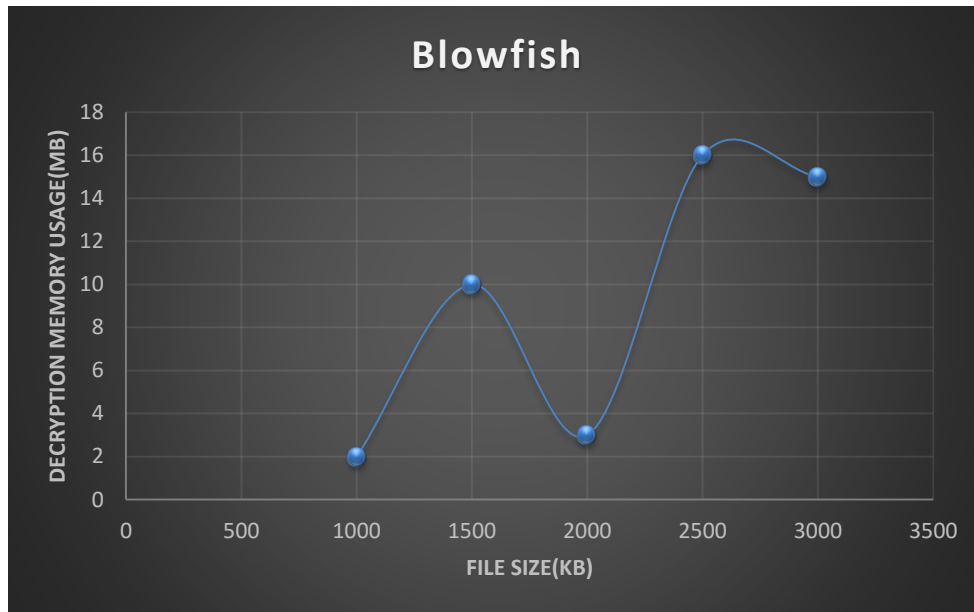


Figure 4.10.6: Graph of File Size against Decryption Memory Usage.

CHAPTER 5

System Screenshots and Result

5.1 Screenshots of DES Algorithm

Following Screenshot shows the output in which encryption is done using DES and various performance metrics like encryption time, decryption time, encryption and decryption memory Usage, encryption and decryption throughput is calculated.

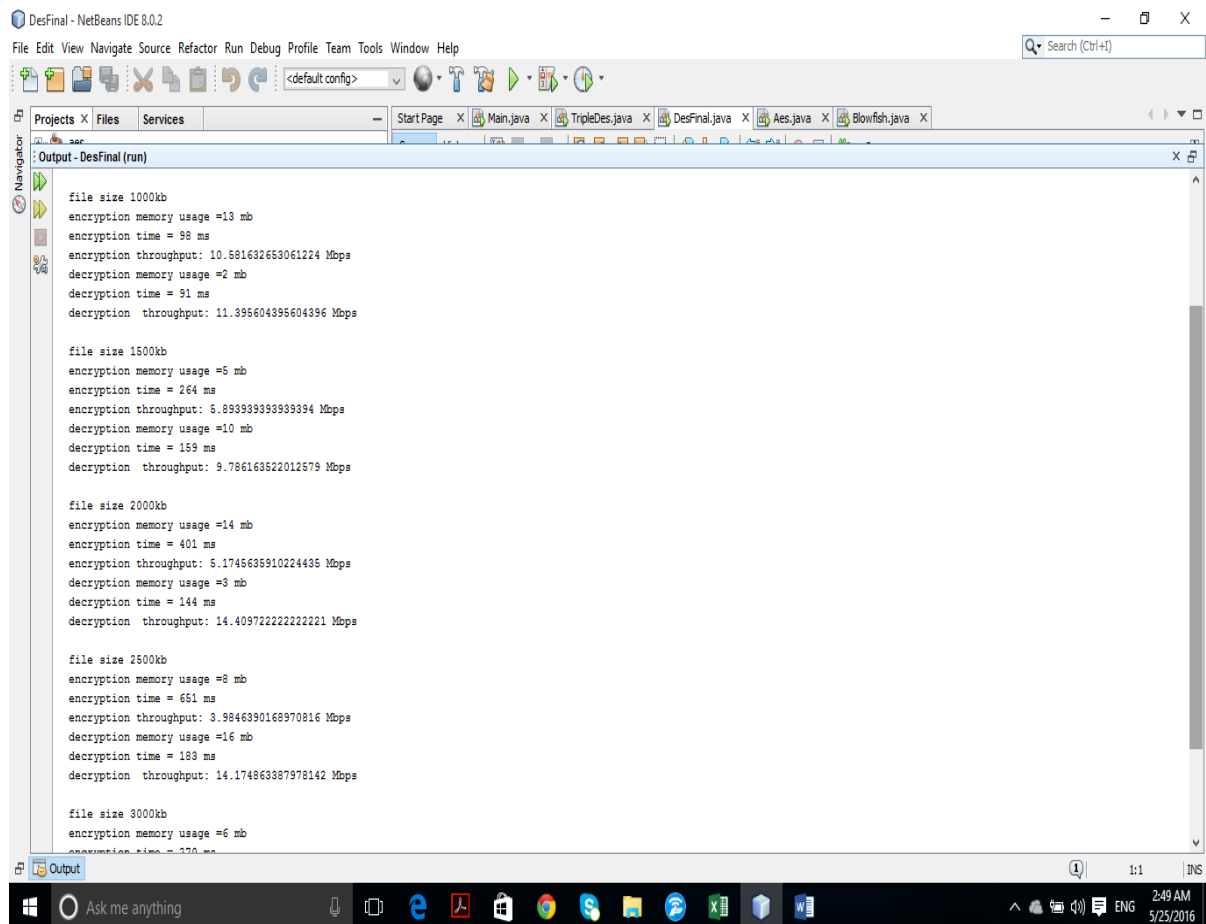


Figure 5.1.1: Screenshot of DES implementation

In following screenshot, des1 is the file to be encrypted .des1.txt is the encrypted file and des1.txt.enc is the file after decryption of des.txt. The experiment is performed on files of different file Sizes.

