Ph. D. Thesis

AN APPROACH TOWARDS THE DEVELOPMENT OF AN EFFICIENT SYMMETRIC KEY ENCRYPTION SCHEME

Submitted by

PAUL A.J.

DIVISION OF ELECTRONICS ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI – 682 022
INDIA

FEBRUARY 2012

AN APPROACH TOWARDS THE DEVELOPMENT OF AN EFFICIENT SYMMETRIC KEY ENCRYPTION SCHEME

A thesis submitted by

PAUL A.J.

for the award of the degree of

DOCTOR OF PHILOSOPHY
(Faculty of Engineering)

Under the guidance of

Dr. P. MYTHILI

And

Under the Co-guidance of

Dr. K. POULOSE JACOB

DIVISION OF ELECTRONICS ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI – 682 022
INDIA
FEBRUARY 2012

AN APPROACH TOWARDS THE DEVELOPMENT OF AN EFFICIENT SYMMETRIC KEY ENCRYPTION SCHEME

Ph. D. Thesis in the field of Computer Data Security

Author

PAUL A.J.
Research scholar
Division of Electronics Engineering
School of Engineering
Cochin University of Science and Technology
Kochi - 682 022, Kerala, INDIA
E-mail: paul_a_j@yahoo.com

Research Advisor

Dr. P. MYTHILI
Associate Professor
Division of Electronics Engineering
School of Engineering
Cochin University of Science and Technology
Kochi - 682 022, Kerala, INDIA
E-mail: mythili@cusat.ac.in

Co-Guide

Dr. K. POULOSE JACOB
Director
School of Computer Science Studies
Cochin University of Science and Technology
Kochi - 682 022, Kerala, INDIA
E-mail: kpj0101@gmail.com

February 2012

CERTIFICATE

This is to certify that the thesis entitled **"An Approach Towards The Development Of An Efficient Symmetric Key Encryption Scheme"** is a bonafide record of research work carried out by Mr. Paul A.J. under my supervision and guidance in the Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, Kochi with Dr. K. Poulose Jacob, Director, School of Computer Science Studies, Cochin University of Science and Technology, Kochi as Co-guide. No part of this thesis has been presented for any other degree from any other university.

Kochi
10th February 2012

Dr. P. Mythili, Ph. D.

(Supervising Guide)
Associate Professor
Division of Electronics Engineering
Cochin University of Science and Technology
Kochi, Kerala, INDIA

CERTIFICATE

This is to certify that the thesis entitled **"An Approach Towards The Development Of An Efficient Symmetric Key Encryption Scheme"** is a bonafide record of research work carried out by Mr. Paul A.J. under the supervision and guidance of Dr. P. Mythili, Associate Professor, Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, Kochi and myself as Co-guide. No part of this thesis has been presented for any other degree from any other university.

Kochi
10th February 2012

Dr. K. Poulose Jacob, Ph. D.

(Co-Guide)

Director
School of Computer Science Studies
Cochin University of Science and Technology
Kochi, Kerala, INDIA

DECLARATION

I hereby declare that the work presented in the thesis entitled "***An Approach Towards The Development Of An Efficient Symmetric Key Encryption Scheme***" is based on the original work done by me under the supervision of Dr. P. Mythili, Associate Professor, Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, Kochi as Research guide and Dr. K. Poulose Jacob, Director, School of Computer Science Studies, Cochin University of Science and Technology, Kochi as Co-guide. No part of this thesis has been presented for any other degree from any other institution.

Paul A.J.

Kochi
10th February 2012

ACKNOWLEDGEMENT

First and foremost, I would like to give special thanks to God Almighty for giving me wisdom and health to complete this research work.

I wish to place on record my profound sense of gratitude and thanks to my guide Dr. P. Mythili, Associate Professor, Division of Electronics Engineering, School of Engineering Cochin University of Science and Technology, Kochi, for her valuable guidance throughout this research work.

I am very much grateful to my co-guide, Dr. K. Poulose Jacob, Director, School of Computer Science Studies, Cochin University of Science and Technology, Kochi, for his valuable suggestions during the research work.

I would like to express my deep gratitude to Dr. R. Gopikakumari, Head, Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, Kochi for the encouragement and suggestions given to me during the research work.

I am very grateful to Dr. Varghese Paul, Professor in Information Technology, School of Engineering, Cochin University of Science and Technology, Kochi for his valuable suggestions during the initial stages of the research work.

I am very much indebted to Dr. T. Santhanakrishnan, Scientist, Naval Physical and Oceanographic Laboratories, Kochi for his valuable suggestions in improving the presentation of the thesis.

My sincere thanks are due to Dr. Babitha Roslind Jose, Asst. Professor, Division of Electronics Engineering, SOE, CUSAT, Mr. Baby Paul and Mr. Shanavaz K.T. research Scholars, SOE, CUSAT for their contributions in the preparation of the thesis.

I am very much indebted to all Faculty and Staff members of Division of Electronics Engineering, School of Engineering, Cochin University of Science and Technology, Kochi for their support and cooperation during the research work.

I acknowledge with gratitude the contributions of my colleagues Ms. Tessy A. Mathew, Ms. Dhanya R., Ms. Surya R., Ms. Lekshmi R. Nair and Mr. Saju A., department of Electronics and Communication Engineering, Musaliar College of Engineering and Technology, Pathanamthitta for their contributions during the preparation of the thesis.

My thanks are also due to my students Mr. Mathew John, Ms. Meenu Anna George, Ms. Meenu Susan, Ms. Devi J. and others for their contributions during the preparation of the thesis.

Paul A.J

ABSTRACT

In the recent years protection of information in digital form is becoming more important. Image and video encryption has applications in various fields including Internet communications, multimedia systems, medical imaging, Tele-medicine and military communications. During storage as well as in transmission, the multimedia information is being exposed to unauthorized entities unless otherwise adequate security measures are built around the information system. There are many kinds of security threats during the transmission of vital classified information through insecure communication channels. Various encryption schemes are available today to deal with information security issues. Data encryption is widely used to protect sensitive data against the security threat in the form of “attack on confidentiality”. Secure transmission of information through insecure communication channels also requires encryption at the sending side and decryption at the receiving side. Encryption of large text message and image takes time before they can be transmitted, causing considerable delay in successive transmission of information in real-time. In order to minimize the latency, efficient encryption algorithms are needed. An encryption procedure with adequate security and high throughput is sought in multimedia encryption applications. Traditional symmetric key block ciphers like Data Encryption Standard (DES), Advanced Encryption Standard (AES) and Escrowed Encryption Standard (EES) are not efficient when the data size is large. With the availability of fast computing tools and communication networks at relatively lower costs today, these encryption standards appear to be not as fast as one would like. High throughput encryption and decryption are becoming increasingly important in the area of high-speed networking. Fast encryption algorithms are needed in these days for high-speed secure

communication of multimedia data. It has been shown that public key algorithms are not a substitute for symmetric-key algorithms. Public key algorithms are slow, whereas symmetric key algorithms generally run much faster. Also, public key systems are vulnerable to chosen plaintext attack.

Symmetric-key cryptography has been and still is extensively used to solve the traditional problem of communication over an insecure channel. The communication technology has advanced over the recent years and as a consequence communication over networks has become faster demanding fast cryptographic transformations for high-speed secure communications. This has been the motivation behind the research work leading to the development of an efficient encryption scheme that is presented in this thesis.

In this research work, a fast symmetric key encryption scheme, entitled “Matrix Array Symmetric Key (MASK) encryption” based on matrix and array manipulations has been conceived and developed. Fast conversion has been achieved with the use of matrix table look-up substitution, array based transposition and circular shift operations that are performed in the algorithm. MASK encryption is a new concept in symmetric key cryptography. It employs matrix and array manipulation technique using secret information and data values. It is a block cipher operated on plain text message (or image) blocks of 128 bits using a secret key of size 128 bits producing cipher text message (or cipher image) blocks of the same size. This cipher has two advantages over traditional ciphers. First, the encryption and decryption procedures are much simpler, and consequently, much faster. Second, the key avalanche effect produced in the ciphertext output is better than that of AES.

The thesis is organized in six chapters. Chapter 1 discusses potential security issues involved in the storage and transmission of digital data in and around an information system and briefly explains cryptography and the various types of cryptographic tools available for information security services. Different mechanisms to deal with security attacks on digital data transmitted over communication networks are also presented. Comparison between the various types of cryptography, their limitations and their applications where they are most suited are given. It is also discussed how the available tools in combination can address various security issues that exist in communication of digital data over insecure channels. String and block ciphers are discussed and a list of popular encryption algorithms is also presented. The need for symmetric key encryption for secure transmission of information over insecure communication channels is indicated.

Chapter 2 explores the history and earlier developmental work on cryptography. The cryptography prevailed since World war-II has been reviewed in brief. Some of the symmetric key ciphers and the popular encryption standards such as DES and AES are discussed. Standard references for classical cryptanalysis are also indicated.

Chapter 3 describes in detail the concept and realization of the proposed MASK encryption technique. The encryption algorithm, based on matrix and array manipulations, using secret key and sub keys is discussed. Three major functional blocks of the encryption scheme viz. matrix initialization, key schedule, substitution and diffusion are explained. Basic test results obtained using plaintext messages and images are presented. Characteristics of the proposed encryption scheme and AES are compared. Results showing improvement on the key avalanche effect produced in AES by replacing the key schedule of AES with that of MASK are also included.

Chapter 4 presents the detailed tests and analysis conducted on the cipher MASK, with gray scale and colour images. Statistical analysis including histogram analysis, adjacent pixel correlation analysis and mean value analysis have been carried out and the results are presented. Comparison of the results obtained from MASK and AES is also presented. Measurements of encryption quality and encryption speed are carried out with different image sizes and the values are tabulated.

Chapter 5 presents security analysis of MASK encryption scheme. Security attacks such as statistical attack, ciphertext only attack, known plaintext attack, chosen plaintext attack, linear and non-linear attacks are considered. Statistical data using images and plaintext are obtained and presented. Results obtained from AES are also shown for comparison.

Chapter 6 gives the conclusions and scope for further research work.

CONTENTS

Page No.

LIST OF FIGURES

Page No.

LIST OF TABLES

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
ANSI	American National Standards Institute
CBC	Cipher Block Chaining
CFB	Cipher Feed Back
DCA	Differential Crypt Analysis
DES	Data Encryption Standard
DEA	Data Encryption Algorithm
ECB	Electronic Code Book
EDI	Electronic Data Exchange
EES	Escrowed Encryption Standard
FEAL	Fast Encryption Algorithm
FIPS	Federal Information Processing Standard
IACR	International Agency for Cipher Research
IDEA	International Data Encryption Algorithm
LCA	Linear Crypt Analysis
MAC	Message Authentication Code
MASK	Matrix Array Symmetric Key
NSA	National Security Agency
PKC	Public Key Cryptography
RSA	Rivest Shamir Adleman (Algorithm)
SKC	Symmetric Key Cryptography
SAFER	Secure And Fast Encryption Routine
TEA	Tiny Encryption Algorithm
TTP	Trusted Third Party

Chapter 1

Introduction

This chapter discusses potential security issues involved in the storage and transmission of digital data in and around an information system. Different mechanisms to deal with security attacks and various types of cryptographic tools available for information security services are discussed. The need for symmetric key encryption for secure transmission of information over insecure communication channels is indicated. The chapter also discusses how the available tools in combination can address various security issues that exist in communication of digital data over insecure channels.

1.1 Security Issues in Data Storage and Transmission

In the recent past information is being handled in digital format because it is easy to store, process and transmit digital data over long distances without loss of quality. Advances in computer science and communication technology have enabled easy access to information and facilitated electronic commerce around the world. The amount of trade conducted electronically has grown extraordinarily with widespread Internet usage. There are various activities happening through networks such as electronic money transfer, supply chain management, Internet marketing, online transaction processing, electronic data interchange (EDI), inventory management systems and automated data collection systems. In all these applications the security of information exchanged through networks is very critical and information security has become a serious matter of concern in recent times [1]. *Cryptography* is an important tool in modern electronic security technologies to protect valuable information resources on intranets, extranets, and the Internet [2]. It has been used historically as a means of providing secure communication between individuals, government agencies and military forces.

Over the centuries, an elaborate set of protocols and mechanisms have been created to deal with information security issues while the information was being conveyed by physical documents. During the last century, mathematical algorithms have been developed to encrypt classified and sensitive information. The objectives of information security cannot be achieved fully through mathematical algorithms and protocols alone, but requires procedural techniques and abidance of laws to achieve the desired result. The way information has been stored did not change much over the time in the past. Information has been typically stored and transmitted on paper. Much of the information presently resides on magnetic, optical or electronic media and is being transmitted via telecommunications systems. During these days, with digital systems, it has become very easy to copy and alter information as one would like. Thousands of identical copies could be made from a piece of information stored electronically and each of them is indistinguishable from the original. This has been very difficult when information was stored on paper. So it has become necessary to incorporate some means to ensure information security that is independent of the physical medium of recording or transmission. This would ensure that the objectives of information security rely solely on digital information itself. One of the fundamental tools used in information security is the signature. It has been (and still is) a building block for many other services such as non-repudiation, data origin, authentication, identification and witnessing to mention a few. With electronic information, the concept of signature is different in a way that it cannot simply be something unique to the signer and independent of the information signed. Electronic replication of it is so simple that appending a signature to a document not signed by the originator is almost a triviality. For dealing this in electronic format, analogues of the paper protocols currently in use are required. There are many aspects to information security associated with applications, ranging from secure commerce, payments through network communications and protecting passwords. Cryptography has been an essential tool for information security during storage and communication.

The objective of modern cryptosystems is not to provide perfect or risk-free security. Rather, the objective of cryptography-based security is to protect information resources by making unauthorized acquisition of the information or tampering with the information more costly than the potential value that might be gained. Because the value of information usually decreases over time, good cryptography-based security protects information until its value is significantly less than the cost of illicit attempts to obtain or tamper with the information. Good cryptography, when properly implemented and used, makes attempts to violate security cost-prohibitive.

1.2 Aspects of Information Security

An information system is said to be secure if three requirements are satisfied. First, the system (hardware and all required software) should be made available to the authorized users whenever it is required (this implies *Availability*). Second, the information should be available only to authorized users of the system (this implies *Confidentiality*). Third, the information available in a system should be authentic (this implies *Integrity*). These security aspects in an information system are shown in Figure 1.1.

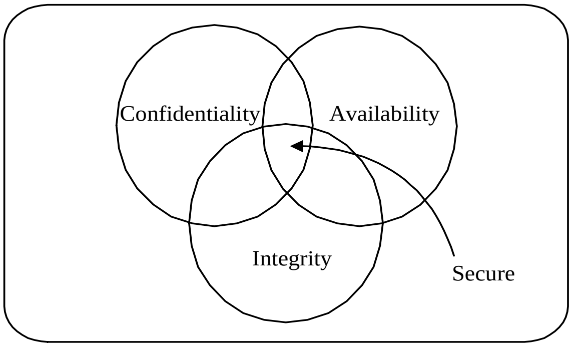
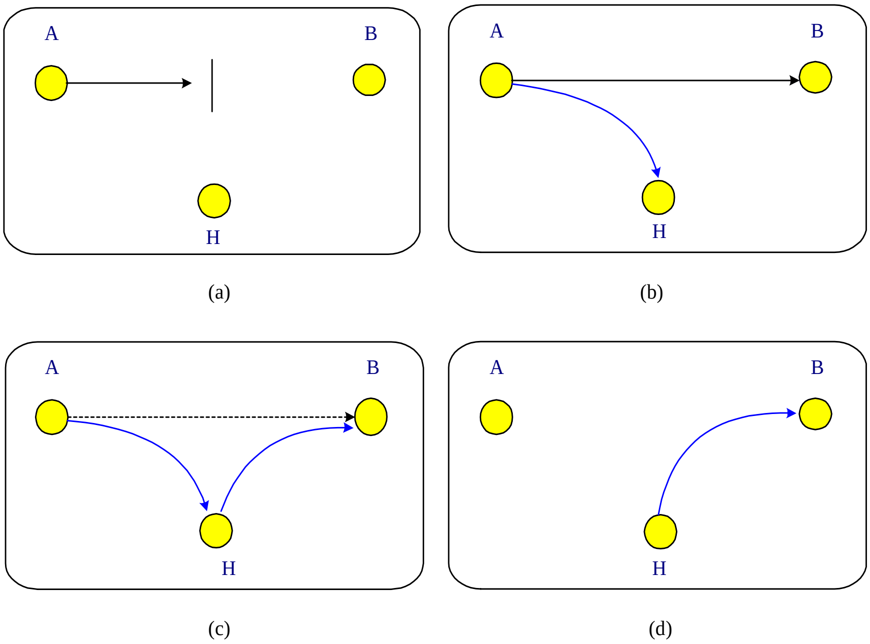


Figure 1.1. Security Aspects of an Information System.

1.2.1 Security Attacks

A security attack is an act that compromise on the security of information owned by an organization. The attacks could be launched when the information exists in a system or while the information is being transmitted over any communication networks or channels. When information is transmitted over networks, there exist certain security threats. Different security attacks during the transmission of information over open networked systems are illustrated in Figure 1.2 and discussed in the following sections.



(a) (b) (c) (d)

Figure 1.2. Security Attacks on Information System. (a) Attack on Availability (b) Attack on Confidentiality (c) Attack on Integrity and (d) Attack on Authenticity.

1.2.1.1 Interruption

It is a kind of attack launched by an opponent, intended to block the communication of information between legitimate users, in a networked system. The opponent tries to make the communication channel unavailable by tampering the communication line or by making the channel busy by continuously transmitting unwanted messages. Figure 1.2(a) depicts this kind of attack. In this Figure, an entity (person or machine) sending information is denoted as A, the intended receiving entity of the information is denoted as B and H represents man-in-the-middle who is trying to launch an attack on the information system.

1.2.1.2 Interception

This is a kind of attack on confidentiality by which a message transmitted by a person (or organization), over a network is being intercepted by a hacker H, for the purpose of releasing the message contents to other parties. The attacker would also be able to make a traffic analysis and find the parties with whom the originator of the message communicates. The intention of the hacker is to provide such information to an opponent (a company or organization) of the originator of the message in order to gain monitory benefits. Figure 1.2(b) depicts this kind of attack.

1.2.1.3 Modification

This is a kind of attack on integrity of message where by an opponent modifies the contents of a message sent by a legitimate user. The opponent intercepts a message sent by a legitimate user and makes modifications on the message. Then the modified message is forwarded to the intended recipient. Figure 1.2(c) depicts the attack on integrity of message.

1.2.1.4 Fabrication

This is a kind of attack on authenticity. A message is created by the attacker and the same is being sent to recipient in such a way that the recipient believes that the message has been originated from an authorized sender. Thus the receiver of the message is being cheated and the attacker could manage to gain monitory benefits or any other personal gains using this kind of attack. Sometimes the attacker may be working as an agent of some organization. This kind of attack is depicted in Figure 1.2(d).

1.2.2 Security Mechanisms

In a networked environment, security mechanisms should be incorporated into the appropriate protocol layer in order to provide some of the Open Systems Interconnect (OSI) security services. Some of the services are discussed in the following sections.

1.2.2.1 Data Encryption

Encryption is a tool that uses mathematical algorithms to transform data into a form that is not readily intelligible. The attack on confidentiality could be effectively addressed by the use of encryption of information.

1.2.2.2 Digital Signature

Digital signature is a kind of cryptographic transformation of a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery. The attack on authenticity could be addressed by the use of digital signature.

1.2.2.3 Access Control

There are many mechanisms such as the use of passwords, biometric information etc. to enforce access rights to information resources. This could be used to address the issues of unauthorized access to information system.

1.2.2.4 Data Integrity

A variety of methods could be used to check the integrity of data unit or stream of data units in an information package. Message digest or hash value of message generated using MAC/Hash algorithms could be attached to a message before transmission. At the receiving side the MAC code or hash value could be computed from the message and compared with the received MAC code or Hash value to ensure integrity of information.

1.2.2.5 Authentication Exchange

Authentication exchange is a mechanism intended to ensure the identity of an entity by means of information exchange. This could be facilitated by the use of User ID and Passwords. The user or entity upon request submits the user ID and password to the system before entering a transaction.

1.2.2.6 Notarization

Digital certificate obtained from a Trusted third party (TTP) could be used to ensure the identity of a person or entity if required before the communication.

1.2.2.7 Traffic Padding

It is a method of insertion of bits into gaps in a data stream to frustrate traffic analysis attempts by an opponent. This would confuse the opponent by making the opponent think that actual data transactions are going on in the channel.

1.3 Cryptography and Information Security

The proliferation of computers and communications systems in the 1960s have brought with it a demand from the private sector for means to protect information in digital form and to provide security services. Beginning with the work of Feistel at IBM in the early 1970s and culminating in 1977 with the adoption as a U.S. Federal Information Processing Standard for encrypting unclassified information, DES, the Data Encryption Standard [3], has become the most well known cryptographic mechanism in history. It remained the standard means for securing electronic commerce for many financial institutions around the world. Achieving information security in an electronic society requires a vast array of technical and legal skills. The technical means is being provided through cryptography.

Many organizations today have computerized information system, where all the computers in one department are being connected to a server via some kind of local area network (LAN). Further, different departments are also being provided with computer connectivity so that information exchanges could be made between the departments as and when required. Some large organizations have offices spread over a wide territory and computers in these offices have been interconnected via wide area network (WAN). Large organizations could use public networks for the connectivity outside its premises and could also depend on Internet for its data transfer requirements. These public networks or the Internet do not guarantee any security of information being communicated over these networks. Individuals using internet for online business and banking applications also encounter security problems leading to diversion of funds and confidential information such as user ID, passwords etc. Anyone who has access to these public networks could (if intended) intercept the information being sent through them. Further, anyone can alter the contents and forward the message to the intended recipient or anyone can masquerade as someone else and send messages to cheat people. Thus, it could be seen that many organizations and individuals depend upon open public networks that are not secure for information transfer. The information being transmitted over these networks are subject to various kinds of security risks as discussed earlier. To counteract these security risks, security mechanisms have to be introduced and security services have be provided while messages are being created and transmitted over insecure communication channels.

Cryptography is central to managing these kinds of risks involved in information communication over insecure networks or communication channels. Information exchange plays a vital role in almost every aspect of human activities. To achieve this, there are server computers and networks through which all other systems are interconnected. Huge amounts of data are moving over these kinds of communication networks. Security of information stored in computer system's storage units as well as that which is being transferred through the communication network have to be ensured so that the information does not reach unauthorized hands for misuse. In the recent years, protection of information in digital form has become more important as there are many kinds of security attacks on information systems. Image and video encryption have applications in various fields including Internet communications, multimedia systems, medical imaging, Tele-medicine and military communications [4]. An encryption procedure with adequate security and high throughput is sought in multimedia encryption applications. Traditional block ciphers like Data Encryption Standard, Advanced Encryption Standard [5] and Escrowed Encryption Standard [6] are not efficient encryption schemes. High throughput encryption and decryption are becoming increasingly important in the area of high-speed networking [7]. Fast encryption algorithms are needed these days for high-speed secure communication of multimedia data [8]. Public-key cryptographic algorithms are slow, whereas Symmetric-key cryptographic algorithms generally run much faster [9]. Symmetric-key cryptography has been and still is extensively used to solve the traditional problem of communication over an insecure channel [10]. During communication, information is being received and misused by adversaries by means of facilitating attacks at various nodes as well as on the lines used in communication [11]. Data encryption using cryptographic methods is the most effective means to counteract the security attacks [12] launched against any information system. The goals of cryptography are given in the following section.

1.3.1 Cryptographic Goals

As mentioned earlier, in Section 1.2, there are four basic security objectives upon which any other objectives could be derived. These are 1) Privacy or confidentiality 2) Data integrity 3) Authentication and 4) Non-repudiation. A fundamental goal of cryptography has been to adequately address these four areas in both theory and practice. Cryptography is about the prevention and detection of cheating and other malicious activities. Cryptography, over the ages, has been an art practiced by many who have devised ad-hoc techniques to meet some of the information security requirements. The last twenty five years have been a period of transition as the discipline moved from an art to a science. There are now several international scientific conferences devoted exclusively to cryptography and also an international scientific organization, the International Association for Cipher Research (IACR), aimed at fostering research in the area. Many research papers have appeared in international journals and conference proceedings. Diffie and Hellman [13] introduced trapdoor one-way functions. Merkle [14] described a means to obtain public-key encryption schemes. The basic concepts of cryptography are being treated quite differently by various authors, some being more technical than others. Brassard [15] provided a concise and technically accurate account. Schneier gave a less technical but very accessible introduction. Saloma [16], Stinson [17] and Rivest [18] presented more mathematical approaches. Diffie and Hellman [19] makes a comparison of encryption scheme with a resettable combination lock. Kerchoffs' desiderata [20] had been originally created in French and the translation made available by Kahn [21]. Shannon [22] suggested desirable features of good cryptographic transformations.

1.3.2 Cryptographic Transformations

A cryptographic transformation is a procedure that changes an intelligible message (or data) into an apparently unintelligible message (or data) by using logical and/or arithmetic operations. Usually, the transformation is performed in association with secret information called *key*. Let *K* denote a set called the *key space.* An element of *K* is called a *key.* Each element $e \varepsilon K$ uniquely determines a bijection from *M* (Message space*)* to *C* (ciphertext space), denoted by *Ee, is* called an encryption function or an encryption transformation. *Ee* must be a bijection if the process is to be reversed and a unique plaintext message recovered for each distinct ciphertext. For each $d \varepsilon K$, *Dd* denotes a bijection from *C* to *M* then *Dd* is called a decryption function or decryption transformation. The process of applying the transformation *Ee* to a message $m \varepsilon M$ is usually referred to as encrypting *m* or the encryption of *m*. The process of applying the transformation *Dd* to a ciphertext *c* is usually referred to as *decrypting c* or the *decryption* of *c*. An encryption scheme consists of a set $\{Ee: e \varepsilon K\}$ of encryption transformations and a corresponding set $\{Dd: d \varepsilon K\}$ of decryption transformations with the property that for each $e \varepsilon K$ there is a unique key $d \varepsilon K$ such that $Dd=(Ee)^{-1}$ that is, $Dd(Ee(m)) = m$ for all $m \varepsilon M$. An encryption scheme is referred to as a *cipher*. The keys *e* and *d* in the preceding definition are referred to as a *key pair* and sometimes denoted by (*e, d*)*, e* and *d* can be same also. To construct an encryption scheme requires one to select a message space *M,* a ciphertext space *C,* a key space *K,* a set of encryption transformations $\{Ee: e \varepsilon K\}$, and a corresponding set of decryption transformations $\{Dd: d \varepsilon K\}$. An encryption scheme could be used as follows for the purpose of achieving confidentiality. Two parties X and Y first secretly choose or secretly exchange a key pair (e, d). At a subsequent point in time, if X wishes to send a message $m \varepsilon M$ to Y, X computes $c = Ee(m)$ and transmits this to Y. Upon receiving c, Y computes $m = Dd(c)$ and hence recovers the original message m. Using transformations that are similar and characterized by keys, if some particular encryption/decryption transformation is revealed then one does not have to redesign the entire scheme but simply change the key. It is a sound cryptographic practice to change the keys (encryption/decryption transformation) frequently. A fundamental premise in cryptography is that the sets M, C, K, $\{Ee: e \varepsilon K\}$, $\{Dd: d \varepsilon K\}$ are public knowledge. When two parties wish to communicate securely using an encryption scheme, the only thing that they keep secret is the particular key pair (e,d) which they plan to use, and which they must decide in advance. One can gain additional security by keeping the class of encryption and decryption transformations secret but one should not base the security of the entire scheme on this approach. History has shown that maintaining the secrecy of the transformations is very difficult indeed. An encryption scheme is said to be breakable if a third party, without prior knowledge of the key pair (e,d), can systematically recover plaintext from corresponding ciphertext within some appropriate time frame. It is possible to break an encryption scheme by trying all possible keys so as to find out the actual key used by the communicating parties (assuming the class of the encryption functions is public knowledge). This is called an exhaustive search of the key space. It follows then that the possible number of keys (i.e. the size of the key space) should be large enough to make this approach computationally infeasible. It is the objective of designer of an encryption scheme to make sure that exhaustive key search method will not help crypt analysis that yield plaintext from ciphertext.

1.3.3 Types of Cryptographic Transformations

There are several ways of classifying cryptographic algorithms. They are categorized based on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms being used are listed below.

1) Symmetric Key Cryptography (SKC) using a single key for both encryption and decryption.

2) Public Key Cryptography (PKC) using separate keys for encryption and decryption

3) Hash Functions that use a mathematical transformation to irreversibly 'encrypt' information without using any key.

1.3.3.1 Symmetric Key Cryptography (SKC)

With symmetric key cryptography, a single key is used for both encryption and decryption. A sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key (or rule set) to decrypt the message and recover the plaintext. With this form of cryptography, it is obvious that the key must be made known to both sender and receiver of information, and that the key must be kept secret.

