Efficient Key Management for Data Security using TDES

A Dissertation Submitted in partial fulfilment of the requirements for the award of the degree of

Masters of Engineering

In

Electronics and Communication Engineering

Submitted by

Gaurav Puri

Roll No. : 801361008

Under the guidance of

Dr. Ajay Kakkar

Assistant Professor, ECED

Thapar University, Patiala

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

(Established under the section 3 of UGC Act, 1956)

PATIALA – 147004 (PUNJAB)

JULY, 2015

DECLARATION

I hereby declare that the work which is being entitled **"Efficient Key Management for Data Security using TDES "** in fulfillment of the requirements for the award of degree of Masters of Engineering in Electronics and Communication Engineering submitted at Electronics and Communication Engineering Department of Thapar University Patiala, is an authentic record of my own work carried out under the guidance of **Dr. Ajay Kakkar (Assistant Professor),** Electronics and Communication Engineering Department and refers others research's work which are duly listed in reference section.

The matter presented in this dissertation has not been submitted in any other University/ Institute for the award of degree.

Date: 14/7/15

Gaurav Puri

Gaurav Puri

Roll No: 801361008

This is to certify that the above statement made by the student is correct to the best of my knowledge and belief.

Date: 14/7/15

Dr. Ajay Kakkar

Assistant Professor, ECED

Thapar University

Countersigned By:

Dr. Sanjay Sharma

Head of Department

ECED, Thapar University

Dr. S.S. Bhatia

Dean of Academic Affairs

Thapar University

ACKNOWLEDGEMENT

First of all, I would like to express my gratitude to **Dr. Ajay kakkar**, **Assistant Professor,** Electronics & Communication Engineering Department, Thapar University Patiala for his patient guidance and support throughout the thesis. I am truly very fortunate to have the opportunity to work with him. I found this guidance to be extremely valuable. I am also thankful to **Dr. Sanjay Sharma, Professor and Head of Department** as well as PG coordinator **Dr. Amit Kumar Kohli, Associate Professor,** Electronics and Communication Engineering Department. I would like thank to entire faculty and staff of Electronics and Communication Engineering Department and then friends who devoted their valuable time and help me in all possible ways towards successful completion of this work. I thank all those who have contributed directly or indirectly to this work

Lastly, I would like to thanks my parents for their years of unyielding love and encourage they have always wanted the best for me and I admire their determination and sacrifice.

Gaurav Puri

Thapar University, PATIALA

ABSTRACT

Data security is the prime requirement for all the organizations in order to keep their important information safe from the hacker. Encryption is process of converting plaintext into cipher text. The strength of this cryptographic technique comes from the fact that no one can read the information without altering its content. There are various techniques which are used to keep the data confidential from hacker; these are passwords, cryptography and biometrics. The passwords are not so good for this task due to their low entropy and Biometrics is too costly and sometimes this technique also produces harmful effects to the human beings. Cryptography is the best solution for the above problem. In this report comparative study of various encryption algorithms has been done. From the Literature Survey; various observations and gaps have also been shown. Objectives have also been derived from the observations and gaps. Using dynamic keys simulation results have been achieve using MATLAB 2013. Finally, comparison of our approach with existing algorithms has also been done in terms of encryption and decryption time.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
CT	Cipher text
PT	Plain text
RL	Resource Lists
REK	Resource Encryption Key
CPA	Chosen Plaintext Attack
CCA	Chosen Ciphertext Attack
TDES	Triple Data Encryption Standard
PGP	Pretty Good Privacy
RSA	Rivest Shamir Adleman
HA	Hash Algorithm
DIFAC	Differential Access Control
KPA	Known Plaintext Attack
ECB	Electronic Code book
TEK	Traffic Encryption Key
DSA	Digital Signature Algorithm
PKI	Public Key Infrastructure
PKCS	Public Key Cryptographic Standards
MA	Modular Arithmetic
CA	Certificate Authority

LIST OF PUBLICATIONS

[1] G. Puri and A. Kakkar, "Development of Key Generation for Data Security using DES", *National Conference on Information Technology in Management and Engineering*, pp. 59-61, Mar. 2015.

[2] G. Puri, N. Kansal and A. Kakkar, "Key Generation in Encryption Algorithm for Data *Security", International Conference on Eco-friendly Computing and Communication Systems, ELSEVIER Publications*, 7-8th Dec 2015, NIT Kurukshetra, Haryana, India. (Communicated)

Chapter 1: Introduction to Cryptography

Cryptography is used to protect the data from an unauthorized access. Method of hiding message is not 100% secured and anyone can access the message; therefore, cryptography with multiple keys has its significance in the field of information security. Encrypting data results in an unreadable form called cipher text and the process of encryption involves set of instructions. It ensures that information is hidden for everyone except the intended users [1].

1.1 Basic Terminology

Plaintext is defined as the original data which one wants to hide from hackers. Encryption is defined as a method of hiding the original data by converting it into coded form with the help of keys. It is defined as a secret word or value which helps to encrypt or decrypt the data. Decryption is defined as a process of reverting cipher text back to its actual form [1]. The basic cryptographic model is shown in figure1.1.

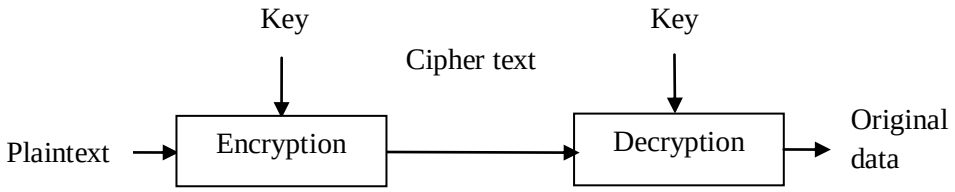

Figure 1.1: Basic cryptographic model

1.2 Security Goals

Cryptography has two processes named as encryption and key managing process. The terms under security goals are as follows:

- **Confidentiality**: It ensures that the secret data could be read or accessed by intended user only. During multiple communications, the aim of confidentiality is to make sure that only the authorized parties can access the data exchange [2].

- **Data integrity**: It overcomes the unauthorized modification of data. It ensures that the actual data has not been modified, destroyed in a malicious node [2,3].

- **Authentication**: It refers to both identifications of user and information. Multiple parties involving in a communication need to identify first, after that the secret data is transmitted over a channel, which makes it genuine. So, cryptography is usually divided into two main parts which are identity authentication and data origin authentication [2].

1.3 Types of Cryptography

There are two different types of cryptography which depends upon the use of key [4] and is explained as follows:

1.3.1 Secret Key Cryptography

Encryption and decryption is done with same key. It is very fast and used whenever data is in bulk [3] and is shown in figure 1.2.

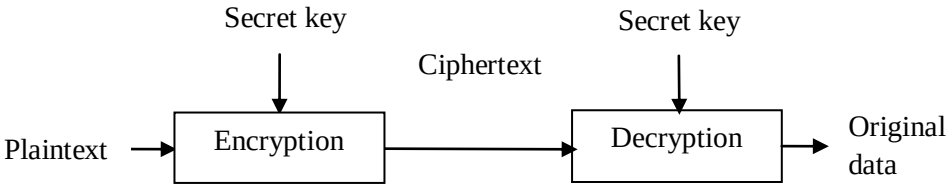

Figure 1.2: Secret Key Cryptography

Classifications of secret or private key cryptography are as follows:

- **Stream ciphers**: A stream cipher is a category of symmetric encryption algorithm or secret key cryptography in which the original data is enciphered bit wise [2].

- **Synchronous stream ciphers**: This scheme produces a key which is independent of message stream, thus same function is used at transmitter and receiver. Also the enciphering of single bit is dependent on the present state, thus known as state ciphers. RC4, TWO PRIME, SEAL, A5, WAKE etc. are commonly used stream ciphers [2, 3].

- **Block ciphers**: Here the input used is of fixed block length, which is transformed into different block of the fixed length. Decryption is performed using the same key by reverse transformation to the cipher text block. Normally block ciphers uses 128 bit of

block length and commonly used block ciphers are RC6, DES, MARS, IDEA, Serpent, AES etc. but the mostly used algorithm is DES [2].

One key is for enciphering and another key is for deciphering and also termed as asymmetrical key cryptography [1]. The basic block diagram of symmetric key cryptography is shown in figure 1.2 and the mostly used symmetric key cryptographic algorithms are described below:

a) Data Encryption Standard

It was developed by IBM in the 1970s but was later adopted by the US government as a national standard. It uses 56 bit key to encrypt 64 bit data block. The computation uses the permuted block as an input to produce the pre output block and for a final interchange of block of 16 iterations in terms of the cipher function *f* operates on two blocks, one of 32 bits and another of 48 bits to produces a final block of 32 bits [1, 2].

b) Triple Data Encryption Standard

As DES work with 56 off parity bits as key, TDES uses 168 off parity bits as key. This is far stronger then original DES algorithm. At first, data is encrypted by 56 bit key using DES, next decrypted with second 56 bit key for DES and at last again encrypted with actual 56 bit key using DES. Triple DES basically uses three iterations of DES scheme to encrypt data with a 168 bit key which is strong enough to provide secret data transmission. DES is very fast algorithm but Triple DES is a little slower. The advantage of TDES scheme is its compatibility with all other software and hardware, which supports DES [3].

c) Rivest Cipher 4

RC4 is an example of stream cipher which accepts one message symbol per unit time and performs encryption. This is variable key length cipher which makes use of operations that are byte-oriented. This algorithm involves random permutation as base and it requires variable length key ranging 1 to 256 bytes or 8 to 2048 bits to form a state vector *S* of size 256 bytes. The *S* contains different permutations of every 8 bit number in the range of 0 to

255 [3]. After the vector *S* is created, there is no need of key. XOR between values of *k* and data is done byte by byte rule and thus encryption is performed.

d) Rivest Cipher 6

Rivest Cipher 6 or RC6 was designed to support block length of 128 bits while the key size could take values of 128, 192 and 256 with a feature to parameterized for different word lengths, keys and permitted round counts. RC6 uses data derived rotatory modular addition and XOR operations [3].

e) Advanced Encryption Standard

It is a symmetric block cipher having three different cipher keys 128, 192 and 256 bits. There are different parameters that depend on key size. If the key length is 128, then 10 rounds exist, while for 192 bit key, rounds are 12 and there are 14 rounds for 256 bit key have been used respectively [3]. The figure 1.4 describes the steps involved in AES.

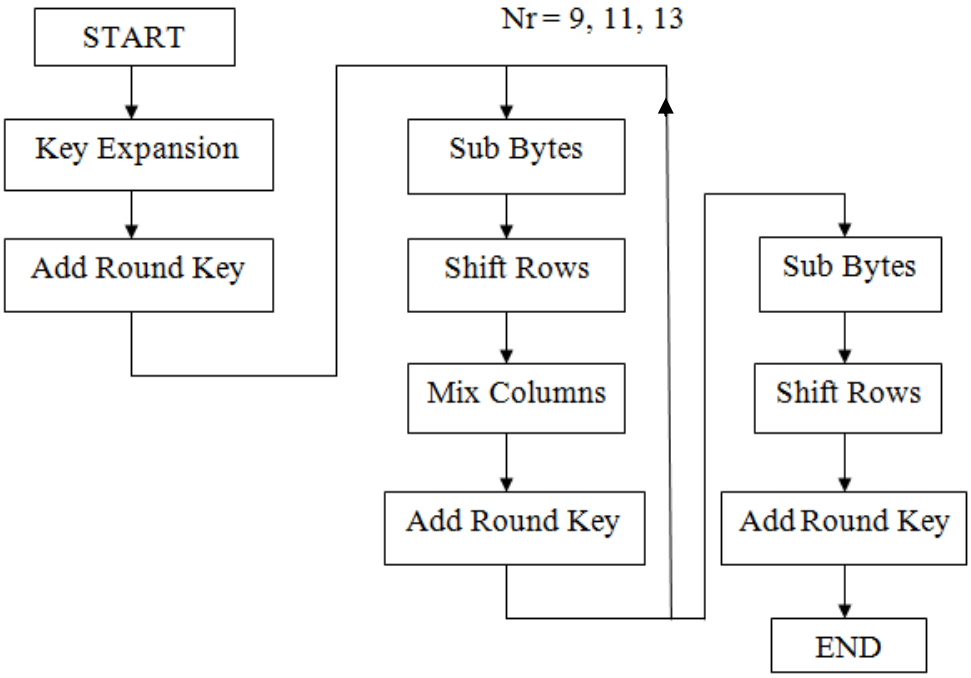

Figure 1.3: Various steps involved in AES [5]

The encryption algorithms/techniques discussed in previous section provides more security of data but these still has few drawbacks which are discussed as follows:

- **Key management**: Systems which have multiple users must generate and manage numerous keys. Thus for security of data, the key must be changed regularly in every session [3].

- **Key distribution and key exchange**: In this kind of cryptosystems, the master key is known to both particular sender and receiver pair. The master key should be known to both the users. Key distribution in a multiple user system is a biggest challenge. There is also a danger of getting the key hacked during the channel; therefore it is always advisable to generate the keys of sender and receiver rather than key exchange [3].

1.3.2 Public Key Cryptography

It employs two keys which are mathematically related to encryption algorithm. It uses a pair of different keys; the first key is termed as public key and is widely available to users. The second key is termed as the private key which is always kept underneath [3].

The commonly used asymmetric key cryptographic algorithms are the followings:

a) Diffie Hellman

It is an extensively used asymmetric key exchange algorithm which was designed in 1976. This algorithm eliminated the need of secure channel for key passing. Users share data for key generation until both ends up with a mutual key. Each user should be able to check the authenticity of provided data but limiting the modification only to the original users [2].

b) Pretty Good Privacy

It was introduced in 1990 by Philip R. Zimmermann. It provides both encryption and signing operations. It also provides utility services like character set encoding and data compression. It includes hybrid encryption in which the transmitted messages are initially enciphered with a symmetric key and then encrypted with a key algorithm [3].

c) Public Key Infra Structure

The public keys of the multiple subscribers are covered in certificates appointed by trusted third parties known as certificate authorities. These certificates are used for recognizing persons and also for documentation of computers and services [2].

d) Rivest, Shamir and Adleman

On the name of authors, they designed a public key cryptographic system which uses the concept of digital signatures. RSA make use of public as well private keys. At encryption end, public key is used, while at decryption side, secret key is used. There is the use of prime numbers denoted as p and q to generate two keys. Product of these p and q determines the security of this algorithm [5].

e) ElGamal

The ElGamal uses asymmetric key for public which was created on the Diffie Hellman key agreement. It is the predecessor of DSA which also accepts elliptical curve variants that provides rely on the hardness of discrete logarithm. As the mathematics are more complex but it provides good performance [5].

f) ECDSA

Elliptic Curve DSA is different from the Digital Signature Algorithm which operates on elliptic curve groups. The scheme helps to improve the performance on the internet and non internet based applications. Breaking of this algorithm requires to solve the elliptical curve discrete logarithm problem thus provides more security as RSA algorithm with small use of keys which termed as its advantage [2].

g) XTR

It is an asymmetric encryption algorithm which uses traces to represent and evaluate powers of a subgroup of finite field. XTR security depends upon the complicated solving of discrete logarithm problems of a finite field [1]. It has very fast key generation mechanism, uses small key size and has fast computational time.

h) Digital Signature Algorithm

DSA is a United States Federal Government standard or FIPS for digital signatures, which was suggested in 1991 by the National Institute of Standards and Technology as Digital Signature Algorithm. A revision was done in 1996, named as FIPS 186, after that the standard was again advanced in 2000 which is finally called as FIPS. Digital Signature Algorithm is similar to ElGamal signature algorithm. A public key signature verification standard designed primarily for digital message authentication [5]. The structural diagram of DSA algorithm is shown in figure 1.7.