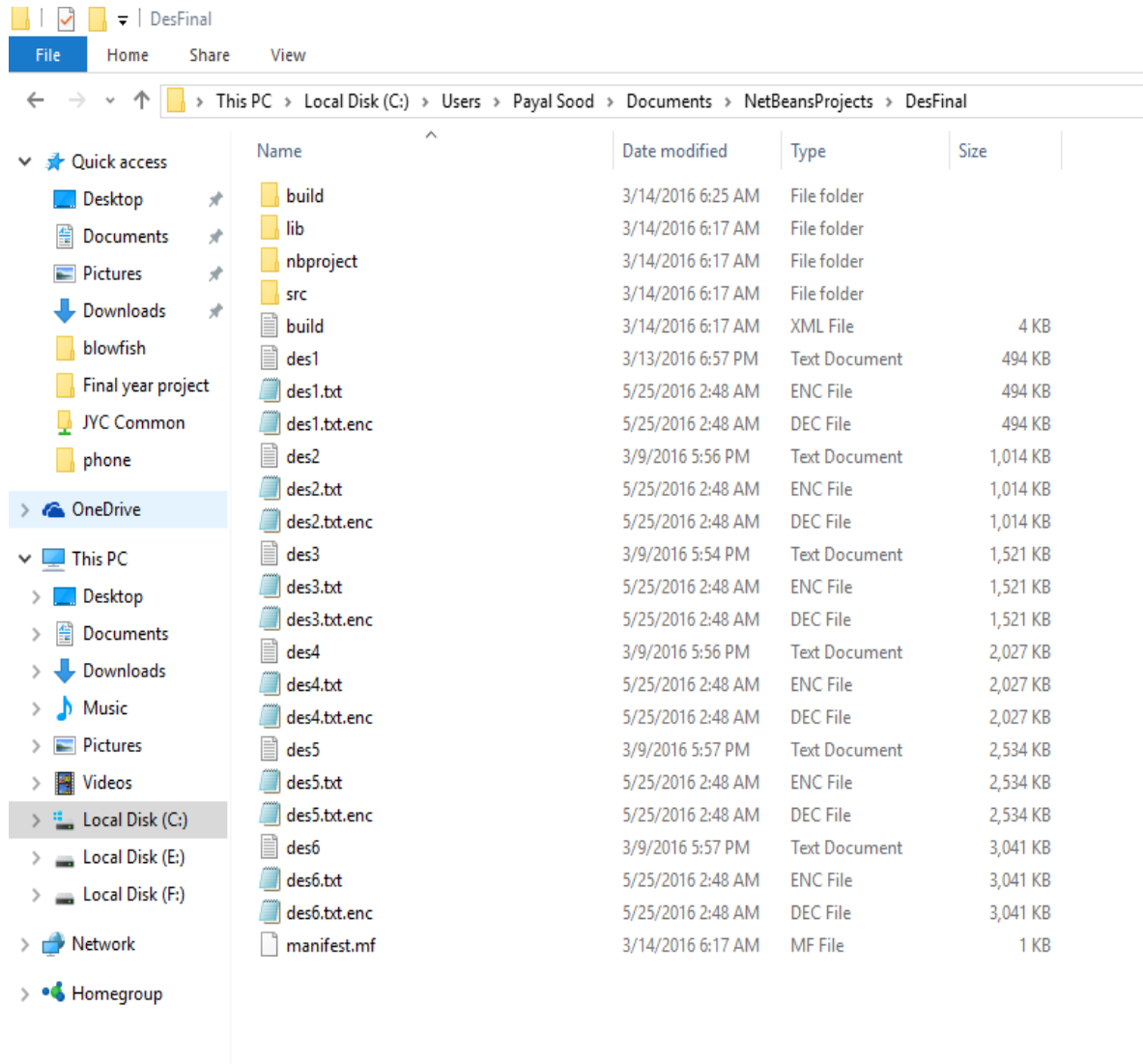
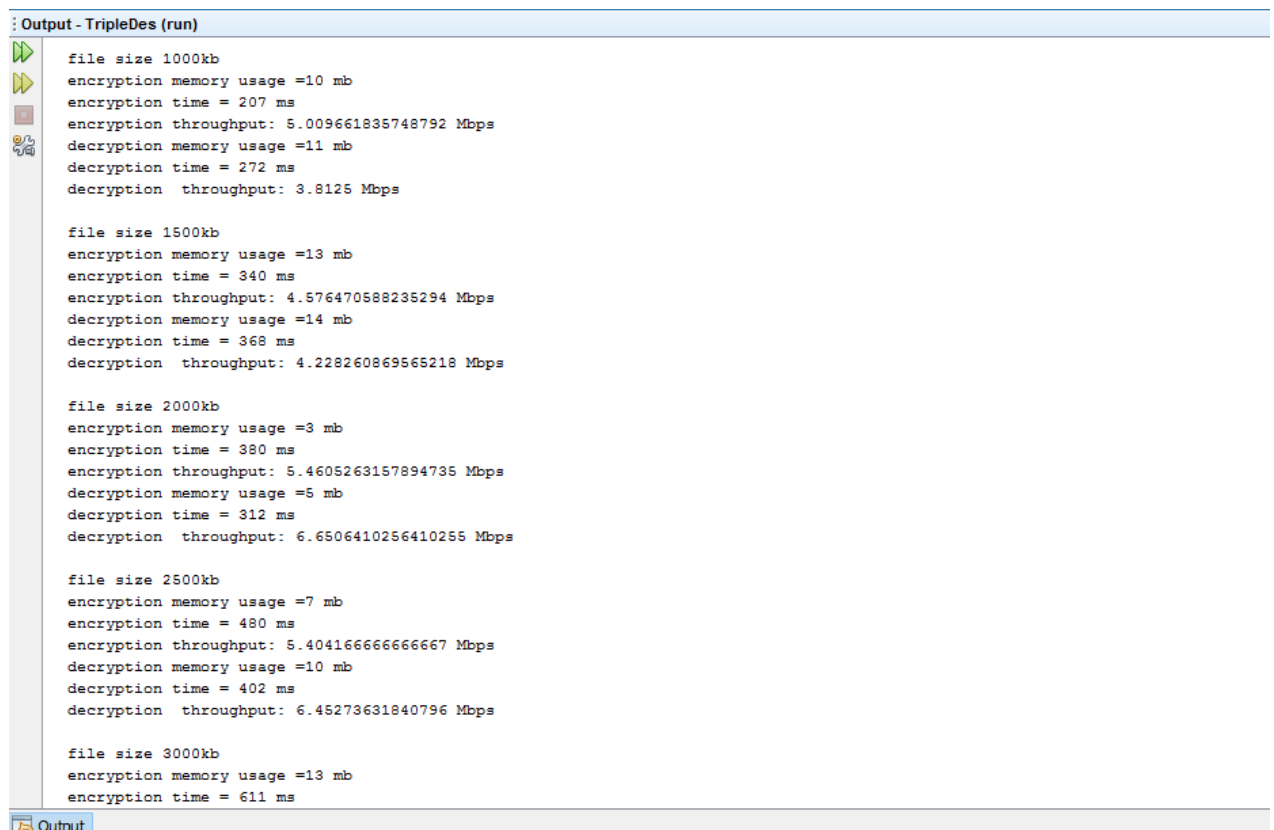