The biggest difficulty with this approach, of course, is the distribution of the key. Symmetric key cryptographic schemes are generally categorized as being either stream ciphers or block ciphers. Stream ciphers operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing. The symmetric key cryptosystem is illustrated in Figure 1.3.

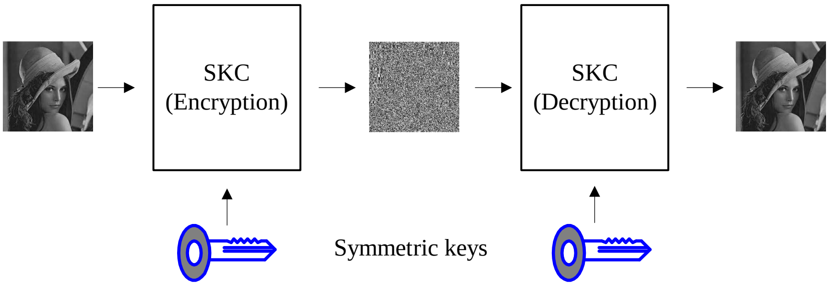


Figure 1.3. Symmetric Key Cryptosystem.

A block cipher is so-called because the scheme encrypts one block of data at a time using the same key on each block. In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

Stream ciphers come in several types but two are worth mentioning here. 1) *Self-synchronizing stream ciphers* calculate each bit in the key stream as a function of the previous *n* bits in the key stream. It is termed “self-synchronizing” because the decryption process can stay synchronized with the encryption process merely by knowing how far into the *n*-bit key stream it is. One problem here is error propagation; a garbled bit in transmission will result in *n* garbled bits at the receiving side. 2) *Synchronous stream ciphers* generate the key stream in a fashion independent of the message stream but by using the same key stream generation function at sender and receiver. While stream ciphers do not propagate transmission errors, they are, by their nature, periodic so that the key stream will eventually repeat.

A block cipher is an encryption scheme that breaks up the plaintext messages to be transmitted into strings, called blocks, of a fixed length and encrypts one block at a time. Most well-known symmetric-key encryption techniques are block ciphers. Two important classes of block ciphers are substitution ciphers and transposition ciphers. Product ciphers combine these two operations in the procedure. Symmetric key block ciphers are the most prominent and important element in many cryptographic systems. Individually, they provide confidentiality. As a fundamental building block, their versatility allows construction of pseudorandom number generators, stream ciphers, and hash functions. They could furthermore serve as a central component in message authentication techniques, data integrity mechanisms, entity authentication protocols and digital signature schemes. No block cipher is ideally suited for all applications, even the one offering a high level of security. This is a result of inevitable trade-offs required in practical applications considering speed requirements, memory limitations and constraints imposed by implementation platforms. In addition, efficiency must typically be traded off against security.

A block cipher can be operated in Electronic Codebook (ECB) mode, Cipher Block Chaining (CBC) mode, Cipher Feedback (CFB) mode and Output Feedback (OFB) mode. ECB mode is the simplest, most obvious application. The secret key is used to encrypt the plaintext block to form a ciphertext block. Two identical plaintext blocks, then, will always generate the same ciphertext block. Although this is the most common mode of block ciphers, it is susceptible to a variety of brute-force attacks. CBC mode adds a feedback mechanism to the encryption scheme. In CBC, the plaintext is

XOR-ed with the previous ciphertext block prior to encryption. In this mode, two identical blocks of plaintext never encrypt to the same ciphertext.

CFB mode is a block cipher implementation as a self-synchronizing stream cipher. CFB mode allows data to be encrypted in units smaller than the block size, which might be useful in some applications such as encrypting interactive terminal input. If we were using 1-byte CFB mode, for example, each incoming character is placed into a shift register the same size as the block, encrypted, and the block transmitted. At the receiving side, the ciphertext is decrypted and the extra bits in the block (i.e., everything above and beyond the one byte) are discarded.

OFB mode is a block cipher implementation conceptually similar to a synchronous stream cipher. OFB prevents the same plaintext block from generating the same ciphertext block by using an internal feedback mechanism that is independent of both the plaintext and ciphertext bit streams. The most popular SKC algorithms are DES, AES, IDEA, CAST 128, RC5, RC6 and Blowfish.

DES [3] is the most common SKC scheme used for encryption for nearly a quarter century. DES was designed by IBM in the 1970s and adopted by the National Bureau of Standards (NBS) [presently the National Institute for Standards and Technology (NIST)] in 1977 for commercial and unclassified government applications. DES is a block-cipher employing a 56-bit key that operates on 64-bit blocks. AES [4] has become the official successor to DES in December 2001. AES uses an SKC scheme called Rijndael, a block cipher designed by Belgian cryptographers Joan Daemen and Vincent Rijmen. The algorithm can use a variable block length and key length; the latest specification allowed any combination of keys lengths of

128, 192, or 256 bits and blocks of length 128, 192, or 256 bits. CAST-128 [76, 77] is a DES-like substitution-permutation crypto algorithm, employing a 128-bit key operating on a 64-bit block. CAST-256 (RFC 2612) is an extension of CAST-128, using a 128-bit block size and a variable length (128, 160, 192, 224, or 256 bit) key. CAST is named for its developers, Carlisle Adams and Stafford Tavares and is available internationally. International Data Encryption Algorithm (IDEA) [41] is a secret-key cryptosystem written by Xuejia Lai and James Massey, in 1992 a 64-bit SKC block cipher using a 128-bit key. RC5 [43] is a block-cipher supporting a variety of block sizes, key sizes and number of encryption passes over the data. RC6 [80] is an improvement over RC5. RC6 was one of the submissions for selection to AES. Blowfish [48] is a symmetric 64-bit block cipher invented by Bruce Schneier optimized for 32-bit processors with large data caches, it is significantly faster than DES on a Pentium / PowerPC-class machine. Key lengths can vary from 32 to 448 bits in length. Blowfish, available freely and intended as a substitute for DES or IDEA, is in use in over 80 products.

Advantages of SKC can be summarized as follows:

1) Symmetric-key ciphers could be designed to have high throughput. Some hardware implementations achieve encryption rates of few megabytes per second, while software implementations may attain throughput rates in the kilobytes per second range.

2) Keys for symmetric-key ciphers are relatively short.

3) Symmetric-key ciphers could be employed as primitives to construct various cryptographic mechanisms including pseudo-random number generators, hash functions and computationally efficient digital signature schemes.

4) Symmetric-key ciphers could be composed to produce stronger ciphers. Simple transformations which are easy to analyze, but on their own weakness, could be used to construct strong product ciphers.

Disadvantages of SKC can be summarized as follows:

1) In a two party communication, key must remain secret at both ends.
2) In a large network, there are many key pairs to be managed. Consequently, effective key management requires the use of an unconditionally Trusted third party (TTP).
3) In a two-party communication between entities A and B, sound cryptographic practice dictates that the key be changed frequently and perhaps for each communication session.
4) Digital signature mechanisms arising from symmetric key encryption typically require either large keys for the public verification function or the use of a TTP.

1.3.3.2 Public Key Cryptography

Public-key cryptography (PKC*)*, also referred as Assymmetric key cryptography, has been said to be the most significant new development in cryptography in the last 300-400 years. Stanford University Professor Martin Hellman and his graduate student Whitfield Diffie [13] have first described modern Public key cryptography publicly in 1976. Their paper described a two-key crypto system in which two parties could engage in a secure communication over a non-secure communications channel without having to share a secret key. PKC depends upon the existence of so-called *one-way functions*, or mathematical functions that are easy to compute whereas their inverse function is relatively difficult to compute.

Multiplication vs. factorization: Consider two numbers, 9 and 16, and that we want to calculate the product; it should take almost no time to calculate the product, 144. But in the contrary if we have a number, 144, and we need to find which pair of integers we have to multiply together to obtain that number. It eventually come up with the solution but calculating the product takes milliseconds, factoring will take longer because first it is necessary to find the eight pairs of integer factors and then determine which one is the correct pair.

Exponentiation vs. logarithms: If we want to take the number 3 to the 6th power; again, it is easy to calculate $3^6 = 729$. But if we have the number 729 and want to find the two integers that we used, *x* and *y* so that $\log_x 729 = y$, it will take longer time to find all possible solutions and select the pair that we used. While the examples above are trivial, they do represent two of the functional pairs that are used with PKC namely, multiplication and exponentiation.

Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key. One key is used to encrypt the plaintext and the other key is used to decrypt the ciphertext. The important point here is that it does not matter which key is applied first, but that both keys are required for the process to work as illustrated in Figure 1.4. Because a pair of key is required, this approach is also called asymmetric cryptography.

In PKC, one of the keys is designated the *public key* and may be advertised as widely as the owner wants. The other key is designated the *private key* and is never revealed to another party. If an entity *A* wants to send a message to entity *B*, then *A* encrypts the message using *B*'s public key and send the encrypted message (ciphertext) to *B*. *B* can decrypt the ciphertext using *B*'s private key. PKC could also be used to prove the identity of a sender of message. Message encrypted by *A* using *A*'s private key and then decrypted using *A*'s public key proves that the message is originated by *A*.

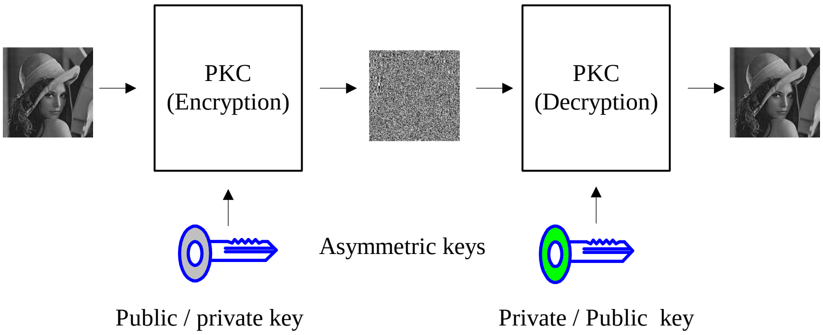


Figure 1.4. Public Key Cryptosystem.

Advantages of PKC can be summarized as follows:

1) Only the private key must be kept secret
2) The administration of keys on a network requires the presence of functionally Trusted Third Party (TTP).
3) Depending on the mode of usage, a private key/public key pair may remain unchanged for considerable periods of time.
4) Many public-key schemes yield relatively efficient digital signature mechanisms. The key used to describe the public verification function is typically much smaller than for the symmetric-key counterpart.

5) In a large network, the number of keys necessary may be considerably smaller than that of symmetric key encryption scheme.

Disadvantages of PKC can be summarized as follows.

1) Throughput rates for the most popular public-key encryption methods are several orders of magnitude slower than the best-known symmetric schemes.
2) Key sizes are typically much larger than those required for symmetric key encryption, and the size of public-key signatures is larger than that providing data origin authentication from symmetric key techniques.
3) No public key scheme has been proven to be secure (the same can be for block ciphers). The most effective public-key encryption scheme found to date has its security based on the presumed difficulty of small set of number-theoretic problems.
4) PKC does not have as extensive a history as SKC being discovered only in the mid 1970s. Symmetric key and public key encryptions have a number of complementary advantages. Current cryptographic systems exploit the strengths of each.

1.3.3.3 Hash Functions

Hash functions, also called message digests and one-way encryption, are algorithms that, in some sense, use no key. Instead, a fixed-length hash value is computed based upon the plaintext that makes it impossible for either the contents or length of the plaintext to be recovered. Figure 1.5 illustrates the functionality of hash function. Hash algorithms are typically used to provide a digital fingerprint of a file's contents. It could be used to check the integrity of a message. Often it is used to ensure that an intruder or a virus has not been able to modify a file. Hash functions provide a measure of the integrity of a file.

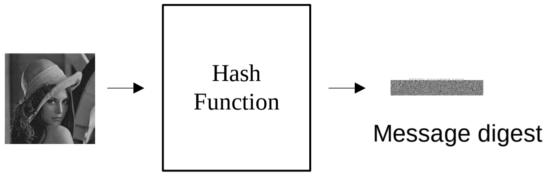


Figure 1.5. Hash Function.

1.3.4 Combined Encryptions for Information Security

Most of the information security issues, discussed earlier, could be solved by the use of SKC, PKC or hash function or any combination of these. Each of these encryption schemes is optimized for some specific application(s). Hash functions, for example, are well suited for ensuring data integrity because any change made to the contents of a message will result in the receiver calculating a different hash value than the one placed in the transmission by the sender. Since it is highly unlikely that two different messages will yield the same hash value, data integrity is ensured to a high degree of confidence. SKC, on the other hand, is ideally suited to encrypting messages. The sender can generate a session key on a per-message basis to encrypt the message. The receiver needs the same session key to decrypt the message. Key exchange is key application of PKC. Asymmetric schemes could be used for non-repudiation. If the receiver can obtain the session key encrypted with the sender's private key, then, only this sender could have sent the message. A hybrid cryptographic scheme combines all of these functions to form a secure transmission comprising *digital signature* and *digital envelope* as shown in Figure 1.6.

In this case, the sender of the message is ‘Alice’ and the receiver is ‘Bob’. Alice uses secret key cryptography to encrypt her message using the session key, which she could generate at random with each session. Alice then encrypts the session key using Bob’s public key. The encrypted message and encrypted session key form the digital envelope. Then Alice generates the fingerprint of the message (Message digest or hash value) using a hash function and encrypts the fingerprint of the message using her private key to form the digital signature which she attaches to the digital envelope.

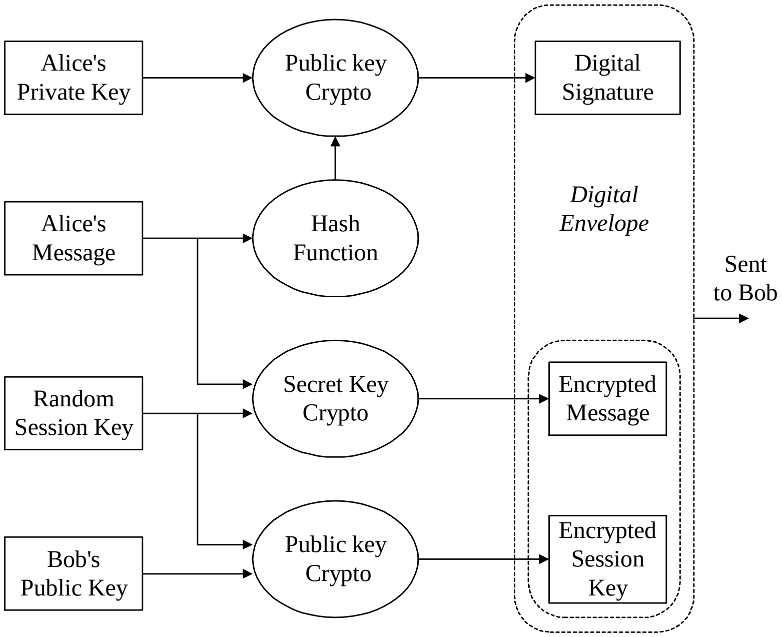


Figure 1.6. Combined Cryptographic Schemes for Information Security.

Upon receipt of the digital envelope, Bob recovers the session secret key using his private key and then decrypts the encrypted message. Bob then computes the hash value or fingerprint of the decrypted message using the same hash function. He also recovers the finger print of the message sent by

Alice by decrypting the encrypted hash value using Alice’s public key. He then compares both the finger prints and can make sure that both the finger prints are matching. Now, if the finger prints are matching, Bob can ensure that the message received is originated by Alice (because the fingerprint of the message is obtained by decrypting the encrypted finger of the original message using Alice’s public key) and that the message has not been altered during the transmission (because both the finger prints are same). Thus, combined use of PKC, SKC and Hash function can ensure confidentiality, integrity and authenticity of information.

Chapter 2

Review of Earlier Work on Cryptography

This chapter explores the history and earlier developmental work on cryptography. The cryptography prevailed since World war-II has been reviewed in brief. Some of the symmetric key ciphers and the popular encryption standards such as Data Encryption Standard and Advanced Encryption Standard are discussed. Standard references for classical cryptanalysis are also indicated.

2.1 Historical Development of Ciphers

By World War II, mechanical and electromechanical cipher machines came in to existence. Since then great advances were made in both cipher design and cryptanalysis, all in secrecy. The Germans have used different versions of an electromechanical rotor machine known as Enigma. Kahn [21] provided a historical reference for classical ciphers and machines up to 1967. The selection of classical ciphers presented, largely followed Shannon's 1949 paper [22]. Vernam Cipher [23] has been developed for telegraph encryption. Poly-alphabetic ciphers have been invented by the Florentine architect Alberti, who devised a cipher disk with a larger outer and smaller inner wheel, respectively indexed by plaintext and ciphertext characters. Recent contributions on homophonic substitution are due to Gunther [24] and Massey [25]. Beker and Piper [26] provided technical details of the Hagelin M-209. Hill [27] proposed matrix cipher by providing a practical method for poly-alphabetic substitution. Diffie and Hellman [28] have presented an instructive overview of rotor machines, used in World War II by the Americans in their high level systems. Davies and Price [29] discussed Enigma, the encryption used by Germans in World War II.

2.2 History of Symmetric Key Block Ciphers

Block ciphers are encryption algorithms that perform transformation on blocks of input data. The history and development of some of the popular symmetric block ciphers are presented in the following sections.

2.2.1 Data Encryption Standard (DES)

DES is a block cipher having a block size of 64 bits and key size of 56 bits. The original specification of DES is the 1977 U.S. Federal Information Processing Standards Publication 46 [30]. Countless papers have analyzed various properties of DES. Subsequent to the discovery of differential cryptanalysis (DC) by Biham and Shamir, Coppersmith et. al [31] explain how DES was specifically designed 15 years earlier to counter DC. Matsui [32] suggested that DES can be strengthened against DC and Linear Cryptanalysis (LC) by re-arranging the order of 8 S-boxes. Matsui and Yemagishi [33] have actually recovered DES key using LC under experimental conditions using 2^{43} known-plaintext pairs from randomly generated plaintexts running twelve 100 MHz machines over 50 days.

2.2.2 Advanced Encryption Standard (AES)

The Rijndael cipher was developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, and submitted by them to the AES selection process. U.S. government selected this cipher as a symmetric-key encryption standard and adopted as Advanced Encryption Standard (AES) in the year 2002 [34]. AES comprises three block ciphers, AES-128, AES-192 and AES-256. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively.

The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor DES. AES is based on a design principle known as a Substitution Permutation (SP) network. Unlike its predecessor, DES, AES does not use a Feistel network. AES operates on a 4×4 matrix of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds is applied to transform ciphertext back into the original plaintext using the same encryption key.

2.2.3 Fast Data Encryption Algorithm (FEAL)

FEAL stimulated the development of a sequence of advanced cryptanalytic techniques of unparalleled richness and utility. While it appears to remain relatively secure when iterated a sufficient number of rounds (e.g., 24 or more), this defeats its original objective of speed. FEAL-4 was found to have certain vulnerabilities by Shimizu and Miyaguchi [35]. Miyaguchi et. al [36] published FEAL-N with *N* rounds, and its extension FEAL-NX with 128-bit key and ascertained that chosen plaintext attacks on FEAL-8 were not practical threats. Langford and Hellman [37] introduced Differential Linear Cryptanalysis by combining linear and differential cryptanalysis to allow a reduced 8-round version of DES to be attacked with fewer chosen-plain texts than previous attacks. Aoki and Ohta [38] refined these ideas for FEAL-8 yielding a differential-linear attack requiring only 12 chosen texts and 35 days of computer time.

2.2.4 International Data Encryption Algorithm (IDEA)

The primary reference for IDEA is Lai [39]. A preliminary version introduced by Lai and Massey [40] was named PES (Proposed Encryption Standard). The analysis of Meier [41] revealed number of attacks feasible against full 8- round IDEA, and supports the conclusion of Lai that IDEA appears to be secure against DC after 4 of its 8 rounds.

2.2.5 Secure And Fast Encryption Routine (SAFER)

Massey and Safer [42] introduced SAFER K-64 with a 64-bit key and initially recommended 6 rounds, giving a reference implementation and test vectors. Massey then published SAFER K-128, differing only in its use of a non-proprietary key schedule accommodating 128-bit keys. Massey gave further justification for design components of SAFER K-64.

2.2.6 RC5

RC5 was designed by Rivest [43] and published along with a reference implementation. The constants used in the algorithm were based on the base of natural logarithms. The data-dependent rotations (which vary across rounds) distinguish RC5 from iterated ciphers that have identical operations in each round.

2.2.7 Other Block Ciphers

LOKI-91 was proposed as a DES alternative with a larger 64 bit key, a matching 64 bit block size and 16 rounds. It differs from DES mainly in key schedule and the F-function. It was introduced by Brown et. al [44]. After the discovery of weaknesses in it they introduced LOKI-91 in the year 1993 [45].

CAST is a design procedure for a family of DES-like ciphers, featuring *m* × 7 1bit S-boxes based on bent functions. Adams and Tavares [46] examined the construction of large S-boxes resistant to differential cryptanalysis and give a partial example (with 64-bit block length and 8 × 32 bit S-boxes) of a CAST cipher.

BLOWFISH is a 16-round DES-like cipher due to Schneier [47] with 64-bit blocks and keys of length up to 448 bits. The computationally intensive key expansion phase creates eighteen 32-bit sub keys plus four 8 × 32 bit S-boxes derived from the input key, for a total of 4168 bytes. Preliminary analysis of BLOWFISH is given in Vaudenay [48].

3-way is a block cipher with 96-bit block size and key size due to Daemen [49]. Daemen et. al [50] have introduced this along with a reference C implementation and test vectors. It was designed for speed in both hardware and software, and to resist differential and linear attacks. It's core is a 3-bit nonlinear S-box and a linear mapping represented as polynomial multiplication.

SHARK is an SP-network block cipher due to Rijmen et. al [51] that may be viewed as a generalization of Safe And Fast Encryption Routine (SAFER) employing highly nonlinear S-boxes and the idea of MDS codes for diffusion to allow a small number of rounds to suffice.

BEAR and LION of Anderson and Biham [52] are 3-round unbalanced Feistel networks, motivated by the earlier construction of Luby and Rackoff [53] which provides a provably secure (under suitable assumptions) block ciphers from pseudorandom functions using a 3-round Feistel structure. SHARK, BEAR and LION all remain to be subjected to independent analysis in order to substantiate their conjectured security levels.

SKIPJACK is a classified block cipher whose specification is maintained by the U.S. National Security Agency (NSA). FIPS185 [54] noted that its specification is available to organizations entering into a Memorandum of Agreement with the NSA, and includes interface details (e.g. it has an 80-bit secret key). Roe [55] gives details regarding curious results on the cyclic closure tests on SKIPJACK and evidence related to the size of the cipher key space.

COST 28147-89 is a Soviet government encryption algorithm with a 32-round Feistel structure and unspecified S-boxes by Charnes et. al [56]. WAKE is a block cipher due to Wheeler [57] employing a key-dependent table, intended for fast encryption of bulk data on processors with 32-bit words. TEA (Tiny Encryption Algorithm) is a block cipher proposed by Wheeler [58].

Zhang Yun-Peng et. al [59] described a digital image encryption algorithm based on chaos and Sai Charan et. al [60] have proposed another method of chaos based image encryption. Dang et. al designed an image encryption scheme for secure Internet multimedia applications [61]. Hua Zhong proposed an image encryption based on chaotic maps [62]. Alireza Jolfaei and Abdolrasoul Mirghadri suggested an image encryption approach using Chaos and Stream cipher [63]. Dawson et. al described strict key avalanche criterion in block ciphers [64]. Aditee Gautam et. al described a new image encryption approach using block based transformation algorithm [65]. Zhang Yun-peng et. al have designed a digital image encryption algorithm based on chaos and improved DES [66].

2.3 Crypt Analysis

Standard references for classical cryptanalysis include Friedman [67], Gaines [68] and Sinkov [69]. The most significant cryptanalytic advances over the 1990-1995 period was Matsui's linear cryptanalysis, and the differential cryptanalysis of Biham and Shamir [78]. Extensions of these included the differential-linear analysis by Langford and Hellman [81], and the truncated differential analysis of Knudsen. Basic theories on various linear cryptanalysis methods are given by Matsui and Yamagishi [79]. Friedman taught how to crypt-analyze running-key ciphers in his Riverbank Publication No. 16, *Methods for the Solution of Running-Key Ciphers.* Additional background on differential cryptanalysis is provided by many other authors including Lai, Massey, Murphy and Coppersmith. Although more efficient 6-round attacks are known, Stinson [70] provided detailed examples of attacks on 3-round and 6-round DES. Kaliski and Yin [71] give an elaborate description regarding both linear and differential cryptanalysis. Regarding text dictionary and matching ciphertext attacks, a vivid description is given by Coppersmith et. al [72]. The 1977 exhaustive DES key search machine proposed by Diffie and Hellman contained 10 DES chips, with estimated cost US$20 million (1977 technology) and 12-hour expected search time. Diffie and Hellman noted the feasibility of a ciphertext-only attack, and found that attempting to preclude exhaustive search by changing DES keys more frequently, doubles the expected search time before success. Subsequently Wiener [73] provided a gate-level design for a machine (1993 technology) using 57600 DES chips with expected success in 3.5 hours. Comparable key search machines of equivalent cost by Eberle [74] and Wayner [75] are respectively 55 and 200 times slower, although the former does not require a chip design and the latter uses a general-purpose machine. Wiener also noted adaptations of the ECB known-plaintext attack to other 64-bit modes (CBC, OFB and CFB) and 1-bit and 8-bit CFB.

2.4 Summary

The literature indicates many cryptographic algorithms that have been developed for information security services. The cryptographic systems prevailed since World War II has been surveyed. PKC algorithms are computationally intensive and hence very slow whereas SKC algorithms are faster. PKC is, therefore, used for key exchanges and digital signature applications and not used for message encryption applications. SKC is used for message encryption applications as it is faster in conversion. In SKC, there are different standard algorithms based on Fiestal networks, substitution and permutation (SP) networks and chaotic maps. Among these

DES and AES are most popular encryption standards and are used worldwide. DES with its 56 bit key size has become insecure in the light of availability of high performance computational facilities at reduced cost. AES with its 128 bit key size has become a benchmark today and it has withstood all known security attacks. Therefore, in this research work, AES has been chosen as a reference. Development of an efficient encryption scheme that has higher conversion speed than AES is pursued while maintaining the security level of AES.

Chapter 3

Matrix Array Symmetric Key Encryption Development

In this chapter the concept and development of the Matrix Array Symmetric Key (MASK) Encryption scheme is presented. The encryption algorithm, which is based on matrix and array manipulations using secret key and sub keys, is discussed. The three major functional blocks of the encryption scheme viz. matrix initialization, key schedule, substitution and diffusion are explained. Basic test results of this scheme obtained using plaintext messages and images are presented. Characteristics of MASK encryption scheme developed in this thesis work and Advanced Encryption Standard (AES) are compared. Results indicating improvement on the key avalanche effect produced in AES by replacing the key schedule of AES with that of MASK encryption are also discussed.

3.1 Introduction

This chapter describes a new, efficient symmetric key block encryption scheme “Matrix Array Symmetric Key (MASK) encryption”. The proposed encryption scheme, is based on matrix substitution mapping and array based diffusion operations depending on the data and key values. It is a block cipher operating on blocks of plaintext message (or image) using a secret key producing blocks of ciphertext message (or cipher image). The block size is 128 bits and the key size is also 128 bits.

This encryption algorithm incorporates substitution and diffusion operations in 16 iterative rounds using sub keys generated from a complex key schedule algorithm. The key schedule incorporated in MASK encryption has been capable of producing a strong key avalanche effect in the ciphertext output of the cipher. Fast conversion, of plaintext data and images into ciphertext data and cipher image, is achieved with matrix based non-linear poly-alphabetic substitution, sub key additions and data based circular shift operations performed in the algorithm. In the following sub sections, the concept and realization of the proposed efficient symmetric cipher is described. Internal parameters, intermediate results and other characteristics of the cipher are obtained and compared with AES. Test results show that the characteristics of the proposed encryption scheme compares well with the characteristics of AES. However, MASK encryption is capable of converting text messages and images faster than AES. This is the main advantage of MASK over AES. A case study also has been conducted to see if there is any improvement in AES by incorporating the MASK key schedule in AES. Enhanced key avalanche effect has been observed in AES with MASK key schedule. The enhanced key avalanche obtained from AES is also discussed.