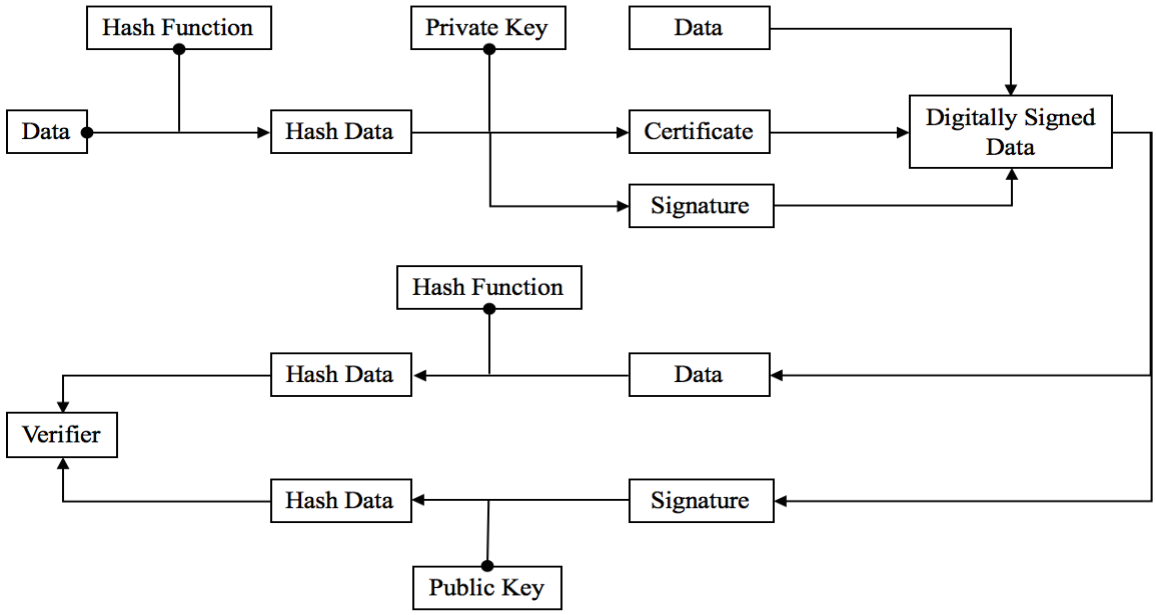

Figure 1.4: Various stages involved in DSA algorithm [5]

i) International Data Encryption Algorithm:

It was originally developed in 1990 as the Proposed Encrypted Standard (PES). In 1992, it was renamed IDEA. IDEA is a block cipher that uses 64 bit data blocks and 128 bit key [3]. This scheme provides more security to data because of more complex key length as well as increases the computational time.

1.4 Comparison between various encryption algorithms

From the previous section, comparison of various encryption algorithms has been done by considering data length, key length, type of algorithm and iterations.

Table 1.1: Comparison of various encryption algorithms

Parameters	**DES**	**RSA**	**IDEA**	**TDES**	**AES**
Year	1970	1978	1991	1998	2001
Data Length	56	Variable	64	168	128
Key Length	64	Variable	128	112	128, 192, 256 bits
Type of Algorithm	Symmetrical	Private key is used to encrypt the data and decryption is done by public key	Symmetrical	Symmetrical	Symmetrical
Iterations	16	Depends upon data and key length	8	3 times to DES	Key size depends upon round functions

1.5 Cryptography Attacks

There are two types of attacks, first is on system and second is on data [13] and the figure 1.8 describes about the type of cryptography attacks as shown in figure 1.8.

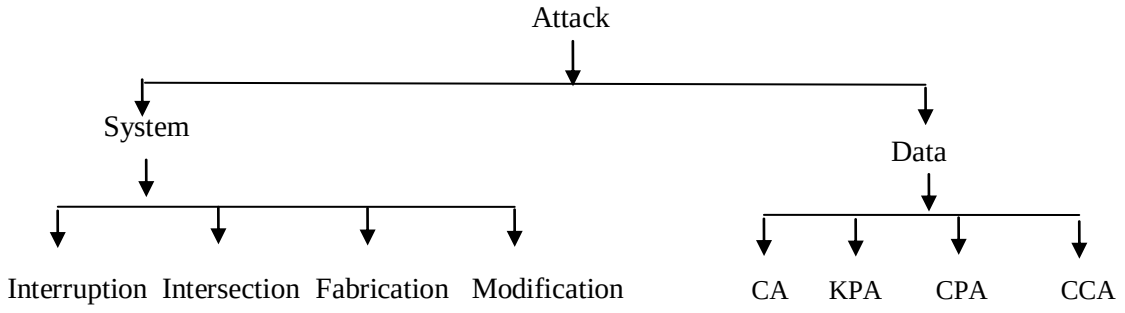

Figure 1.5: Types of cryptography attacks [3]

1.5.1 System Attacks

These attacks are performed for data acquisition. For any cryptanalysis operations to work there must be the available a copy of cipher-text [2]. This section will deal with acquiring data by adversary for cryptanalysis, which is shown in figure 1.9:

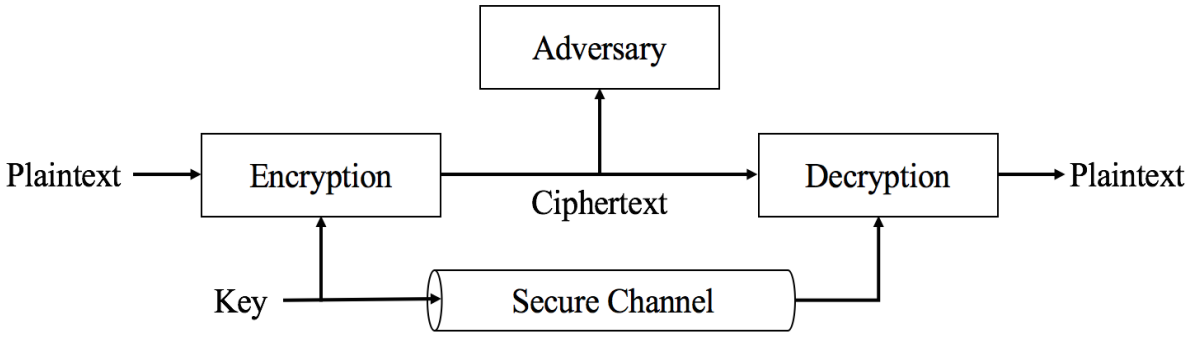

Figure 1.6: Typical data flow scenario [3]

- **Interruption:** This attack is done when the normal flow of data through a route is unavailable for some time and is shown in figure 1.10.

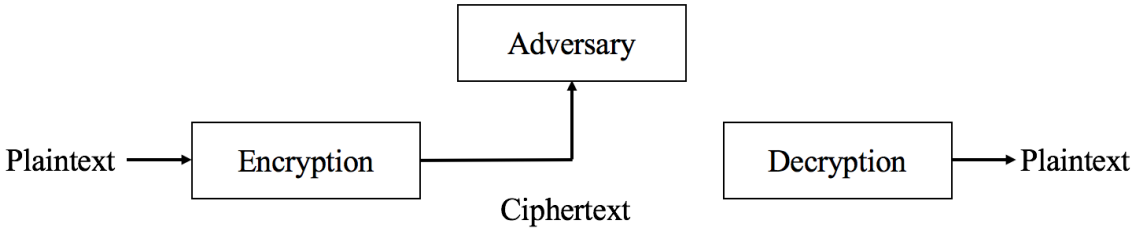

Figure 1.7: Interruption in data flow [1]

- **Interception:** This is an attack on the confidentiality of the system. Adversary receives the copy of data transmitted without showing its presence at all. This can be shown in figure 1.11 which is as follows:

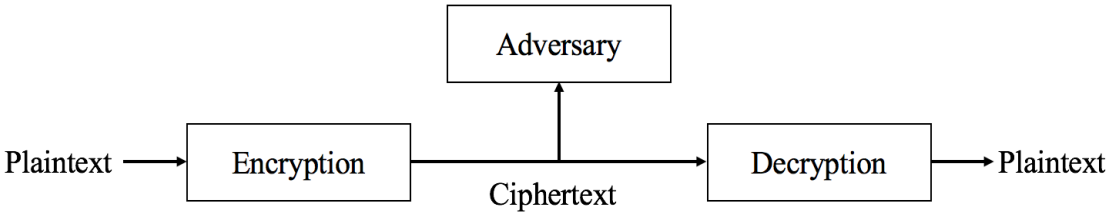

Figure 1.8: Interception attack [1]

- **Fabrication:** When the authenticity of the system is challenged, the adversary is able to feed the receiver any false information. This attack allows to insert forget data into the system and is shown in figure 1.12.

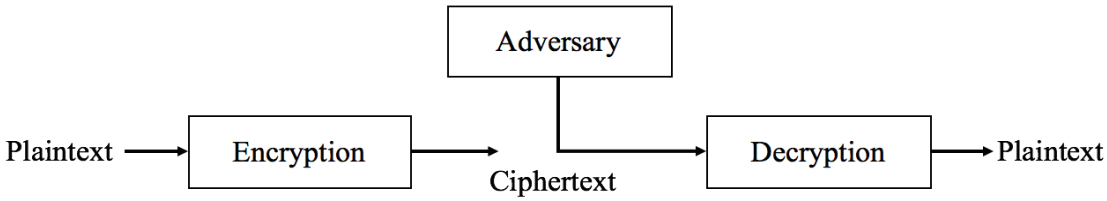

Figure 1.9: Data under fabrication attack [1]

- **Modification:** Under this attack the integrity of data is compromised. The adversary has the power to modify data according to the needs and plant that into system as shown in figure 1.13.

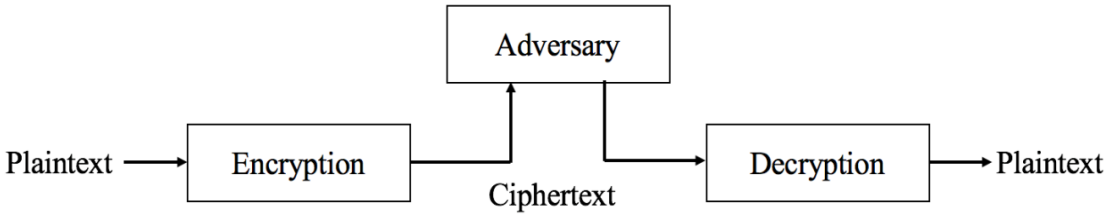

Figure 1.10: Data modification by adversary [1]

1.5.2 Data Attacks

Hacking the data is known as attack. It may also stop the flow of data that decoder easily recovers from the crypto system [2]. It has three ways of decryption which are given as follows:

- **Cipher-text only attack**: The crypto-analyst has cipher text of several messages encrypted using the same encryption algorithm. The main objective is to obtain the key used to encrypt and decrypt the original message [2].

- **Known Plaintext Attack**: Crypto-analysts try to have an access of both plain-text and cipher-text. Aim is to hold the key used to encrypt the messages or an algorithm to decrypt messages [2].

- **Chosen Plaintext Attack**: The crypto-analyst holds both the cipher-text but also some parts of selected plain-text [2]. By knowing this information, hacker could be able to change the encrypted data.

- **Chosen Cipher-text Attack**: This attack is commonly associated with decryption process, the hackers obtains the temporary access to the decryption machine and may easily recover the original data by selecting a particular cipher-text string.

1.6 Organization of the Thesis

Chapter1 discuss the introduction of cryptography and types of the cryptography. A brief introduction about the security of the data has also been included. Chapter 2 includes the literature survey. From this literature survey the observations have been drawn, from which the objectives have been developed. Chapter 3 includes the critical analysis of the TDES algorithm. Detailed description of TDES algorithm has also been explained. Chapter 4 includes the detailed analysis of proposed work. In this chapter the modified TDES approach on the basis of key management has been successfully achieved. Chapter 5 shows the results and discussion of the proposed work. Comparison of proposed scheme has been done with TDES in terms of encryption and decryption time. Chapter 6 contains the conclusion of the proposed work and its future scope has also been discussed.

Chapter 2: Literature Survey

This chapter covers the work done by the various scholars in the area of data security. From the literature survey various interpretations have been drawn. Objectives have also been drawn from these observations and enlisted in the end.

2.1 Literature Survey

J. Borst [14] *et al.* worked on cryptographic algorithm implemented on smart cards that was used in the security environment. RSA and elliptical curve algorithm used to generate a pseudo random generator were also analyzed. Key length was optimized but the complexity of the scheme designing termed as its weakness.

Stefan Mangard [15] *et al.* introduced a highly consistent and accessible AES hardware architecture. They also evaluated a scalable throughput and key size for a better performance. This performance was attained by balancing the combinational paths of the design. This architecture provides a throughput of 241 Mbps on a 0.6μm CMOS process by standard cells. High level of performance was provided by using similarities of encryption and decryption with relatively small area. The disadvantage was limited performance rate as the key size varies.

Wong yang [16] *et al.* worked on a providing the tool for addressing the problem with the help of RFID technology. They also suggested a method, useful for the point of scale application. The protection and authentication of tags result in the detection of errors, which improved the correctness and robustness. They also developed 96 bits programmable tags which were very complex.

Joseph Ren [17] *et al.* proposed ring signature set of possible signers, so that, the verifier was unable to communicate members actually produced the signature. They also introduced a generalized multi-signer ring signature scheme to for joint message singing, in which same prime number was shared by all ring members. The limitation of this scheme is that it

achieves unconditional signer ambiguity but it was secured against adaptive selected message attack in the random oracle model.

Xiaojiang Du [18] *et al.* proposed a heterogeneous sensor network model which has routing driven key management scheme. Shared key was used for neighboring sensor which establishes communication with each other. Hence, it results in better performance and security stability and energy consumption in comparison to other management schemes. Because security of data was an important factor; therefore there is still need to work on the proposed scheme for more secured key management.

Massimo Alioto [19] *et al*. analyzed the conditions which maximize the effectiveness of DPA attacks. The algorithm could be improved by increasing iterations and key length. The inadequacy was that the very high values of *m* deteriorate the inter signal to noise ratio which increases spike amplitude.

S. P. Kumar [20] *et al.* proposed a secured grid based route driven Public Key Cryptography (PKC) scheme for heterogeneous sensor networks. The analysis for energy and throughput efficiency for the dynamic position of sensor nodes was covered in the work. They tested the route driven scheme for the scalability by varying the node density from 100 nodes to 1000 nodes in the network. The main limitation of their work was that in case of node failure only neighboring nodes were used to hold the data of faulty node.

Abdulhadi Shoufan [21] *et al.* proposed a hardware architecture for Mc-Eliece cryptosystem which showed the advantage of modern FPGAs. The complexity of the system was ruled by defining a layer model and by the classification of the different operations of Mc-Eliece into the layers of this new and original model. Hence, highly modular design resulted, which provided optimization and was efficient to support future extensions. The crypto processor used undergoes an optimization process regarding resource usage and performance. It also handles two hash modules; first in the PRNG and the second for CCA2. This provides a future work to reduce these two modules to just one module with corresponding rescheduling to avoid performance losses.