Figure 5.1.1: Screenshot of DES encrypted and decrypted files

5.2 Screenshots of TripleDES Algorithm

Following screenshot shows the output in which encryption is done using TripleDES and various performance metrics like encryption time, decryption time, encryption and decryption memory Usage, encryption and decryption throughput is calculated.



```
Output - TripleDes (run)
file size 1000kb
encryption memory usage =10 mb
encryption time = 207 ms
encryption throughput: 5.009661835748792 Mbps
decryption memory usage =11 mb
decryption time = 272 ms
decryption throughput: 3.8125 Mbps

file size 1500kb
encryption memory usage =13 mb
encryption time = 340 ms
encryption throughput: 4.576470588235294 Mbps
decryption memory usage =14 mb
decryption time = 368 ms
decryption throughput: 4.228260869565218 Mbps

file size 2000kb
encryption memory usage =3 mb
encryption time = 380 ms
encryption throughput: 5.4605263157894735 Mbps
decryption memory usage =5 mb
decryption time = 312 ms
decryption throughput: 6.6506410256410255 Mbps

file size 2500kb
encryption memory usage =7 mb
encryption time = 480 ms
encryption throughput: 5.404166666666667 Mbps
decryption memory usage =10 mb
decryption time = 402 ms
decryption throughput: 6.45273631840796 Mbps

file size 3000kb
encryption memory usage =13 mb
encryption time = 611 ms
```

Figure 5.2.1: Screenshot of Triple DES implementation

In following screenshot, tdes1 is the file to be encrypted .tdes1.txt is the encrypted file and tdes1.txt.enc is the file after decryption of tdes.txt. The experiment is performed on files of different file Sizes

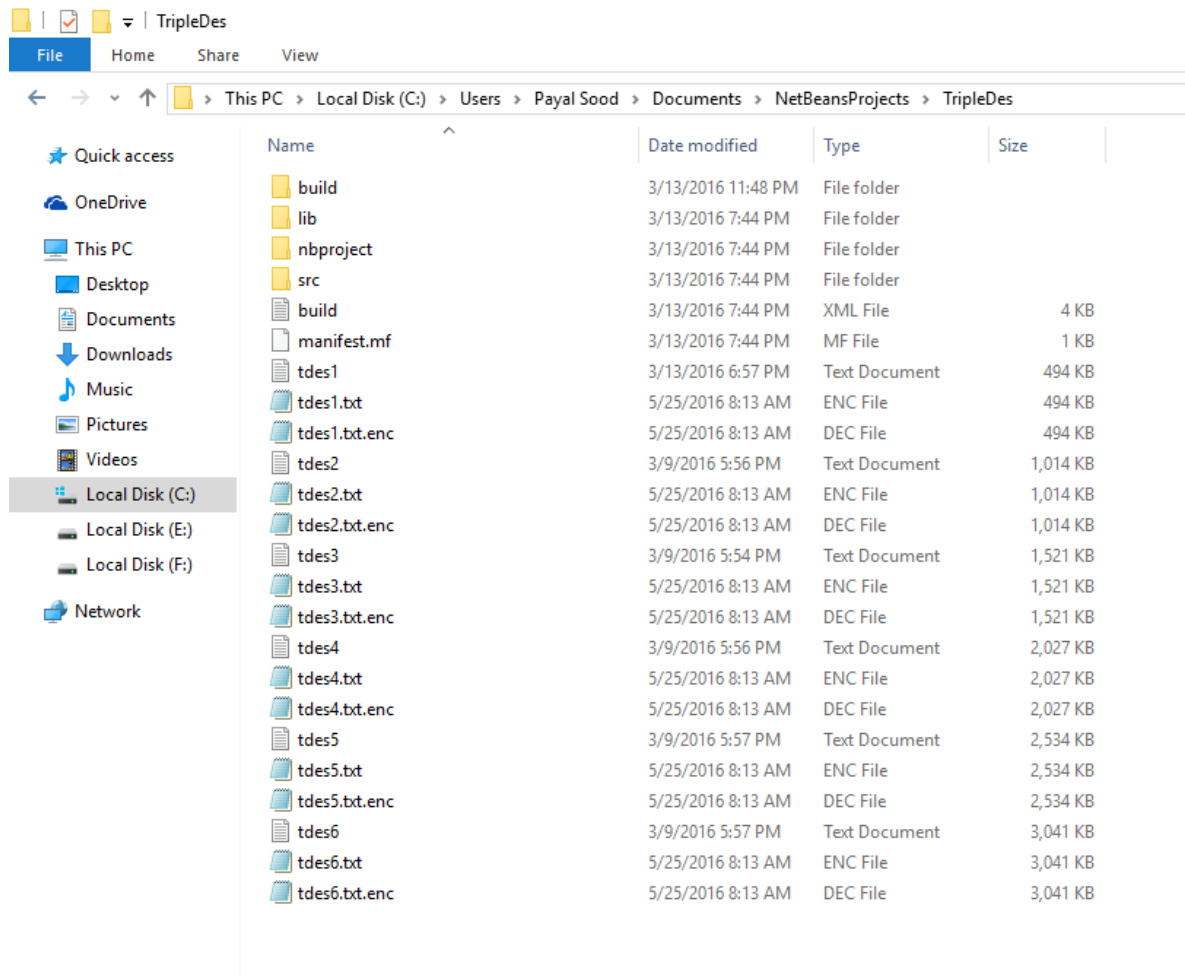
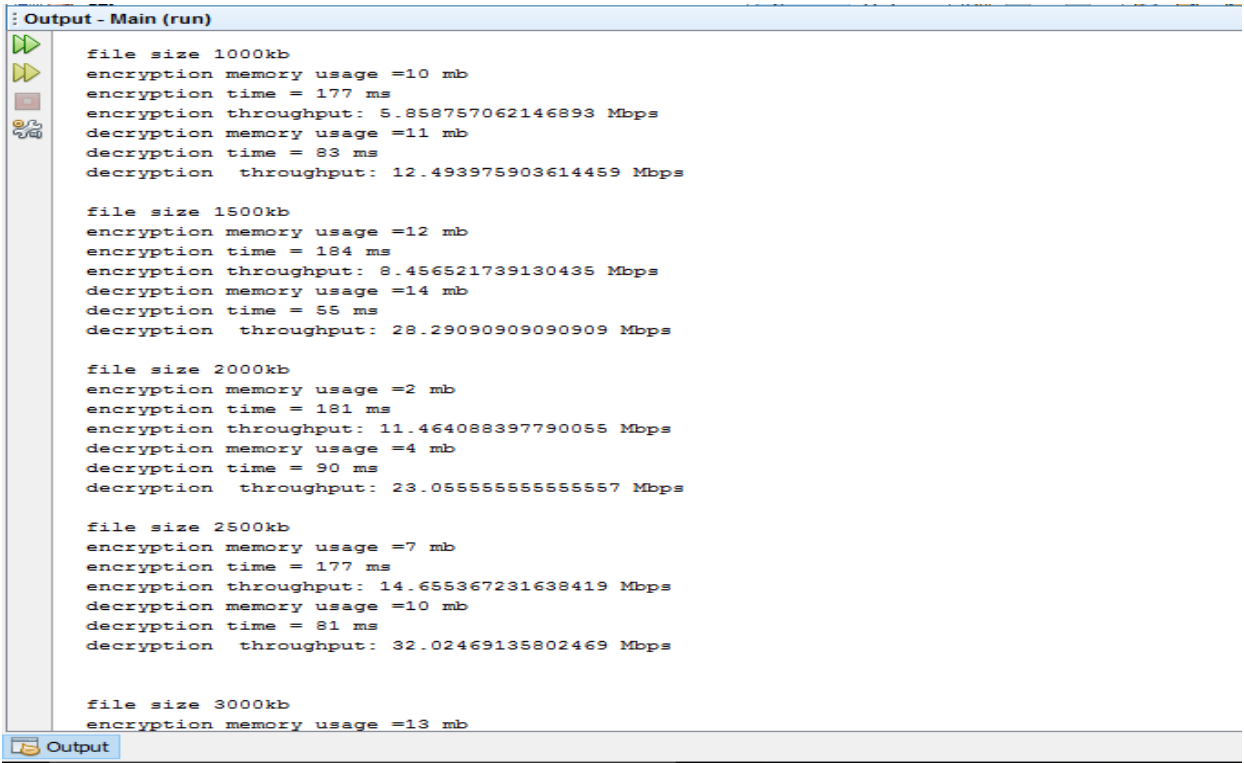