3.2 Nomenclature

$P(i)$ – i^{th} plain text character in input plain text character block

$C(i)$ – i^{th} cipher text character in output cipher text character block

$M_e(i,j)$ – Element of encryption matrix M_e with row i and column j

$M_d(i,j)$ – Element of decryption matrix M_d with row i and column j

$K_e(i)$ – i^{th} character of encryption key, K_e

$K_d(i)$ – i^{th} character of decryption key, K_d

$E(K_e,P)$ – Encryption of plain text P with secret key K_e

$D(K_d,C)$ – Decryption of cipher text C with secret key K_d

$K_{s1e}(n)$, $K_{s2e}(n)$ – n^{th} round encryption sub keys

$K_{s1d}(n)$, $K_{s2d}(n)$ – n^{th} round decryption sub keys

3.3 The Encryption Process

The MASK encryption algorithm consists of three functional sections. The first section, “Matrix initialization”, creates an encryption matrix, $M_{e,}$ (of size 16 × 256 bytes) with numbers ranging from 0 through 255, arranged in 16 rows in an order depending on the decimal values of the characters of secret key selected. The columns of this matrix are shuffled using a table look-up procedure. This matrix is being used for two purposes. First, it is being referred by the key schedule algorithm for generating sub keys to be used in diffusion round operations. Second, it is being referred by the substitution and diffusion section for substituting a value obtained from a selected row of the matrix to a given input data byte. The second section “Key schedule”, is being used to generate 16 pairs of sub keys, K_{s1e} and K_{s2e}, referring to the matrix, to be used by 16 diffusion round operations in the encryption transformation. The third section, “Substitution and Diffusion”, transformation converts the plaintext (or image) data into ciphertext (or cipher image) data in blocks of 16 bytes. Figure 3.1 shows these functional blocks of the encryption algorithm.

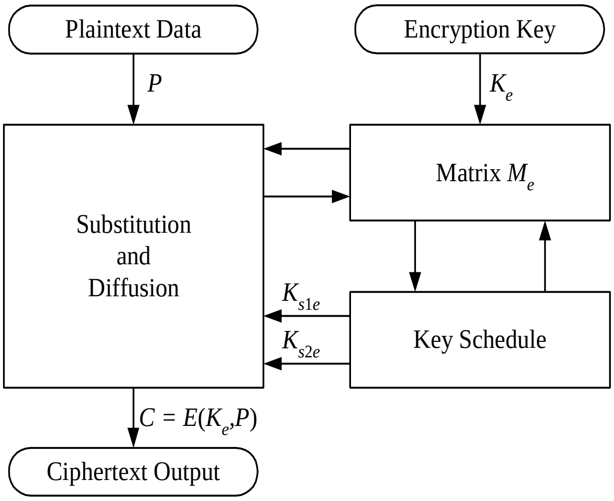


Figure 3.1. Functional blocks of MASK Encryption.

3.3.1 Matrix Initialization

A matrix $M_{e1,}$ having 16 rows and 256 columns with element values ranging between 0 and 255 is created. These values represent ASCII characters of plaintext and pixel gray scale (intensity) values. The values are stored in the columns of each row of the matrix in such a way that it depends on the encryption key K_e. Further, the elements in the columns of every row are shuffled so that the numbers represented by the elements arrange itself in a non-linear fashion.

3.3.1.1 Matrix Creation

A matrix $M_{e1,}$ is created with columns of every row of the matrix filled with numbers between 0 and 255 (both the numbers included) in an order depending on the characters of secret key. The first column in the i^{th} row of the matrix is filled with integer value of i^{th} character of the secret key K_e. The subsequent columns of the i^{th} row of the matrix are filled with numbers that have increments of 1 from the previous value till the number becomes 255. Remaining columns are filled with numbers starting from 0 and ends with integer value of the i^{th} character of secret key minus 1.

The distribution of numbers (or equivalent characters) in the columns of all the sixteen rows of the matrix thus becomes key dependent. Without knowing the secret key the element in a column of any row of the matrix M_{e1} cannot be determined by an adversary. Plate 3.1 shows the matrix initialization pseudo code. The matrix M_{e1} initialized with secret key ‘Godiseternalyes!’ looks like the one shown in Figure 3.2.

```
For $i$ ← 1 to 16   // rows
For $j$ ← 1 to 256   // columns
        $M_{e1}(i,j) = \text{int}(K_e(i)) + (j-1)$
                If $M_{e1}(i,j) > 255$
        { $M_{e1}(i,j) = M_{e1}(i,j) - 256$ }
EndFor  // columns
EndFor  // rows
```

Plate 3.1. Matrix initialization pseudo code.

Columns →

1	2	3	4	5	6	7	8	9	10	11	.	.	.	.	256
G	H	I	J	K	L	M	N	O	P	Q	R	.	.	.	F
o	p	q	r	s	t	u	v	w	x	y	z	.	.	.	n
d	e	f	g	h	i	j	k	l	m	n	o	.	.	.	c
i	j	k	l	m	n	o	p	q	r	s	t	.	.	.	h
s	t	u	v	w	x	y	z	{	|	}	~	.	.	.	r
e	f	g	h	i	j	k	l	m	n	o	p	.	.	.	d
t	u	v	w	x	y	z	{	|	}	~		.	.	.	s
e	f	g	h	i	j	k	l	m	n	o	p	.	.	.	d
r	s	t	u	v	w	x	y	z	{	|	}	.	.	.	q
n	o	p	q	r	s	t	u	v	w	x	y	.	.	.	m
a	b	c	d	e	f	g	h	i	j	k	l	.	.	.	.
l	m	n	o	p	q	r	s	t	u	v	w	.	.	.	k
y	z	{	|	}	~	.	.	.	.	.	.	.	.	.	x
e	f	g	h	i	j	k	l	m	n	o	p	.	.	.	d
s	t	u	v	w	x	y	z	{	|	}	~	.	.	.	r
!	“	#	$	%	&	‘	(	)	*	+	,	.	.	.	.

Figure 3.2. Matrix M_{e1} created using secret key ‘Godiseternalyes!’.

3.3.1.2 Matrix Column Shuffling

The matrix already initialized M_{e1}, is further subjected to column shuffling in order to achieve non-linearity in substitution. The non-linearity could present confusion to the crypt analyst while attempting to decrypt an encrypted message, by linear crypt analysis. The bytes stored in the columns of each row are mixed according to a pre-defined pattern using a look-up table. The pseudo code of shuffling operation is given in plate 3.2.

```
$Ai$ = [3 10 1 6 2 14 16 15 8 12 9 13 4 11 5 7]
for $i$ ←1 to16
    for $k$ ← 1 to 256 step 16
        $n1$=1;
        for $j$ ← $k$ to ($k$+15)
            $M_e(i,j) = M_{e1}(i,(Ai(n1)+(k-1)))$
            $n1 = n1+1$
        EndFor
    EndFor
EndFor
```

Plate 3.2. Matrix shuffling pseudo code.

An indexing array Ai, of 16 elements with decimal values ranging from 1 to 16 with no values repeated (Ai = [3 10 1 6 2 14 16 15 8 12 9 13 4 11 5 7]) is created. The element values in this array are used as index to select 16 columns of the matrix. Another matrix M_e of size 16 × 256 is created such that columns 1 through 16 of each row of the matrix M_{e1}, are copied to columns 1 through 16 of each row of the matrix M_e using elements of Ai as column index. Figure 3.3 illustrates the column shuffling procedure.

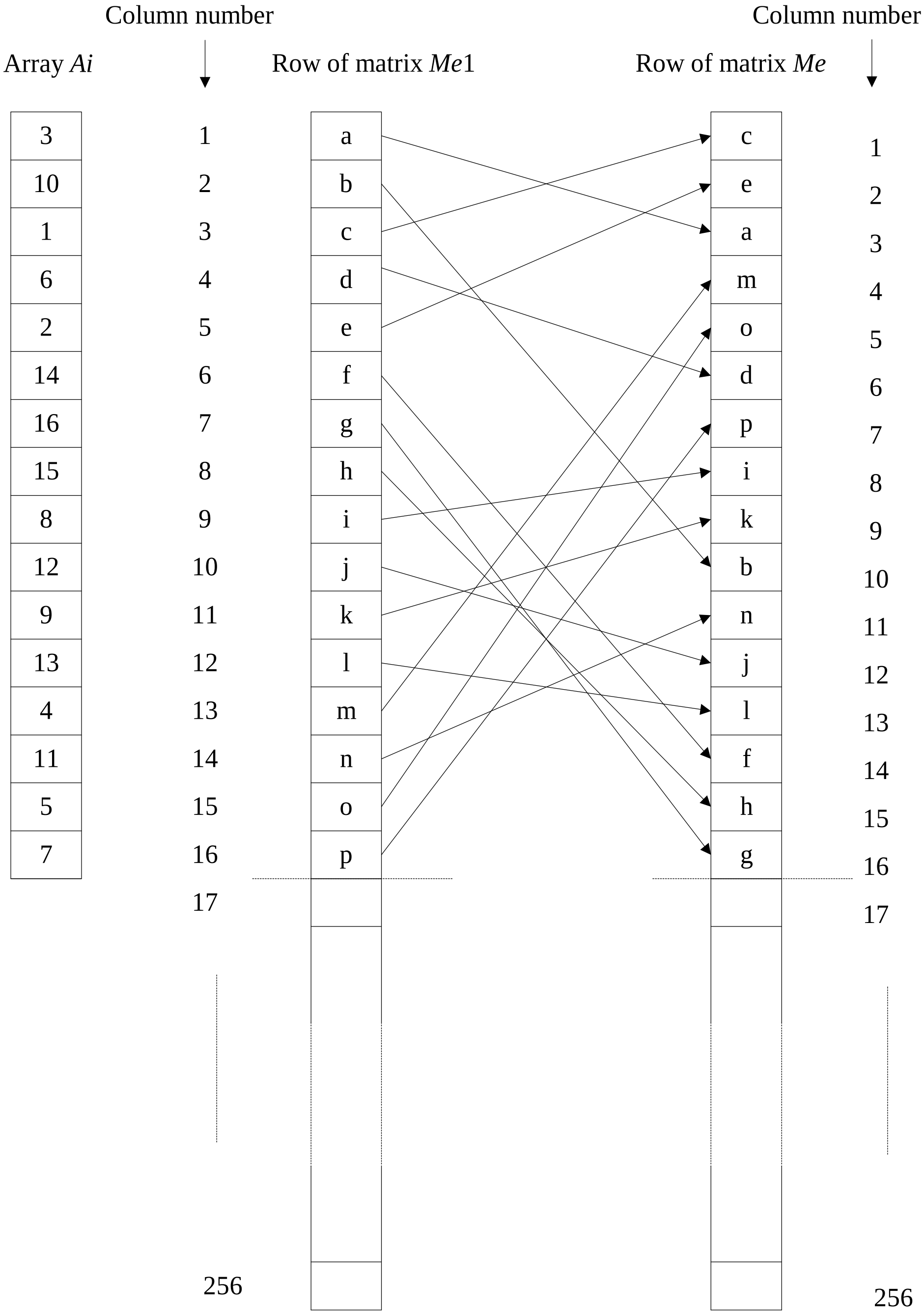


Figure 3.3. Matrix column shuffling.

Columns 17 through 32 of each row of the matrix M_{e1}, are copied to columns 17 through 32 of each row of the matrix M_e using decimal value of elements of Ai + 16 as column index. In this way, blocks of 16 columns of every row of the matrix M_{e1}, are copied to corresponding blocks of 16 columns of every row of matrix $M_{e.}$ This operation facilitates a shuffling effect on the elements stored in the columns of matrix.

3.3.2 The Key Schedule

Sub key matrices, K_{s1e} and K_{s2e} used in diffusion round operations are generated by the key schedule procedure. In this procedure the two sub key matrices K_{s1e} and K_{s2e} of size 16 × 16 are derived from the matrix M_e. It is desirable that the key schedule be a complex procedure so that an adversary must find it extremely difficult to derive the sub-keys during crypt analysis. Another desirable feature of key schedule is that a small change in the secret key should get well diffused into the sub keys. One bit change in secret key should cause many bits of sub keys to change (key avalanche effect). The key schedule procedure is explained in steps as follows:

1) Transpose (T) the secret key K_e, to get K_{a1}. This is achieved by a byte-level transposing operation where by the least significant (LS) byte takes the place of most significant (MS) byte position and the MS byte takes the LS byte position after the transpose operation.
2) XOR K_{a1} with K_e to get K_{a2}. This operation can cause up to two bits change in K_{a2} when 1 bit is changed in secret key K_e.
3) XOR Left half (LH) 8 bytes of K_{a2} and right half (RH) 8 bytes of K_{a2} to get K_{a3}.
4) XOR transposed LH of K_{a2} and transposed RH of K_{a2} to get K_{a4}.

5) Concatenate K_{a3} *and* K_{a4} to get K_{a5}. With this operation, a one bit change in secret key, K_e, can cause up to 4 bits to change in K_{a5}.

6) Calculate Sum of integer values of bytes in K_{a5} to get L

7) Calculate K_{se1} such that $K_{se1} = L \% 23$. When the secret key has a one bit change, K_{se1} can have up to 4 counts change.

8) K_{se2} *is calculated such that* $K_{se2} = L \% 15$.

   When secret key has 1 bit change, K_{se2} can have up to 4 counts change and ($K_{se1} + K_{se2}$) can have up to 8 counts change.

9) Derive two matrices K_{s1e} *and* K_{s2e} of size 16 × 16 from the base matrix M_e, as $K_{s1e}(row,column)=M_e(row,(K_{se1}+K_{se2}+column))$

   $K_{s2}e(row,column)=M_e(row,K_{s1e}(row,column))$

   Columns of K_{s1e} matrix are chosen from the base matrix M_e depending upon $K_{se1,}$ K_{se2} values. Here, an element of K_{s1e} can have up to eight counts change with one bit change in secret key.

   Columns of K_{s2e} matrix are chosen from the base matrix M_e depending upon element values of columns of K_{s1e} matrix. An element of K_{s2e} can have up to eight counts change with one bit change in secret key.

10) Rotate vertically down i$^{\text{th}}$ column of matrix K_{s1e} number of times equal to $((int(K_e(i)) \% 12) + K_{se1})$.

11) Rotate vertically down i$^{\text{th}}$ column of matrix K_{s2e} number of times equal to $((int(K_e(i)) \% 10) + K_{se2})$. The rotations shuffle the elements of sub-key matrices thereby providing more changes in the sub-key values while one bit change is applied on the original secret key, K_e.

Plate 3.3 shows the pseudo code of the key schedule procedure.

```
$K_{a1} = K_e$ Transposed ;  $K_{a2} = K_{a1}$ XOR $K_e$
$K_{a1} =$ Left 8 characters of $K_e$
$K_{a2} =$ Right 8 characters of $K_e$
$K_{a3} = K_{a1}$ XOR $K_{a2}$
$K_{a1} = K_{a1}$ Transposed; $K_{a2} = K_{a2}$ transposed
$K_{a2} = K_{a2}$ XOR $K_{a1}$
$K_{a3} = K_{a2}$ and $K_{a3}$ concatenated
SUM = Integer sum of Elements of $K_{a3}$
$K_{se} = SUM \% 23$;   $K_{se1} = SUM \% 15$
For r ← 0 to 15
    For i ← 0 to 15
        $K_{s1e}(i,r) = M_e(i,(K_{se}+K_{se1}+r))$
        $K_{s2e}(i,r) = M_e(i, int(K_{s1e}(i,r)))$
        $K_{s1e}(i,r) = M_e(i, int(K_{s2e}(i,r)))$
        $Q(i) = int(K_e(i)) \% 12$
    EndFor
EndFor
For i ← 0 to 15
    Circlar shift down $i^{th}$ column of $K_{s1e}$ & $K_{s2e}$
    number of times equal to $K_{se1} + Q(i)$
EndFor
```

Plate 3.3. Key schedule pseudo code.

The two sub keys K_{s1e} and K_{s2e} (16 bytes each) generated from an encryption key K_e, for each of 16 rounds using the key schedule are given in plate 3.4 and plate 3.5.

Secret key K_e = 4C 69 66 65 27 73 20 62 65 61 75 74 69 66 75 6C

Key schedule (K_{s1e}):

1: 6D 6A 49 40 6D 52 F2 49 40 3D 83 67 6E 34 3D 31
2: 68 6F 35 3E 32 6E 6B 4A 41 6E 53 F3 4A 41 3E 84
3: 3F 33 6F 6C 4B 42 6F 54 F4 4B 42 3F 85 69 70 36
4: 34 70 6D 4C 43 70 55 F5 4C 43 40 86 6A 71 37 40
5: 6E 4D 44 71 56 F6 4D 44 41 87 6B 72 38 41 35 71
6: 39 42 36 72 6F 4E 45 72 57 F7 4E 45 42 88 6C 73
7: 74 3A 43 37 73 70 4F 46 73 58 F8 4F 46 43 89 6D
8: 50 47 74 59 F9 50 47 44 8A 6E 75 3B 44 38 74 71
9: 39 75 72 51 48 75 5A FA 51 48 45 8B 6F 76 3C 45
10: 49 76 5B FB 52 49 46 8C 70 77 3D 46 3A 76 73 52
11: 71 78 3E 47 3B 77 74 53 4A 77 5C FC 53 4A 47 8D
12: 79 3F 48 3C 78 75 54 4B 78 5D FD 54 4B 48 8E 72
13: 73 7A 40 49 3D 79 76 55 4C 79 5E FE 55 4C 49 8F
14: 4A 3E 7A 77 56 4D 7A 5F FF 56 4D 4A 90 74 7B 41
15: 75 7C 42 4B 3F 7B 78 57 4E 7B 60 00 57 4E 4B 91
16: 7C 61 01 58 4F 4C 92 76 7D 43 4C 40 7C 79 58 4F

Plate 3.4. Secret key K_e and Sub key K_{s1e} for 16 rounds.

Each byte of the secret key and sub keys generated for 16 diffusion rounds are shown in hexadecimal values. It may be noted that the key schedule generates distinct sub keys for all the 16 rounds from a given secret key value.

Secret key K_e = 4C 69 66 65 27 73 20 62 65 61 75 74 69 66 75 6C

Key schedule (K_{s2e}):

1: D8 D0 F8 F6 E0 DA F8 E6 A6 E0 DA D8 5C F4 4E D2

2: D1 F9 F7 E1 DB F9 E7 A7 E1 DB D9 5D F5 4F D3 D9

3: E2 DC FA E8 A8 E2 DC DA 5E F6 50 D4 DA D2 FA 8

4: DD FB E9 A9 E3 DD DB 5F F7 51 D5 DB D3 FB F9 E3

5: F8 52 D6 DC D4 FC FA E4 DE FC EA AA E4 DE DC 60

6: D5 FD FB E5 DF FD EB AB E5 DF DD 61 F9 53 D7 DD

7: E6 E0 FE EC AC E6 E0 DE 62 FA 54 D8 DE D6 FE FC

8: 55 D9 DF D7 FF FD E7 E1 FF ED AD E7 E1 DF 63 FB

9: E2 00 EE AE E8 E2 E0 64 FC 56 DA E0 D8 00 FE E8

10: DB E1 D9 01 FF E9 E3 01 EF AF E9 E3 E1 65 FD 57

11: DC E2 DA 02 00 EA E4 02 F0 B0 EA E4 E2 66 FE 58

12: E3 DB 03 01 EB E5 03 F1 B1 EB E5 E3 67 FF 59 DD

13: DC 04 02 EC E6 04 F2 B2 EC E6 E4 68 00 5A DE E4

14: ED E7 05 F3 B3 ED E7 E5 69 01 5B DF E5 DD 05 03

15: E0 E6 DE 06 04 EE E8 06 F4 B4 EE E8 E6 6A 02 5C

16: 5D E1 E7 DF 07 05 EF E9 07 F5 B5 EF E9 E7 6B 03

Plate 3.5. Secret key K_e and Sub-key $K_{s2\text{e}}$ for 16 rounds.

3.3.2.1 Key Avalanche Effect on Sub-keys and Round Outputs

A desirable feature of any block cipher is that a small change either in the plaintext data or in the secret key should produce a significant change in the output ciphertext data block. This is called avalanche effect. The key avalanche is achieved using a complex key generation procedure and data avalanche is achieved by using powerful encryption primitives in cryptographic transformation algorithms. Even though the Data Encryption Standard, with its key size of 56 bits, is not secured enough today due to small key size, it exhibits strong avalanche properties that any good cipher is expected to have.

Tests conducted to obtain the effect of 1 bit change in secret key on sub-keys (sub key avalanche) would give an indication of the effectiveness of the key schedule. In this test, first the key schedule procedure was executed with a given secret key and the sub-keys generated for 16 rounds were recorded. Then, another secret key with a difference of only 1 bit (one count) from the first key was used to execute the key schedule procedure and the sub-keys generated for 16 rounds were recorded. The number of bits changed in sub-keys, in each round, was calculated from the recordings and the results were plotted. Figures 3.4 and 3.5 show the number of bit changes in sub-keys K_{s1e} and K_{s2e} in various rounds due to one bit change in the secret key. It can be seen that one bit change in secret key value causes 40 to 60 bits to undergo changes in the sub-keys (128 bit size) of every round. The sub-key avalanche in turn can cause many bits to toggle in the ciphertext output block of the cipher called key avalanche effect. The key avalanche effect of MASK encryption and AES encryption have been evaluated and presented.

Figures 3.6 and 3.7 show the number of bit changes in the ciphertext output, due to one bit change in the secret key, produced in MASK compared with DES and AES, respectively. It can be seen that one bit change in secret key brings many bit changes in the sub keys. Nearly 50% of the bits in the cipher output change with one bit change in secret key. This indicates a strong key avalanche effect on output data that enhances the security of the cipher.

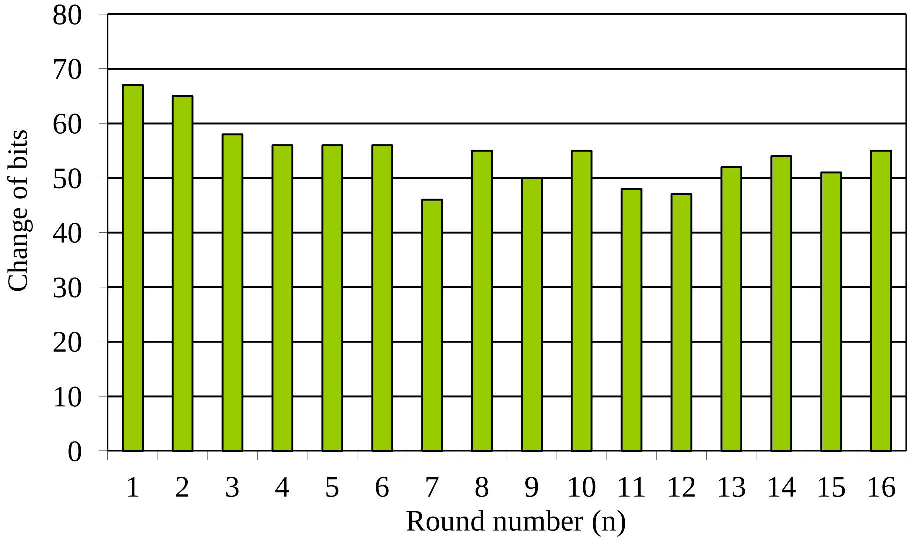


Figure 3.4. Sub-key Avalanche in K_{s1e} sub-key.

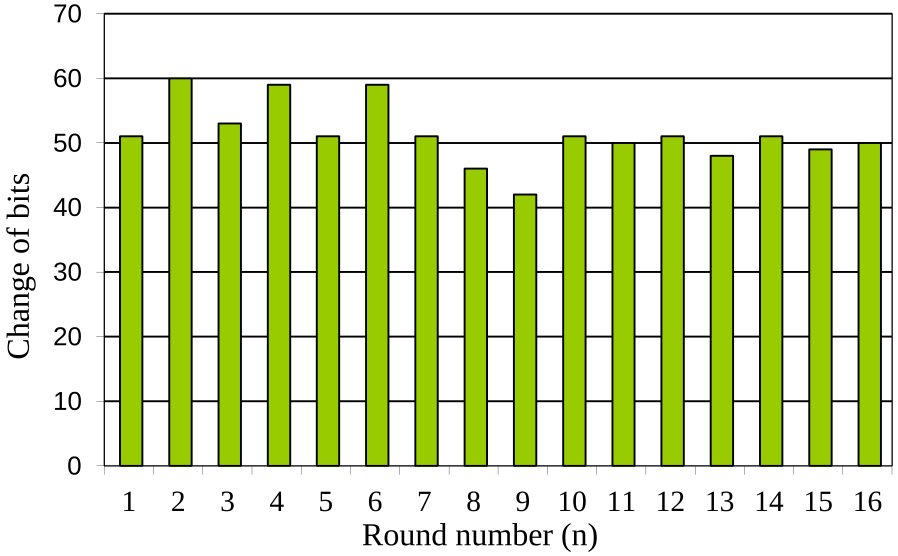


Figure 3.5. Sub-key Avalanche in K_{s2e} Sub-key.

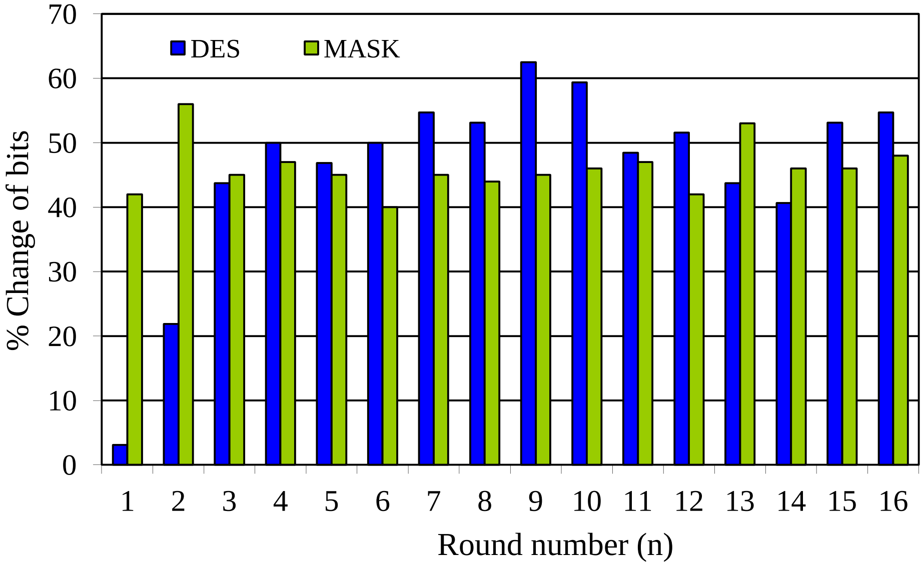


Figure 3.6. Key Avalanche produced on output data in DES and MASK.

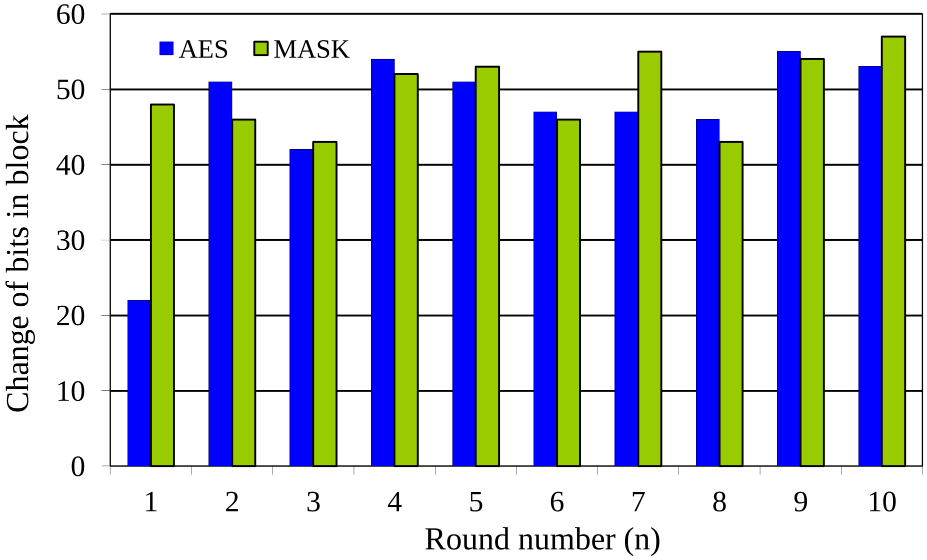


Figure 3.7. Key Avalanche produced on output data in AES and MASK.

3.3.3 Substitution and Diffusion Rounds

The transformation of plaintext data block, consisting of 16 bytes (128 bits) into ciphertext block is carried out in the substitution and diffusion round operations. There are up to 16 user selectable iterations of substitution and diffusion in the cipher. The simplified block diagram of the substitution and diffusion round is shown in Figure 3.8.

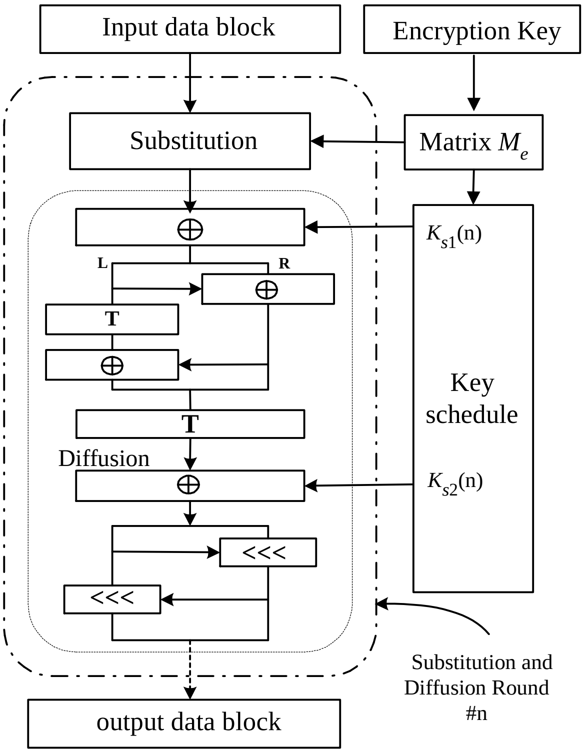


Figure 3.8. Simplified block diagram of Substitution and Diffusion.