Mao Wang [22] *et al.* worked on single and multi-core configurable AES architectures for flexible security. Each AES processor provides block cipher which had a key expansion design. In this multi-core architecture; the memory controller of each AES processor was designed for the maximum overlapping between data transfer and encryption. It reduces interrupt handling load of the host processor. It had independent data paths and also reduces the bandwidth problem. The main limitation of this was the architecture complexity.

T. Monoth [23] *et al.* worked on two aspects, visual cryptography and biometrics for the data security. Visual cryptography scheme allowed confidential messages to be encrypted into k out of n secret sharing schemes. The scheme was efficient enough to keep the information secure and reliable. More research is required to be carried out in this area for the better results in terms of data confidentially, as the image reconstruction through VCS was limited.

R. H. Torres [24] *et al.* worked on identification of keys and cryptographic algorithms using genetic algorithm and graph theory which provides more accurate results. Evidence of patterns and an efficient signatures generated by the algorithms under test were also detected; therefore this scheme has complexity issues and also takes more computational time.

L.J. Garcia Villalba [25] *et al*. worked on secured extension to the Optimized Link State Routing Protocol (OLSR). It was assumed that the nodes in the network were reliable, making it susceptible to random attacks. The behavior of extended OLSR against different attackers in different situations was also assessed and showed that the result of their proposed scheme adds a slight overhead to OLSR and does not affects performance. This scheme was not so much secured as required and can be easily hacked.

Shengrong Bu [26] *et al.* worked on distributed combined authentication and intrusion detection with data fusion in high security mobile Ad hoc networks. Multimodal biometrics was deployed to work with intrusion detection systems to improve the short comings of uni-modal biometric systems. The decisions were taken in a fully distributed manner by each authentication device and IDS. Combining continuous authentication and intrusion detection

could be an efficient approach to improve the security issues in high security MANETs. This scheme however, suffers from crossover interference effect.

Jing Li [27] *et al*. They introduced one way function tree which provides the better security and minimizes the collision attacks. They also introduced a new cryptographic construction method named as HOFT which was found with the general idea for obtaining one way function tree. They also proposed two structure preserving graph operations on HOFT called as: tree product and tree blinding. The cost in providing the communication to the different protocol put the main constraint on this method.

Hao Yang [28] *et al.* studied a case on DNSSEC, so that to deploy cryptography in Internet scale systems. They also provided the systematic analysis of the design, deployment, and operational challenges. Their work revealed the gap between cryptographic designs and operational Internet systems. The limitation is that are not able to deploy cryptography in internet scale systems successfully.

Chang Yuan Yan [29] *et al* investigated a cryptographic technology which helps in providing secured information and was a kernel of information safety. Block cipher was the core part for the data encryption, key management, authentication and digital signature in information security. Its design part was very complex and it affects the performance of the optimized algorithms.

A. Dogan [30] *et al.* evaluated and compared the power consumption of AES architectures and observed that AES had gained substantial advantage for its security. Thus, the authors introduced low power AES model, which had performance oriented designs and thus gained more importance over an entire area, which could also reduce power consumption in FPGA. They showed its efficiency by discussing the results that carry details about area, time and power dissipation.

Zhiguo Wan [31] *et al.* extended the attribute set based encryption and proposed the hierarchical attribute set based encryption scheme. To deal with user revocation more

efficiently than the previous schemes, the hierarchical attribute set based encryption scheme was employed which has multiple value assignments for access expiration time. The constraint of this structure is the computation complexity provided by hierarchical structure. It becomes difficult to decrypt the original data, if hacker made very small modification in the algorithm.

Tiancheng Li [32] *et al.* showed which slicing preserves better data utility that could be used for attributed disclosure protection than the previous generalized techniques. They also developed an algorithm for computing the sliced data which handle high dimensional data. The limitation is that the random grouping used was not very effective. Hence, it provides the future work to design more effective grouping algorithms.

Kazuo Sakiyama [33] *et al.* critically analyzed DFA attacks on the AES from an information theoretic perspective. They also considered DFA attacks by making them equivalent to a passive attack where attackers could get leaked information without measuring noise. They also presented a new DFA methodology to achieve the optimum DFA attack by deriving the amount of the leaked information for several fault models. The scheme suffered in terms of security and thus there much scope to work on this scheme.

Isaac Woungang [34] *et al.* proposed AES based routing algorithm for securing AODV based eMANETs against wormhole attacks. The scheme consists of substituting the AES part of the scheme by the TDES based upon AODV-WADR-TDES routing algorithm. To observe the performance of the algorithm where mobile devices are incompatible with AES as a part of eMANET nodes was used. They also observed that the markers in the form of hash codes are included in the data packets to help consolidating the data integrity. Scheme was better in terms of end toend delay, packet delivery ratio and number of packets. Thus, it is concluded that there is still a need to work on this scheme to achieve better performance of the system.

S. Verma [35] *et al.* proposed an efficient symmetric key cryptography algorithm for information security. This block encryption algorithm offers the enhanced security features compared to other symmetric algorithms. It also helps in achieving data security and message

authentication. For large packet size, the scheme takes much less encryption time. It was also effective whenever there was a change in data type such as image, audio or video instead of text. It also proved that higher key size leads to change in the battery and time consumption.

Yang Li [36] *et al.* worked on new fault based side channel attack called fault sensitivity analysis attack using fault sensitivity. They also explained the successful FSA attacks against three AES. Hardware implementations, where two of them were resistant to the differential fault analysis. They also discussed the countermeasures against the proposed FSA attacks. They suggested a new fault based attack called FSA attack. The FSA attack was the first one that introduced the concept of fault injection intensity to the fault based attacks. In the FSA attack to test fault sensitivity, fault injection were used to test out the sensitive information. The hardware implementation was very complex.

Darpan Anand [37] *et al.* worked on the identity based cryptography which includes encryption and the techniques of digital signature for authentication. Thus, the technique was useful for the identity based encryption applications in the field of various networks as ad-hoc networks. Time required for user identification was large which a major drawback of this work.

Junbeom Hur [38] proposed an attribute based secured data sharing scheme in smart grid. The key escrow problem was resolved such that the key generation and storage center cannot decrypt the cipher-text. The computation overhead of receivers for decryption was independent of the size of the attribute. The limitation of this scheme is that the desired data cannot be recovered due to collusion attack, as the blinding value is randomized for a particular user.

Koji Nuida [39] et al. worked on the security of pseudo-randomized information theoretically secure schemes and projected methods for assessment between random and pseudorandom cases in PRG based randomness cryptographic schemes. There is scope of further improving the effect of PRG based randomness reduction in order to secure and recover the original data.

Shuangqing Wei [40] *et al.* worked on DES based block ciphers operating in cipher feedback mode to prove quantitatively the pros and cons of exploiting voluntarily. A serially concatenated scheme with both outer and inner encoders encipher pairs was proposed. The security improvement factor which reflects the required plaintext and cipher-text pairs for eaves dropper's known plaintext attack was also conducted. The simulation results demonstrated the accuracy of derived approximation of the post decryption performance for the legitimate receiver. The key generation scheme used in this technique has to be dynamic key generation, in order to make the scheme more secured.

Krishna Kumar Pandey [41] *et al* worked upon an enhanced symmetric key cryptography algorithm to improve data security which was essential for block cipher method. It takes less time for twenty iterations, providing security for bully files. It becomes very tough to break the encryption algorithm without knowing the exact key due to internal key generation with the reference of entered key. The method for both encryption and decryption could be applied to any type of public application for sending confidential data. It could also be useful to send internal key to the sender using another secured path to the receiver. It prevents data from attackers and claim for less time complexity for large data files. Large chunk of data required for this method was difficult to handle.

Dominik Schurmann [42] *et al.* worked on the likelihood to utilize contextual information to create a secured communication channel among devices and their approach was exemplified for ambient audio that could be similarly applied to alternative features or context sources. This scheme faces difficulties to achieve sufficiently accurate time synchronization among wireless devices. But the accuracy and computational complexity of the approach can be further reduced by more exact time synchronization.

Zhao [43] *et al.* used different AES implementation and carried the multiple deductions based algebraic side channel attack, which was created on the cache attacks. With the use of key length the MDATDCA technique also evaluated that TDCAs can become possible AES-192 and AES-256 as well and thus, achieved many enhancements of trace driven cache

attacks on AES when compared to the previous work done in this area. Hence, the work could be extended to improve TDCAs on other block ciphers too.

Z. Cica [44] discussed implementation of AES with TDM multiplexing for providing internet routers. The author suggested a FPGA design for the AES encryption algorithm, which provides high throughput rate and can be used for multiple data encryption with the help of key expansion module. He also introduced a TDM technique which was highly efficient as it provided sharing in the key expansion module. The limitation of this scheme is that the channel could not be fully utilized and synchronization is an important parameter in this system.

Chun I Fan [45] *et al.* presented an attribute based encryption scheme which was the first ABE scheme. It along with binary states also aims at dynamic membership management with arbitrary states. They keep high flexibility of the constraints on attributes and allow users to join, leave and update their attributes without using random oracles. But the limitation is that it is unnecessary for those users who do not change their attribute statuses to renew their private keys.

Jin Li [46] *et al.* proposed a new secured outsourced attribute based encryption system which offloads all access policy and attribute related operations in the key issuing process. It leaves only a constant number of simple operations for the attribute authority and eligible users to perform locally. It also facilitated the check ability of the outsourced computed results in an efficient way but it takes more time than the original ABE scheme which seems as its limitation.

Joseph K. Liu [47] *et al.* gives a concreted construction in the random oracle model and created a linkable ring signature scheme with unconditional anonymity. It was more secured even in case where the pairing operation was to be used. Thus, it leaves the scope of research and provides the further research to shorten the size of the signature and to build arrangement with unconditional anonymity in the standard model.

Seung Hyun Seo [48] *et al.* worked on mediated certificate-less public key encryption scheme without using pairing operations. User encrypts the crucial data using the public keys of the cloud and uploads it. Upon successful authorization; the cloud partially decrypts it for the users; therefore, the users subsequently decrypt the partially decrypted data using their private keys. This scheme faces the problem of collusion attack that termed as its limitation; therefore, it is still needed to work on the data security.

Huang Lu [49] *et al.* worked on secure data transmission for cluster based WSN where the clusters were formed dynamically and periodically. They also proposed a secure and efficient data transmission protocols for CWSNs, known as Identity Based Digital Signature scheme and the Identity Based Online/Offline Digital Signature scheme. SET-IBOOS reduces the computational overhead for protocol security for WSNs. These protocols meet the security requirements and security analysis against various attacks but the limitation is that the SET-IBS protocol is less secured as compared to SET-IBOOS protocol and extra energy was consumed by the auxiliary security overhead.

Jinguang Han [50] *et al.* operated on identity based secured distributed data storage schemes and notes the following properties: a) the access permission could be decided independently without the help of the private key generator by the file owner, b) for one query; one file is accessed by the receiver, rather accessing from all files of the owner. Data owner remains to be online to provide access permission to receiver and the scheme was more time consuming for one query.

Zarko Stanisavljevic [51] *et al.* represented a new and original software system for cryptographic algorithm's visual representation, industrialized to provide a data security. The system allows users to follow the execution of several complex algorithms on real world with the possibility of forward and backward navigation. It was spontaneous and user approachable but had limitation as the system security was less than that was predicted.

Daojing He [52] *et al.* demonstrated that the system could be implemented the protocols on real mobile devices and sensor platforms with limited resource. It was observed that the

execution time of proxy signing a message and the signature verification were 1.57 ms and 1.72 ms for a network user and network server on a 1.6 GHz PC having the length of 1024 bits respectively. Hence, this operation takes relatively longer time and more memory was required.

Xiaofeng Chen [53] *et al.* carried research work on new algorithms for secured outsourcing of modular exponentiations. The scheme has only single round of conversation between the client and the servers. The communication complexity was only a few kilobytes for each instance of an outsourcing algorithm. This scheme requires so much memory for its operation, which makes it more expensive.

Seung Hyun Seo [54] *et al.* supports an efficient key revocation for compromised nodes which minimizes the impact of a node compromise of other communication links. It was observed that the technique was effective in defending against various attacks and using COOJA simulator to assess its time, energy, communication and memory performance. This scheme will provide the future work to formulate a mathematical model for energy consumption, based on certificate-less effective key management with various parameters related to node movements.

Attila Altay Yavuz [55] *et al.* worked on broadcast authentication scheme for command and control messages termed as rapid authentication which simultaneously achieves several desirable properties. It includes fast signature generation and verification, immediate verification without message buffering, small public key and signature size, high scalability, high packet loss tolerance, provable security. It was observed that rapid authentication was an ideal choice for broadcast authentication of command and control messages in large distributed systems with time critical application. Rapid Authentication scheme has limitation that it cannot provide authentication for command and control messages without predefined structures but it also provides the future to overcome the limitation of the scheme.

Mehran Mozaffari Kermani [56] *et al.* worked upon the two fault diagnosis approaches for the lightweight block cipher which include parity based structure and RERO structure and

their results of the simulations proved very high error coverage for the present error detection structures for the injected faults. It was observed that by utilizing this scheme; the FPGA implementation analysis show acceptable overheads for the XTEA thus this scheme can be used to protect the extremely sensitive and resource constrained applications.

Taeho Jung [57] *et al.* worked on privacy preserving sum and product calculation protocols without secure communication channels which allow up to *k* collusive participant. They also analyzed the security of protocol and showed that the protocol was more secured, if the CDH problem was assumed to be intractable. This scheme provides the future work to design privacy preserving data releasing protocols, such that general function of data could be evaluated correctly while preserving individual's data privacy.

Donald Donglong Chen [58] *et al.* proposed a high speed pipelined design for FFT multiplication that presented efficient implementation of ring-LWE and encryption cryptosystems. They also proposed multipliers on Spartan-6 FPGA which showed that the architecture achieves a 3.5 times speedup on average as compared to the state of art and their result also showed that the selected parameters support efficient modular design reduction. The system provides the future work that will exploit the full usage of the processing units of the proposed pipelined architecture, the control logic design and there is still a scope to propose the FFT multiplier in the ring LWE and SHE cryptosystems.

Sha Ma [59**]** *et al.* proposed a public key encryption which supports four types of authorization; a) user level authorization, b) cipher text level authorization, c) user specific authorization, and d) cipher text to user authorization. It has been observed that this scheme provides flexible authorization without sacrificing much efficiency and introduces flexible authorization policies to the PKEET primitive but there is need to construct a scheme which supports additionally the other two types of authorization.

Feihu Xu [60] *et al.* proposed a novel approach called measurement device independent QKD that could remove all side-channels from the measurement unit and was the most vulnerable part in QKD system. It offers a clear avenue towards secure QKD realizations. It

was still necessary to improve the performance of the mdiQKD implementation that will prove the feasibility of mdiQKD for free space communication. It was observed that this scheme puts a first step towards future satellite based mdiQKD networks, in which an un-trusted satellite could be shared by many users and the mdiQKD enables developments in the field of quantum optics, as well as advanced novel applications for quantum information and quantum communication.