Figure 5.2.2: Screenshot of Triple DES encrypted and decrypted files.

5.3 Screenshots of RC4

Following Screenshot shows the output in which encryption is done using RC4 and various performance metrics like encryption time, decryption time, encryption and decryption memory Usage, encryption and decryption throughput is calculated.



```
Output - Main (run)
file size 1000kb
encryption memory usage =10 mb
encryption time = 177 ms
encryption throughput: 5.858757062146893 Mbps
decryption memory usage =11 mb
decryption time = 83 ms
decryption throughput: 12.493975903614459 Mbps

file size 1500kb
encryption memory usage =12 mb
encryption time = 184 ms
encryption throughput: 8.456521739130435 Mbps
decryption memory usage =14 mb
decryption time = 55 ms
decryption throughput: 28.2909090909090909 Mbps

file size 2000kb
encryption memory usage =2 mb
encryption time = 181 ms
encryption throughput: 11.464088397790055 Mbps
decryption memory usage =4 mb
decryption time = 90 ms
decryption throughput: 23.055555555555557 Mbps

file size 2500kb
encryption memory usage =7 mb
encryption time = 177 ms
encryption throughput: 14.655367231638419 Mbps
decryption memory usage =10 mb
decryption time = 81 ms
decryption throughput: 32.02469135802469 Mbps

file size 3000kb
encryption memory usage =13 mb
```

Figure 5.3.1: Screenshot of RC4 implementation

In following screenshot, test1 is the file to be encrypted .test1.txt is the encrypted file and test1.txt.enc is the file after decryption of test.txt. The experiment is performed on files of different file Sizes.

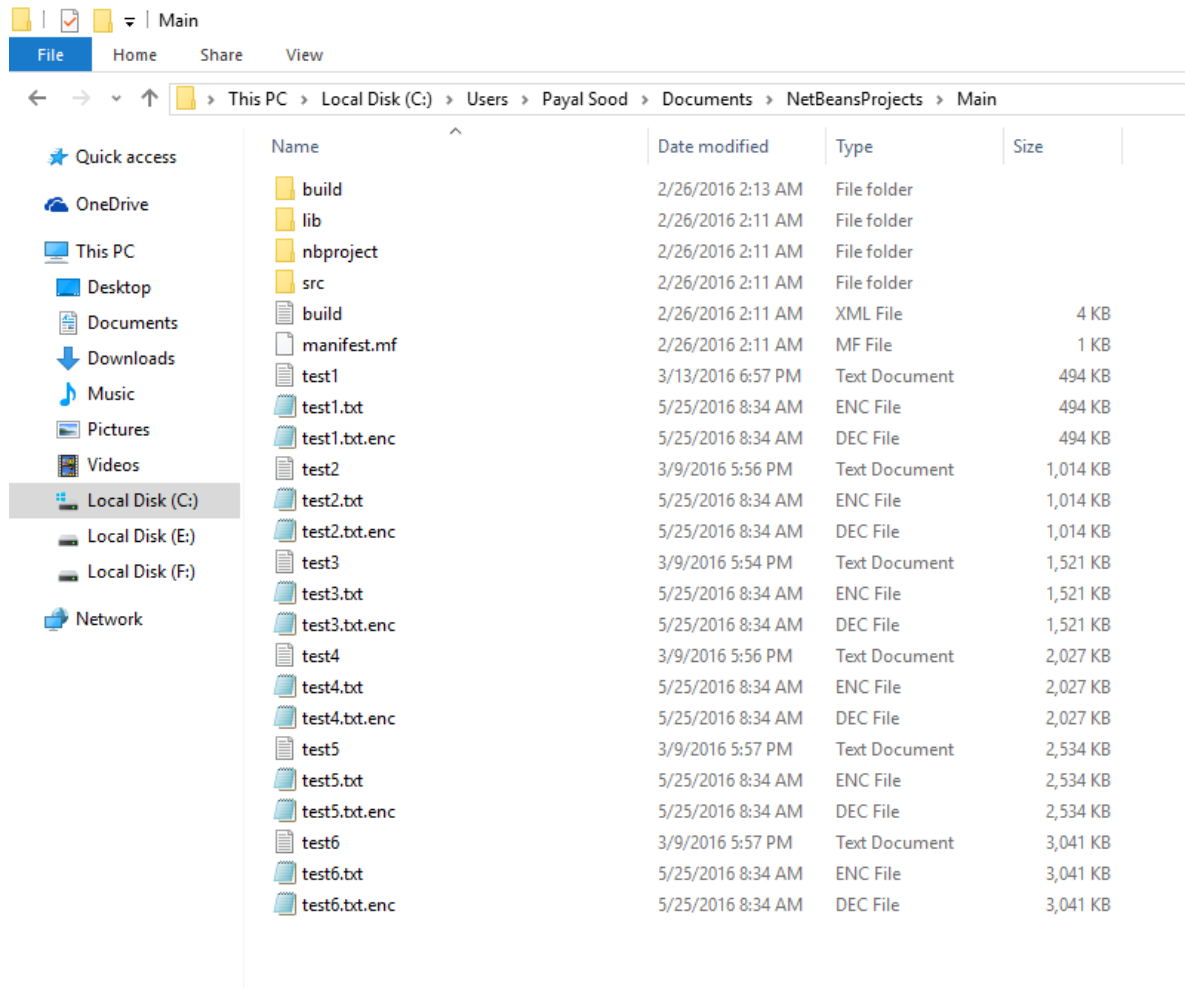
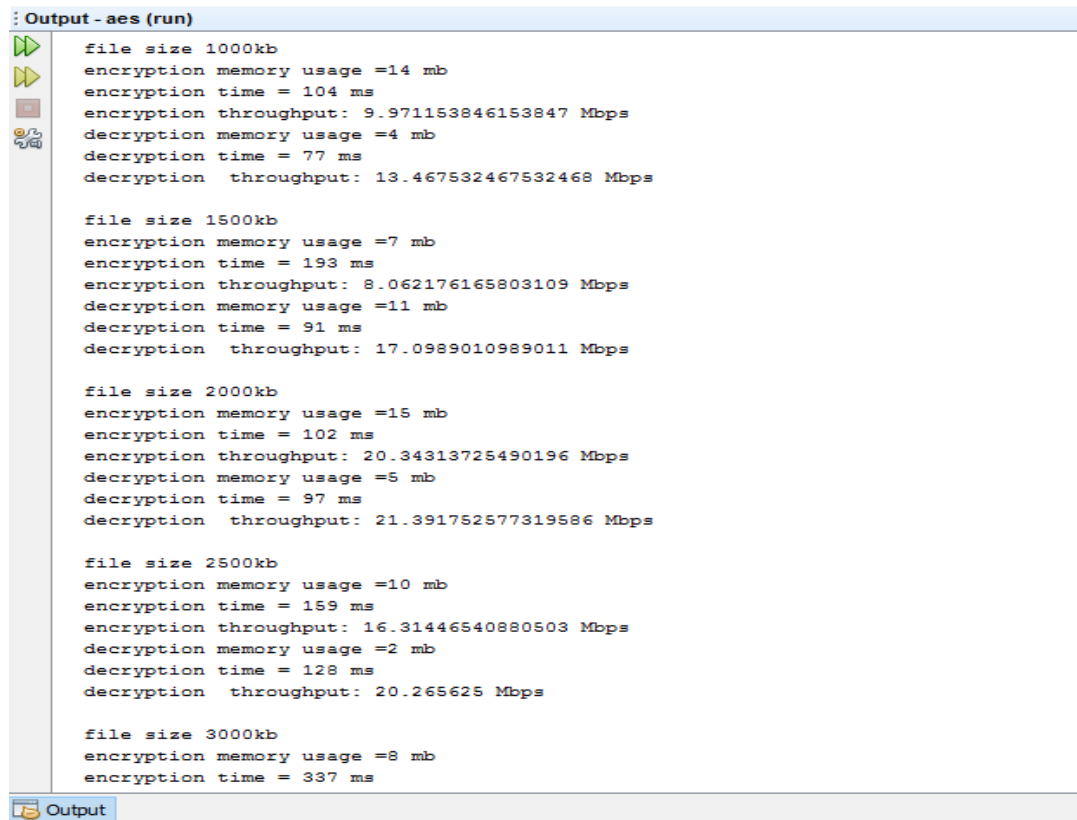