The input data block is applied to the substitution section. The substitution section converts the data block into intermediate ciphertext data block using the contents of matrix, M_e. This forms the input to the diffusion section. The diffusion section scrambles the intermediate ciphertext block using sub keys obtained from the key schedule program and produce diffused output data block. This function (substitution and diffusion) is performed up to 16 times before the final ciphertext data block is produced. The initial round input data to the substitution and diffusion round operation is the plaintext input data block for encryption and the final round output data is the ciphertext output data block.

3.3.3.1 Substitution Section

The substitution incorporated in the algorithm is secret key dependent, non-linear and poly-alphabetic. The block schematic of the substitution process is shown in Figure 3.9. Recollect that the matrix $M_e(i,j)$ has 16 rows (row 1 through row 16) and 256 columns (column 1 through column 256). An input data block *P*, consisting of 16 bytes, *P*(1)-*P*(16), is applied at the input of substitution section. The input byte can have a decimal value between 0 and 256, both numbers included. Data byte, *P*(*i*) is taken and the decimal value of (*P*(*i*)+1) is used as column number *j*, of the i^{th} row of matrix M_e to read the value $M_e(i,j)$. (Here, 1 is added to *P*(*i*) to ensure that the column number obtained falls in the range 1 through 256 that will be used as column index to read the content of that column of the i^{th} row). This value is taken as the substitute for *P*(*i*). For example, for the byte *P*(1) in a block, $i = 1$ and j = decimal value (*P*(1)+1) is used to find the value *M*(1,*j*) as substitute, *C*(1), for *P*(1). For the byte *P*(2) in a block, i = 2 and j = decimal value (*P*(2)+ 1) are used to find the value *M*(2,*j*) as substitute, *C*(2), for *P*(2). In this way, all the 16 bytes of data in a block are substituted by a value taken from selected column and row of the matrix depending on the position of data in the block and the data value. Figure 3.10 depicts the substitution procedure and Plate 3.6 shows the substitution pseudo code. The substitution section has been tested with a block of input data having all byte values equal to 97 (corresponding to the ASCII character ‘a’) and different secret key values. The substitution section output data block is obtained and printed in decimal format. Plate 3.7 and 3.8 show these test results.

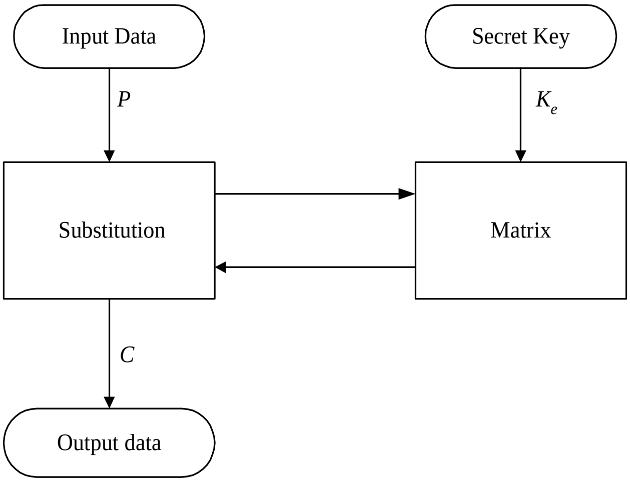


Figure 3.9. Block schematic of substitution process using matrix.

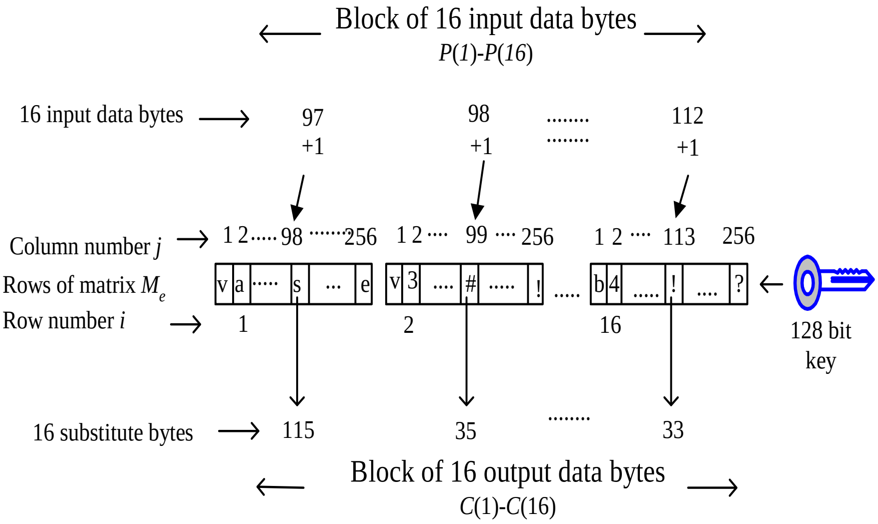


Figure 3.10. Substitution process using matrix.

```
For $i \leftarrow$ 1 to 16
        $j = P(i)+1$
        $C(i) = M_e(i,j)$
EndFor
```

Plate 3.6. Substitution pseudo code.

It can be seen that the substitution procedure is poly-alphabetic in nature as the same plaintext character ‘a’ throughout the input data block given to the substitution section has produced distinctly different substitute characters at the output data block.

```
Input (char) → aaaaaaaaaaaaaaaa
 Secret key → Godiseternalyes!
 Output → 167  207  196  201  211  197  212  197
          210  206  193  204  217 197  211  129
```

Plate 3.7. Substitution result with input data and secret key ‘Godiseternalyes!’.

```
Input (char) → aaaaaaaaaaaaaaaa
Secret key → Whenthewindblows
Output → 183  200  197  206  212  200  197  215
         201  206  196  194  204  207  215  211
```

Plate 3.8. Substitution test with input data and secret key ‘Whenthewindblows’.

Figures 3.11 and 3.12 show the plot of input data vs. output data for different key values produced by the substitution section. The substitution is key dependant as the output changes when the key is changed.

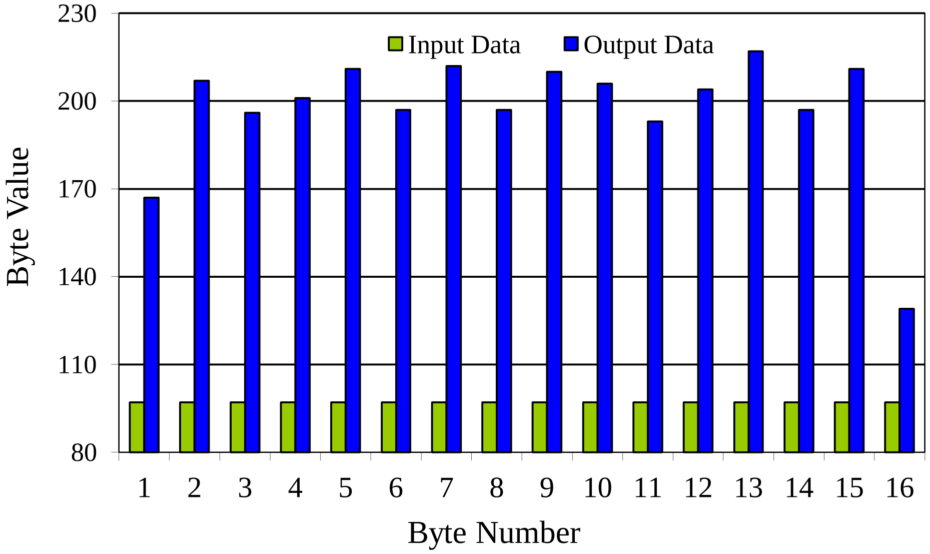


Figure 3.11. Substitution output vs. input with data and key 1.

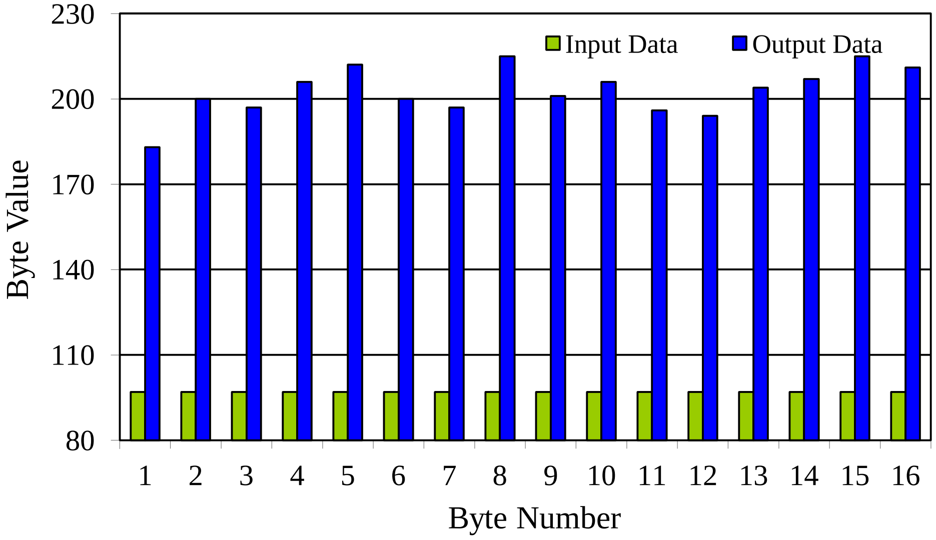


Figure 3.12. Substitution output vs. input with data and key 2.

The results indicate that the substitution is poly-alphabetic. For the same character ‘a’ throughout the input data block, the substitute characters produced at the output block are distinctly different. To check non-linearity property of the substitution, test has been conducted with a block of 16 input data bytes. Starting with value 97 (ASCII character ’a’) for the first byte and the successive bytes with linear increments of 1 is chosen. Then, a key is selected such that each character of the key is same, e.g., ‘A’ (the byte value is 65). This is to ensure that the elements in each column of all the rows of the matrix M_e, are identical. This arrangement would provide same substitute byte for same input byte at any location in the input byte block. As the input byte values linearly increase from first byte to the last byte in the input block, the output block of substitution is expected to have non-linear variation in byte values starting from the first byte to the last byte. This has been achieved as a result of shuffling of columns of matrix in the matrix column shuffling procedure. Figure 3.13 clearly indicates that the substitution is non-linear.

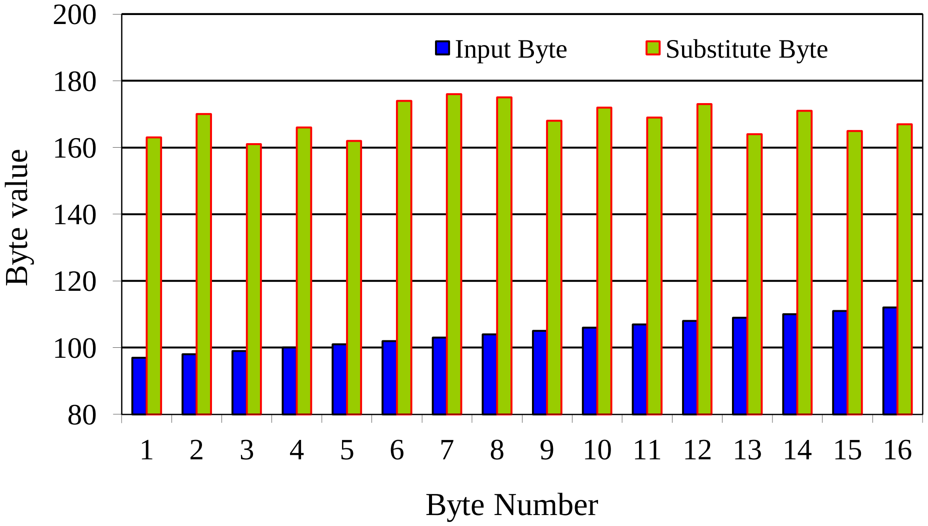


Figure 3.13. Non-linear Substitution Characteristic.

3.3.3.2 Diffusion Section

The diffusion section facilitates data avalanche, key avalanche and random differential data and differential key propagation characteristics in addition to encryption transformation of input data block. Data avalanche means many bits in the output ciphertext block undergo change when one bit change is introduced in the input data block. Key avalanche means many bits in the output block undergo change when one bit change is introduced in the secret key. The data avalanche is achieved mainly by data bifurcation and data based rotation (>>>) operations. Key avalanche is achieved by addition (XOR) of sub-keys to data in each round. The differential data propagation refers to how a difference in data value propagates through the diffusion rounds in a cipher. The differential key propagation refers to how difference in secret key value propagates through the diffusion rounds in a cipher. The diffusion section consists of key based XOR, data based XOR, transpose (T) and data based rotation (>>>) operations. The simplified block diagram of the diffusion section is shown in Figure 3.14. The input data to this section is a data block DB (16 bytes long), $\mathrm{K}_{s1e}(n)$ and $K_{s2e}(n)$ the n^{th} round sub keys. The operations performed on an input data to this section is described in steps as follows:

1. The input data block, DB, is XOR-ed with sub key $K_{s1e}(n)$ and the resulting data block is bifurcated into left half data block (LHDB) and right half data block (RHDB), each 8 bytes long.
2. The LHDB is XOR-ed with RHDB to get RHDB1.
3. The LHDB is transposed (byte level transpose operation) to get LHDBT.
4. The LHDBT is XOR-ed with RHDB1 to get LHDB1.
5. LHDB1 and RHDB1 are concatenated to get the data block DB1.

6. The data block DB1 is transposed to get DB1T and is XOR ed with sub key $K_{s2e}(n)$ to get DB2.
7. DB2 is bifurcated to left half data block, LHDB2 and right half data block, RHDB2.
8. The RHDB2 is rotated right number of times equal to the sum of decimal value of bytes of LHDB2 MOD 6 to get RHDB2R. This number lies in the range 0 to 5. The value of this integer depends on the decimal value of left half data block LHDB2. Left half data block LHDB2 is rotated right number of times equal to the sum of decimal value of bytes in RHDB2 MOD 6 to get LHDB2R.
9. LHDB2R and RHDB2R are concatenated to get DB, which is the output data block generated from the diffusion section.

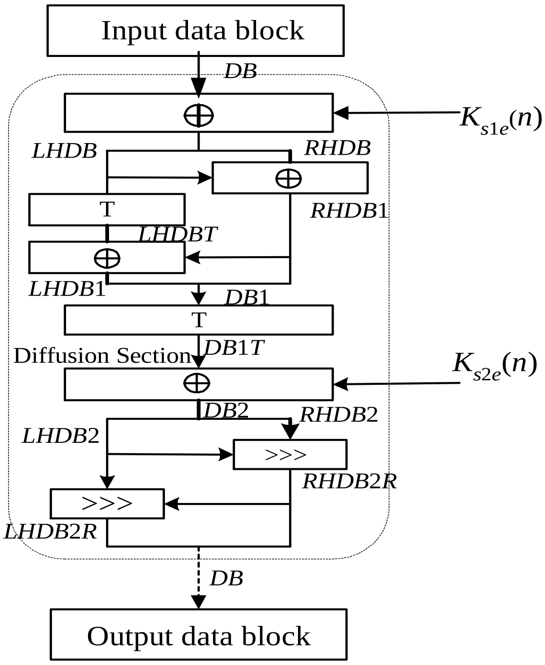


Figure 3.14. Simplified block diagram of diffusion section.

Key avalanche effect on ciphertext block: An important property for a secure block cipher is the key avalanche effect. A block cipher satisfies the key avalanche effect if for a fixed plaintext block a small change in the key causes a large change in the resulting ciphertext block [64]. To determine how a small change (usually 1 bit change) in secret key gets diffused in the data blocks in each round output, we perform the key avalanche test. Here, a block of plaintext data is applied as input to the diffusion section and with a given secret key, K_e, and the derived sub keys, the diffusion section is executed for 16 rounds. Then, with the same block of input plaintext, a secret key value that differs by one bit is used to execute the section. The number of bit changes that occur in each round has been determined. The number of bit changes, in each round due to one bit change in the secret key value is plotted in Figure 3.15. It can be seen that around 50 bits undergo changes in each data output in round for one bit change in secret key.

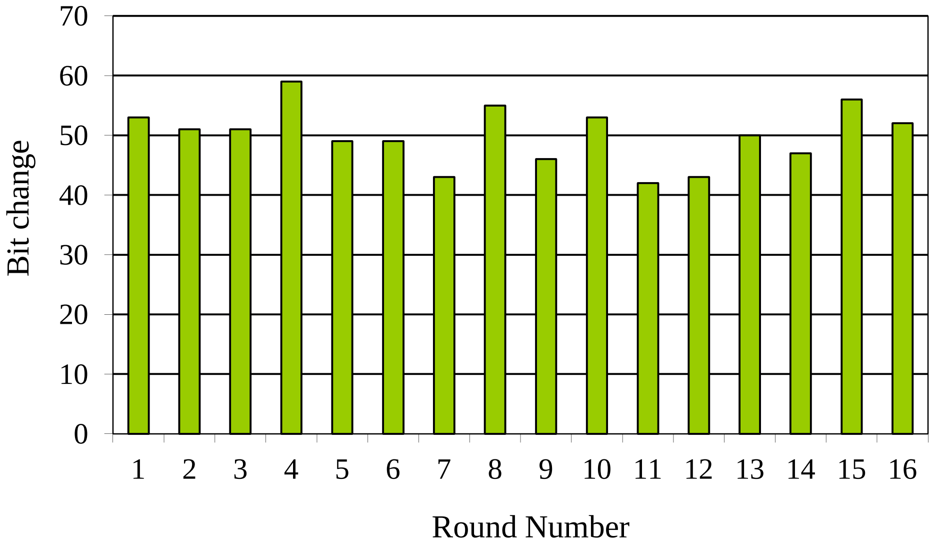


Figure 3.15. Key Avalanche in 16 rounds for a change in one bit in secret key.

This indicates the ability of the diffusion section to produce a good key avalanche effect. In a good block cipher, for a given plaintext input block, it is desirable that up to 50% of the bits in a block of data in the output of the cipher undergo changes due to one bit change in secret key [64].

Data avalanche effect on ciphertext block: Another important property for a secure block cipher is the plaintext avalanche effect. A block cipher satisfies the plaintext avalanche effect if for a fixed key, a small change in the plaintext causes a large change in the resulting ciphertext block [64]. To determine how a small change (usually 1 bit change) in input data block gets diffused in the data blocks in each round output, we perform the data avalanche test. The number of bit changes, in each round, due to one bit change in input data block has been obtained and shown in Figure 3.16.

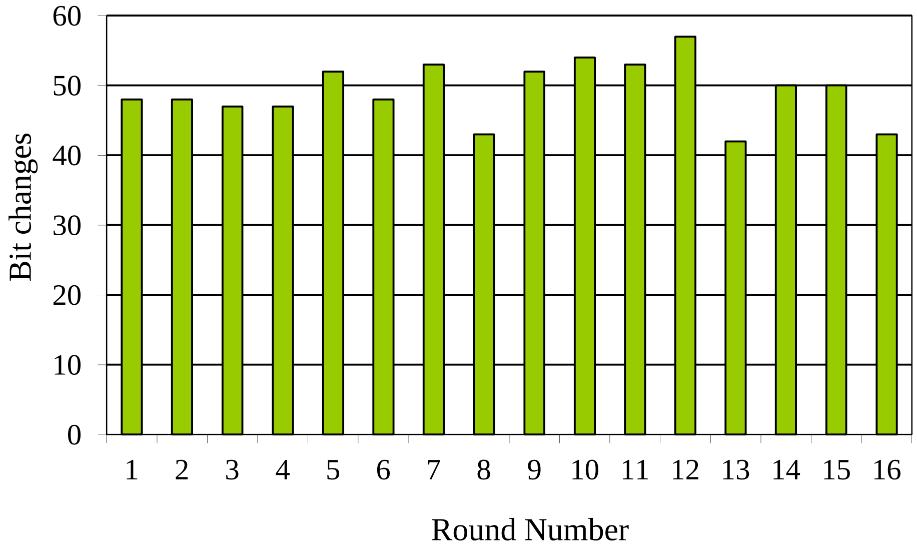


Figure 3.16. Data Avalanche in 16 rounds for a change in one bit in plaintext data.

With a given secret key $K_{e,}$ the cipher has been executed and the output block produced in each round has been recorded. Then, with 1 bit change in the input data block and with the same key, the cipher has been executed. The number of bit changes that occur in each round has been determined. Around 40% bits undergo changes in output data block in each round indicating data diffusion effect of the cipher. In a block cipher, for a given secret key, it is desirable that up to 50% of the bits in the output data block undergo changes due to one bit change in input data block [64]. The MASK encryption algorithm is shown in plate 3.9. Block diagram of MASK encryption scheme is shown in Figure 3.17.

```
MASK ENCRYPTION ALGORITHM

Input:      Plaintext block placed in 128 - bit register $A_1$
            Number of rounds $n$ (maximum 16),  Secret key $K_e$ 128 bits
Output:     Ciphertext block placed in register $A_1$
Procedure:

Initialize Matrix $M_e$ using secret key $K_e$
Run Key schedule to generate Sub-key matrices $Ks_{1e}$, $Ks_{2e}$
for  r = 1 to $n$  do
 $A_1 = f(K,A_1)$   ; Key dependant Poly-alphabetic substitution
 $A_1 = A_1 (+) K_{s1e}(r)$   ; bit-wise XOR, key dependent
 $A_{1R} = A_{1L} (+) A_{1R}$   ; Left half of $A_1$ & Right half of $A_1$bit-wise XOR
 $A_{1L} = ( A_{1L} )^T$ : Transpose Left half of $A_1$ . byte-wise transpose
 $A_{1L} = A_{1L} (+) A_{1R}$   ; bit-wise XOR, data dependent
 $A_1 = A_{1L} || A_{1R}$   ; $A_1 = (A_1)^T$  : Transpose bytes
 $A_1 = A_1 (+) K_{s2e}(r)$   ; bit-wise XOR, key dependent
 $A_{1R} = A_{1R}$ >>> (sum of integer value of bytes of $A_{1L}$ % 6)
 $A_{1L} = A_{1L}$ >>> (sum of integer value of bytes of $A_{1R}$ % 6)
$A_{1L} = A_{1L} || A_{1R}$   ; Concatenate $A_{1L}$ and $A_{1R}$
```

Plate 3.9. MASK encryption algorithm.

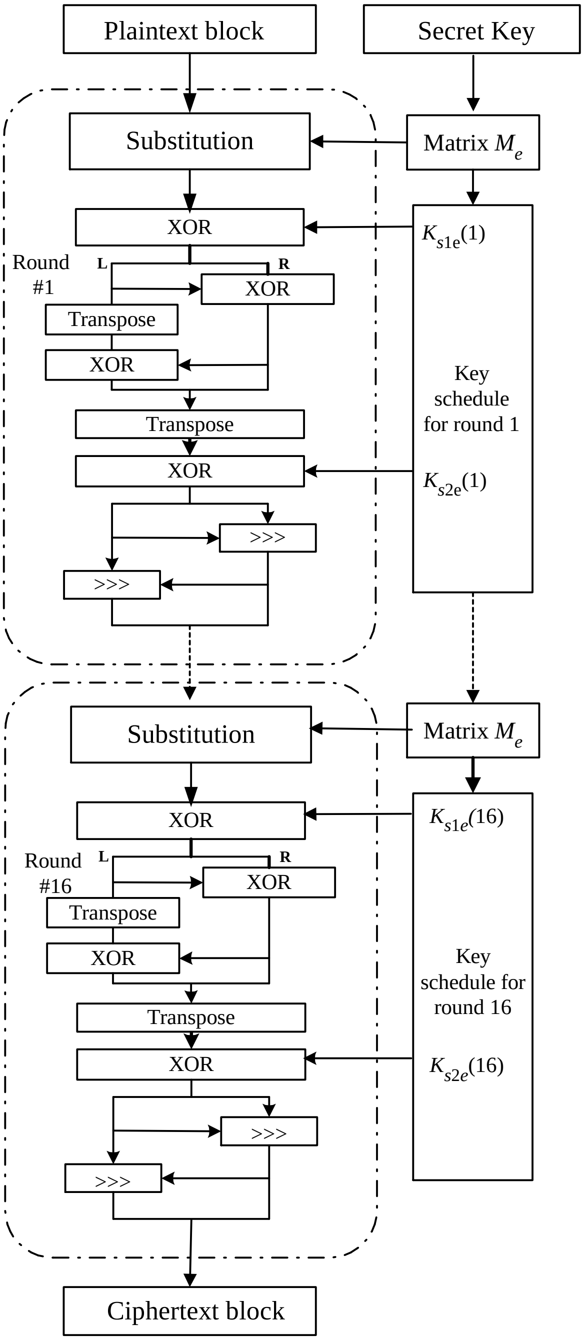


Figure 3.17. Block diagram of MASK encryption scheme.

3.4 The Decryption process

In order to obtain plaintext data blocks from ciphertext data blocks, inverse of encryption, $(E(K_e,P))^{-1}$, is performed. The encryption transformation produced ciphertext $C = E(K_e,P)$. Thus the decryption of C yields P such that $P = (E(K_e,P))^{-1} = D(K_d, E(K_e,P))$. Note that in symmetric key encryption $K_e = K_d$. At the decryption site, it is necessary to have the same matrix, $M_d = M_e$, initialized using the same secret key $K_d = K_e$ for the correct decryption of the message.

Thus we have the same number of sections in the decryption algorithm. These sections are Matrix formation section, Key schedule section, Inverse diffusion and Inverse substitution sections. The block diagram of decryption process is given in Figure 3.18.

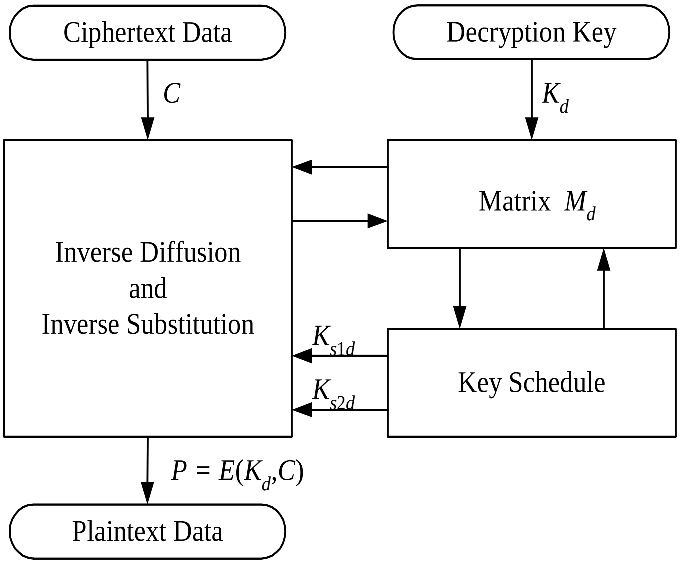


Figure 3.18. Block diagram of Decryption process.

3.4.1 Matrix Initialization (Decryption)

A matrix, M_{d1}, with 16 rows and 256 columns is created using the decryption key, K_d, in the same way as the matrix M_{e1} had been created. The matrix, M_{d1}, is further shuffled in the same way as the matrix M_{e1} had been shuffled to obtain another matrix, M_d.

3.4.2 Decryption Key Schedule

Sub-keys used in decryption round operations are generated by a key scheduling procedure exactly similar to the one used in the encryption algorithm. In this procedure, sub-key matrices K_{s1d} and K_{s2d} (of size 16 × 16) are derived from the base matrix M_d. These pairs of key are used in the inverse diffusion operations performed in the block cipher. As the procedure is same as that of the key schedule in the encryption section, it is not discussed again in this section.

3.4.3 Inverse Diffusion and Inverse Substitution Rounds

The transformation of ciphertext data block, consisting of 16 bytes into plaintext data block is carried out in the inverse diffusion and inverse substitution round operations. This block performs the reverse operations carried out in the substitution and diffusion round of the encryption section. Since, the sequence of operations carried out on data block in the encryption process are substitution and diffusion, in the decryption process, it has to be performed in a reverse sequence. Therefore, in the decryption process, the first operation to be performed on data block is inverse diffusion and the second operation performed on data block is inverse substitution. The simplified block diagram of the inverse diffusion and inverse substitution round operation section is shown in Figure 3.20.