2.2 Observations

From the above section, following observations have been drawn:

- The security level of any cryptographic model decreases whenever a) data block is increased, b) single key of short length is used to encrypt data, and c) access points are increased.
- If the key management is properly done, i.e. either multiple keys are used or number of iterations are increased in which key is generated from them, the security level could also be increased.
- Random attacks make the cryptographic model more weak and computational complexity is also increased due to these attacks.

2.3 Gaps and Problem formulation

From the above observations, it has been concluded that there is still need to develop an algorithm which provides more data security and provides efficient key management system for utilizing the various keys to increase the security of the data as well as reduces the computational complexity of the systems.

2.4 Objectives

From the above section, following objectives have been drawn:

a) To analyze DES and TDES algorithm and study the performance of these algorithms.

b) To improve the key management in TDES scheme in order to reduce encryption time.

c) To compare the proposed scheme and compare with existing cryptographic algorithm.

CHAPTER3: ANALYSIS OF TDES ALGORITHM

This chapter provides critical analysis of TDES algorithm and its various performance measuring parameters such as round function, key utilization, block size, rounds, key length, computational speed, power consumption and memory usage.

3.1 Triple DES Scheme

Triple DES is based upon the DES algorithm. It is a symmetric block cipher that basically operates DES algorithm three times. It consumes three 56 bit keys, thus a complete key length of 168 bits. The procedure followed for encryption is the same as DES scheme but it is repeated three times, hence it is named as TDES. The process has three parts where the data is initially encrypted with the first key then decrypted with the second key and finally encrypted again with the third key. It is more consistent and has a longer key length that eliminates many of the attacks which are used to reduce the amount of time it takes to break DES, thus the DES algorithm become outdated [13]. Many security systems support both Triple DES and DES, whereas Triple DES is used for backward compatibility and the block diagram of this scheme is shown figure 3.1:

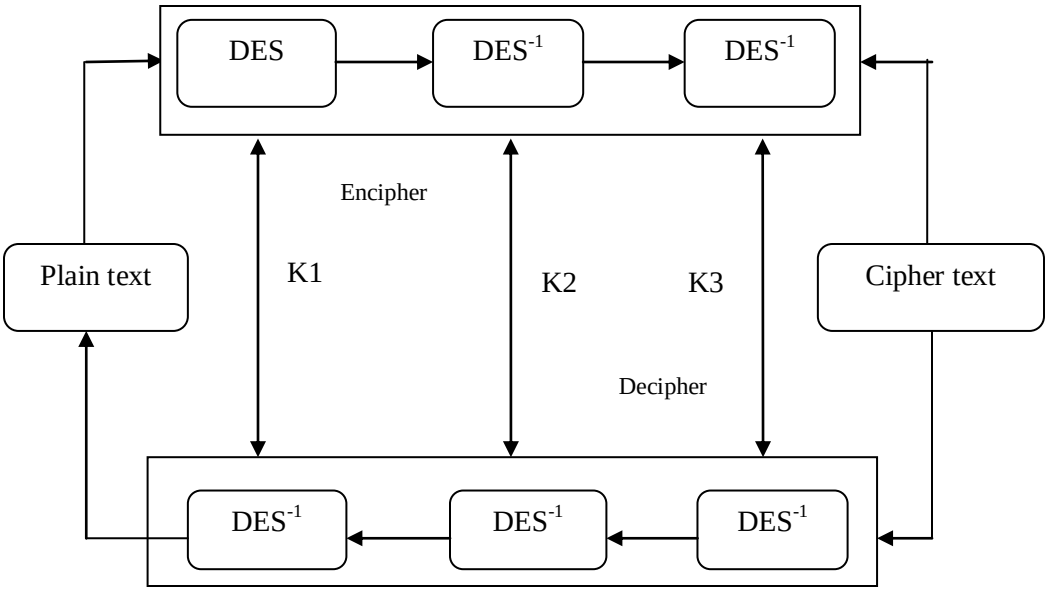

Figure 3.1: Block diagram of TDES encryption and decryption [9]

3.2 Triple DES Algorithm

The *EK(I)* and *DK(I)* signifies the DES encryption and decryption of *I* using DES key *K* respectively. The TDEA encryption/decryption process is a composite operation of DES encryption and decryption operations [13]. The procedures performed in TDES are as follows:

- TDEA encryption operation: It transforms a 64 bit block *I* into a 64 bit block *O*.

$$O = EK3(DK2(EK1(I))) \tag{3.1}$$

- TDEA decryption operation: It transforms a 64 bit block *I* into a 64 bit block *O*.

$$O = DK1(EK2(DK3(I))) \tag{3.2}$$

3.2.1 Modes of Operation

Triple ECB: The Triple ECB works exactly the same way as the ECB mode of DES [10]. This is the most commonly used mode of operation. The initial 64 bit key acts as the initialization vector to DES. Triple ECB is then implemented for a single 64bit block of plaintext [7]. The resulting cipher text is then XORed with the next plaintext block to be encrypted and the procedure is repeated. This technique increases the security by adding an extra layer to TDES and is more protected than triple ECB [10].

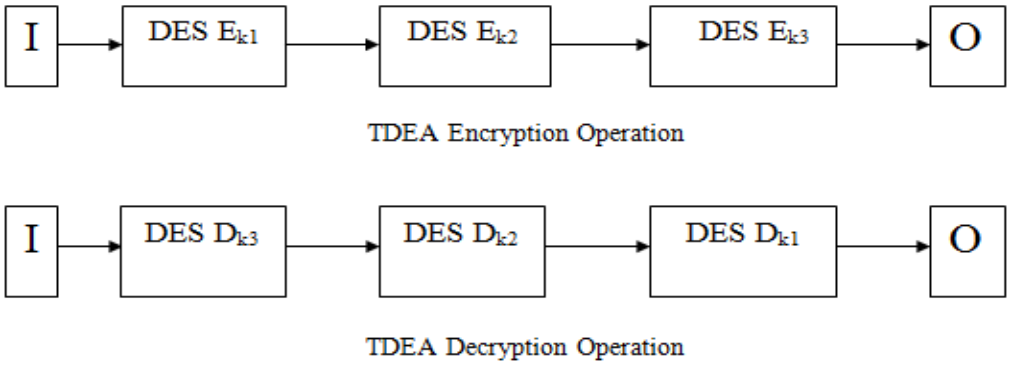

Figure 3.2: TDES block diagram (ECB mode)

3.2.2 Applications of TDES

The TDEScore can be used for number of encryption applications comprisingof [3] secure data transfer, electronic funds transfer, encrypted data storage andsecure communications.

3.2.3 Features of TDES Algorithm

The various features of TDES algorithm is shown in table 3.1:

Table 3.1: Features of TDES Algorithm

FEATURES	**TDES**
Cipher Type	Symmetric Block Cipher
Algorithm Structure	Feistel Network
Block Size	64 bits
Rounds	48
Key Length	112, 168 bits
Computational Speed	Moderate
Throughput	Low
Power Consumption	Highest
Memory usage	Very high
Confidentially	High
Cryptanalysis	Vulnerable to Differential, and linear cryptanalysis

The TDES algorithm has been critically analyzed on the basis of round function, key utilization, block size, rounds, key length, computational speed, power consumption and memory usage. The development of modified key management has been done in next chapter in order to optimize the encryption time.

Chapter 4: Efficient Key Management Scheme using TDES

The most important parameter in cryptography is to protect the data from data attacks thus termed as data security. The most practical attack is a brute force attack. The other data attacks have less theoretical complexity in comparison to brute force attack. Hence, it requires an unrealistic number of known plaintext to be carried out. The length of the key determines the number of possible keys and thus based upon the key length the security of data matters. As a result, the key size was reduced from 128 bits to 56 bits to fit on a single chip as well provides data security.

4.1 Modified TDES Approach

In order to overcome the problem of data attacks by the differential cryptanalysis, this scheme aims to design the modified TDES approach. Initially, it breaks a block of 96 bits into 3sub-blocks then performs different *f* functions on each of the 3 sub blocks and at last increases the S-box from S_1 to S_{16} which makes it less vulnerable to attack by differential cryptanalysis. In order to increase the cryptographic security, the 128 key bits that are provided to increase the key from 56 bits to 112 bits and are equally divided into 64 bits, K_1, K_2 [13]. According to the key scheduling of the DES scheme there are 64 bits in Initial Combination 1 on the left and then after removing 8 parity bits through the Permuted Choice l (PC l), the 56 bits are output. These 56 bits are then divided into 28 bits on the left and on the right. The sub key is shifted according to the number of times of the left shift of the key schedule in each cycle, hence produce the $K_{1,1}$ to$K_{1,16}$are the sub keys of 48bits through the PC 2. $K_2$is the 64 bit on the right and $K_{2,1}$to $K_{2,16}$ is the sub keys of 48 bits are also produced by the key schedule. As a result, when applying $K_{1,1}$and $K_{2,1}$to the *f* functions on the left and the right, the following encryption and decryption formula are derived.

Encryption:

$$Ai = Bi - 1 \tag{4.1}$$

$$Bi = Ci - l \oplus f(Bi - 1, K2, i) \tag{4.2}$$

$$Ci = Ai - l \oplus f(Bi - 1, K1, i) \tag{4.3}$$

Decryption:

$$Ai-1=Ci\oplus f(Ai,K2,i) \quad (4.4)$$

$$Bi-1=Ai \quad (4.5)$$

$$Ci-1=Bi\oplus f(Ai,K1,i) \quad (4.6)$$

4.1.1 Single Round of Encryption and Decryption

The single round function of encryption and decryption is presented in figure 4.1

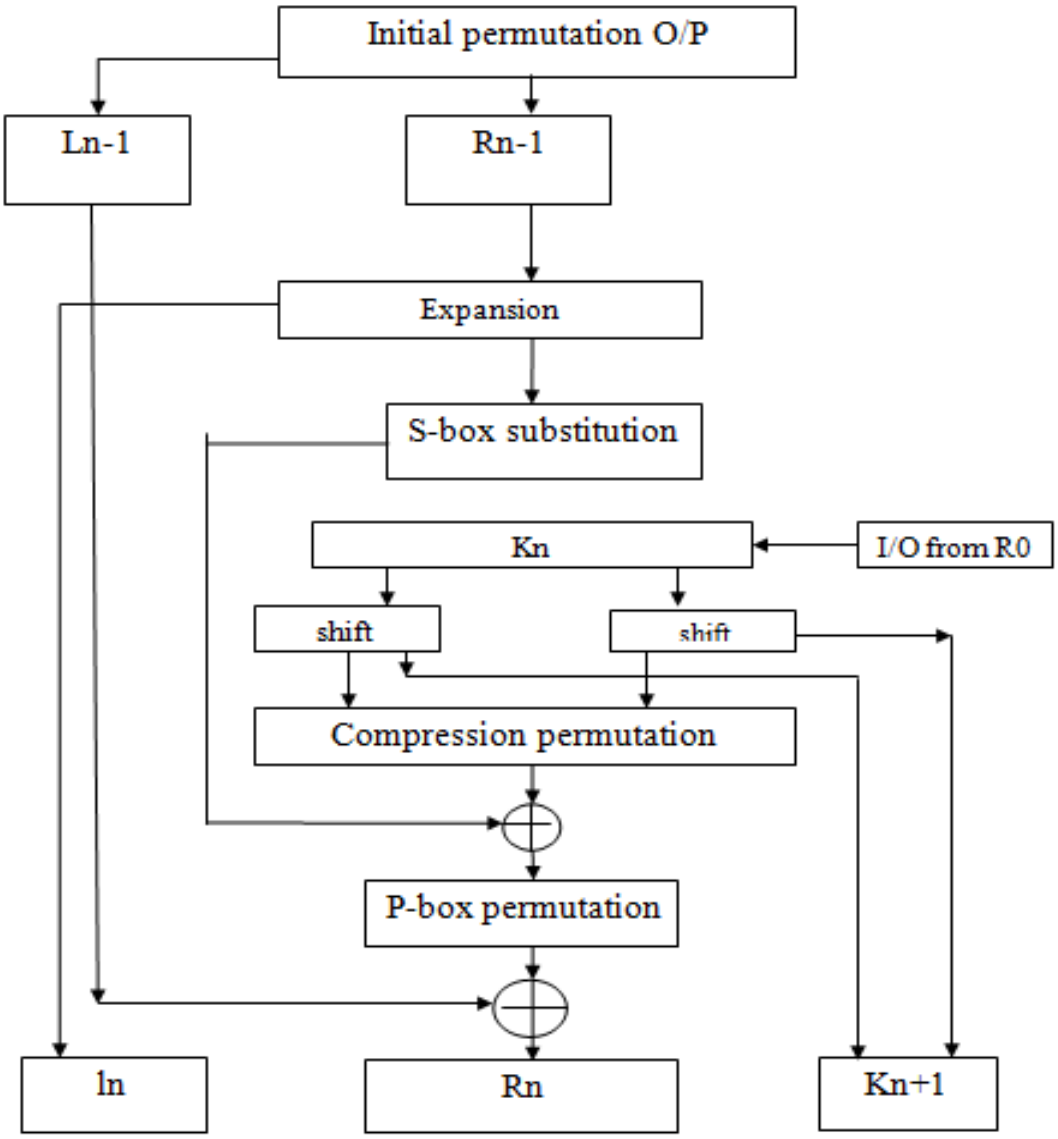

Figure 4.1: Single round of encryption and decryption

4.1.2 Key Transformation

The keys reduced from 64 bit to 56 bit by removing every eighth bit, which are sometimes used for error checking. 16 different 48 bit sub-keys are then generated. This is achieved by splitting the 56 bit key into the two halves, and then circularly shifting them left by one or two bits, depending on the round. The 48 of the bits are selected. They are shifted in different groups of key bits which are used in each sub key [15]. This process is called compression permutation due to transposition of bits and reduction of the overall size, which is shown in figure 4.2.