Figure 5.3.2: Screenshot of RC4 encrypted and decrypted files

5.4 Screenshots of AES

Following Screenshot shows the output in which encryption is done using AES and various performance metrics like encryption time, decryption time, encryption and decryption memory Usage, encryption and decryption throughput is calculated.



```
Output - aes (run)
file size 1000kb
encryption memory usage =14 mb
encryption time = 104 ms
encryption throughput: 9.971153846153847 Mbps
decryption memory usage =4 mb
decryption time = 77 ms
decryption throughput: 13.467532467532468 Mbps

file size 1500kb
encryption memory usage =7 mb
encryption time = 193 ms
encryption throughput: 8.062176165803109 Mbps
decryption memory usage =11 mb
decryption time = 91 ms
decryption throughput: 17.0989010989011 Mbps

file size 2000kb
encryption memory usage =15 mb
encryption time = 102 ms
encryption throughput: 20.34313725490196 Mbps
decryption memory usage =5 mb
decryption time = 97 ms
decryption throughput: 21.391752577319586 Mbps

file size 2500kb
encryption memory usage =10 mb
encryption time = 159 ms
encryption throughput: 16.31446540880503 Mbps
decryption memory usage =2 mb
decryption time = 128 ms
decryption throughput: 20.265625 Mbps

file size 3000kb
encryption memory usage =8 mb
encryption time = 337 ms
```

Figure 5.4.1: Screenshot of AES implementation

In following screenshot, aes1 is the file to be encrypted .aes1.txt is the encrypted file and aes1.txt.enc is the file after decryption of aes.txt. The experiment is performed on files of different file Sizes.