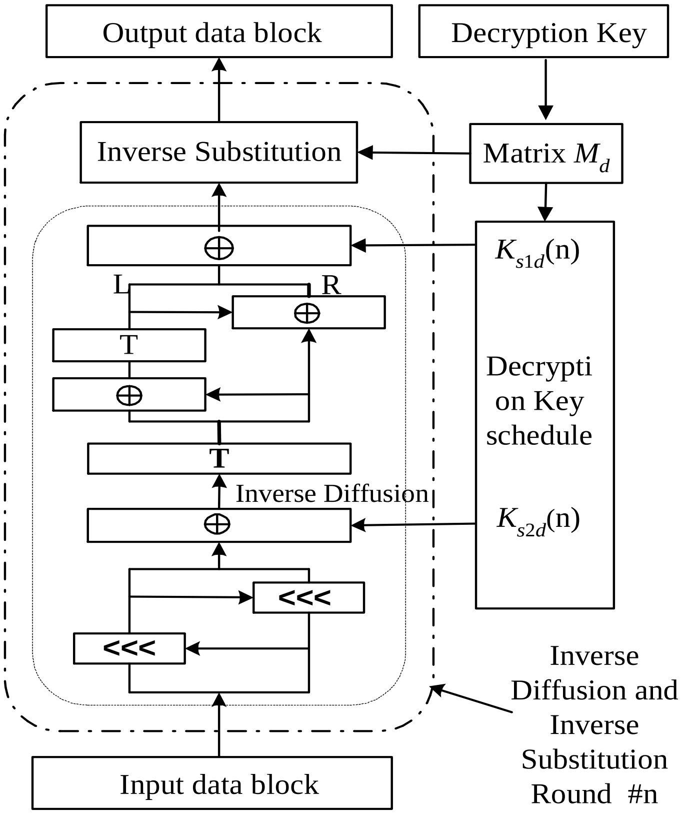


Figure 3.19. Block diagram of Inverse Substitution and Inverse Diffusion round.

3.4.3.1 Inverse Diffusion Section

The inverse diffusion section performs the reverse of operations carried out in the diffusion section described in the encryption process. Input data to this section is a 16 byte (128 bits) data block DB. The input data is manipulated by data based rotations, addition of sub keys, byte transpose operations and data additions as shown in the block diagram of one round of diffusion section given in Figure 3.20. It may be noted, here, that the sequence of operations carried out in the inverse diffusion section is in the reverse order as carried out in the diffusion section of encryption process. These operations are described in steps as shown below after the figure.

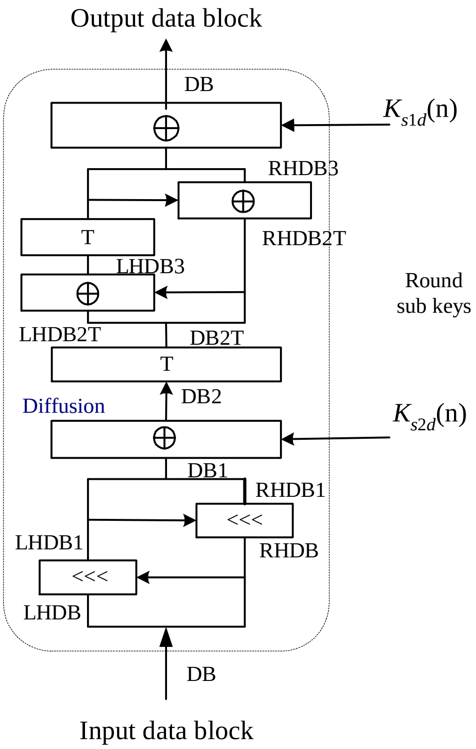


Figure 3.20. Block diagram of one round of Inverse Diffusion.

1) Input data block DB, is bifurcated to left half (8 bytes B0, B1,.., B7) data block LHDB and right half (8 bytes B8, B9, …, B15) data block RHDB.
2) The left half data is left circular shifted number of times equal to the integer value given by the sum of integer values of the right half data *MOD* 6 to get LHDB1.
3) The right half data is left circular shifted number of times equal to the integer value given by sum of integer values of LHDB1 *MOD* 6 to get RHDB1.
4) LHDB1 and RHDB1 are concatenated to get 16 byte data block DB1.
5) DB1 is XOR-ed with the n^{th} round sub key $K_{S2d}(n)$ to obtain DB2.

6) DB2 is transposed to get DB2T.
7) DB2T is bifurcated to left half data, LHDB2T and right half data RHDB2T, each 8 bytes long.
8) LHDB2T is XOR-ed with RHDB2T to obtain LHDB3.
9) LHDB3 is transposed to get LHDB3T.
10) LHDB3T is XOR-ed with RHDBT2 to obtain RHDB3.
11) LHDB3T and RHDB3 are concatenated and XOR-ed with n^{th} round sub key $K_{s1d}(n)$ to obtain DB which is the output data block of the inverse diffusion section.

3.4.3.2 Inverse Substitution Section

The inverse substitution section performs the reverse operations performed in the substitution section of the encryption process. The sequence of operations performed in this section is also in the reverse order. $C(i)$ is the i^{th} data byte in the input data block of this section. Figure 3.21 shows the block diagram of inverse substitution section. The inverse substitution operation is described in steps as follows:

1) Let $i = 1$.
2) Search and locate the byte value represented by $C(i)$ in the i^{th} row of the matrix M_d.
3) Obtain the column number j, in the i^{th} row where the byte $C(i)$ has been located.
4) Assign value (j-1) to $P(i)$ which gives inverse substitution to $C(i)$.
5) Increment i.
6) Go to step 1 till i becomes > 16 (the block size is 16).

For example, let the 1st byte in the input data block is, $C(1)$, in the input data block be '#'. We proceed to search '#' in the matrix M_d to find the column number j in the 1st row where $C(1) = M_d(1,j)$. Then we use the byte value (j-1) as the data output $P(1)$ corresponding to input data byte $C(1)$. It may be noted that the matrix M_d has columns 1 through 256 and the byte value in each location is in the range 0 through 255 only. And this is the reason why we assign (j-1) for the output byte value corresponding to an input byte in a block of input data to the inverse substitution section. Let the 2nd byte, $C(2)$, in the input data block be '%'. We proceed to search '%' in the matrix M_d to find the column number j in the 2nd row where $C(2) = M_d(2,j)$. Then we use the byte value of (j-1) as the output data $P(2)$ corresponding to $C(2)$.

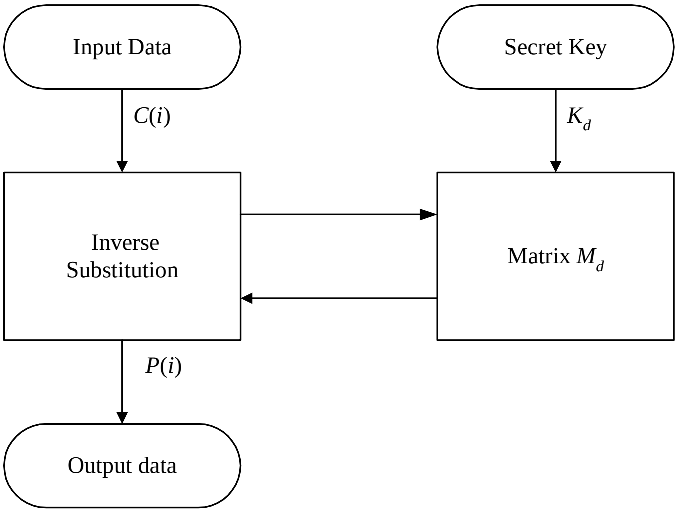


Figure 3.21. Block Diagram of Inverse Substitution.

In this way all input data bytes in a block of input data to the inverse substitution section are converted. Figure 3.22 depicts the inverse substitution procedure. Plate 3.10 shows the inverse substitution pseudo code. Figure 3.23

shows the simplified block diagram of decryption procedure. The MASK decryption algorithm is shown in plate 3.11.

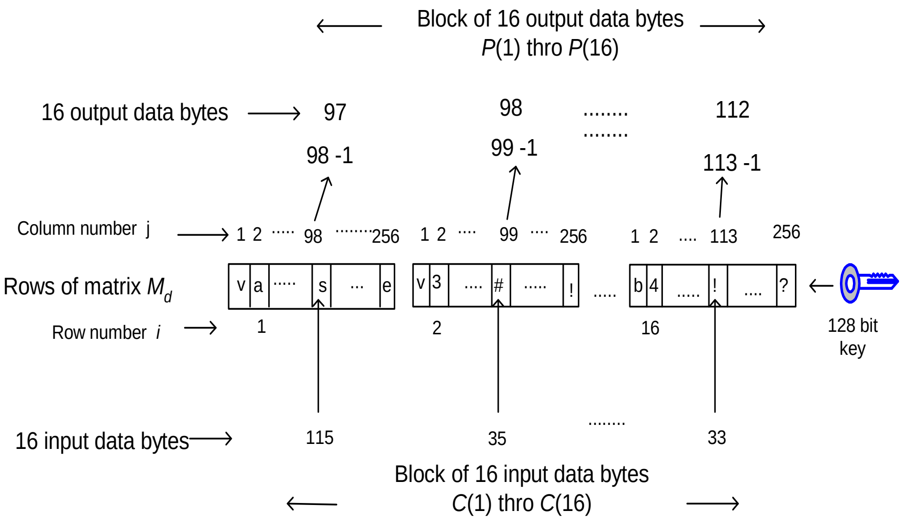


Figure 3.22. Inverse Substitution mapping procedure.

```
For i ← 1 to 16
 j = 1
  while $C(i)$ not equal to $M_d(i,j)$
    j = j+1
  Endwhile
$P(i)$ = ( j-1)
EndFor
```

Plate 3.10. Inverse substitution pseudo code.

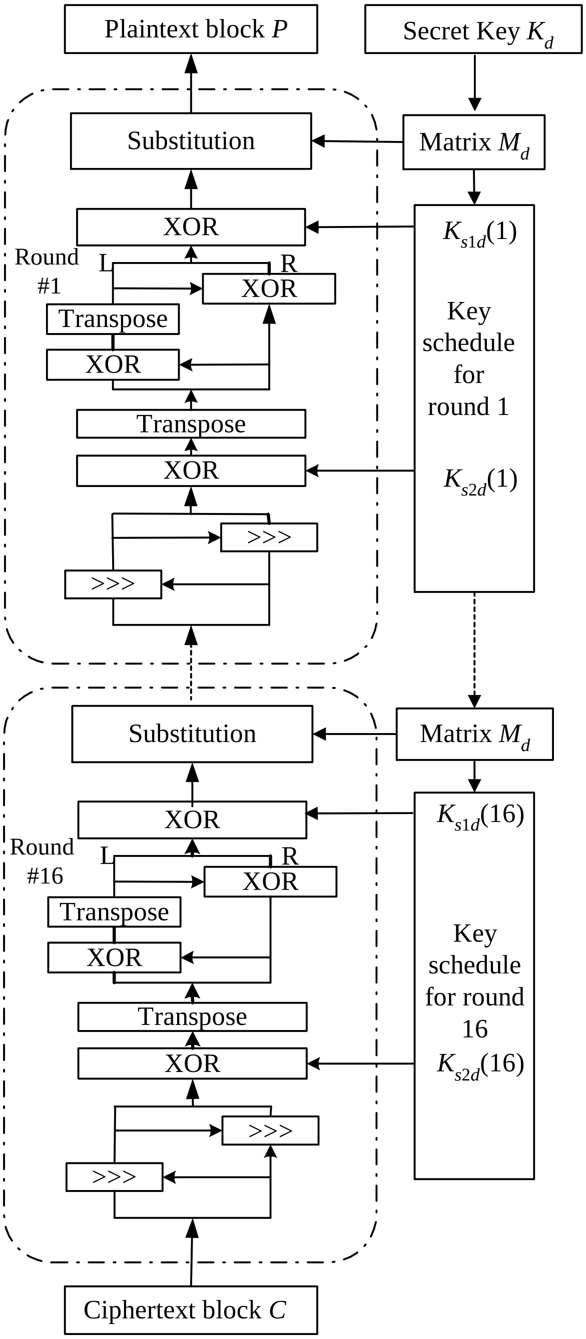


Figure 3.23. Simplified Block Diagram of Decryption Procedure.

MASK DECRYPTION ALGORITHM

Input: Ciphertext block placed in 128 - bit register A_1

Number of rounds n (maximum 16)

Secret key K_d 128 bits

Output: Plaintext block placed in register A_1

Procedure:

```
Initialize Matrix $M_d$ with secret key $K_d$
Run Key schedule to obtain $Ks_{1d}$, $Ks_{2d}$ sub-key matrices
For $r = 1$ to $n$ do
$A_{1L} = A_{1L}$ <<< (sum of integer value of bytes of $A_{1R}$ % 6)
$A_{1R} = A_{1R}$ <<< (sum of integer value of bytes of $A_{1L}$ % 6)
$A_1 = A_{1L} || A_{1R}$    : Concatenate $A_{1L}$ and $A_{1R}$
$A_1 = A_1 (+) Ks_{2d}(r)$   ; bit-wise XOR, key dependent
$A_1 = (A_1)^T$  : Transpose bytes
$A_{1L} = A_{1L} (+) A_{1R}$   ; bit-wise XOR, data dependent
$A_{1L} = ( A_{1L} )^T$ : Transpose Left half of $A_1$ . byte wise transpose
$A_{1R} = A_{1L} (+) A_{1R}$   ; Left half of $A_1$ & Right half of $A_1$bit-wise XOR
$A_1 = A_{1L} || A_{1R}$    : Concatenate $A_{1L}$ and $A_{1R}$
$A_1 = A_1 (+) Ks_{1d}(r)$   ; bit-wise XOR, key dependent
$A_1 = f(K_d,A_1)$   ; Key dependant Poly-alphabetic inverse  substitution
```

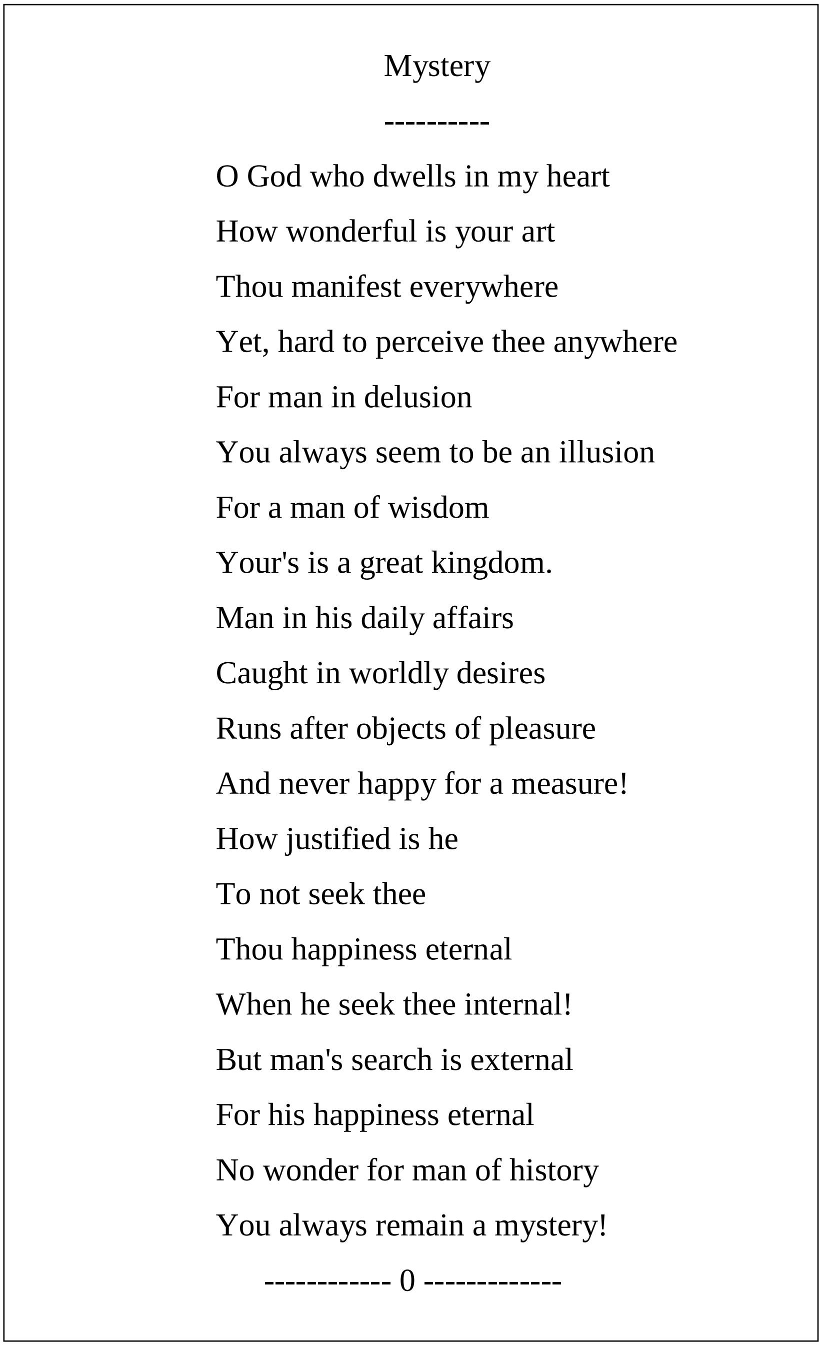
Plate 3.11. MASK Decryption Algorithm.

3.5 Testing of Encryption Algorithm

In this section, basic tests results are discussed. Tests have been conducted for the qualitative evaluation of the encryption algorithm. Test 1 has been conducted to obtain cipher text generated by the encryption algorithm with different number of diffusion rounds. Test 2 has been conducted to observe the plain text produced by the decryption algorithm with the closest key (one bit difference between encryption key and decryption key) and to ensure that the decryption is totally unintelligible. Test 3 has been conducted to reveal the poly-alphabetic property of the encryption algorithm. Test 4 has been conducted to measure the number of bit changes in a block of ciphertext characters produced with one bit change in the encryption key or plaintext data in various rounds (avalanche characteristics). Test 5 has been conducted to obtain the propagation of key change through round outputs. Test 6 has been conducted to obtain the propagation of data change through round outputs. Test 7 has been conducted to obtain the encryption speed for comparison with AES. Test 8 has been conducted to obtain image encryption and decryption. They are explained in the following sub sections.

3.5.1 Test 1- Ciphertext Generation from Plaintext Message

In this section, results are generated to show the difference in cipher text produced for different number of diffusion rounds using the same plain text message. For a given plaintext message, the MASK encryption algorithm have executed with one diffusion round and sixteen diffusion rounds and the ciphertext messages obtained. Plate 3.12 shows plain text message used for the test and plates 3.13 and 3.14 show cipher text messages generated with 1 and 16 diffusion rounds. The secret key used for these encryptions is ‘Life’s beautiful’.



Mystery

O God who dwells in my heart
How wonderful is your art
Thou manifest everywhere
Yet, hard to perceive thee anywhere
For man in delusion
You always seem to be an illusion
For a man of wisdom
Your's is a great kingdom.
Man in his daily affairs
Caught in worldly desires
Runs after objects of pleasure
And never happy for a measure!
How justified is he
To not seek thee
Thou happiness eternal
When he seek thee internal!
But man's search is external
For his happiness eternal
No wonder for man of history
You always remain a mystery!

------------ 0 -------------

Plate 3.12. Plain text message used for encryption.

```
;k%ZbCM`S~}\L\=&6WFRZF
:"$c|9<=f`8(:krso8-[DK)S
~2N3S@(SaD<3IU ??+Ff43:7" I^
?t23%gJZ0{{o!2^tW2+gaIe
" sn13%1/X@yBly/Gm/;Gnu
)7Gz&3>,`U@,2cW8n|=!=^]
?')%kc"□a/ A+wd#'8c1~(9×Ny5? Vx_jrVg/g□U<
□ P"IUB?!$$|8&4[i7Le)pA
9+&+ )U&5FBFLt4by)/|'+d;
-J×=>YoaM )7u!PnJ.=r"y
$D#v^I[b"×4!2K3mP'3D3 w
tKQpc<bd8×.@#bE{&7G/rG
$m@?3&+×3IH× ?k0x;-6 (
$q|8;+9_u]B!C□1^
Q5e?(._n7)<M@3<&!tpF
'B-8i,1b8xP$=-m'YHD z@mH?=6oj
{ %a!;2N=)@{×>It2vbSjX
k/hx[@=vgBqNI
```

Plate 3.13. Cipher text message with 1 round of encryption.

It may be noted that the number of lines of text in plaintext and apparent number of lines of ciphertext generated are not same. It is because of the fact that in the ciphertext generated, there exists special control characters such as carriage return and line feed etc. Wherever there is a linefeed character in the ciphertext output, it goes to the next line in the ciphertext printout. This can cause more lines in the ciphertext printout. Similarly, after every line of plaintext there is a line feed character and thus the plaintext printout will show exact number of lines as found in the plaintext message. But in the ciphertext output, there can have less number of characters that represent line feeds. In that case there can be less number of lines in the ciphertext message corresponding to a plaintext message. However the number of characters (including special characters) will be same in both plaintext message and ciphertext message.

```
×6z!2-+eDjY)o6q~@KIhDrZ
06W)?4~3Ox] j32E-OcpNoD,w
tz?,>r^ZV; @!1^Bec,s5<B)92':A-c/Hc6Gb;8_>tgEu<
z02^p:6w.I!5=MV@dY6{Z
@!-?^^dE,|6Jz), ~ImU:@5U1<,_OWOxS-j] W1U4J
Wdc0v%"6R)4X!OPD6|(hTbjc[BxD(@U
u6T(u8IZU.AEvI!;<!yDxD,v3FN6}weE9jP>8M#DMmE(v
c0MSpkc/H    #<z39M:□8pZW<w(
sU3;,J3!tWY,Ez>z)U9tNec:D~6G
)g.Q7@xX(y@P?,ZOZ\jRz
), ^^eX,D# G<=DwfT|A/F!?,U-^RWDE-?g
3h^QY:ju"K0-lAVV5H 9gJ!;< fT^WS9jc
z(,PNkijPN6T;+ MpnA0{
6H?7V!!xc<5q)o6q~MI.|<a)z!b,MI^xcj
```

Plate 3.14. Ciphertext Message with 16 Rounds of Encryption.

3.5.2 Test 2- Decryption with Closest Key

In this section decryption of an encrypted message is attempted with a decryption key that is very close to the encryption key (difference between encryption key and decryption key is only one bit). The aim is to show that the decrypted message is totally unintelligible even if the assumed secret key is closest to the original secret key. It can be seen clearly that the decrypted message is totally unintelligible without leaving any trace of the words in the original plain text message. Plate 3.15 and 3.16 show decrypted messages with one round and 16 rounds of operations, respectively.

```
[(dK!wþ)cHu_À-v~]2XX(□áG)C□R
<l7¢'W]Ù)üJG+X×!yH4'×`Y(6˜EzG3J ×PDd1F6Ù`s0ó-
,<□péQ5sZQ!□›G+G+Gy,G7dv{     „˜Ì#X
66PnDC7ZfL-„s)±ó-68l,Bue1L!6{Ì{ó
G,< □M'W[S1R-P×l~<K □°?o]Ù+5JX/å□<T
□PDd6L\:&Ì7C’#<a/°=W]@(„{§K×,1□i5°8WdÙ(2-
K~C,8OaQN}TF14Sxu66Ôo1K4Z+Ù(6&J}
+(ku5¢'ouK!wJP.÷z9MsDP62_S!2&V}
LKGdDM7)vM\;zO(ãG8FrDB'[vF7˜˜Fwl!<Ô 9ÄNbZA-
6¶Ì/åG6<vP'``K\R˜F%C,6<eO38afÙ
$27O/Îz¶J□7d]F0&=G+ÎG6<0F7]1K$6˜Ì/Î,7Kn□
KG}QJ8□{B×J+6Je□Q5V1>7□˜W{×18lO
eAd1?-RJG7÷83Ge0¢'sc:6izKK376Ôo5C7d1A
+wJN7ÎG<O  JNb`M^&3PzC,JÔa×¢'dvR%:sÌ7
C5³JtéQu/üIi^GUcHu_À-v°_2{Ii^GUcHu_À-D°
```

Plate 3.15. Decrypted message with 1 bit difference in key; rounds = 1.

```
Rn:RÃ×yo>dG1C[)(Phn»:twlkgNf«{I2~~GBí# l×!igôBÎ
~pFI:Nr§W%ut¦p<5x}e
-Ö8 NÖ3C;g+:RPNv¦ô2}u«"ä(T+9VPSa¸?g~^H
ä4j+D+Ãtt
l.ulçj:>h}Pa6%rW¼×%NÔ}2Bh$@à%no$8
NÅ{:Ct|eÃNmWZ8uk«p8~~A)Þ)iY@,^5p
•p'GB®M
ºd<N^ÖTö5x}ž×PRsWFgM[«3A•r~GA:%mEvXS_HzC
•e~:B:ZiÎ
gSw434zGjÞLh7M"%NP!?:i$üB:Ns□ô."H}'C2~'ORÃ×
Ù#Oz8%25ghE)ÃZs□ô.mvç428hqP9PSa□2uwÕ"2C~#P
~rªJ@×^PpG˜z{D~àNdWd<uu«T'5~yFRP)eª¹g~u«7aRex
@BÈ
ZpD7Oj¦pä□h}GïiWl7uu«p>?h%eRÈNeWd7~x5}
:}Gk=<tm Y^"N¦74j e>teÌZ.!_Å{a`n}eNà)
lF9#tç7>¼~+9&aÎvu&NP!CBh}e~®×
¹F7u^4p%;z{F9JiY
WgSaP3<2~}P¾ZRnWFg(d¦$ä4tg+Y=F-
T>dG1C[)•/hžY=F-T>dG1C[2•
```

Plate 3.16. Decrypted message with 1 bit difference in key; rounds = 16.

3.5.3 Test 3- Poly-alphabetic Nature Test

In a block cipher with block size of 128 bits (16 characters), blocks of 16 plain text characters are converted to cipher text characters at a time. In this test the poly-alphabetic substitution capability of the encryption is revealed. For the same plain text characters within one block, a poly-alphabetic encoding should produce distinct cipher text characters in the ciphertext output block. To test this feature, a block of plaintext input data with 16 identical characters (‘A’ → Decimal 65, hexadecimal 41) is chosen. The cipher is executed for 1 to 8 ciphering rounds and output data block of each round is obtained. The ciphertext output block is shown in hexadecimal format in plate 3.17.

```
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41        ← Input data block (hex)
-----------------------------------------------------------------------------------------------
E9 EC C5 F5 D2 D3 DA E6 3A 32 20 17 30 00 3F 36  ← 1 round  o/p data block
2C 0C 22 32 0A 2E 61 65 46 79 56 21 30 31 25 69       ← 2 rounds  o/p data block
B4 B7 33 BF 8D EF F3 2E 70 B7 43 60 C6 30 CB 80  ← 3 rounds  o/p data block
2E 33 66 FD AF 50 A8 B0 B3 9F F8 EE C5 5C E9 F3 ← 4 rounds  o/p data block
85 97 D2 5E 0B B8 0E 1A 1E 00 62 50 3A C1 5A 65  ← 5 rounds  o/p data block
F9 43 95 7F 74 57 37 DE 45 7B 1A 88 16 4C 41 DD   ← 6 rounds  o/p data block
A5 F6 C6 45 F5 28 F2 E0 2F 51 D1 6B E2 2B 66 C0   ← 7 rounds  o/p data block
9D 59 71 D0 23 84 A1 9C 49 70 BF 12 E5 4F 80 78    ← 8 rounds  o/p data block
```

Plate 3.17. Poly-alphabetic test result.

3.5.4 Test 4- Avalanche Property Test

In a block cipher that is said to have better cryptographic strength, a change of one bit in the key or in the plain text block affects many bits in the cipher text output block. This is called avalanche effect. The change is diffused to various parts of the cipher text block instead of confining only to one part. DES has a very good avalanche effect due to the expansion and contraction permutations performed in its various rounds. In the diffusion test, first a block of plaintext characters is converted into cipher text characters and the corresponding bit pattern is obtained. Then with a change of only one bit in the key and using the same plain text character block the output cipher text character block is obtained and the corresponding bit pattern is analyzed. The number of bits changed with one bit change in the key can be determined. Similarly, a change of one bit in the plain text block is introduced to determine the changed number of bits for the same encryption key. The test is carried out with different number of rounds of diffusion operations. Plate 3.18 shows the bit pattern in the cipher text outputs obtained using two keys that differ by only one bit for one round of substitution and diffusion operation. Plate 3.19 shows the bit pattern in the cipher text outputs obtained using two keys that differ by only one bit for sixteen rounds of substitution and diffusion operation. Table 3.1 shows the number of bits changed with one bit change in the key for different number of diffusion rounds compared with the diffusion characteristics of AES. Plate 3.20 shows the bit pattern in the cipher text output obtained using a plain text and a particular key. Plate 3.21 shows the bit patterns obtained for the same key but a plain text block that differs by only one bit for different diffusion rounds. Table 3.2 shows the number of bits changed with one bit change in the plaintext for different number of diffusion rounds compared with the diffusion characteristics of AES.

```
1001001010111110011001111001011001011100100001001110001111100010
0001011001001110000010100100101010100111000011010110110100001101
Plaintext  = abcdefghijklmnop, Key = Godiseternalyes!, Number of rounds = 1
1100011011011010111010100000111000101100100100101110000011101110
0110111001000010100101000101000111100101010010000100001101110001
Plaintext = abcdefghijklmnop, Key = Hodiseternalyes!, Number of rounds = 1
```

Plate 3.18. Key diffusion test result with one round operation.

```
1111011011011110100100000010010000001111010010111000010000001011000
1100001011111001110110011000111101011110100010101100001000111001
Plaintext = abcdefghijklmnop, Key =Godiseternalyes!, Number of rounds = 16

1111000001101100010001101010010110000001100110001000000110101101
0100011111001010110010101010000000100110111011001000010010110101 0
Plaintext = abcdefghijklmnop, Key = Hodiseternalyes!, Number of rounds = 16
```

Plate 3.19. Key diffusion test result with 16 round operations.