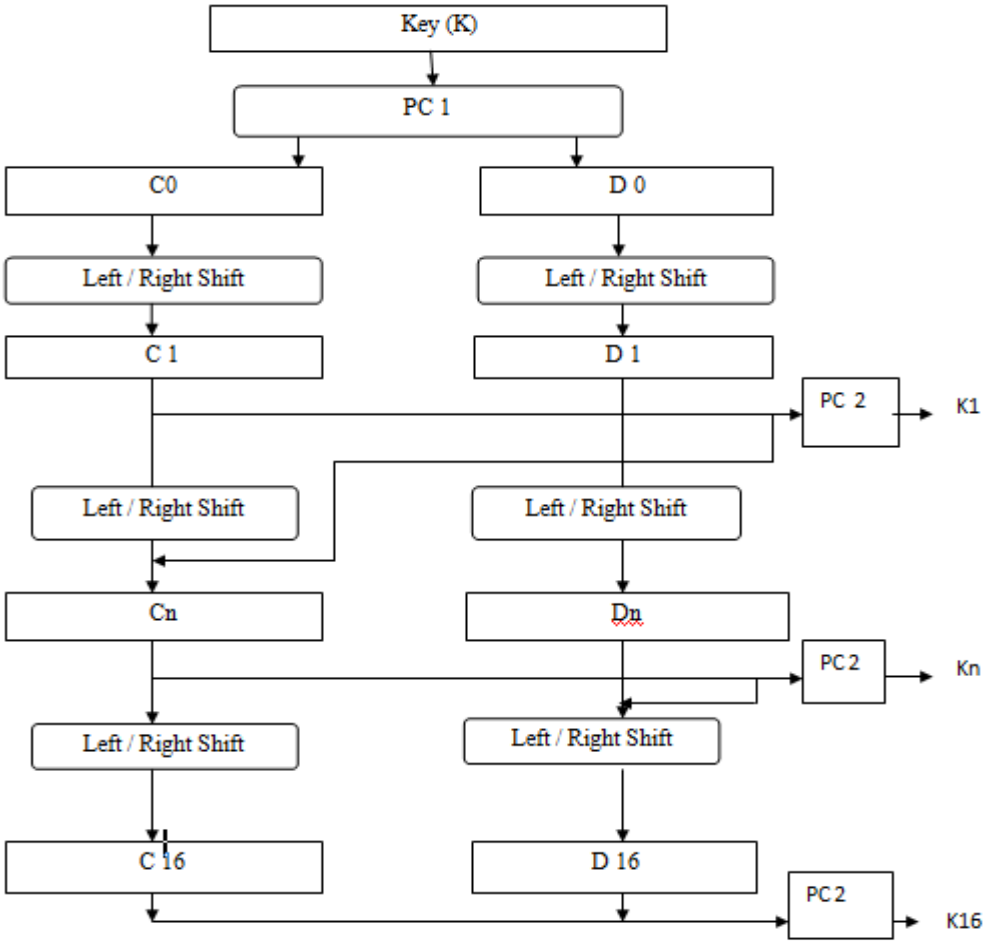

Figure 4.2: Key Transformation

4.1.3 Permutation

The substitution process consists of 32 bit output that undergoes a straight forward transition using P-box. After completing all rounds 64 bit output is obtained by recombining two 32 bit

half blocks. This transposition attracts first and fourth bits to appear twice in the output which becomes the fourth input bit and results the fifth and the seventh output bit. Similarly final permutation is performed on it and the resulting 64 bit block is the desired DES encrypted cipher text of the input plain text block [15]. The initial permutation and the inverse permutation used in the proposed algorithm are as follow:

Table 4.1: Initial Permutation

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Table 4.2: Inverse Permutation

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	2	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

4.1.4 Expansion Permutation

The half of the block undergoes a permutation after key transformation. In this function, the expansion and the transposition are achieved simultaneously by allowing the first and fourth bits in each block for bit block to appear twice in the output that is the fourth input bit becomes the fifth and the seventh output bit [16], which is discussed in table 4.3.

Table 4.3: Expansion permutation

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	33

The expansion permutation achieves 2 things:

- It increases the size of the half block from 32 to 48 bit. These same number of bit as in compressed key subset is important because the next operation is to XOR the two new blocks together. It produces a long string of data for the substitution operation.
- The first and the 4^{th} bits appearing in the second S-boxes affect two substitutions. This effect is caused by the dependency of the 25^{th} output bit on the input bits.

4.1.5 XOR Operation

XOR operation performed with the appropriate subset key for that round and gives 48 block codes, which is sent to the substation block for further operation.

4.1.6 Substitution

After XOR operation the next operation is to perform substitution on the expanded block. There are eight substitution boxes called S-boxes. The first S-box operates on the first 6 bits

of the 48 bit expanded block, the second S-box on the next 6 and so on. Each S-box operates from a table of 4 rows and 16 columns, whereas each entry on the table is a 4 bit number. The first and the 6 bits combine to form a two bit number corresponding to a row number, the second and fifth bit combine to form a 4 bit number corresponding to a particular column [16]. The net result of the substitution phase is 48 bit blocks that are then combined to form a 32 bit block. It is the non-linear relationship of the S-boxes that really provides DES with its security.

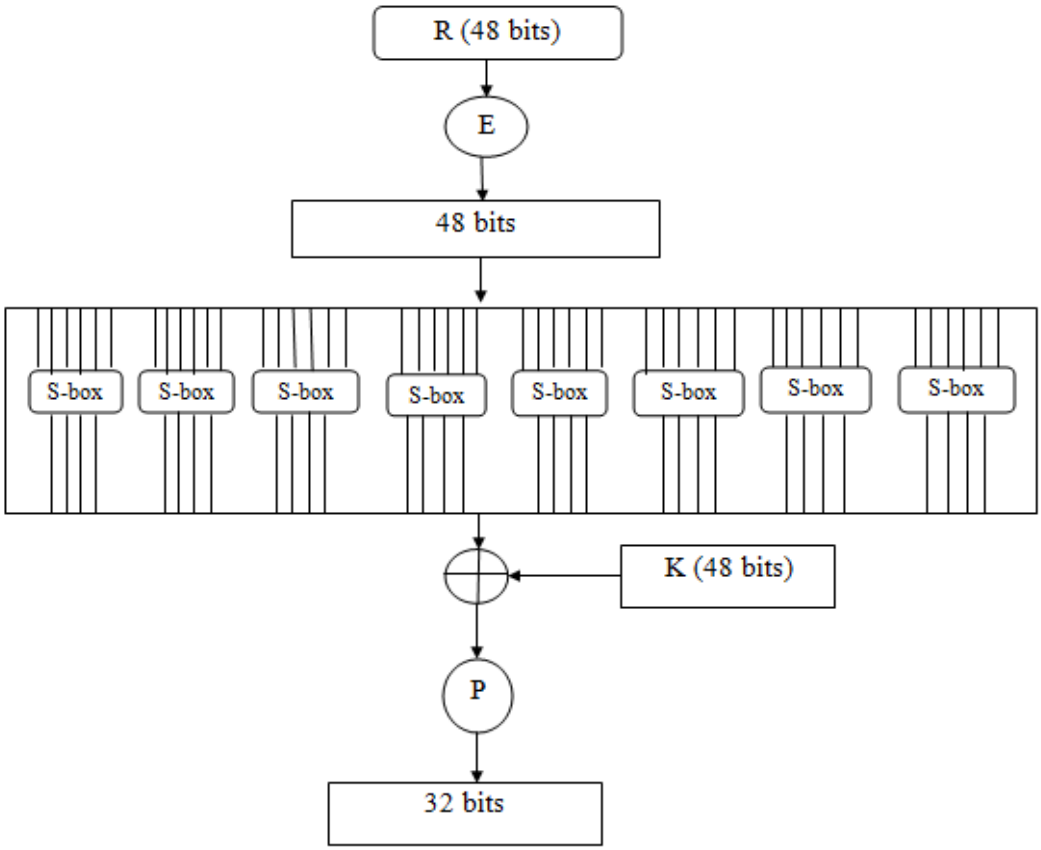

Figure 4.3: Calculation function f(R, K)

4.1.7 Explanation of modified TDES scheme

During the encryption process, the A_{16} and B_{16} of the final iteration has to be interchanged as shown in the fig, whereas the decryption process remains the same of the encryption process except that A1 and B1 should be interchanged and passed to the sub block. The key has to be used in the reverse order of $K_{1,16}$, $K_{1,15}$,$K_{1,14}$ - - - $K_{1,1}$ with $K_{1,1}$ on the left and $K_{2,1}$ on the right. The S_1 to S_8that placed on the left and S_9 to S_{16}that placed on the right have also to be

interchanged. Against the differential cryptanalysis attacks, the iteration number of the *f* function performed in each sub block during 16 rounds should be different, which decrease the probability of *N* round. In DES the *f* function performed in the sub block is repeated 8 times in the DES and it repeated 11 times during the performance of A_0 to B_{16}, 10 times during from B_0 to A_{16}, and 11 times during C_0 to B_{16}. This provides higher security as compare to DES. The modified TDES has a different iteration number of *f* functions in each sub block, it resists differential cryptanalysis. In modified 3DES, the S_1 to S_8 of the S-box is enlarged to S_1 to S_{16} and the S-box is chosen when each entry is suitable both for the SAC and the correlation. The cryptographic security is further improved in modified 3DES algorithm where each S-box is arranged randomly, so that it agrees with the condition as well as the correlation coefficient is increased to S_1 to S_{16}.

Therefore, it is known that when designing the S-box, SAC and the correlation coefficient, the S-box of the modified TDES is better than the DES. Hence, the modified TDES developed in this research work has been implemented in MATLAB software and it has been verified that its cryptographic security is superior to that of the TDES scheme.

Chapter 5: Results and Discussion

This chapter discusses the simulation results which have been obtained using MATLAB 2013, further, comparison of proposed algorithm with other cryptographic algorithms such as DES and TDES. It all has been done on the basis of throughput for the different file size.

5.1 Performance Evaluation Parameters

The time taken by the algorithm for the encryption and decryption of input file size that is computational encryption time and computational decryption time used for the processing of file is called as performance evaluation parameters.

5.1.1 Encryption Computational Time

The time taken by the algorithm for conversion of plaintext to cipher text is called as encryption computational time. The encryption throughput of the algorithms is then used to calculate this encryption time.

Table 5.1: Encryption Time for Different File Size

	Encryption Time (sec)		
INPUT FILE SIZE(Kb)	**DES**	**TDES**	**PROPOSED**
20	18	32	21
40	40	69	65
80	121	146	142
160	260	295	270
320	459	483	465
640	751	755	743

The input file size of 20Kb is the encryption time for DES, TDES and proposed algorithm which are as 18, 32, 21sec respectively as shown in table 5.1. Therefore, for the file size of 640 Kb the encryption time are 751, 755, 743 sec respectively. Thus, it shows that the proposed algorithm consumes more time than DES scheme but less time as compare to TDES scheme, for all the file sizes which in results increases the security of the system. Figure 5.1 has been created which shows the dependence of encryption execution time on the

file size. Therefore, the algorithm proposed in this work is faster than TDES. Finally, comparison of proposed approach with other algorithms has been carried out. Figure 5.1 compares the three schemes; DES, TDES and proposed algorithm.

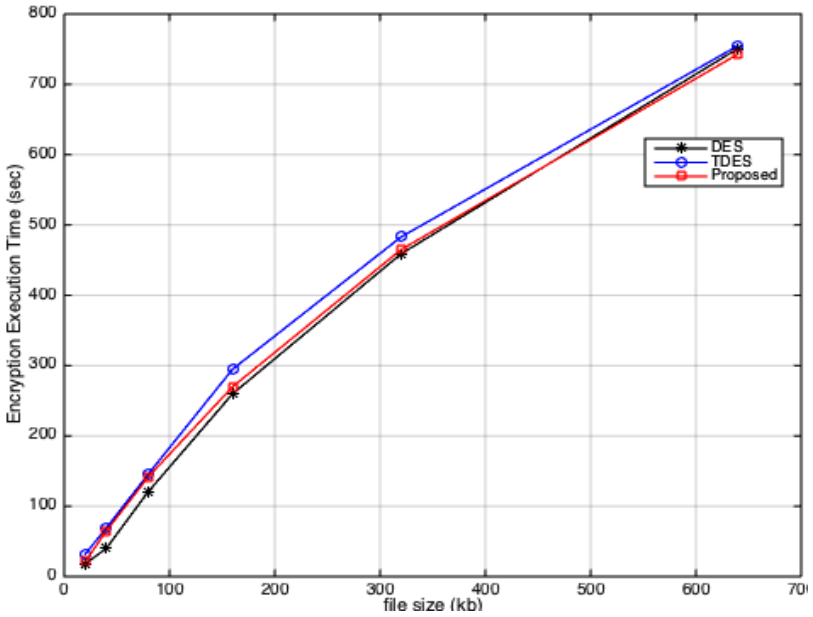

Figure 5.1: Encryption Time of different file sizes

In this figure 5.1 DES for file of 320kb takes 459 seconds and TDES takes 483 seconds whereas proposed algorithm takes 465 seconds.

5.1.2 Calculation of Encryption Throughput

Encryption Throughput (Kb/sec) = ∑ Input file size/ ∑ Encryption Execution time

∑ Input file size = 20+40+80+160+320+640

∑ Input file size = 1260 Kb

Encryption Throughput for DES

∑ Encryption Execution Time [DES] = 18+40+121+260+459+751

∑ Encryption Execution Time [DES] = 1649

Encryption Throughput [DES] = 1260/1649

Encryption Throughput [DES] = 0.764 Kb/sec

Encryption Throughput for TDES

∑ Encryption Execution Time [TDES] = 32+69+146+295+483+755

∑ Encryption Execution Time [TDES] = 1780

Encryption Throughput [TDES] = 1260/1780

Encryption Throughput [TDES] = 0 .707 Kb/sec

Encryption Throughput for Proposed Algorithm

∑ Encryption Execution Time [Proposed] = 21+65+142+270+465+743

∑ Encryption Execution Time [Proposed] = 1706

Encryption Throughput [Proposed] = 1260/1706

Encryption Throughput [Proposed] = 0.738 Kb/sec

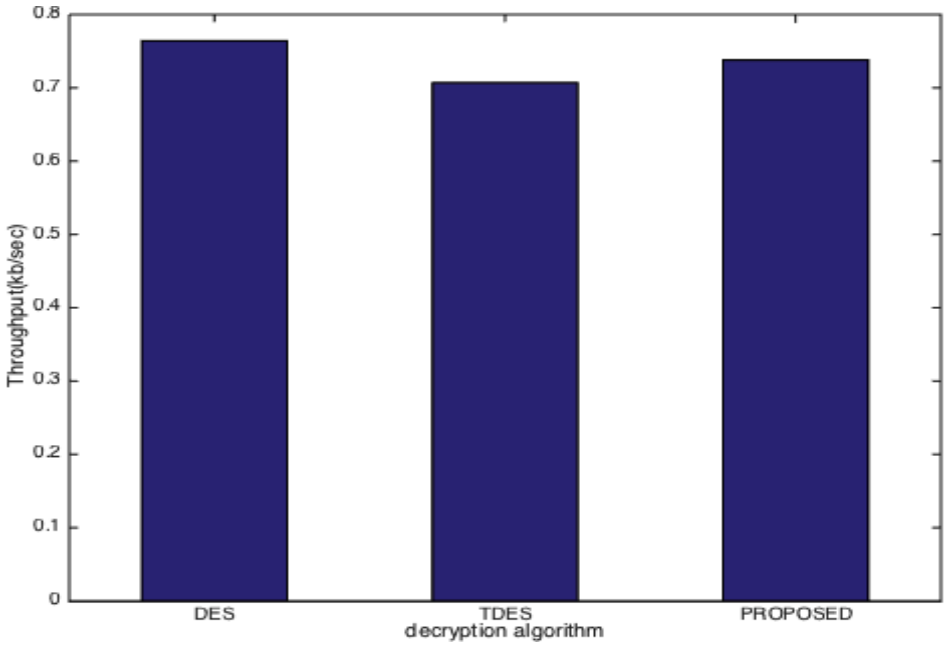

Figure 5.2: Throughput of Various Encryption Algorithms

From the above calculated values of throughput, it has been clear that the proposed algorithm provides optimized results in comparison to the other encryption algorithm and results are shown in figure 5.2. Throughputs of various encryption algorithms have been compared and it is clearly shown that proposed algorithm has higher throughput than that of TDES.

5.1.3 Outcome of work:

From the MATLAB plots it has been observed that the proposed approach takes more encryption time which increases the security of system for all types of file size when

compared with the other algorithms. Therefore, the results prove that when compared in terms of hacking and processing time, the proposed algorithm is optimized in comparison with other algorithms

5.2 Variations of Input File Size and Encrypted File Size in KB:

Table 5.2 shows linear relationship between input file size and the corresponding cipher-text file. Figure 5.3 is result obtained on proposed algorithm. Input file size and Encrypted file size follow an almost linear relationship.