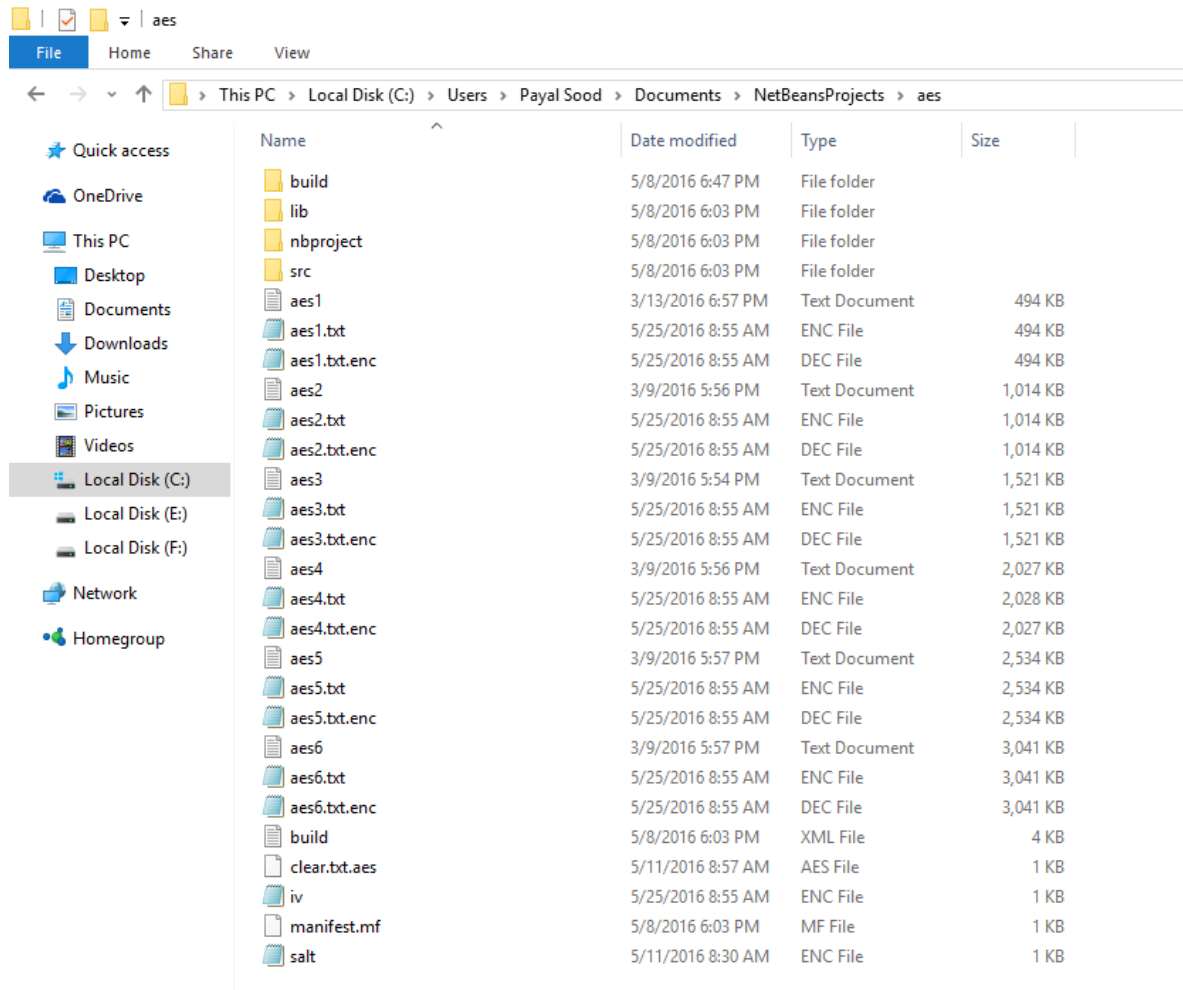
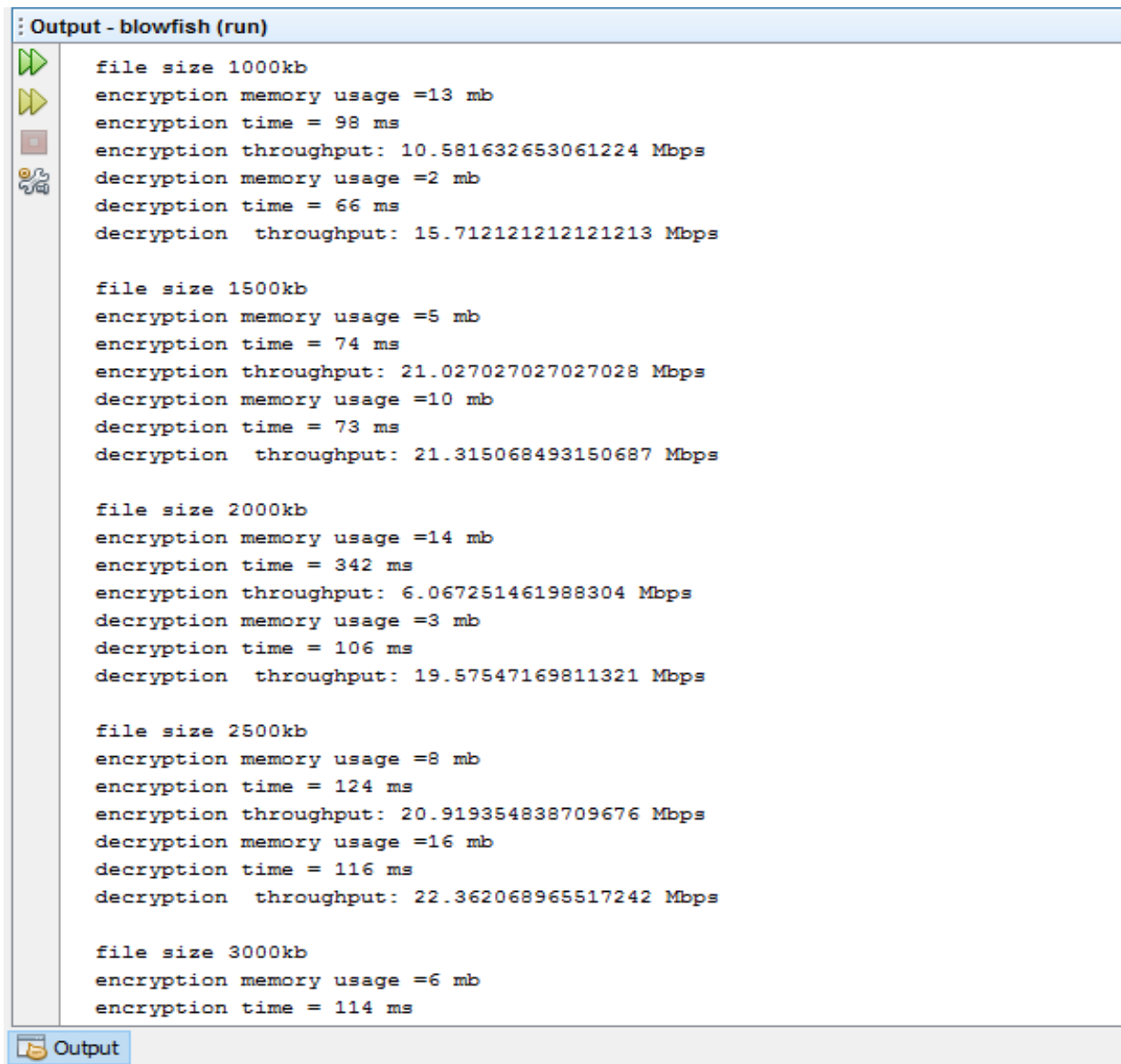


Figure 5.4.2: Screenshot of AES encrypted and decrypted files

5.5 Screenshots of Blowfish

Above Screenshot shows the output in which encryption is done using Blowfish algorithm and various performance metrics like encryption time, decryption time, encryption and decryption memory Usage, encryption and decryption throughput is calculated.



```
Output - blowfish (run)
file size 1000kb
encryption memory usage =13 mb
encryption time = 98 ms
encryption throughput: 10.581632653061224 Mbps
decryption memory usage =2 mb
decryption time = 66 ms
decryption throughput: 15.712121212121213 Mbps

file size 1500kb
encryption memory usage =5 mb
encryption time = 74 ms
encryption throughput: 21.027027027027028 Mbps
decryption memory usage =10 mb
decryption time = 73 ms
decryption throughput: 21.315068493150687 Mbps

file size 2000kb
encryption memory usage =14 mb
encryption time = 342 ms
encryption throughput: 6.067251461988304 Mbps
decryption memory usage =3 mb
decryption time = 106 ms
decryption throughput: 19.57547169811321 Mbps

file size 2500kb
encryption memory usage =8 mb
encryption time = 124 ms
encryption throughput: 20.919354838709676 Mbps
decryption memory usage =16 mb
decryption time = 116 ms
decryption throughput: 22.362068965517242 Mbps

file size 3000kb
encryption memory usage =6 mb
encryption time = 114 ms
```