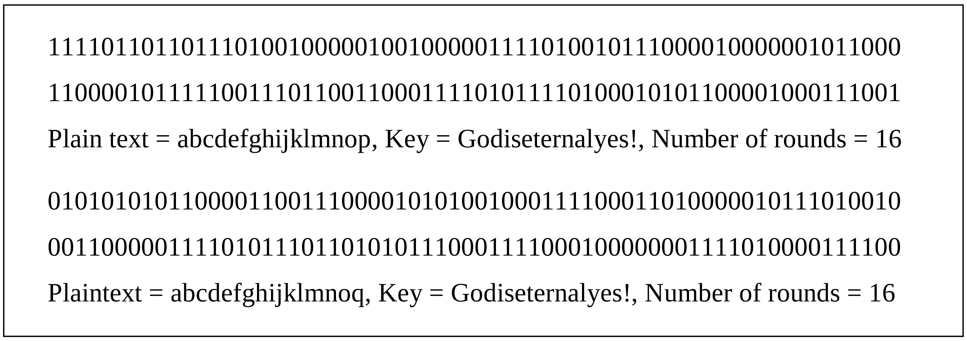
Table 3.1. Comparison of Key Diffusion Characteristics of AES and MASK.

No. of rounds	No. of bit changes for one bit change in key	
	AES	MASK
1	22	48
2	51	46
3	42	43
4	54	52
5	51	53
6	47	46
7	47	55
8	46	43
9	55	54
10	53	57

```
1001001010111111001100111100101100101110010000100111000111100010
0001011001001110000010100100101010100111000011010110110100001101
Plain text = abcdefghijklmnop,Key = Godiseternalyes!, Number of rounds = 1

1001011001011100100001001110001111100010100100101011110101100111
0100101010100111000011010110110100001101000101100100110100001010
Plaintext  = abcdefghijklmnoq, Key = Godiseternalyes!, Number of round = 1
```

Plate 3.20. Data diffusion test result with 1 round operation.



```
1111011011011110100100000100100000111101001011100001000000001011000
1100001011111001110110011000111101011110100010101100001000111001
Plain text = abcdefghijklmnop, Key = Godiseternalyes!, Number of rounds = 16

0101010101100001100111000010101001000111100011010000010111010010
0011000001111010111011010101110001111000100000000111101000011110
Plaintext = abcdefghijklmnoq, Key = Godiseternalyes!, Number of rounds = 16
```

Plate 3.21. Data diffusion test result with 16 rounds operation.

Table 3.2. Comparison of Data Diffusion Characteristics of AES and MASK.

No. of rounds	No. of bit changes for one bit change in key	
	AES	MASK
1	10	46
2	46	50
3	54	42
4	57	51
5	53	50
6	46	47
7	44	50
8	49	43
9	49	50
10	50	46

3.5.5 Test 5- Propagation of *Delta-K* through Data in Rounds

A block of plaintext data (128 bits or 16 characters of plaintext) is used as input to the diffusion section in this test. With a given secret key *K*, the section is executed. The output block produced in each round is recorded. Then, with the same block of input plaintext and a secret key value that differs by one bit (*Delta-K*) is used to execute the section. The output block produced in each round is recorded. The difference in byte values of the data blocks produced in respective round is calculated. The differences in byte values show how one bit change in secret key propagates through data in various rounds. The differential key propagation is very important in connection with the resistance of the cipher against differential attacks. If the difference in byte value between round outputs due to one bit change (or for a given difference) in key value is not consistent then the cipher exhibits strength against differential crypt analysis. Figure 3.24 shows the variation of difference in byte values (only byte 1 and byte 2 are shown in the graph. All bytes in a block exhibit similar characteristics) of the data blocks produced by each round due to one bit change in secret key.

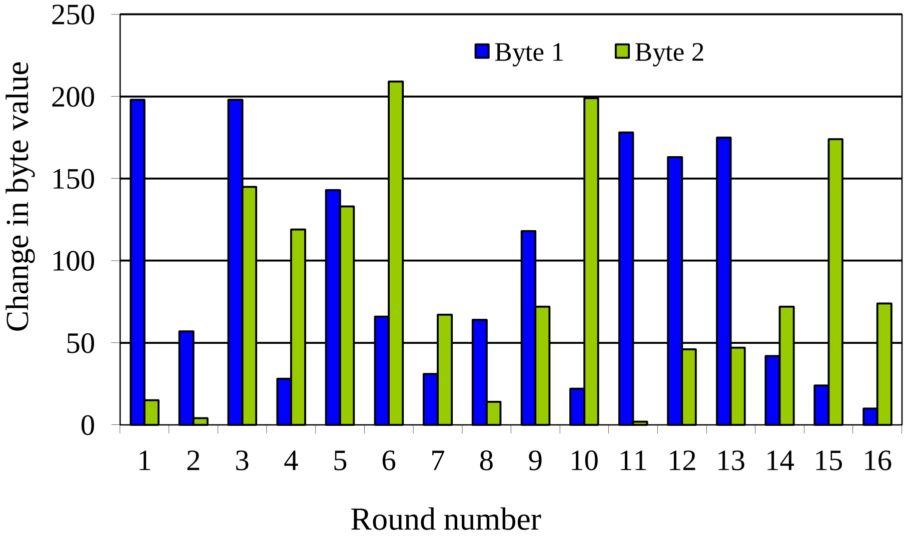


Figure 3.24. Propagation of *Delta-K* through rounds.

The Figure indicates that the difference propagation is inconsistent, meaning good differential characteristics, through rounds revealing resistance against differential attacks on the cipher.

3.5.6 Test 6- Propagation of *Delta-P* through Data in Rounds

The differential data (*Delta-P***)** propagation is also important in connection with the resistance of the cipher against differential attacks. With a block of plaintext data and a given secret key, *K*, the diffusion section is executed in 16 rounds. The output block produced in each round is recorded. Then, with the same key, and a plaintext block that differ by 1 bit (*Delta-P*) is used to execute the diffusion section. The output block produced in each round is recorded. The difference in byte values of the data blocks produced in respective round is calculated. The differences in byte values show how one bit change in plaintext data block propagates through data in various rounds. If the difference in byte value in round outputs due to one bit change in plaintext data block is not consistent then the cipher exhibits strength against differential crypt analysis. Figure 3.25 shows the propagation of *Delta-P* through the two bytes of data blocks produced in each round due to one bit change in plaintext data.

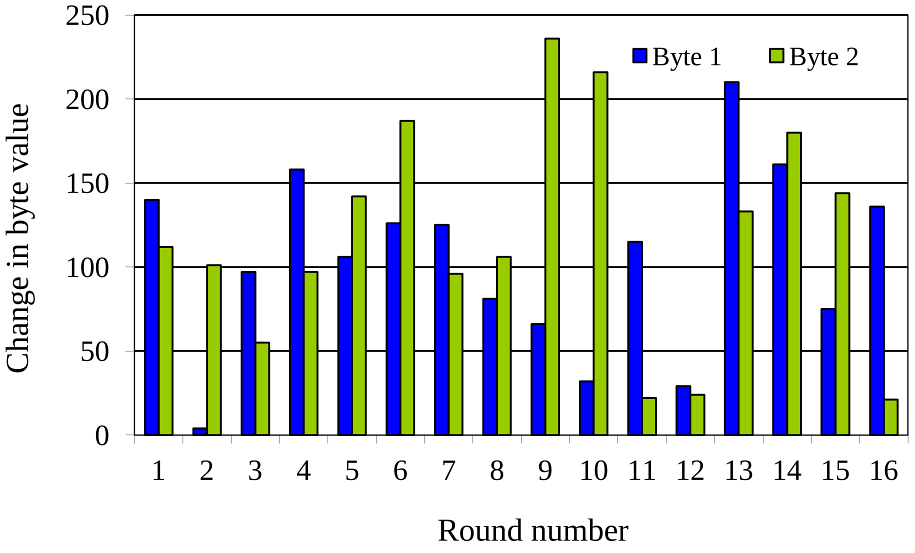


Figure 3.25. Propagation of *Delta-P* through Rounds.

The Figure indicates that the difference propagation is random, through rounds, revealing resistance of the cipher against differential attacks.

3.5.7 Test 7- Throughput Comparison

Throughput of an encryption algorithm is the number of bytes of plaintext data it can convert to ciphertext data in one second. The throughput refers to encryption speed. In any encryption scheme, higher conversion speed is an advantage while remaining secure. The encryption speed of MASK is compared with AES while both algorithms running with a plaintext message having size 45,612 bytes and secret key Godiseternalyes!. The throughputs of these algorithms are recorded. Table 3.3 shows the comparison of performance of MASK and AES. It can be seen that MASK encryption algorithm is 8 times faster than AES. The tests have been conducted using Turbo C in an Intel Atom 1600 MHz processor with Windows-XP operating system.

TABLE 3.3. Comparison of throughput of MASK and AES on an Intel Atom 1600 MHz processor.

Encryption Algorithm	AES	MASK
Throughput Bytes/sec	16,941	103,767

3.5.8 Test 8- Image Encryption and Decryption

An encryption scheme should be capable of encrypting plaintext messages and images to generate ciphertext messages and cipher images without leaving any trace of the plaintext or the image in the encrypted output. An image contains redundant information and there is strong correlation between adjacent pixels in horizontal, vertical and diagonal directions of the image. A weak encryption may not be able to hide these aspects of the original image in the ciphered image. Therefore, even if the ciphertext message generated from a plaintext message by an encryption scheme is secure, the cipher image generated from a plain image may not be hiding certain characteristics of the original image. This can give some clues to the crypt analyst regarding the nature of the original image there by making crypt analysis easier. Therefore, it is important that an encryption scheme should be analyzed using images. Moreover, encrypted and decrypted images facilitate statistical analysis viz. histogram analysis, adjacent pixel correlation analysis cross correlation analysis and key space analysis. Cross correlation analysis, between input and output is also made easily possible with images.

The encryption algorithm has been implemented in MATLAB7 for testing the algorithm using images. Different standard images available in MATLAB were used as input and the encrypted images and decrypted images were obtained. These images have different size and texture besides having different background illuminations, histograms and adjacent pixel correlation values. Encryption quality measurement and encryption speed measurement using images were also carried out for comparison with AES. The following image encryption and decryption results are presented for the observation on the outcome of encryption and decryption processes. Detailed analysis, with images, is presented in chapter 5. Figure 3.26 shows the input image “Lifting body”, the cipher image and the decrypted image. The secret key used for encryption and decryption operation is ‘Godiseternalyes!’. Similarly, Figure 3.27 and Figure 3.28 shows the input image “Cameraman” and “Saturn” along with their corresponding ciphered and decrypted images for the same secret key.

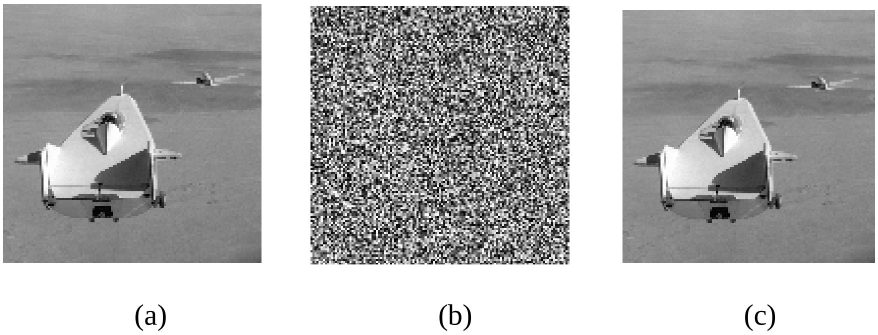


Figure 3.26. Encryption and decryption of image ‘Lifting body’.
(a) Original image (b) cipher image and (c) decrypted image.

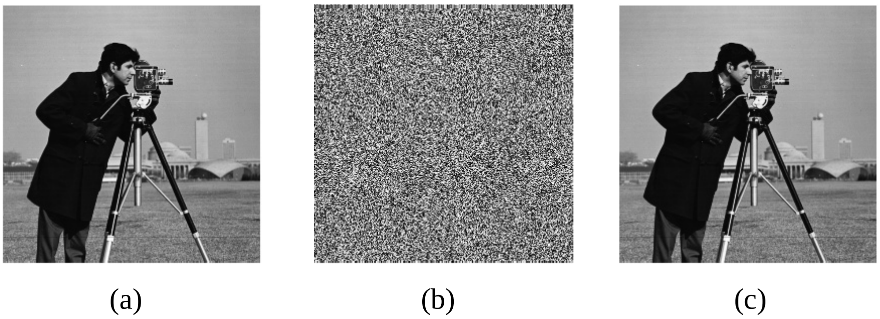


Figure 3.27. Encryption and decryption of image ‘Cameraman’.
(a) Original image (b) cipher image and (c) decrypted image.

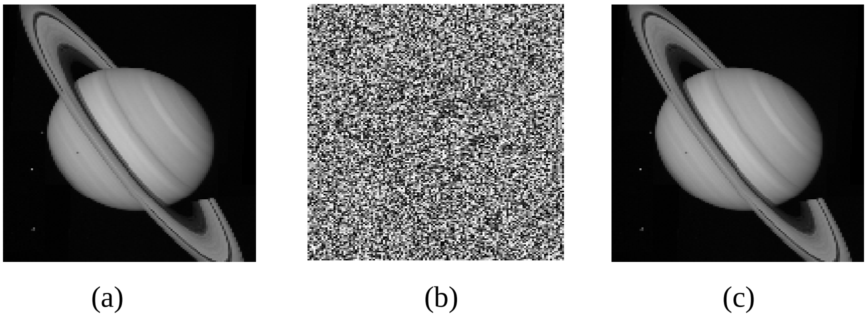


Figure 3.28. Encryption and decryption of image ‘Saturn’.
(a) Original image (b) cipher image and (c) decrypted image.

3.6 Improved Performance of AES using MASK Key Schedule

Encouraged by observing faster encryption and decryption performance of MASK encryption algorithm, an attempt was made to see if the key avalanche and differential key propagation characteristics of AES could be improved by using MASK key schedule in AES.

As a case study AES algorithm has been modified by replacing the original AES key schedule with the matrix based key generation procedure used in MASK. It may be noted, here, that AES use only one set of 10 sub keys for the 10 diffusion rounds when the secret key size chosen is 128 bits. Using sub key set, K_{s1} of MASK, AES has been tested to evaluate the following performance criteria.

- Effect of 1 bit key change on sub-keys
- Key avalanche characteristics of AES
- Propagation of *Delta-K* through data in AES

It has been shown that, the key avalanche effect and differential key propagation characteristics of AES have improved by replacing the AES key schedule with the Matrix based key generation procedure.

3.6.1 Effect of 1 bit Key Change on Sub-keys

Tests conducted to obtain the effect of 1 bit change in secret key on sub-keys would give an indication of the effectiveness of the key scheduling procedure. The number of bit changes in sub-keys due to one bit change in secret key, called sub key avalanche, has been observed. The Matrix based key schedule of MASK encryption has been executed to obtain the sub key values. Then the values are compared with the sub keys produced in the original AES. First, the key scheduling procedure has been executed with a given secret key and the sub-keys generated for 10 rounds have been recorded. Then, with another secret key, with a difference of only 1 bit (one count) from the first key, has been used to execute the key schedule and the sub-keys generated for 10 rounds has been recorded. The number of bits changed in sub-keys, in each round, has been calculated from the recordings and the result has been plotted. Figure 3.29 shows the comparison of the bit changes in sub-keys generated by the MASK key schedule with the number of bit changes produced in sub-keys in AES.

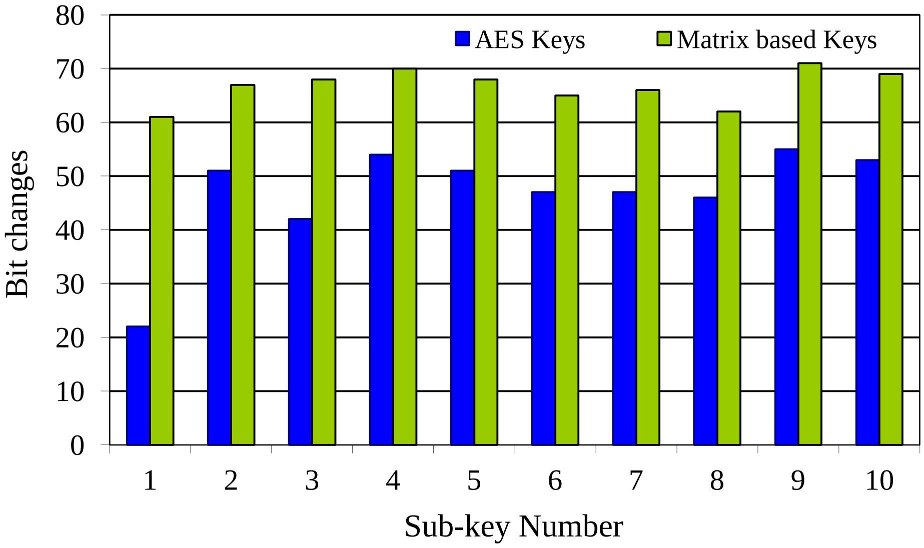


Figure 3.29. Effect of 1 bit change in secret key on sub-keys.

It can be seen from the plot that the sub key avalanche is more prominent in MASK key schedule. This feature can facilitate stronger diffusion of key in to the ciphertext. The key avalanche produced in AES due to MASK key schedule is discussed in the following section.

3.6.2 Key Avalanche Characteristics

In this test, a block of plaintext data (128 bits or 16 characters of plaintext) has been used as input to the cipher. With a given secret key *K*, the cipher has been executed. The output block produced in each round has been recorded. Then, with the same block of input plaintext, and a secret key value that differs by one bit has been used to execute the cipher. The output block produced in each round has been recorded. The number of bit changes that occurred in each round has been calculated to plot the key avalanche in the case of original AES and AES modified with Matrix based key generation procedure. Figure 3.30 shows the change of bits in a block of output data in each ciphering round for one bit change in key produced in AES and the same in AES with Matrix based key generation.

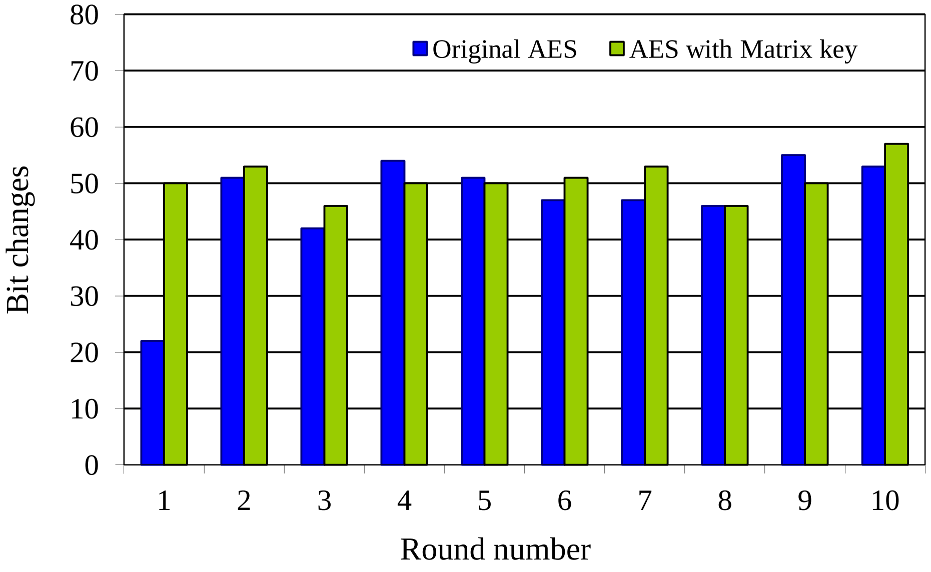


Figure 3.30. Key Avalanche in AES with Matrix key schedule.

It can be seen that AES with Matrix based key generation procedure has an enhanced key avalanche characteristics in the initial and final round outputs.

3.6.3 Improvement in Propagation of *Delta–K* through Data in Rounds

In this test, the differential propagation of key through diffusion rounds of original AES and modified AES with MASK key schedule is obtained. If the difference in byte value in round outputs due to one bit change (or for a given difference) in key value is not consistent then the differential crypt analysis will not yield decryption of encrypted message by the crypt analyst. Figure 3.31 shows the variation of difference in byte values of the data blocks produced by each round due to one bit change in secret key. The results indicate that the differential key propagation is better in AES with matrix based key schedule.

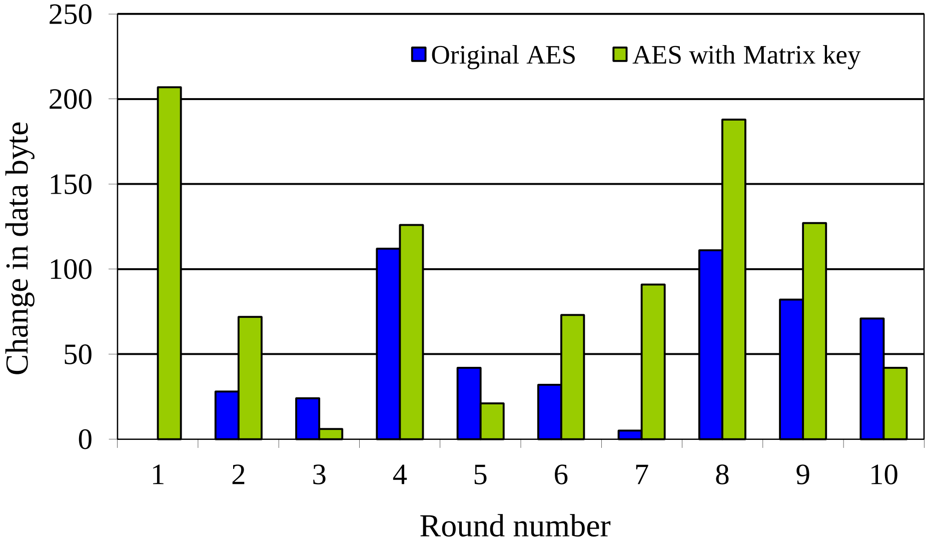


Figure 3.31. Propagation of *Delta–K* through data block in rounds.

Chapter 4

MASK Encryption: Results with Image Analysis

This chapter discusses the tests conducted and analysis made on MASK encryption, with gray scale and colour images. Statistical analysis including histogram analysis, adjacent pixel correlation analysis and mean value analysis have been carried out and the results are compared with AES. Encryption quality and encryption speed are obtained with images of different sizes and the values are presented.

4.1 Results with Image Analysis

Tests have been conducted using images of different sizes and textures for statistical analysis and comparison with AES. These include 1) Encryption of gray scale images and colour images, 2) Histogram analysis, 3) Adjacent pixel correlation analysis, 4) Mean value analysis, 5) Encryption quality, 6) Key space analysis and 7) Encryption speed comparison.

4.1.1 Image Encryption and Decryption

Image encryption and decryption tests have been carried out using standard images of different sizes in gray scale and colour. Encrypted and decrypted outputs have been obtained from MASK and AES and are presented in the following figures. Figures 4.1 to 4.3 show the original gray scale images and the cipher images and decrypted images by MASK and AES, respectively.

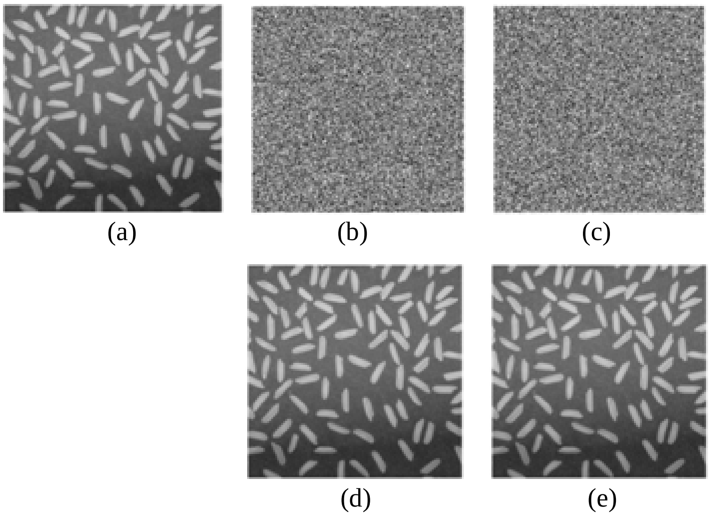


Figure 4.1. Encryption and decryption of Image ‘Rice’ by MASK and AES.
(a) Original Image ‘Rice’, (b) MASK cipher image, (c) AES cipher Image,
(d) MASK decrypted image and (e) AES decrypted image.

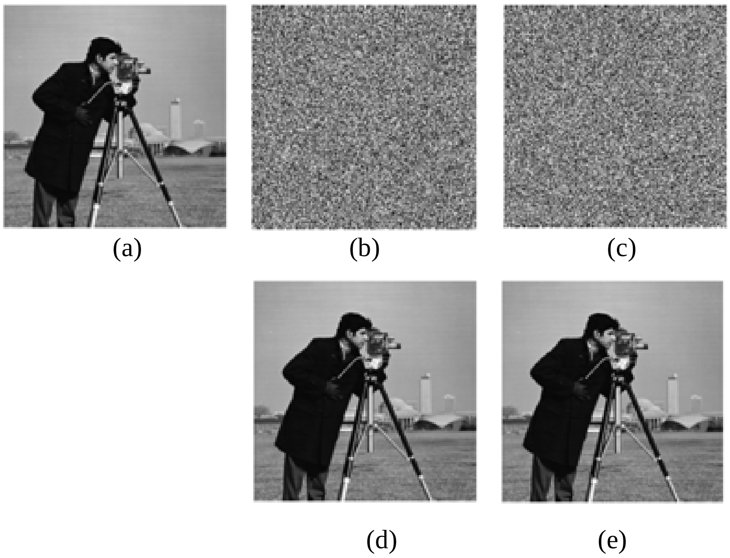


Figure 4.2. Encryption and decryption of Image ‘Cameraman’ by MASK and AES.
(a) Original Image ‘Cameraman’, (b) MASK cipher image, (c) AES cipher Image, (d) MASK decrypted image and (e) AES decrypted image.

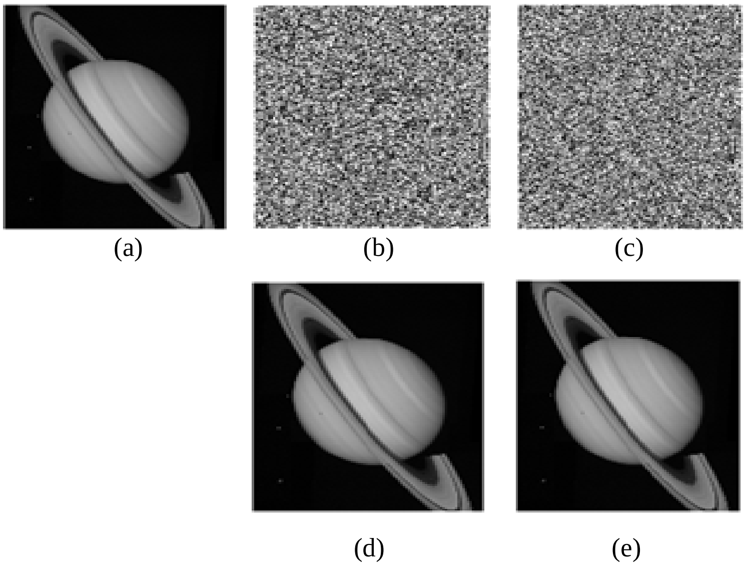


Figure 4.3. Encryption and decryption of Image ‘Saturn’ by MASK and AES.
(a) Original Image ‘Saturn’, (b) MASK cipher image, (c) AES cipher Image, (d) MASK decrypted image and (e) AES decrypted image.

Figures 4.4 and 4.5 show the original colour images and the corresponding cipher images and decrypted images by MASK and AES respectively. It may be noted that in all the encrypted images obtained from MASK and AES no trace of original image is visible.