Table 5.2: Variation of Input File Size and Encrypted File Size

S No.	INPUT FILE SIZE (kb)	ENCRYPTED FILE SIZE (kb)
1	20	31.5
2	40	55.9
3	80	150.3
4	160	223.5
5	320	401
6	640	966

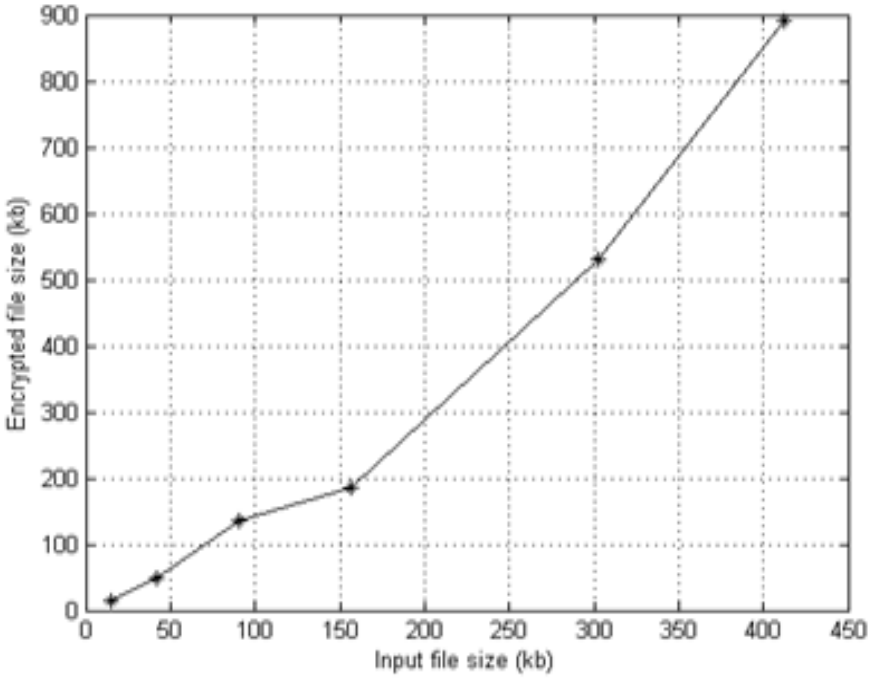

Figure 5.3: Variation between input file size and encrypted file size

For node = 5; the table 5.3 shows the various parameters having variable key length, constant file size and the table5.4 is the vice versa of table 5.3 shaving constant key length and variable file size. From these parameters the observations like execution time, hacking time are successfully obtained and their corresponding results are shown in the figure 5.4 and 5.5 respectively.

Table 5.3: For node 5 having variable key length and constant file size

Data size (KB)	Key length (bit)	Encryption time (sec)	Hacking time per key (msec)
50	8	87	52.8
50	16	113	2
50	32	140	0.043
50	64	339	0.003
50	128	408	$2.9x10^{-39}$

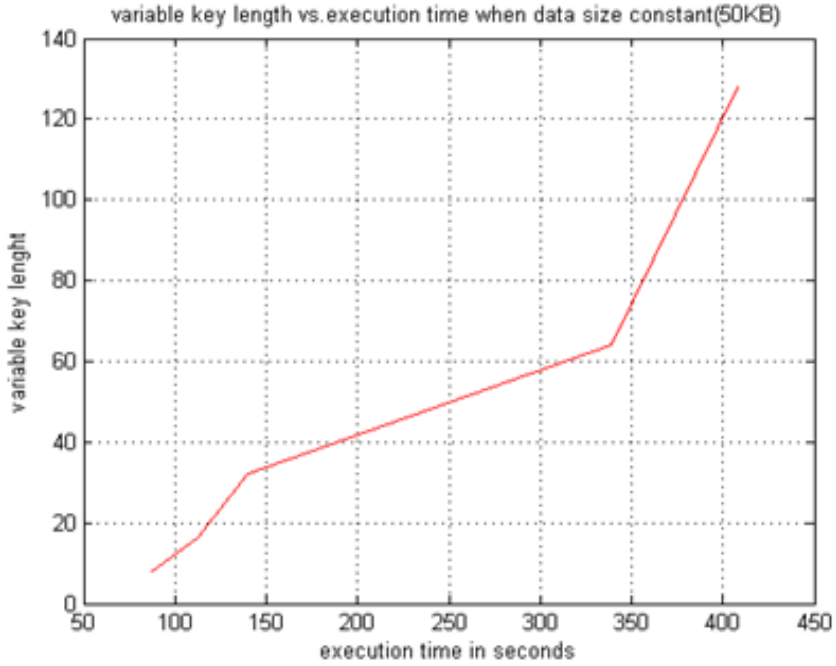

Figure 5.4: Variable key length v/s execution time having constant file size

Figure 5.4 has been created using table 5.3, input data size is fixed at 50kB with key length varying. The result shows that as key length increases in size execution time also increases.

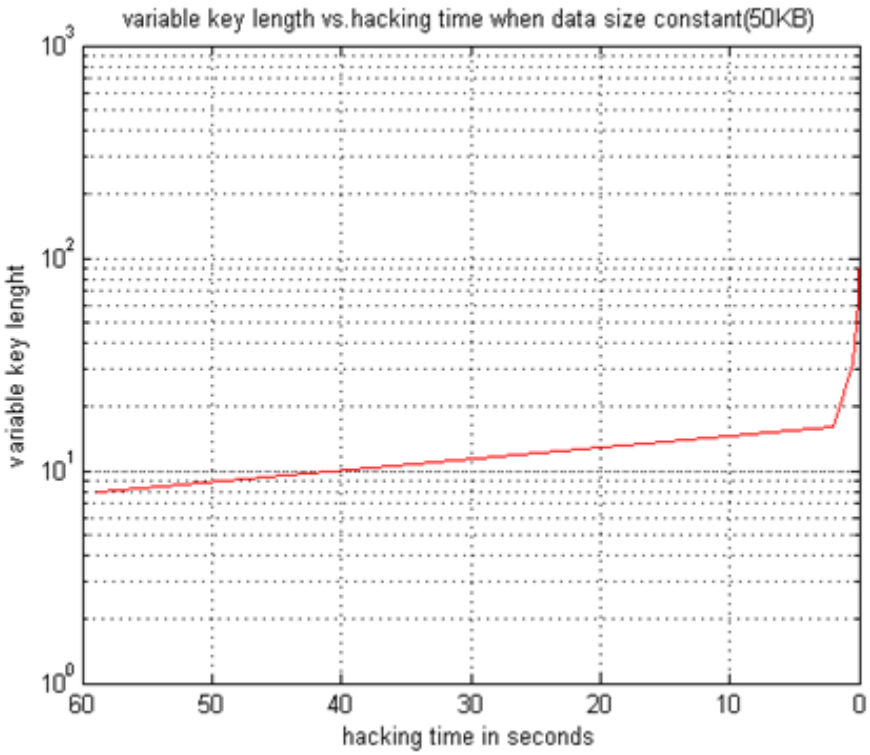

Figure 5.5: Variable key length v/s hacking time having constant file size

Figure 5.5 is plotted using column 2 and 4 of table 5.3. This plot is simulated in MATLAB for the proposed algorithm. In this input data file size is fixed at 50kB keeping key length variable. The result shows that as key length increases, the hacking time decreases. With 16 bit key length hacking time taken per key is 2 milliseconds, whereas it reduces to 2.9×10^{-39} milliseconds with128 key length.

Table 5.4: For node 5 having 8 bit constant key length and variable file size

Data size (KB)	**Key length (bit)**	**Encryption time (sec)**	**Hacking time per key (msec)**
50	8	87	0.017
100	8	188	0.67
150	8	243	11
200	8	367	153
250	8	592	1.2×10^{14}

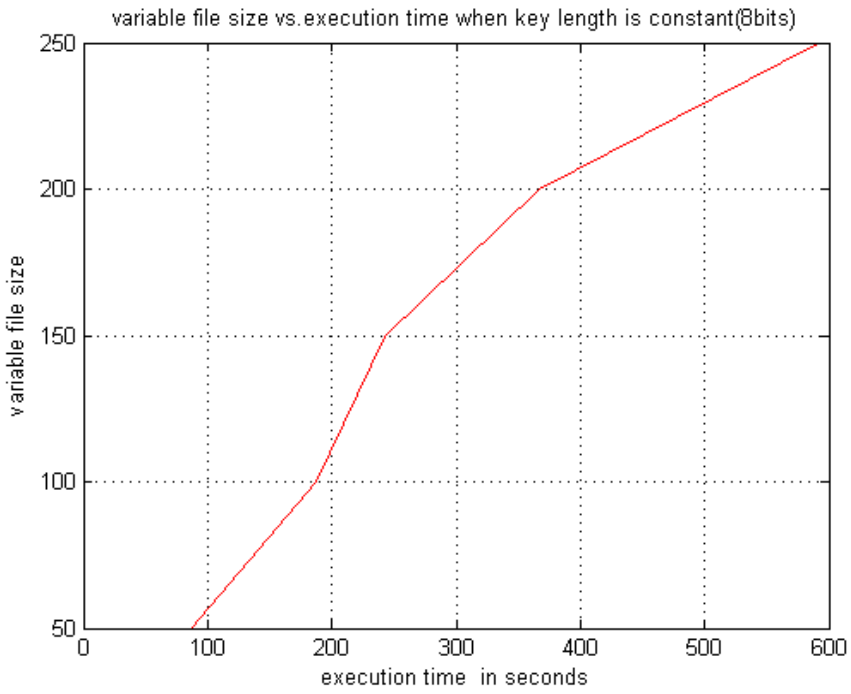

Figure 5.6: Variable file size v/s execution time having 8 bit constant key length

Figure 5.6 is result plotted between file size and execution time taken. The result shows that with 8 bits fixed key length the 50KB file size takes 87 seconds encryption time.

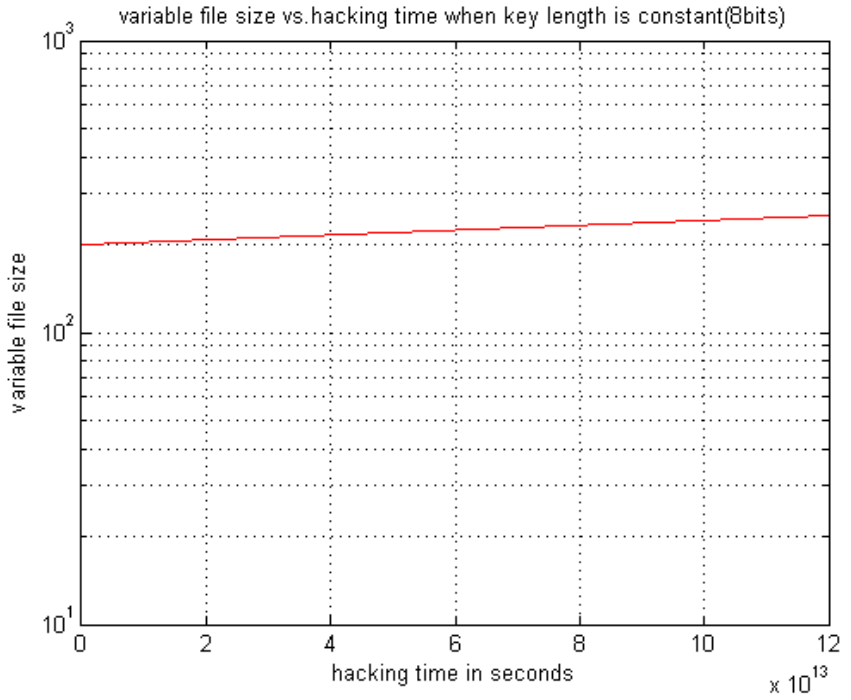

Figure 5.7: Variable file size v/s hacking time having 8 bit constant key length

Figure 5.7 shows the comparison between file size and hacking time per key. For 50KB file size the hacking time per key is 0.017 second. Thus, it is concluded that as the file size increases the hacking time per key also increases.

Table 5.5: For node 5 having 16 bit constant key length and variable file size

Data size (KB)	Key length (bit)	Encryption time (sec)	Hacking time per key (msec)
50	16	148	0.0289
100	16	306	1.139
150	16	418	12.1
200	16	549	260.1
250	16	1019	2.04×10^{14}

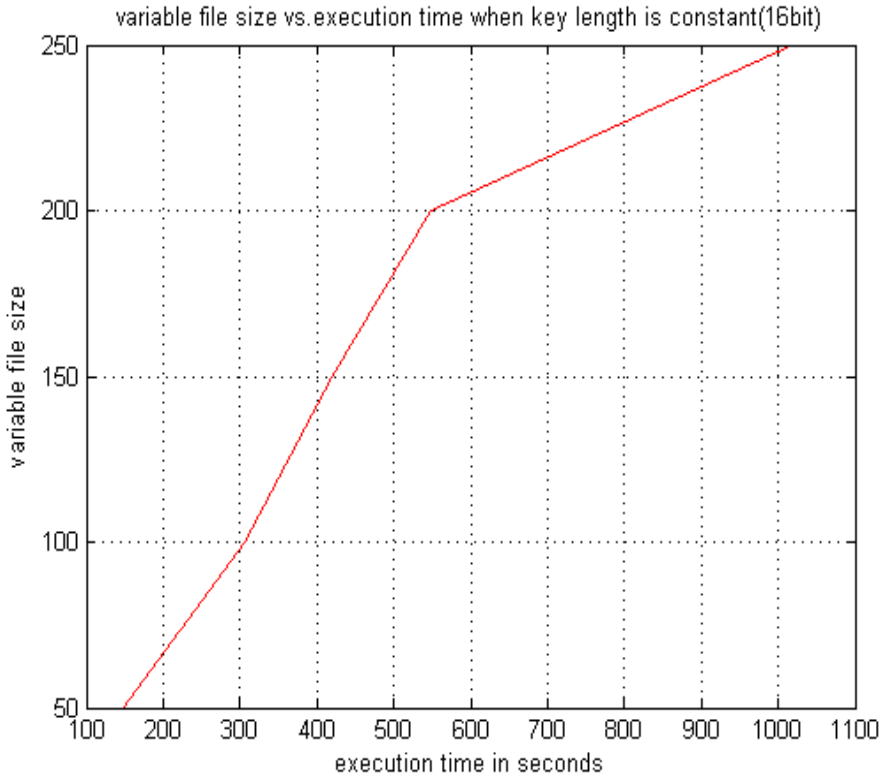

Figure 5.8: Variable file size v/s execution time having 16 bit constant key length

It is demonstrated in figure 5.8 that for proposed algorithm with fixed key size of 16 bits and input data file of 50KB file size, cryptographic model takes 148 second for encryption.

Figure 5.9 shows the comparison between file size and hacking time per key. For 50KB file size the hacking time per key is 194 second. Thus, it is concluded that as the file size increases the hacking time per key also increases.