Figure 39: Screenshot of Blowfish implementation

In following screenshot, blowfish1 is the file to be encrypted .blowfish1.txt is the encrypted file and blowfish1.txt.enc is the file after decryption of blowfish.txt. The experiment is performed on files of different file Sizes

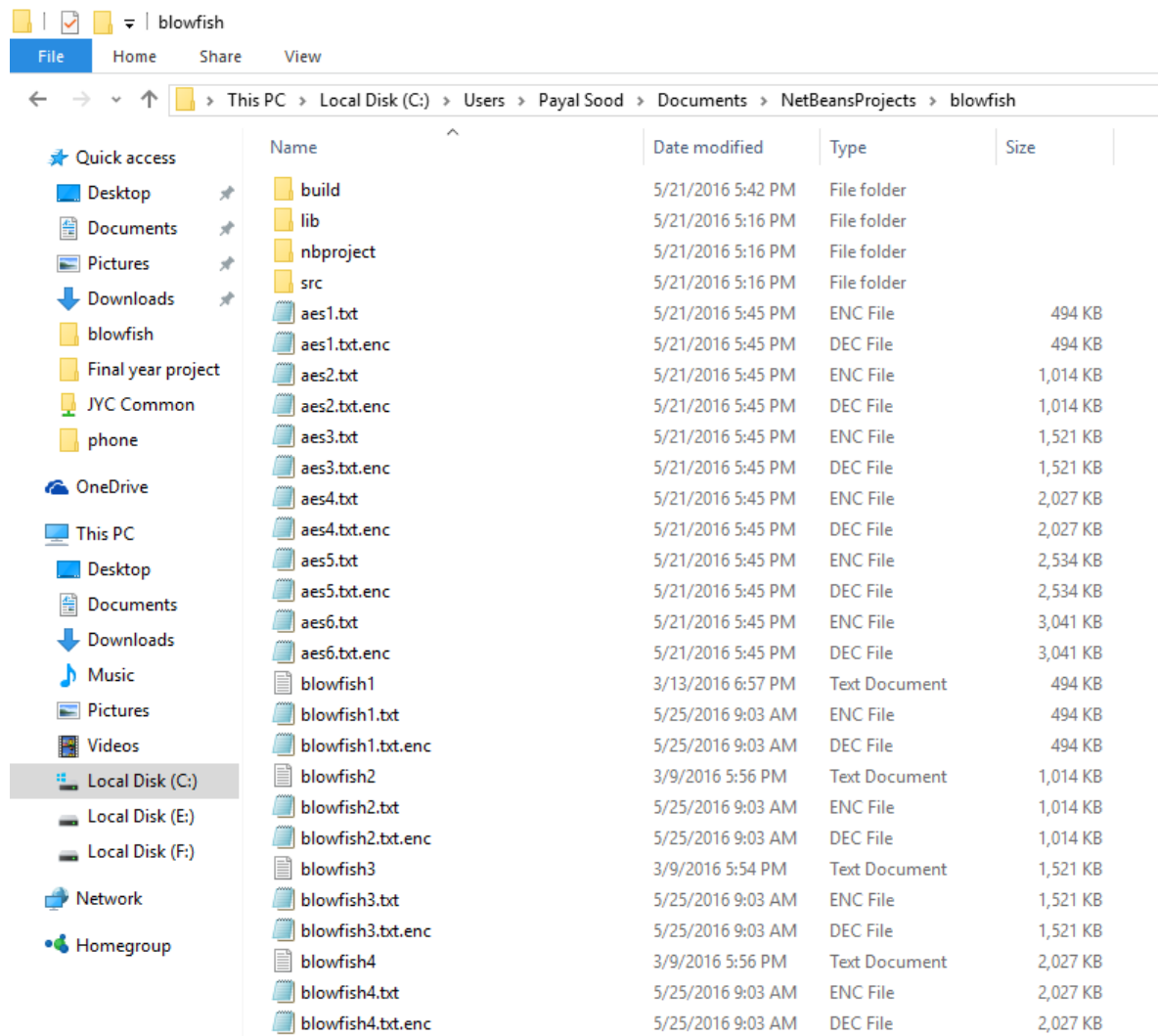


Figure 5.5.2: Screenshot of Triple DES encrypted and decrypted files

5.6 Result

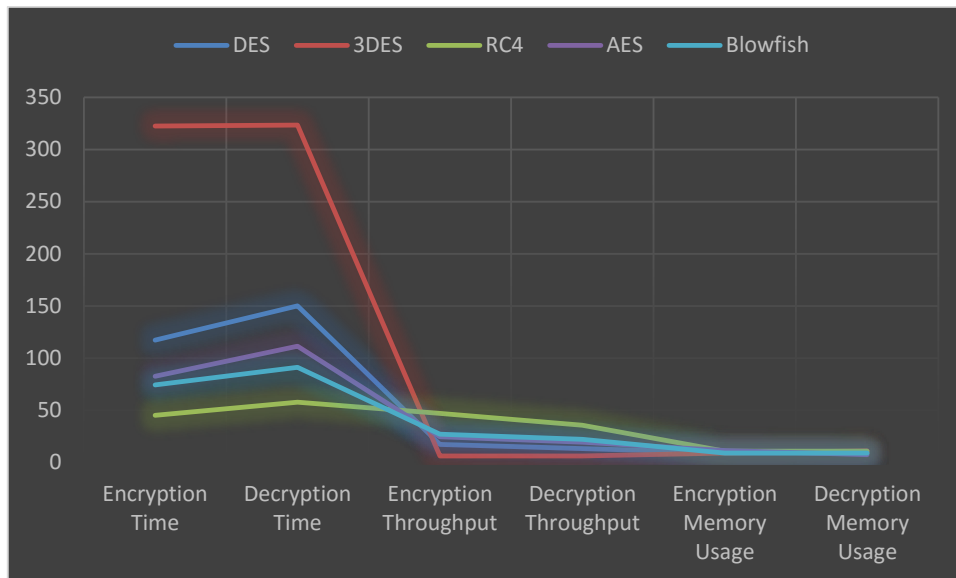


Figure 5.6.1: Comparative analysis of all the algorithms.

According to the Analysis done, RC4 being stream cipher is fastest among other symmetric encryption technique. Other than that blowfish is fast, easy and compact algorithm than other block cipher encryption techniques. AES being government Standard is widely used providing adequate amount of security. Triple DES is slowest among all the other encryption standards.