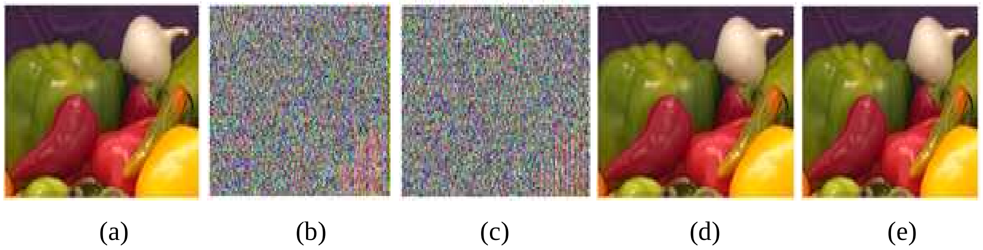


Figure 4.4. Encryption and decryption of Colour Image ‘Onion’ by MASK and AES. (a) Original Colour image ‘Onion’, (b) MASK cipher image, (c) AES cipher Image, (d) MASK decrypted image and (e) AES decrypted image.

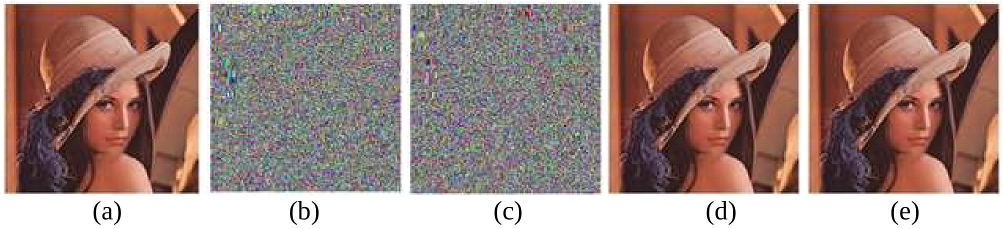


Figure 4.5. Encryption and decryption of Colour Image ‘Lena’ by MASK and AES. (a) Original Colour image ‘Lena’, (b) MASK cipher image, (c) AES cipher Image, (d) MASK decrypted image and (e) AES decrypted image.

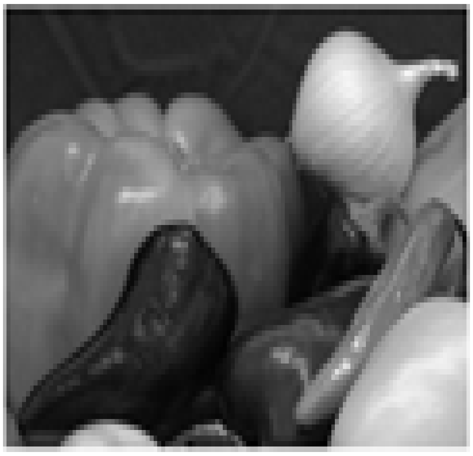
4.1.2 Statistical Analysis

Digital images, accounting for 70% of the information transmitted on the Internet, are important parts of network exchanges [61]. However, the image information, which is different from text message, has larger scale of data, higher redundancy and stronger correlation between pixels [63]. Statistical analysis of encrypted images provides much information about the security of a cipher with reference to statistical attacks that could be launched against the cipher. There are two important methods of statistical analysis of encrypted images. The first is histogram analysis and the second is the adjacent pixel correlation analysis. In the following section, analysis carried out on MASK and AES based on these two methods is discussed.

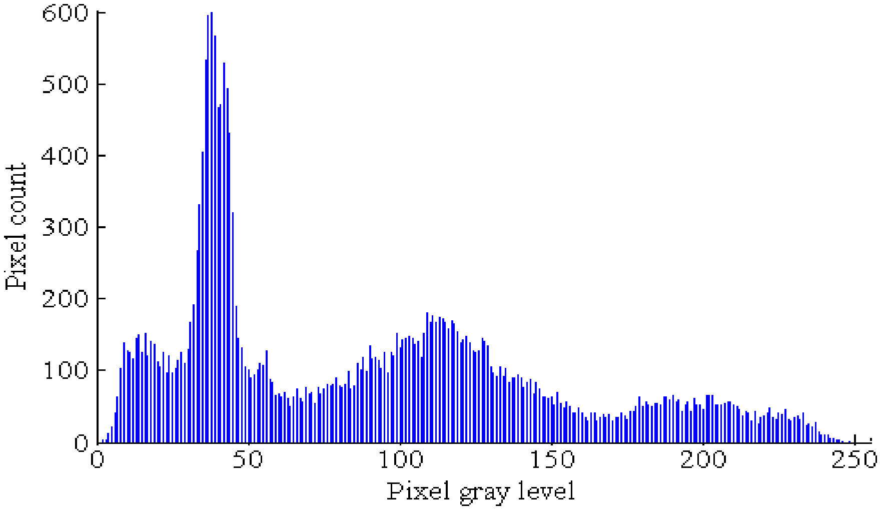
4.1.2.1 Histogram Analysis

In image processing context, the histogram of an image normally refers to a histogram of the pixel intensity values. This histogram is a graph showing the number of pixels in an image at each different intensity value found in that image. For an 8-bit grayscale image, there are 256 different possible intensities, and so the histogram will graphically display 256 numbers showing the distribution of pixels amongst those grayscale values. The image is scanned in a single pass and a running count of the number of pixels found at each intensity value is kept. This is then used to construct a suitable histogram.

Histograms can also be taken of color images - either individual histogram of red, green and blue channels can be taken, or a 3-D histogram can be produced, with the three axes representing the red, blue and green channels, and brightness at each point representing the pixel count. For a good encryption, the distribution of gray scales in the encrypted image should be fairly uniform [65]. Using gray scale images of different sizes and textures, histograms of encrypted images obtained from MASK encryption and AES encryption have been analyzed. It has been observed that the histograms of encrypted images have fairly uniform distribution of pixel gray values and are significantly different from the histograms of the original images. Figure 4.6 shows original gray scale image ‘Onion’ and its image histogram and Figure 4.7 shows the corresponding histograms of MASK cipher image and AES cipher image.

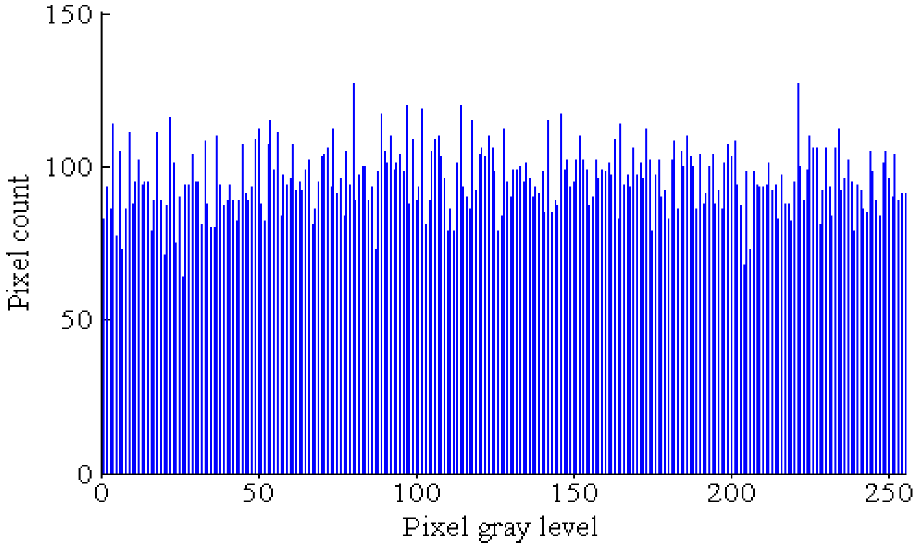


(a)

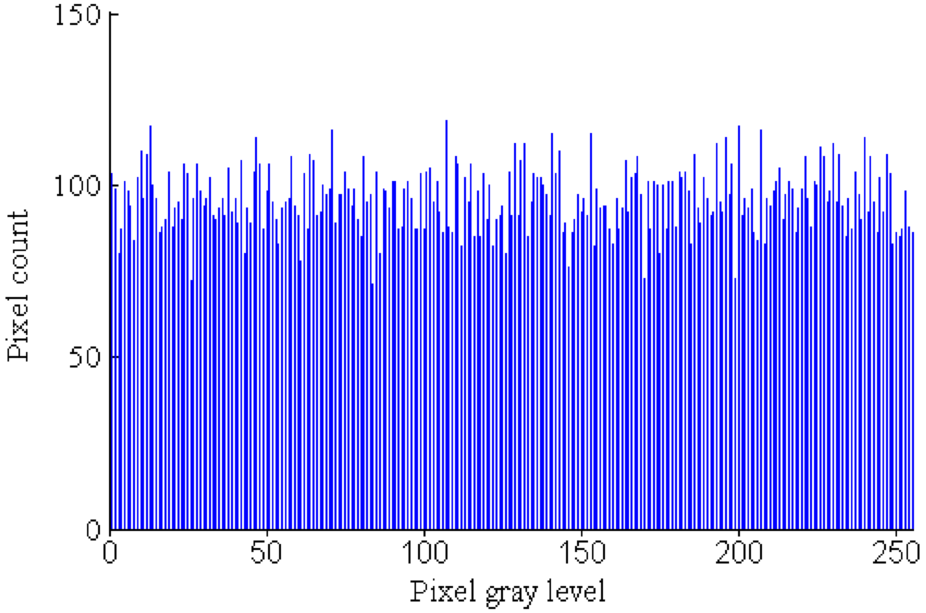


(b)

Figure 4.6. Image ‘Onion’ and Histogram. (a) Original Image and (b) Histogram of image.



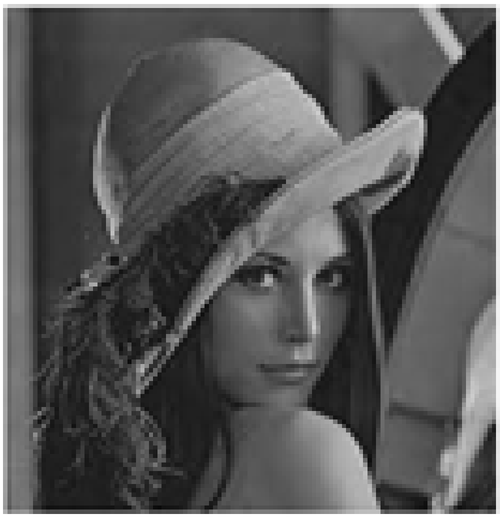
(a)



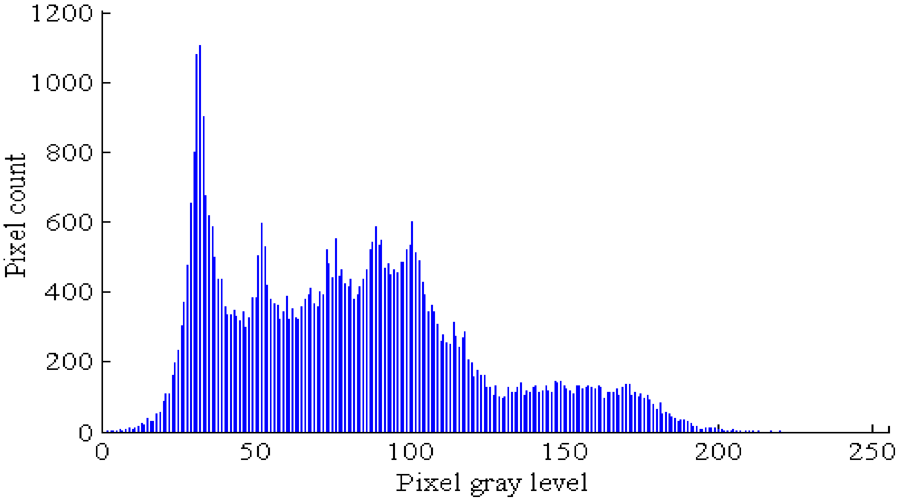
(b)

Figure 4.7. Histograms of cipher images of ‘Onion’. (a) Histogram of MASK Cipher image of ‘Onion’ and (b) Histogram of AES Cipher image of ‘Onion’.

Figure 4.8 shows original gray scale image ‘Lena’ and its image histogram. Figure 4.9 shows the corresponding histograms of MASK cipher image and AES cipher image.

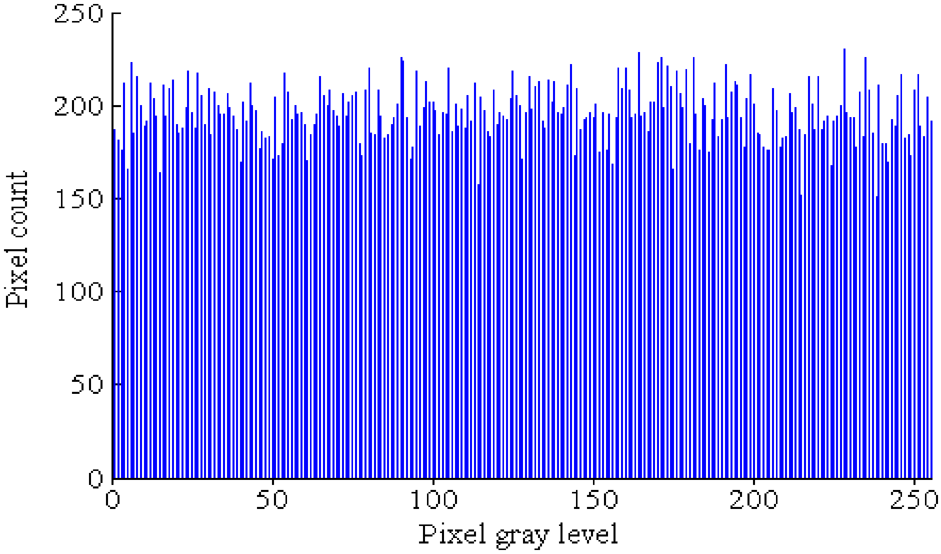


(a)

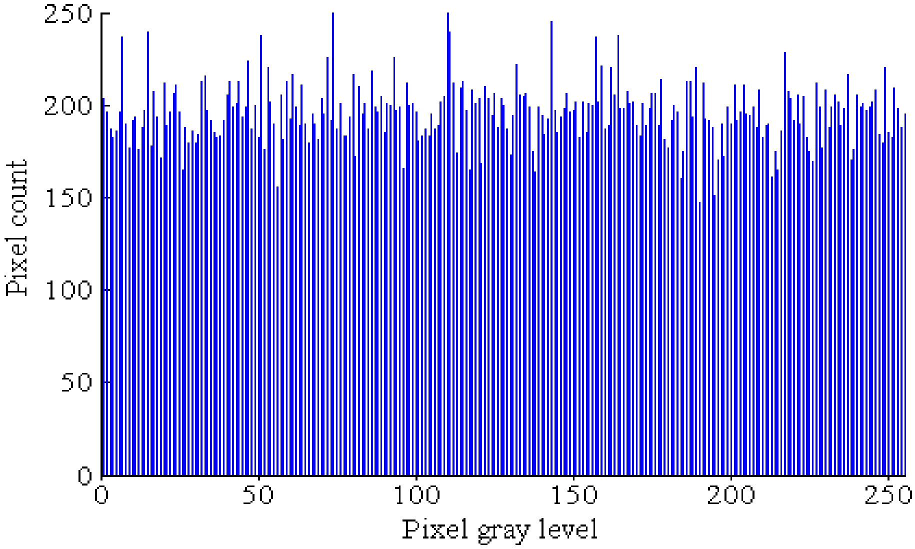


(b)

Figure 4.8. Image ‘Lena’ and Histogram. (a) Original Image and (b) histogram of the image.



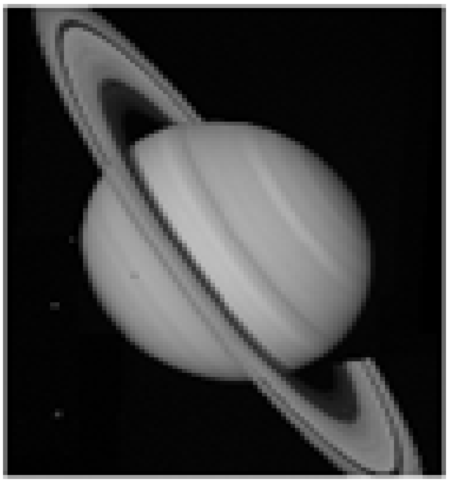
(a)



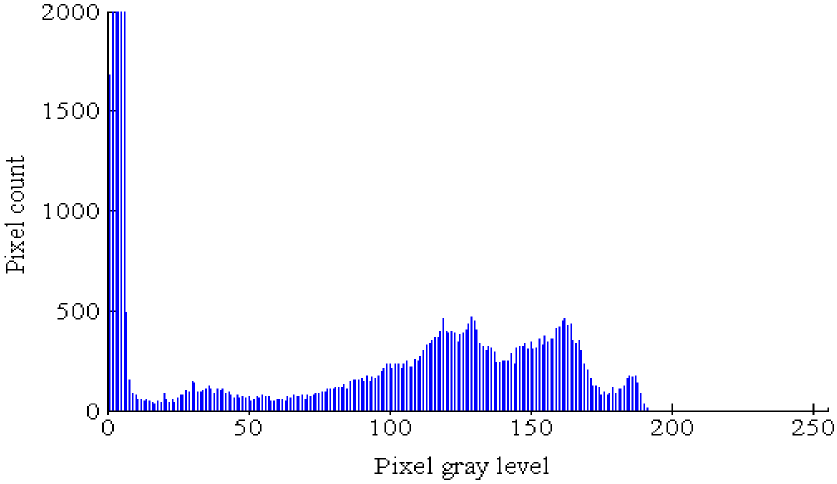
(b)

Figure 4.9. Histograms of cipher images of ‘Lena’. (a) Histogram of MASK cipher image of ‘Lena’ and (b) Histogram of AES cipher image of ‘Lena’.

Figure 4.10 shows original gray scale image ‘Saturn’ and its image histogram. Figure 4.11 shows the corresponding histograms of MASK cipher image and AES cipher image.

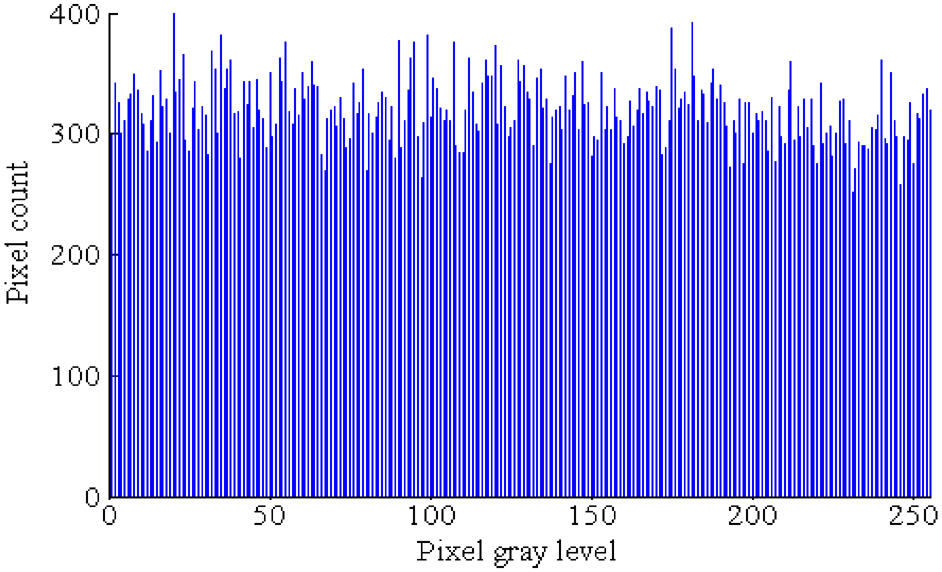


(a)

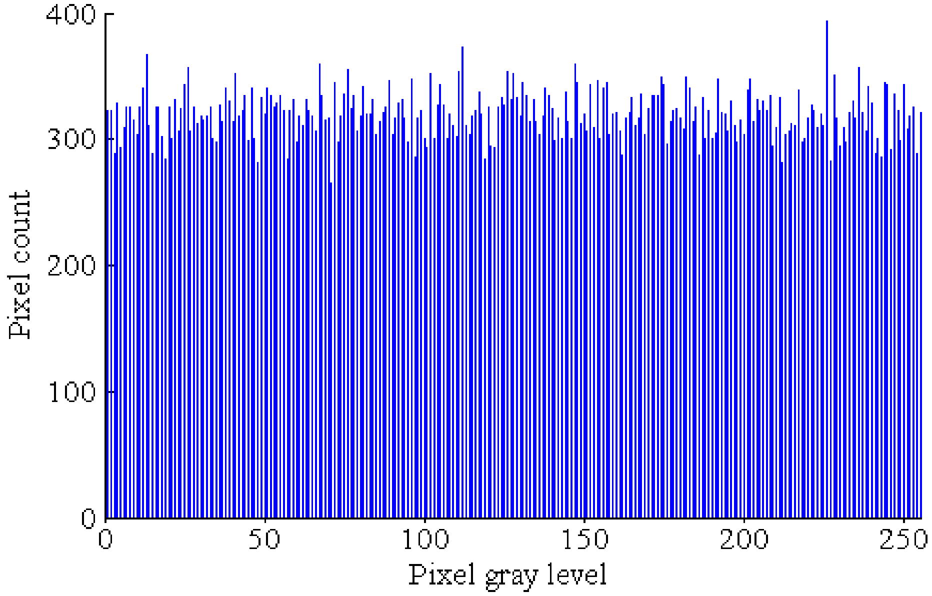


(b)

Figure 4.10. Image ‘Saturn’ and Histogram. (a) Original Image and (b) histogram of the image.



(a)



(b)

Figure 4.11. Histograms of cipher images of ‘Saturn’. (a) Histogram of MASK cipher image of ‘Saturn’ and (b) Histogram of AES cipher image of ‘Saturn’.

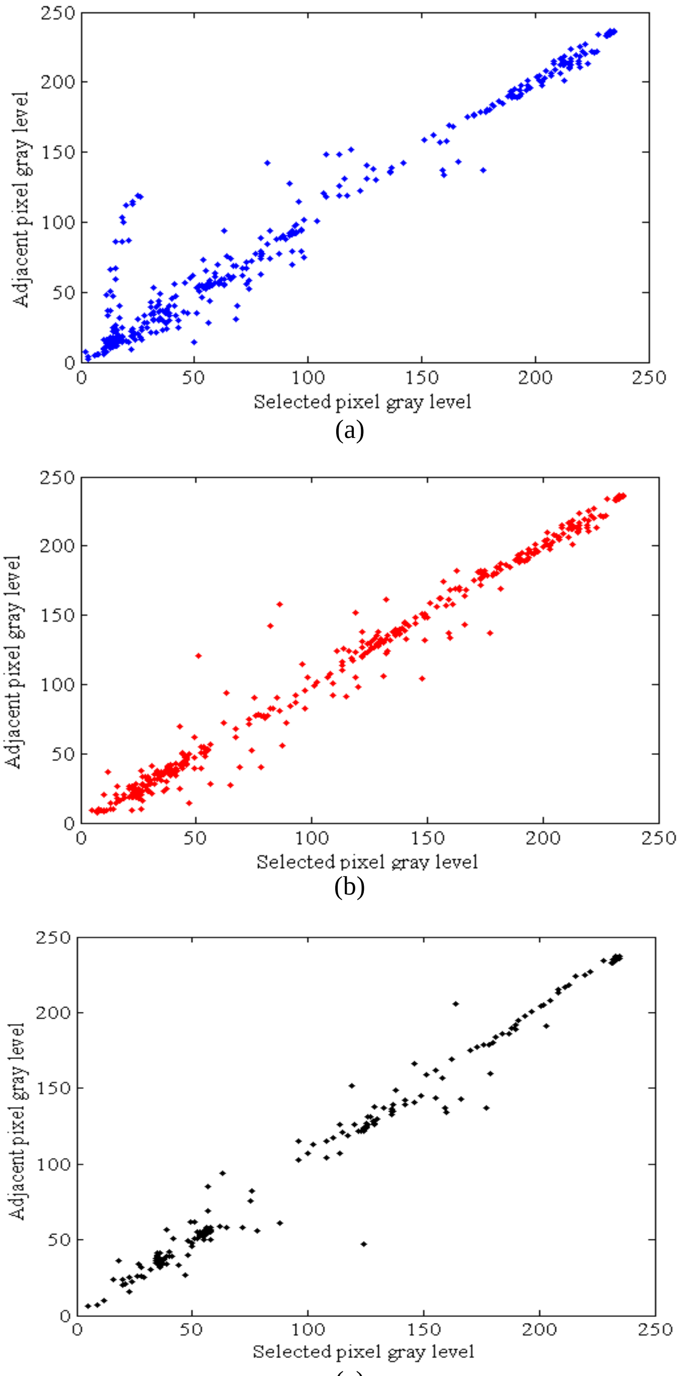
It is clear from the above histograms of the encrypted images by MASK and AES, that the distribution of gray scale values is uniform, and significantly different from the respective histograms of the original images. In the original image some gray-scale values in the range 0 to 255 do not exist but every gray-scale values in the range 0 to 255 exist and are uniformly distributed in the encrypted image. So, the encrypted image does not provide any clue to employ statistical attack on MASK encryption procedure. This gives MASK encryption high security against statistical attacks.

4.1.2.2 Adjacent Pixel Correlation Analysis

Correlation is a measure of the relationship between two variables. If the two variables are the two neighboring pixels in an image, then there is a very close correlation between them. This is called adjacent pixel correlation in an image. The correlation coefficient $C_{r,}$ is computed using the equation (4.1) given in [63].

$$C_\mathrm{r} = \frac{N\sum_{j=1}^{N}(\mathrm{X}_j * \mathrm{Y}_j) - \sum_{j=1}^{N}\mathrm{X}_j * \sum_{j=1}^{N}\mathrm{Y}_j}{\sqrt{[N\sum_{j=1}^{N}\mathrm{X}_j^2 - (\sum_{j=1}^{N}\mathrm{X}_j)^2\,][N\sum_{j=1}^{N}\mathrm{Y}_j^2 - \left(\sum_{j=1}^{N}\mathrm{Y}_j)^2\right]}}\,, \qquad (4.1)$$

where X and Y are gray values of two adjacent pixels in the original and encrypted image and N is the total number of adjacent pixels selected from the image. The correlation coefficient, $C_{r,}$ has been computed using direct MATLAB command. The adjacent pixel correlation plots are obtained by using 512 randomly selected pairs of adjacent pixel gray scale values of two standard images and the corresponding cipher images generated by MASK and AES. Figures 4.12 to 4.17 show adjacent pixel correlation plots of images ‘Onion’ and ‘Lena’, adjacent pixel correlation plots of corresponding MASK cipher images and AES cipher images along horizontal, vertical and diagonal directions.

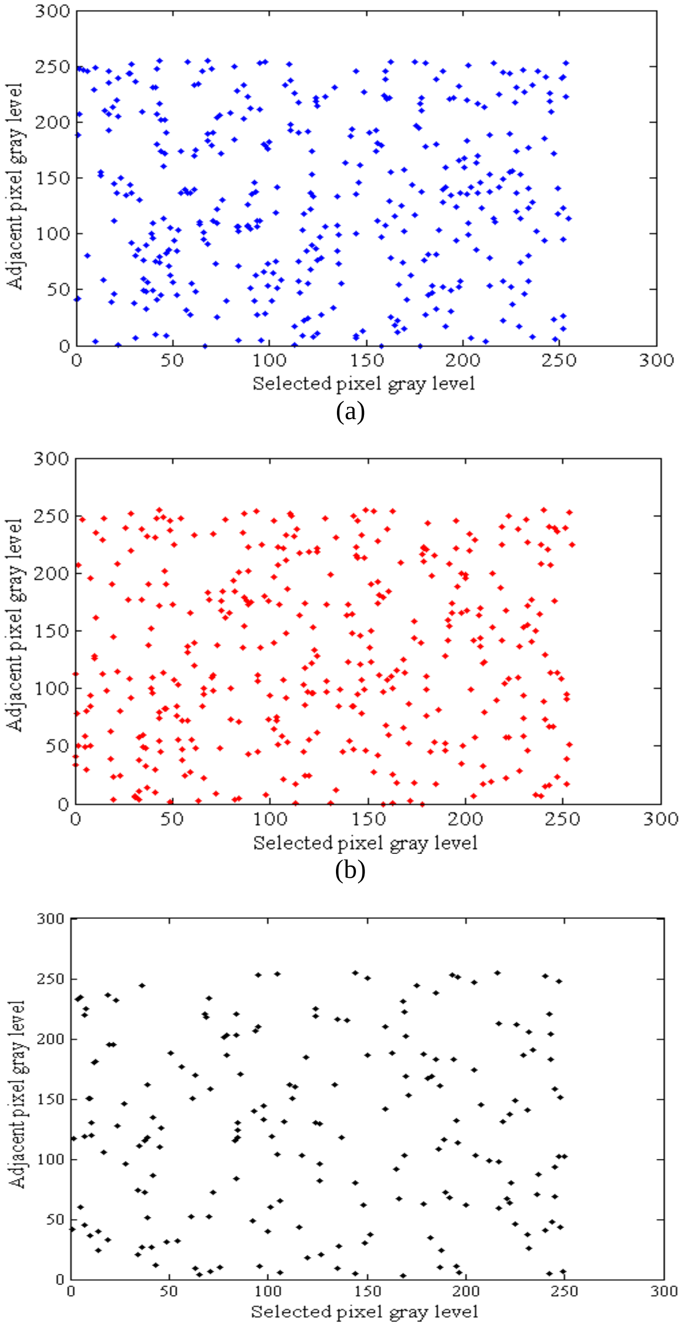


(a)

(b)

(c)

Figure 4.12. Adjacent pixel correlation plots of image ‘Onion’. (a) Horizontal direction, (b) Vertical direction and (c) Diagonal direction.