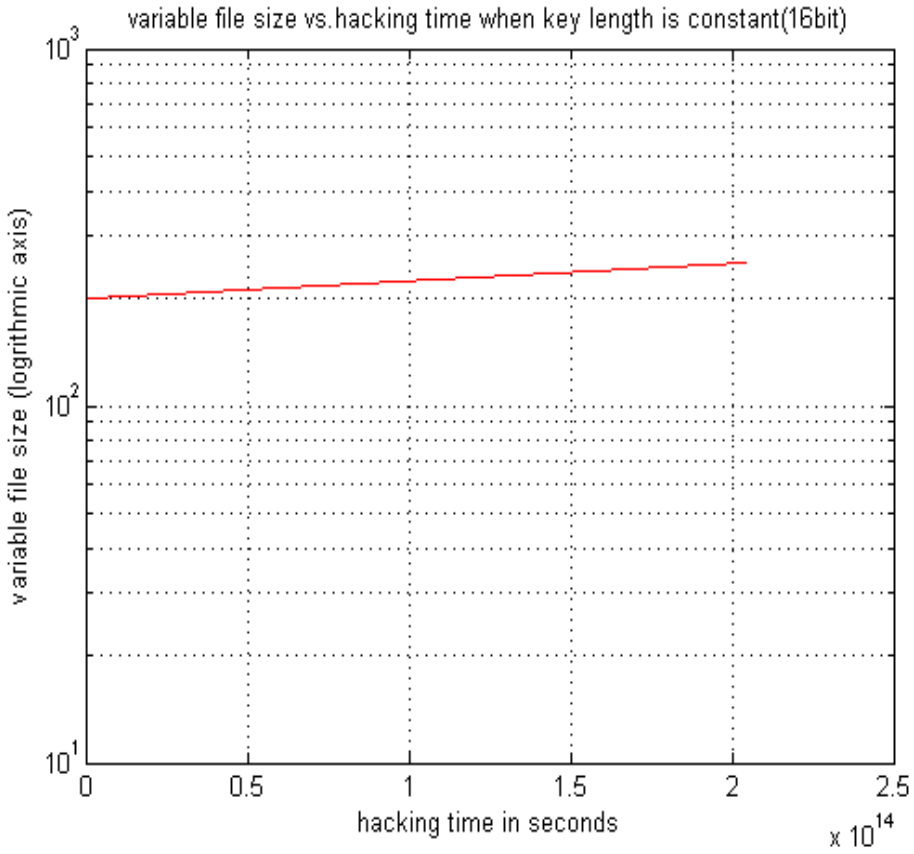

Figure 5.9: Variable file size v/s hacking time having 16 bit constant key length

For Node = 10

Table 5.6: For node 10 having variable key length and constant file size

Data size (KB)	Key length (bit)	Encryption time (sec)	Hacking time per key (msec)
50	8	288	194
50	16	361	6.6
50	32	468	0.141
50	64	1182	0.0099
50	128	1370	9.57×10^{-39}

Table 5.6 shows the results of proposed algorithm with variable key length and constant file size of 50KB for node 10.

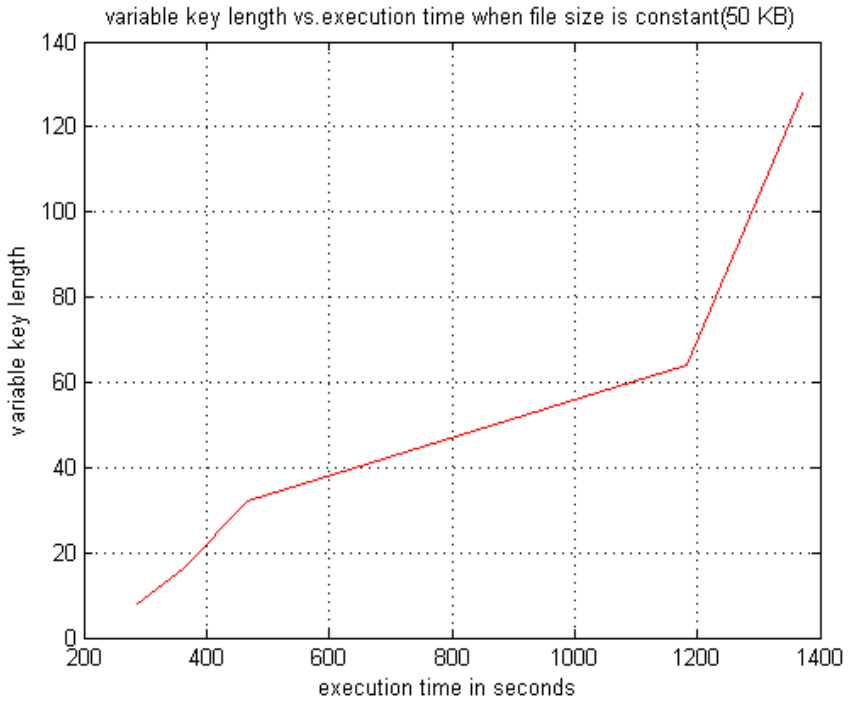

Figure 5.10: Variable key length v/s execution time having constant file size

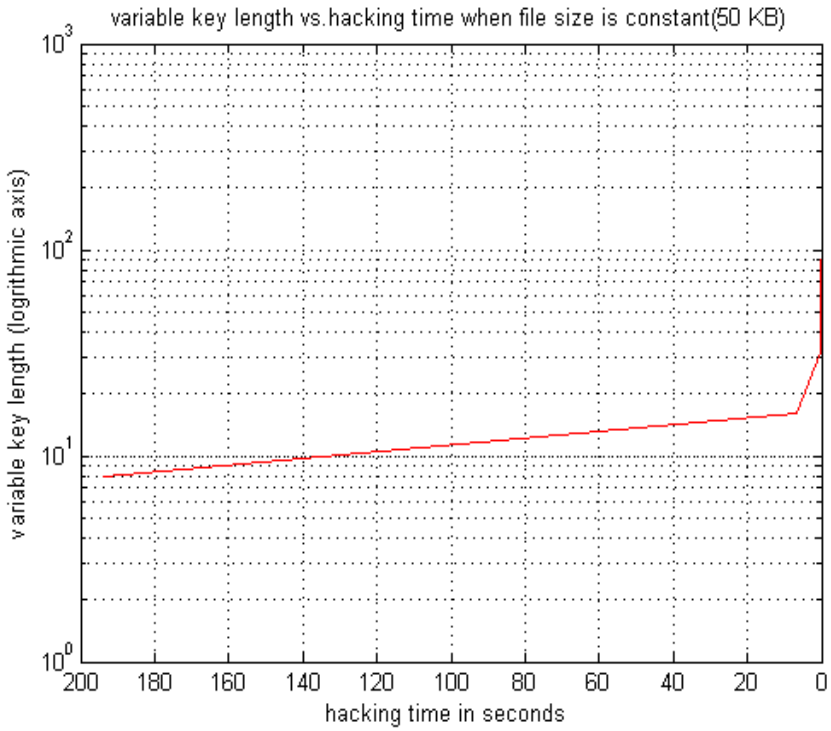

Figure 5.11: Variable key length v/s hacking time having constant file size

Figure 5.10 shows the plot between key length and execution time of constant file size of 50KB. When key length is 8 bit the execution time taken is 288 seconds. Figure 5.11 shows the key length versus hacking time results for proposed algorithm with data size fixed at 50KB. For 8-bit key length the hacking time per key is 194 milliseconds. Table 5.7 shows results of encryption time and hacking time for variable data size and key length fixed at 8 bits.

Table 5.7: For node 10 having 8 bit constant key length and variable file size

Data size (KB)	**Key length (bit)**	**Encryption time (sec)**	**Hacking time per key (msec)**
50	8	297	0.056
100	8	243	2.211
150	8	530	36.3
200	8	1008	504.9
250	8	1683	3.96×10^{14}

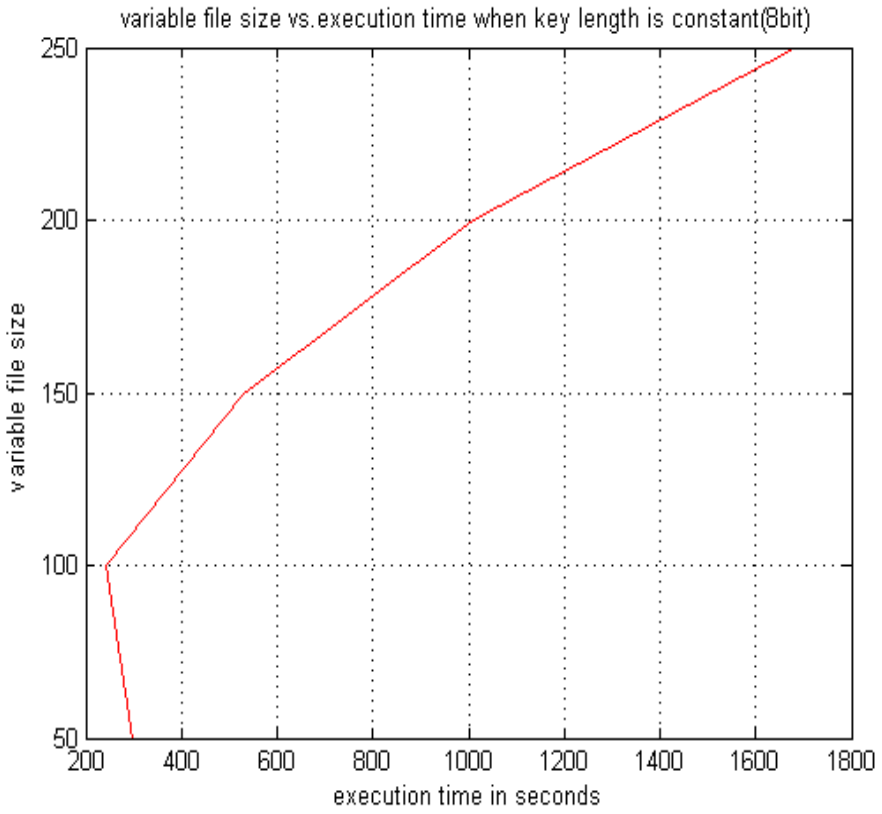

Figure 5.12: Variable file size v/s execution time having 8 bit constant key length

Figure 5.12 is MATLAB result which plots file size and execution time with key length fixed. The result is plotted using table 5.7. When input data size is 50KB, encryption time taken is 297 seconds for key length fixed at 8 bits.

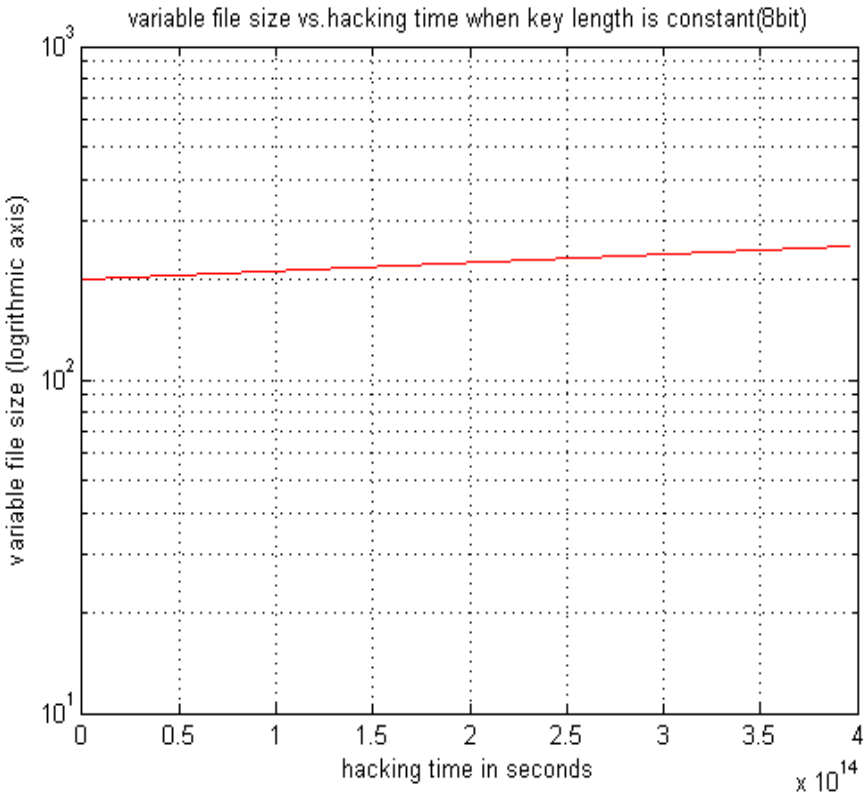

Figure 5.13: Variable file size v/s hacking time having 8 bit constant key length

Figure 5.13 is MATLAB result which plots file size and hacking time with key length fixed. The result is plotted using table 5.7. When input data size is 50KB, hacking time taken is 0.056 milliseconds for key length fixed at 8 bits.

Table 5.8: For node 10 having fixed key length and variable file size

Data size (KB)	**Key length (bit)**	**Encryption time (sec)**	**Hacking time per key (msec)**
50	16	485	0.095
100	16	1003	3.72
150	16	1376	39.9
200	16	1815	853.53
250	16	3061	6.73×10^{14}

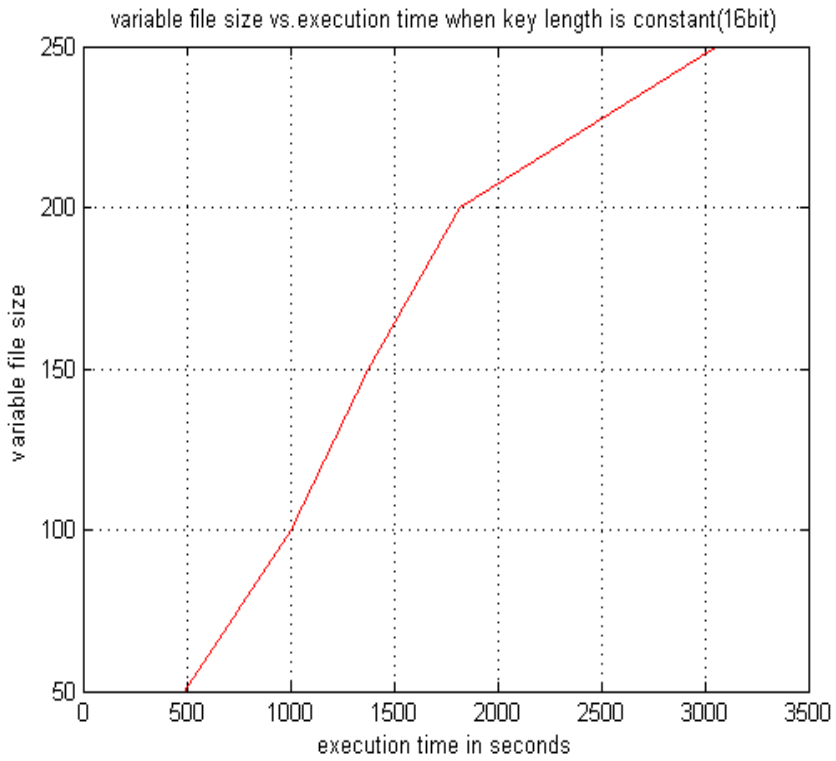

Figure 5.14: Variable file size v/s execution time having 16 bit constant key length

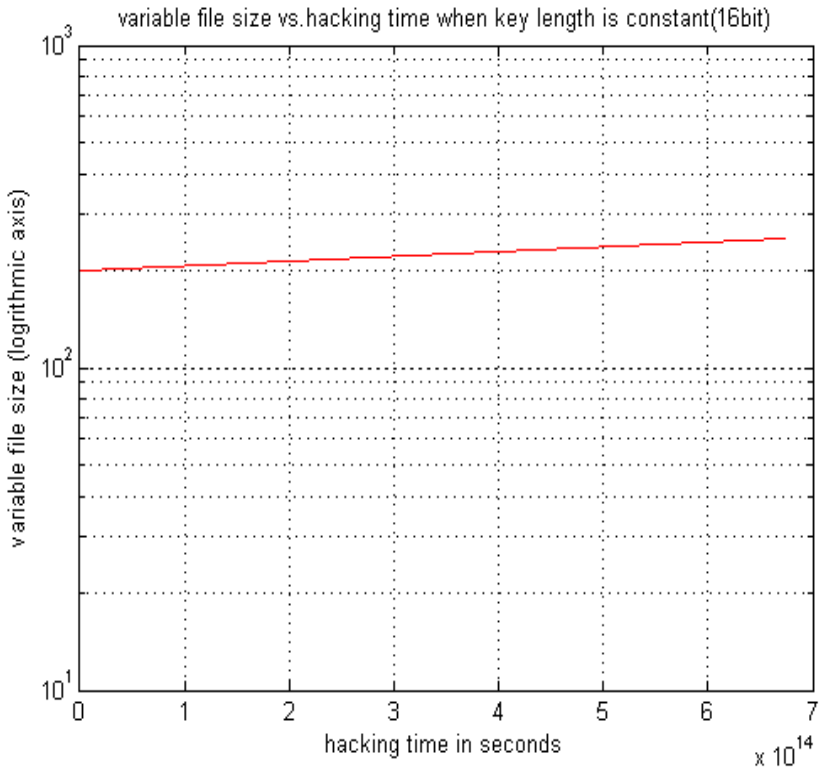

Figure 5.15: Variable file size v/s hacking time having 16 bit constant key length

Figure 5.14 is MATLAB result which plots file size and execution time with key length fixed. The result is plotted using table 5.8. When input data size is 50KB, encryption time taken is 485 seconds for key length fixed at 16 bits. Figure 5.15 is MATLAB result which plots file size and hacking time with key length fixed. The result is plotted using table 5.8. When input data size is 50KB, hacking time taken is 0.095 milliseconds for key length fixed at 16 bits.