CHAPTER 6

Conclusion & Future Directions

Internet is mainly used by Individuals, Co-operatives and Governments. They have send information through internet. But there is a possibility to hack the information. So to protect information, we need to encrypt/decrypt information by using cryptography algorithms. In this work the existing encryption techniques are studied and analysed to promote the performance of the encryption methods also to ensure the security proceedings. To sum up, all techniques are unique in its own way, which might be suitable for different applications. Everyday new encryption technique is evolving hence fast and secure conventional encryption techniques will always work out with high rate of security.

The research revealed that; encryption only depends upon the number of bytes present in the file. It also revealed that encryption time and data size is proportional to each other. As the size of data increase the encryption time also increase proportional to data size and vice versa. I have studied many papers on cryptography. Some papers were very good and effective and can be used for future work.

DES is the old "data encryption standard" from the seventies. DES is the most widely used encryption scheme, especially in financial applications. Its key size is too short for proper security (56 effective bits; this can be brute-forced, as has been demonstrated more than ten years ago). Also, DES uses 64-bit blocks, which raises some potential issues when encrypting several gigabytes of data with the same key (a gigabyte is not that big nowadays).

Triple DES is a trick to reuse DES implementations, by cascading three instances of DES (with distinct keys). Triple DES is believed to be secure up to at least " 2^{112} " security (which is quite a lot, and quite far in the realm of "not breakable with today's technology"). But it is slow, especially in software (DES was designed for efficient hardware implementation, but it sucks in software; and Triple DES sucks three times as much). Triple DES runs three times slower than DES, but is much more secure if used properly. In Triple DES memory required for implementation is the highest means it is the slowest algorithm. This is the

main drawback of Triple DES. It is having a sufficient value of avalanche effect. Several internet-based applications have adopted triple DES. But because of various drawbacks it is not a reasonable candidate for long term use.

AES is the successor of DES as standard symmetric encryption algorithm for US federal organizations. AES uses keys of 128, 192 or 256 bits, although, 128 bit keys provide sufficient strength today. It uses 128 bit blocks, and is efficient in both software and hardware implementations. It was selected through an open competition involving hundreds of cryptographers during several years.

RC4 being stream cipher is better than other encryption techniques .According to the analysis done, it is faster than AES.

Blowfish attempts to make a brute-force attack more difficult by making the initial key setup a fairly slow operation. For a normal user, this is of little consequence because we're talking about fractions of a second, but if you're trying out millions of keys per second to hack into the data, the difference is quite substantial.

Here we have used text files of variable Size for encryption and decryption process using the algorithms but based on same steps, we can compare the encryption algorithm based on video and audio files and developing a stronger encryption algorithm with high speed and minimum energy consumption.

Our future work will explore this concept and a combination of algorithms will be applied either sequentially or parallel, to setup a more secure environment for data storage and retrieval.

In future this analysis can be implemented in better simulators to get better results. This analysis can be done in another simulator by taking networking into consideration to show which algorithm performs better in network. The simulators which can be used are: MATLAB, NS2, ns3, OPNET, NetSim etc. These simulators will give better results for cryptographic applications in network.

References

- [1] Nataranjan R, 'Types of Cryptography',2003 .[Online].
Available : <http://www.thegeekstuff.com/2012/07/cryptography-basics/> [Accessed: 25-Sep- 2015]
- [2] Coppersmith D, Johnson DB, Matyas, SM. A proposed mode for triple-des encryption. IBM Journal of Research and Development. pp. 253-262, vol 1, issue 9,1996.
- [3] Mohtashim M, 'Data Encryption Standard',2016.[Online].
Available:http://www.tutorialspoint.com/cryptography/data_encryption_standard.html/
[Accessed : 30-Sep-2015]
- [3] S. Praveen, M. Nagesh, "Implementation of the Triple DES Block Cipher using VHDL", International Journal of Advances in Engineering & Technology, pp. 117-128, vol 3, issue 1, 2012.
- [4] S. Mandeep, S. Simarpreet, " Implementation of Triple Data Encryption Standard using Verilog ", International Journal of Advanced Research in Computer Science and Software Engineering ,pp. 667-672, vol 4, issue 1,2014.
- [5] W. Lee, T. Chen and C. Chieh Lee,"Improvement of an encryption scheme for binary images," Pakistan Journal of Information and Technology, pp. 191-200 , vol 2 , issue 2 ,2003.
- [6] V. Singh and S. K. Dubey, "Analyzing Space Complexity Of Various Encryption Algorithms", International Journal of Computer Engineering and Technology (IJCET),pp. 127-135 , Vol 4, Issue 1, 2013.
- [7] R. Manju , K. Sudesh," Analysis on Different Parameters of Encryption Algorithms for Information Security", International Journal of Advanced Research in Computer Science and Software Engineering, pp. 104 -107, vol 8 , issue 8,2015.
- [8] Chethan Kumar K V, S Sujatha, "VLSI Implementation of DES and TDES Algorithm with Cipher Block Concept" International Journal of Emerging Science and Engineering (IJESE) Volume 2, Issue 7, May 2014.

- [9]. Mandeep Singh Narula and Simarpreet Singh, "Implementation of Triple Data Encryption Standard using Verilog", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, January 2014, ISSN: 2277 128X.
- [10] Himani Agrawal and Monisha Sharma "Implementation and analysis various symmetric cryptosystems " Indian Journal of Science and Technology ,pp. 1173-1176,vol. 3,issue 12,2010.
- [11] Jawahar Thakur and Nagesh Kumar, "DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis", International Journal of Emerging Technology and Advanced Engineering, pp.6-12,vol 1, issue 2,2011.
- [12] S. Majithia , Kumar. Dinesh ,” Implementation and Analysis of AES, DES and Triple DES on GSM Network” , IJCSNS International Journal of Computer Science and Network Security, pp. 298-302 , vol 10 , issue 1 ,2010.
- [13] P. Kitsos, S. Goudevenos, "VLSI implementations of the triple-DES block cipher", Electronics, Circuits and Systems, ICECS 2003, Proceedings of the 2003 10th IEEE International Conference, pp. 76-79, vol. 1, 2003 .
- [14] A. Nadeem, "A performance comparison of data encryption algorithms, "IEEE information and Communication Technologies , pp. 84-89, 2006.
- [15] William Stallng, "Cryptography and Network Security Principles and Practice 5th Edition", Pearson.
- [16] Shashi Mehrotra Seth, Rajan Mishra, " Comparative Analysis Of Encryption Algorithms For Data communication " in IJCST,pp.292-294,vol. 2, Issue 2, 2011.
- [17] O. Hamdan, B. Zaidan, "New Comparative Study between DES, Triple DES and AES within Nine Factors", Journal of Computing, pp. 152-157, vol 2, issue 3, 2010
- [18] Aamer Nadeem, Dr M. Younus Javed, "A Performance Comparison of Data Encryption Algorithms", First International Conference on IEEE Information and Communication Technologies (ICICT), pp. 84-89,vol 1, issue 6, 2005.