Chapter 6: Conclusion and Future Scope of Research

The objectives of the thesis have been successfully achieved. Comparative study of various encryption algorithms has been done. From the Literature Survey; various observations and gaps have also been shown. Objectives have been drawn from the observations and gaps. The modified TDES approach has been successfully achieved based on the various performance parameters like key management, efficient key utilization, computational speed, memory usage and data security. The advancement in computational speed, power consumption and memory usage in this scheme makes the algorithm fast and more secured. Simulation results have been achieved using MATLAB 2013. The results clearly highlight that the proposed scheme takes less time as compared with TDES algorithm. In the proposed algorithm the available data has been used to generate the key and it also avoids the need of transmitting additional bits along with cipher text. Finally comparison of our approach with existing algorithms has also been done in terms of encryption and decryption time. The work can be further extended by considering the more round functions in the key generation process. The number of S-Boxes can be increased for a better secured model by keeping an eye on the processing time.

References

[1] B. Schneider, *Applied Cryptography*. 2nd ed. Wiley India, 2006.

[2] R. E. Klima, Neil P. Sigmon, *Classical Cryptology with Maplets*. 2nd ed. CRC press, 2010.

[3] W. Stallings and E. Schaefer, *An Introduction to Cryptography and Cryptanalysis*. 6th ed. Pearson International Publishing, 2013.

[4] A. W. Dent, "Choosing Key Sizes for Cryptography", *Elsevier Information Security Technical Report,* vol. 15, no. 1, pp. 21-27, 2010.

[5] H. Wang, M.Y. Wang and C.W. Wu, "Threshold Ring Signature into Certificate-less Public Key Cryptography", *IEEE Transaction on Computers*, vol. 18, no. 5, pp. 673-682, 2010.

[6] O. P. Verma, R. Agarwal, D. Dafouti and Shobha, "Performance Analysis of Data Encryption Algorithms", *Information Technology Delhi Technological University*, 2011.

[7] L. Buttyan, L. Czapand, I. Vajda, "Detection and Recovery from Pollution Attacks in Coding based Distributed Storage Schemes", *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 824-838, 2011.

[8] Y. Jin, B. Yangand Y. Makris, "Cycle Accurate Information Assurance by Proof Carrying Based Signal Sensitivity Tracing", *IEEE Transaction on Hardware-Oriented Security and Trust*, vol. 13, no. 7, pp. 373-384, 2013.

[9] L. Cheng, Q. Wen, Z. Jin, H. Zhang, "Cryptanalysis and improvement of a certificate-less encryption scheme in the standard model", *Springer Computer Science*, pp.163-173, 2013.

[10] O. Karaahmetoglu and M. T. Sakall, "A New Method to Determine Algebraic Expression of Power Mapping Based S-Boxes", *Elsevier Information Processing Letters*, vol.113, pp. 229-235, 2013.

[11] J. Chu, M. Benaissa," Error detecting AES using Polynomial Residue Number Systems", *IEEE Microprocessors and Microsystems*, vol. 37, no. 2, pp. 228-234, 2013.

[12] A. Barenghi, G. M. Bertoni, L. Breveglieri, and G. Pelosi, "A Fault Induction Technique Based on Voltage Underfeeding with Application to Attacks Against AES and RSA", *Journal of Systems and Software*, vol. 86, no. 7, pp. 1864-1878, 2013.

[13] H. Xu, S. Guoand K. Chen, "Building Confidential and Efficient Query Services in the Cloud with RASP Data Perturbation", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 2, pp. 322-335, 2014.

[14] J. Borst and M. Wamng, "Cryptography using Smart Cards", *Elsevier Computers in Industry*, vol. 36, no. 4, pp. 342-349, 2001.

[15] S. Mangard, M. Aigner and S. Dominikus, "A Highly Regular and Scalable AES Hardware Architecture", *IEEE Transactions on Computers*, vol. 52, no. 4, pp. 483-491, 2003.

[16] K. H. M. Wong, P. C. L. Hui, and A. C.K. Chan, "Cryptography and Authentication on RFID Passive Tags for Apparel Products", *Elsevier Computers in Industry*, vol. 57, no. 4, pp. 342-349, 2005.

[17] J. Ren and L. Harnsciences, "Generalized Ring Signatures," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 3, pp. 155-163, 2008.

[18] X. Du, M. Guizani, Y. Xiao, H. H. Chen, "A Routing-Driven Elliptic Curve Cryptography Based Key Management Scheme for Heterogeneous Sensor Networks", *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1223-1229, 2009.

[19] M. Alioto, M. Poli and S. Rocchi, "Differential Power Analysis Attacks to Pre-Charged Buses and General Analysis for Symmetric Key Cryptographic Algorithms", *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no 3, pp. 226-239, 2010.

[20] S. Pradheep kumar, R. Fareedha, M. Jeniefer kavetha, A. Geanremona and R. Juliajoyce, "A Secured Grid Based Route Driven PKC Scheme for Heterogeneous Sensor Network", *International Journal of Grid Computing and Applications*, vol. 1, no. 2, pp. 15-25, 2010.

[21] A. Shoufan, T. Wink, H. G. Molter, S. A. Huss, E. Kohnert, "A Novel Crypto processor Architecture for the Mc-Eliece Public-Key Cryptosystem", *IEEE Transactions on Computers*, vol. 59, no. 11, pp. 1533-1546, 2010.

[22] M. Wang, C. Su, C. Horng, C. Wu and C. Huang, "Single and Multi Core Configurable AES Architectures for Flexible Security," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 18, no. 4, pp. 541-552, 2010.

[23] T. Monoth, B. Anto, "Tamperproof Transmission of Fingerprints using Visual Cryptography Schemes", *Science Direct Computer Science*, vol. 2, pp. 143-148, 2010.

[24] R. H Torres, G. A. Oliveira, W. A. R Souza and R. Linden, "Identification of Keys and Cryptographic Algorithms using Genetic Algorithm and Graph Theory", *IEEE Latin America Transactions*, vol.9, no.2, pp. 178-183, 2011.

[25] L. J. G. Villalba, J. G. Matesanz, D. R. Canas and A. L. S. Orozco, "Secure Extension to the Optimized Link State Routing Protocol", *IET Information Security*, vol. 5, no. 3, pp. 163-169, 2011.

[26] S. Bu, F. R. Yu, X. P. Liu, P. Mason and H. Tang, "Distributed Combined Authentication and Intrusion Detection with Data Fusion in High-Security Mobile Networks", *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 1025-1036, 2011.

[27] J. Liu and B. Yang, "Collusion Resistant Multicast Key Distribution Based on Homomorphic One Way Function Trees", *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, pp. 980-991, 2011.

[28] H. Yang, E. Osterweil, D. Massey, S. Lu and L. Zhang, "Deploying Cryptography in Internet Scale Systems on DNSSEC", *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 5, pp. 656-669, 2011.

[29] C. Y. Yan, R. H. Xiao, "Study of Block Algorithms Implement on Hardware in Information Security System", *IEEE Transactions on Computer and Information Security*, vol. 11, no. 1, pp. 598-593, 2011.

[30] A. Dogan, S. Bernaor and G. Saldamli, "Analyzing and Comparing the AES Architectures for their Power Consumption", *Springer Computer Science journal*, vol. 45, vol. 5, pp. 263-271, 2011.

[31] Z. Wan, J. Liu and R. H. Deng," A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing", *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 743-754, 2012.

[32] T. Li, N. Li, J. Zhang and I. Molloy, "Slicing; A New Approach for Privacy Preserving Data Publishing", *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 561-574, 2012.

[33] K. Sakiyama, Y. Li, M. Iwamoto and K. Ohta, "Information-Theoretic Approach to Optimal Differential Fault Analysis", *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp.109-120, 2012.

[34] I. Woungang, S. K. Dhurandher, V. Koo and I. Traore, "Comparison of Two Security Protocols for Preventing Packet Dropping and Message Tampering Attacks on AODV-Based Mobile Networks", *IEEE International Workshop on Mobility Management in the Networks*, Vol. 12, No. 1, pp. 1037-1041, 2012.

[35] S. Verma, R. Choubey and R. Soni, "An Efficient Developed New Symmetric Key Cryptography Algorithm for Information Security," *International Journal of Emerging Technology and Advanced Engineering,*" vol.2, no. 7, pp. 18-21, 2012.

[36] Y. Li, K. Ohta and K. Sakiyama, "New Fault Based Side-Channel Attack using Fault Sensitivity", *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 88-97, 2012.

[37] D. Anand, V. Khemchandani, R.K. Sharma. "Identity Based Cryptography Techniques and Applications", *IEEE International Transaction on Computational Intelligence and Communication Networks*, vol.5, no.1, pp. 343-348, 2013.

[38] J. Hur, "Attribute Based Secure Data Sharing with Hidden Policies in Smart Grid", *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2171-2180, 2013.

[39] K. Nuida and G. Hanaoka, "On the Security of Pseudo randomized Information Theoretically Secure Schemes", *IEEE Transactions on Information Theory*, vol. 59, no. 1, pp. 635-653, 2013.

[40] S. Wei, J. Wang, R. Yin, J. Yuan, "Trade-off between Security and Performance in Block Ciphered Systems with Erroneous Cipher-texts", *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 636-635, 2013.

[41] K. K. Pandey, V. Rangari and S. K. Sinha, "An Enhanced Symmetric Key Cryptography Algorithm to Improve Data Security," *International Journal of Computer Applications,* vol. 74, no. 20, pp. 29-33, 2013.

[42] D. Schurmann and S. Sigg, "Secure Communication Based on Ambient Audio", *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 358-370, 2013.

[43] X. Zhao, S. Guob, F. Zhangc, T. Wanga, Zhijie Shic, Zhe Liud and J. Franc, "A Comprehensive Study of Multiple Deductions based Algebraic Trace Driven Cache Attacks on AES", *Elsevier Computers and Security*, vol. 39, pp. 173-189, 2013.

[44] Z. Cica, "AES Implementation with TDM Multiplexing for Internet Routers", *Elsevier Journal Lightwave Technology*, vol. 19, no. 5, pp.405-408, 2013.

[45] C. I. Fan, "Arbitrary-State Attribute Based Encryption with Dynamic Membership", *IEEE Transactions on Computers*, vol. 63, no.8, pp. 1951-1961, 2014.

[46] J. Li, X. Huang, X. Chen and Y. Xiang, "Securely Outsourcing Attribute Based Encryption with Check Ability", *IEEE Transaction on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2201-2210, 2014.

[47] J. K. Liu and M. H. Au, "Linkable Ring Signature with Unconditional Anonymity*",* *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 157-165, 2014.

[48] S. H. Seo, M. Nabeel, X. Ding and E. Bertino, "An Efficient Certificate-less Encryption for Secure Data Sharing in Public Clouds", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2107-2119, 2014.

[49] H. Lu, J. Li, and M. Guizani, "Secure and Efficient Data Transmission for Cluster-Based Wireless Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 750-761, 2014.

[50] J. Han, "Identity-Based Secure Distributed Data Storage Schemes", *IEEE Transactions on Computers*, vol. 63, no.4, pp. 941-953, 2014.

[51] Z. Stanisavljevic, J. Stanisavljevic, P. Vuleticandand, Z. Jovanovic, "COALA System for Visual Representation of Cryptography Algorithms", *IEEE Transactions on Learning Technologies*, vol.7, no. 2, pp. 178-190, 2014.

[52] D. He, S. Chan, S. Tang, "A Novel and Lightweight System to Secure Wireless Medical Sensor Networks", *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 1, pp. 316-326, 2014.

[53] X. Chen, J. Li, J. Ma, Q. Tang and W. Lou, " New Algorithms for Secure Outsourcing of Modular Exponentiations", *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 9, pp. 2386-2396, 2014.

[54] S. H. Seo, M. Nabeel, X. Dingand and Elisa Bertino, "An Efficient Certificate-less Encryption for Secure Data Sharing in Public Clouds", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2107-2119, 2014.

[55] A. A. Yavuz, "An Efficient Real Time Broadcast Authentication Scheme for Command and Control Messages", *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 10, pp. 1733-1742, 2014.

[56] M. M. Kermani, K. Tian, R. Azarderakhsh, S. B. Sarmadi, "Fault Resilient Lightweight Cryptographic Block Ciphers for Secure Embedded Systems", *IEEE Transactions on Embedded Systems Letters*, vol. 6, no. 4, pp. 89-92, 2014.

[57] T. Jung, X. Y. Li, M. Wan, "Collusion Tolerable Privacy Preserving Sum and Product Calculation without Secure Channel", *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 45-57, 2015.

[58] D. D. Chen, N. Mentens, F. Vercauteren, S. S. Roy, R. C. C. Cheung, D. Pao and I. Verbauwhede, "High Speed Polynomial Multiplication Architecture for Ring-LWE and SHE Cryptosystems", *IEEE Transactions on Circuits and System*, vol. 62, no. 1, pp. 157-166, 2015.

[59] S. Ma, Q. Huang, M. Zhang and B. Yang, "Efficient Public Key Encryption with Equality Test Supporting Flexible Authorization", *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 458-470, 2015.

[60] F. Xu, M. Curty, B. Qi and H. K. Lo. "Measurement Device Independent Quantum Cryptography", *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 21, no. 3, pp. 6603-6614, 2015.

.