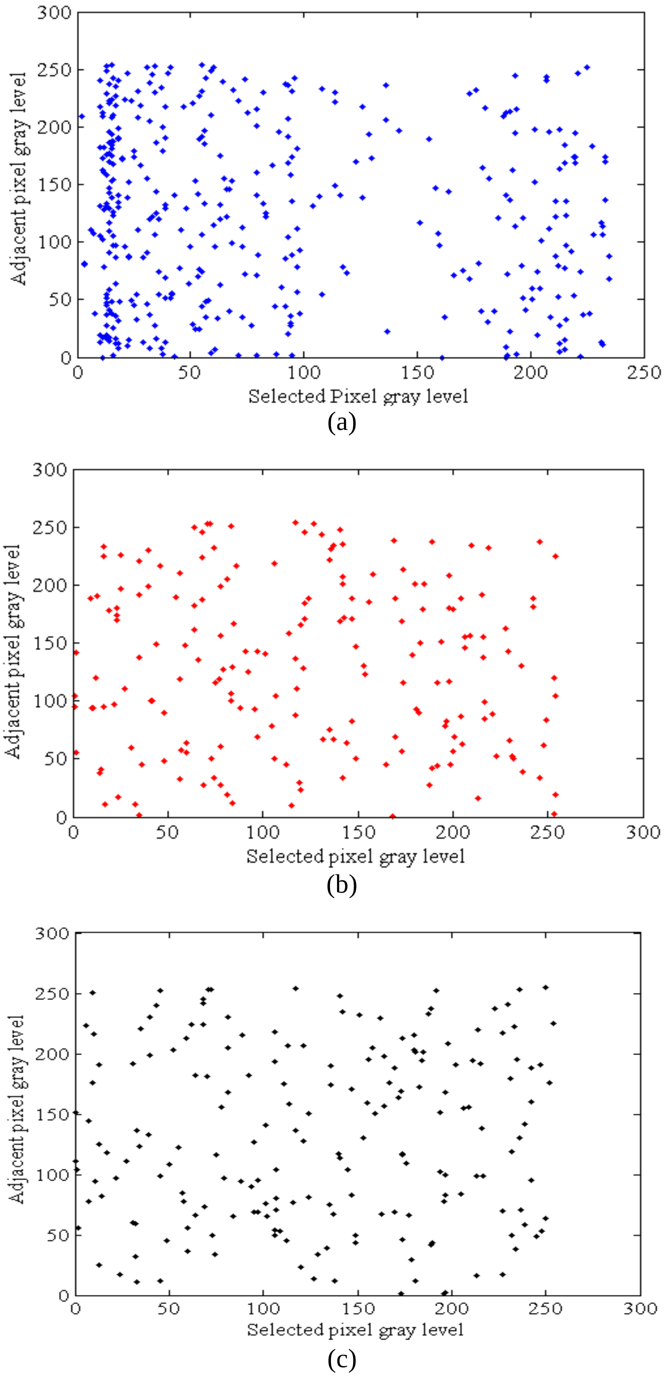


(a)

(b)

(c)

Figure 4.13. Adjacent pixel correlation plots of MASK cipher image of ‘Onion’. (a) Horizontal direction, (b) Vertical direction and (c) Diagonal direction.

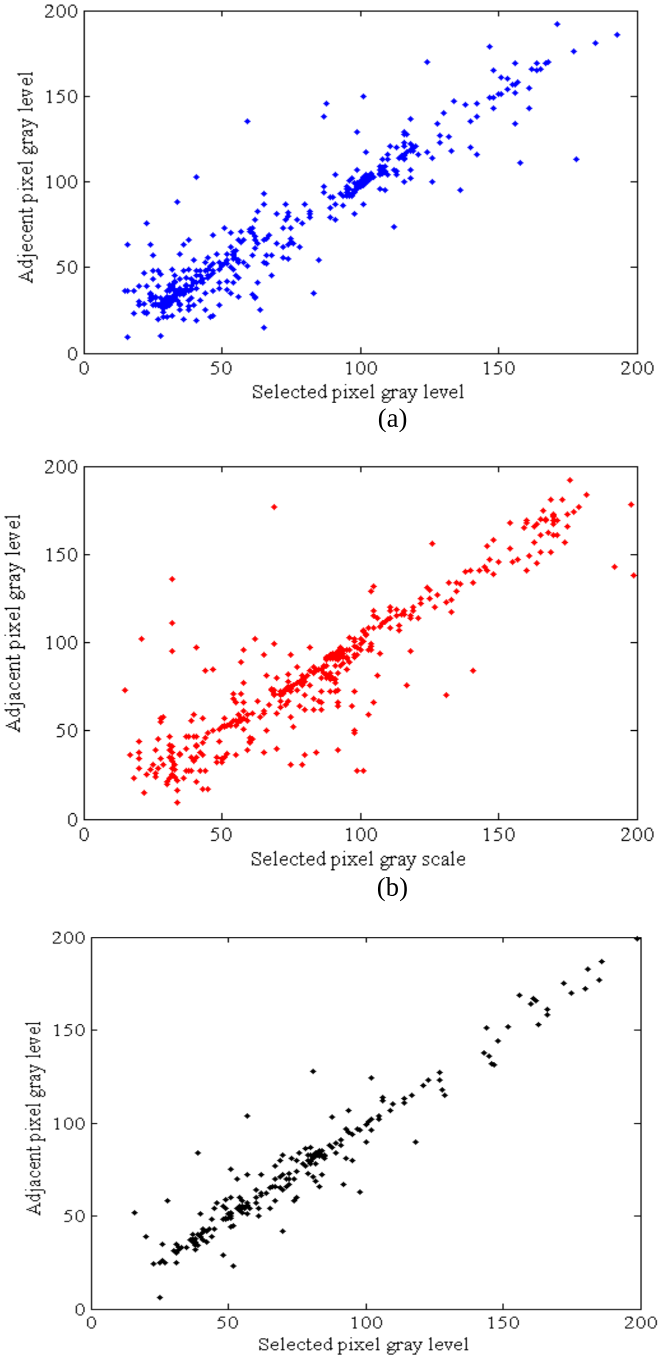


(a)

(b)

(c)

Figure 4.14. Adjacent pixel correlation plots of AES cipher image of ‘Onion’.
(a) Horizontal direction, (b) Vertical direction and (c) Diagonal direction.

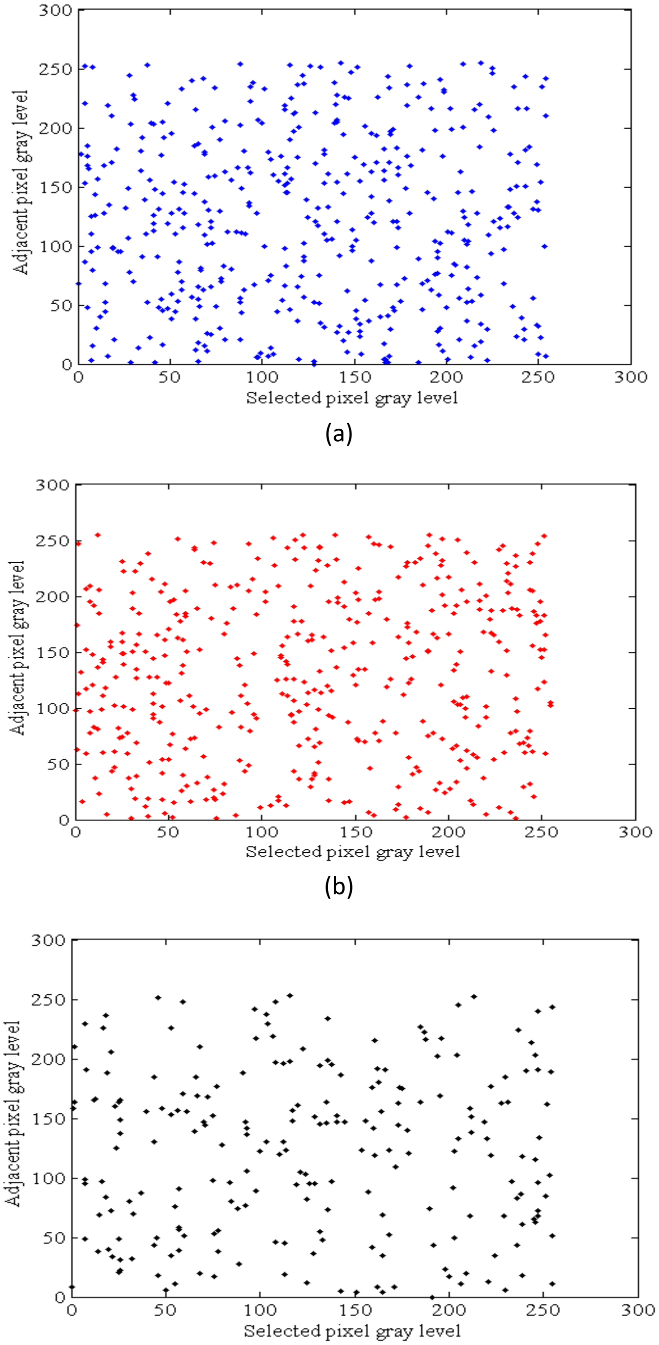


(a)

(b)

(c)

Figure 4.15. Adjacent pixel correlation plots of image ‘Lena’. (a) Horizontal direction, (b) Vertical direction and (c) Diagonal direction.

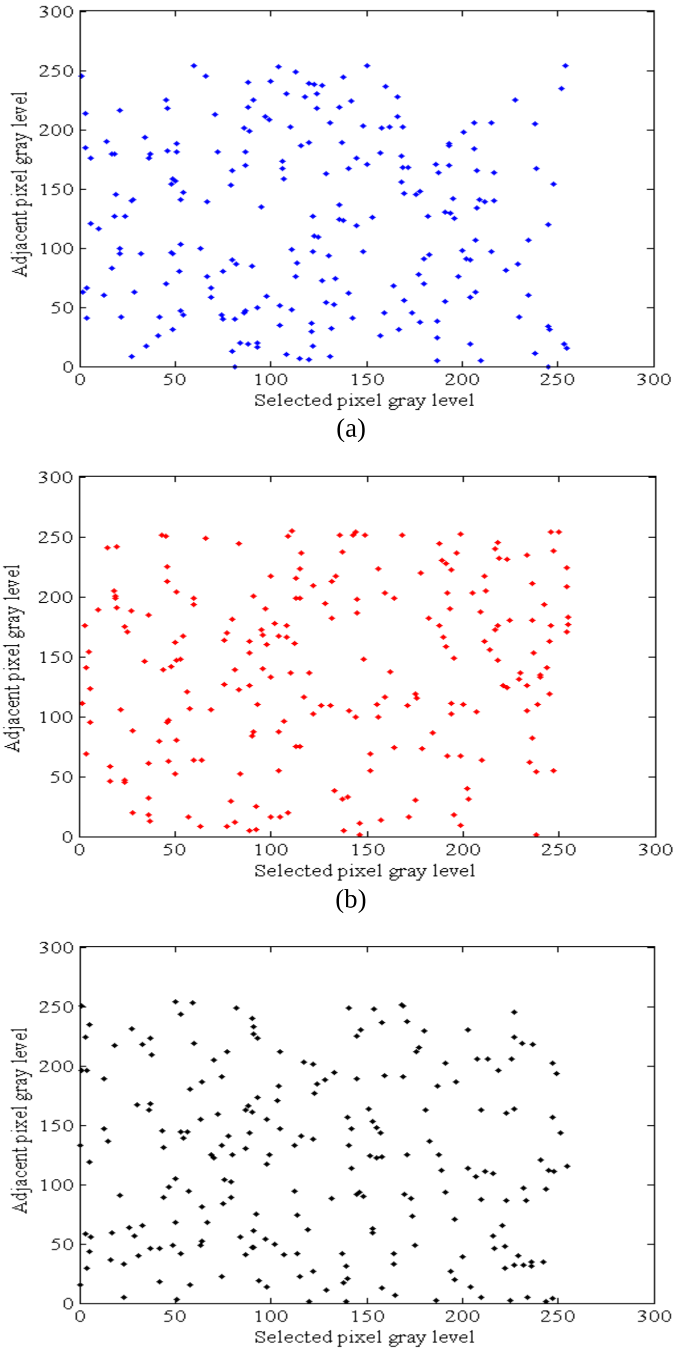


(a)

(b)

(c)

Figure 4.16. Adjacent pixel correlation plots of MASK cipher image of 'Lena'. (a) Horizontal direction, (b) Vertical direction and (c) Diagonal direction.



(a)

(b)

(c)

Figure 4.17. Adjacent pixel correlation plots of AES cipher image of ‘Lena’. (a) Horizontal direction, (b) Vertical direction and (c) Diagonal direction.

In the above plots, gray scale values of selected pixels are applied as *x* axis input data and adjacent pixel gray scale values are applied as *y* axis input data of the plotting procedure. If the correlation is very high, then the plot appears like a concentration of points along the diagonal of the *x*-*y* plane. However, if the correlation is very weak, the plot represents scattered points throughout the *x*-*y* plane. In the case of an encrypted image, the adjacent pixel correlation should be very small if the encryption process is successful in hiding the details of the original image [65]**.** It can be seen that in the correlation plots of the encrypted images by MASK and AES the correlation is very low in all the three directions as the plot represents scattered points throughout the *x*-*y* plane. The correlation between the adjacent pixels in the original image is strong as there is concentration of points along the diagonal of the *x*-*y* plane. Comparison of correlation coefficients in selected images and their cipher images obtained from AES and MASK encryption is given in Table 4.1.

Table 4.1. Adjacent Pixel Correlation Coefficients of Original Images and Cipher Images Generated by MASK and AES.

Image Name	Direction	Correlation Coefficient C_r		
		Plain Image	MASK cipher image	AES cipher image
Onion	Horizontal	0.973	0.053	0.072
	Vertical	0.989	0.032	0.007
	Diagonal	0.988	0.016	0.043
Lena	Horizontal	0.943	0.006	0.004
	Vertical	0.897	0.101	0.026
	Diagonal	0.966	0.012	0.054
Saturn	Horizontal	0.997	0.247	0.209
	Vertical	0.997	0.059	0.019
	Diagonal	0.999	0.014	0.106
Rice	Horizontal	0.918	0.020	0.046
	Vertical	0.890	0.079	0.032
	Diagonal	0.954	0.079	0.076

4.1.2.3 Mean Value Analysis

This test is intended to verify the distribution of mean pixel gray value in every vertical line of an image. This gives the average intensity of pixels along the horizontal direction in the image. In a plain image, the mean value will vary along the horizontal direction and appears as a signal with wide variations in the mean across the width of the image. Whereas in a well encrypted image the mean value along the horizontal should remain more or less consistent, indicating uniform gray level distribution along all vertical lines of the encrypted image. Mean value data has been collected from the encrypted images obtained from MASK and AES using different images. Figures 4.18 to 4.21 show the mean values obtained from the gray scale image ‘Lena’, ‘Cameraman’, ‘Galaxy’ and, ‘Saturn’ along with the mean values of the encrypted images obtained from MASK and AES encryption schemes. In all the mean value plots of encrypted images, the mean value across the image remains nearly consistent. Also it can be seen that the mean values of the encrypted images generated by MASK and AES are close to each other.

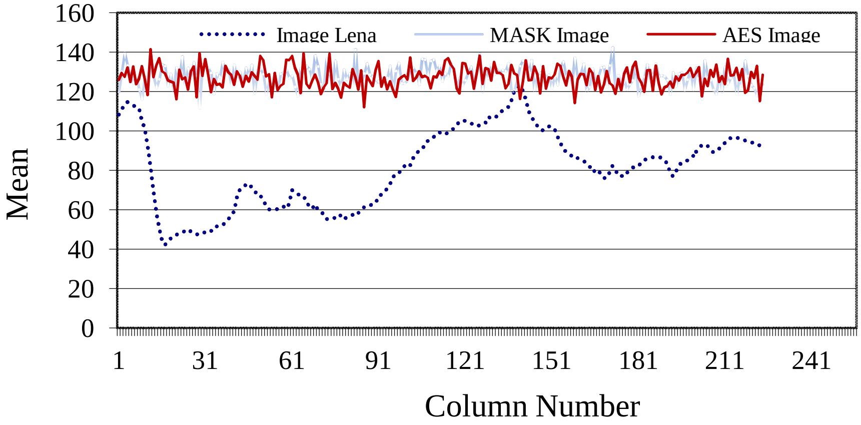


Figure 4.18. Mean value plots of image ‘Lena’ and encryptions.

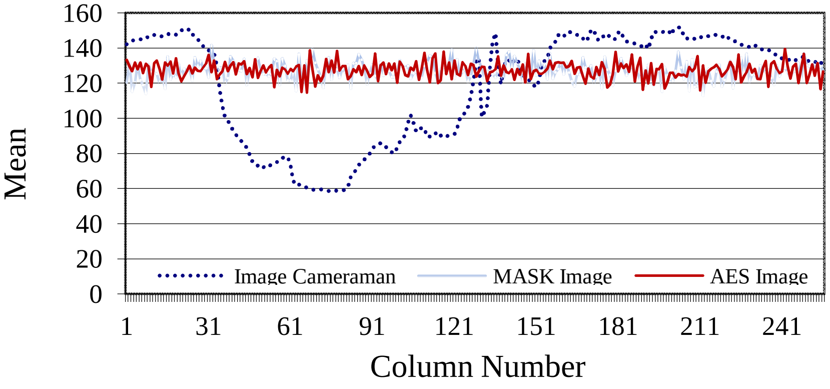


Figure 4.19. Mean value plots of image ‘Cameraman’ and encryptions.

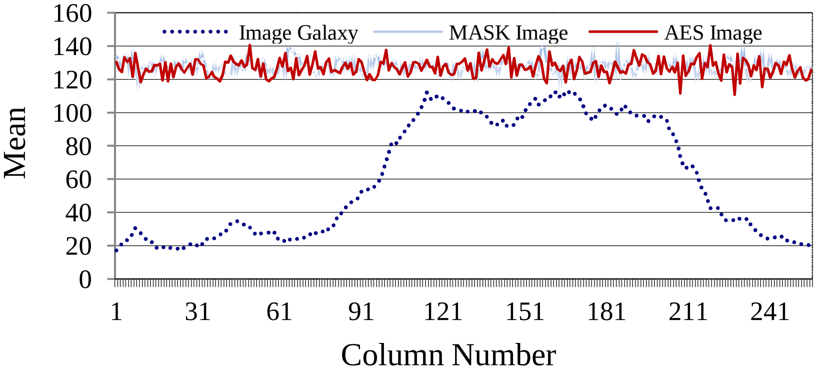


Figure 4.20. Mean value plots of image ‘Galaxy’ and encryptions.

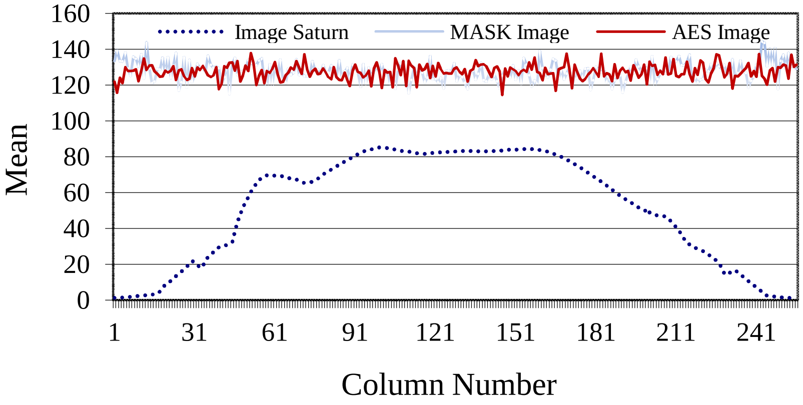
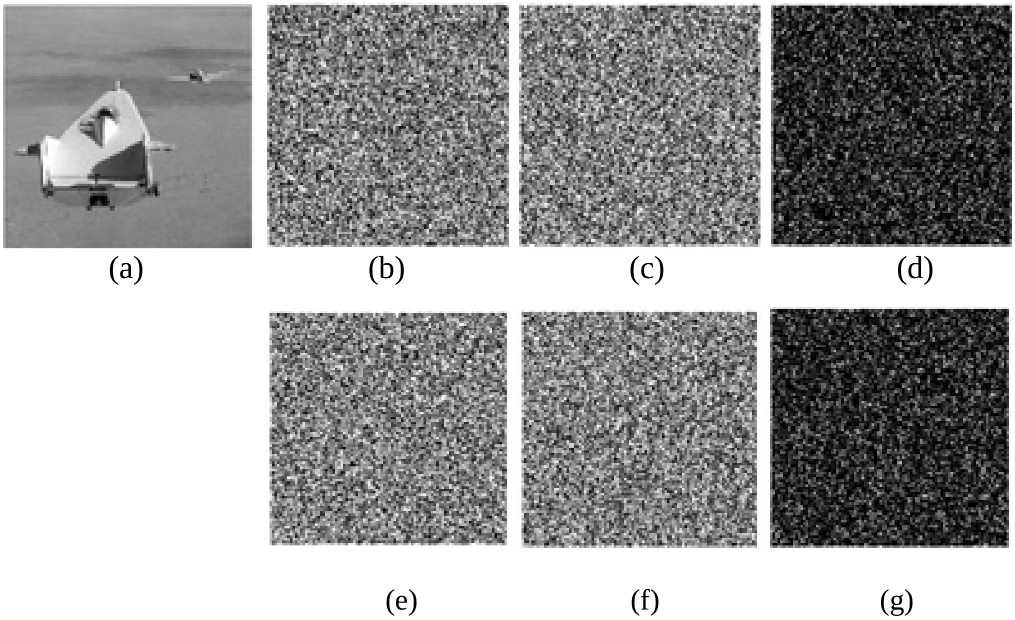


Figure 4.21. Mean value plots of image ‘Saturn’ and encryptions.

4.1.3 Key Sensitivity Analysis

This test reveals how much change is produced in the encrypted output of a cipher due to a small change (1 bit) in the secret key. To determine this we first run the encryption program, MASK, with an input image I and secret key K_{e1} and obtain the cipher image, C_1. Then we run the program with the same input image and another secret key $K_{e2,}$ that is different by one bit (closest key) with respect to K_{e1} and obtain the cipher image, C_2. Using the two encrypted images we obtain the difference image, $|(C_1\text{-}C_2)|$. Figure 4.22 shows the encryption on an original image ‘Lifting body’ using closest keys by MASK and AES and the difference images.


(a) (b) (c) (d)

(e) (f) (g)

Figure 4.22. Encryptions using closest keys by MASK and AES.
(a) Original image ‘Lifting body’, (b) MASK cipher image using key $K_{e1,}$ (c) MASK cipher image using key K_{e2} , (d) MASK difference image, (e) AES cipher image using key $K_{e1,}$ (f) AES cipher image using key K_{e2} and (g) AES difference image.

The percentage intensity difference I_d, using the encrypted images generated by MASK and AES encryptions is given by

$$\% \, I_d = \frac{\{\sum_{i=1}^{M}\sum_{j=1}^{N} I_1(i,j) - \sum_{i=1}^{M}\sum_{j=1}^{N} I_2(i,j)\}}{\sum_{i=1}^{M}\sum_{j=1}^{N} I_1(i,j)} \times 100, \tag{4.2}$$

where M and N are the dimensions of encrypted image in pixels and I_1 and I_2 are the pixel gray scale values in encrypted images C_1 and C_2 respectively. It has been observed that the image encrypted by the first key has 33.63% difference from the image encrypted by the second key in terms of pixel gray scale values in the case of MASK although there is only one bit difference in the two keys. Whereas the image encrypted using the first key has 33.72% difference from the image encrypted by the second key in terms of pixel gray scale values in the case of AES.

4.1.4 Measurement of Encryption Quality

The encryption quality, expressed in terms of total changes in pixel gray values between the original image and the encrypted image, is given by [65]

$$Q = \frac{\sum_{L=0}^{255} |H_L(F) - H_L(F')|}{256}, \tag{4.3}$$

where L is the pixel gray level, $H_L(F)$ the number of pixels having gray level L in the original image and $H_L(F')$ the number of pixels having gray level L in the encrypted image. The encryption quality values of MASK and AES have been evaluated, using images of different sizes and textures, for all the ciphering rounds. Tables 4.2 to 4.4 show comparison of encryption quality measured in AES and MASK using three different images of sizes 128 × 128 pixels, 256 × 256 pixels and 512 × 512 pixels respectively. Table 4.5 shows comparison of encryption quality measured in AES and MASK using same image having three different sizes (128 × 128, 256 × 256 and 512 × 512 pixels).

Table 4.2. Encryption Quality Measured in AES and MASK with Different Images having Dimension 128 × 128 Pixels.

Encryption Quality of AES and MASK using Different Images of Size 128 × 128 Pixels						
Ciphering rounds	Image name					
	Rice		Liftingbody		Cameraman	
	Algorithm type					
	AES	MASK	AES	MASK	AES	MASK
1	58.26	61.72	78.60	80.30	63.88	69.33
2	58.11	58.65	79.68	81.18	63.69	64.73
3	57.41	57.69	79.80	79.15	63.03	63.86
4	58.01	57.93	79.58	79.84	63.24	62.83
5	58.19	58.17	80.07	79.17	63.87	63.51
6	58.08	58.10	78.64	78.99	63.75	63.55
7	57.55	57.29	79.00	79.40	64.54	62.88
8	56.87	58.26	79.67	79.00	63.66	62.54
9	57.56	58.03	78.69	79.30	63.94	62.83
10	58.44	58.48	79.11	79.06	63.55	64.10
Average	57.848	58.432	79.284	79.539	63.715	64.016

Table 4.3. Encryption Quality Measured in AES and MASK with Different Images having 256 × 256 pixels.

Encryption Quality of AES and MASK using Different Images of Size 256 × 256 Pixels						
Ciphering rounds	Image name					
	Rice		Liftingbody		Cameraman	
	Algorithm type					
	AES	MASK	AES	MASK	AES	MASK
1	230.191	229.77	318.1	320.99	253.34	253.23
2	230.98	228.48	318.25	321.45	250.83	253.70
3	230.789	229.38	316.77	317.73	252.59	250.78
4	229.832	230.41	317.72	317.19	251.47	250.4
5	230.41	230.61	317.86	316.95	249.25	249.8
6	230.48	230.43	316.21	317.0	251.99	250.42
7	229.312	231.02	316.87	318.95	249.83	252.88
8	229.543	229.7	317.17	318.09	250.7	251.89
9	230.101	230.13	317.65	317.2	251.24	251.16
10	230.902	227.61	318	316.36	249.36	250.85
Average	230.254	229.754	317.46	318.191	251.06	251.511

Table 4.4. Encryption Quality Measured in AES and MASK with Different Images having Dimension 512 × 512 Pixels.

Encryption Quality of AES and MASK using Different Images of Size 512 × 512 Pixels						
Ciphering rounds	Image name					
	Rice		Liftingbody		Cameraman	
	Algorithm type					
	AES	MASK	AES	MASK	AES	MASK
1	761.08	761.95	1265.9	1268.7	847.77	852.19
2	755.46	759.26	1267.8	1269.3	844.46	851.11
3	759.75	758.65	1268.6	1262.5	842.67	842.47
4	755.55	755.32	1265.9	1269.5	843.79	846.25
5	759.59	759.91	1268.8	1269.3	848.70	844.54
6	755.51	751.97	1271.0	1266.6	843.15	845.30
7	754.05	757.56	1268.4	1265.3	842.79	845.37
8	760.15	759.04	1266.4	1268.6	846.91	845.87
9	756.35	753.66	1270.3	1269.0	843.04	846.18
10	757.06	757.77	1269.1	1266.2	844.80	846.26
Average	757.455	770.409	1268.22	1270.8	844.808	861.954

Table 4.5. Encryption Quality Measured in AES and MASK with Same Image having Different Dimensions.

Encryption Quality of AES and MASK using Same Image with Different Sizes						
Ciphering rounds	Image: Liftingbody					
	Size 128 × 128 Pixels		Size 256 × 256 Pixels		Size 512 × 512 Pixels	
	Algorithm type					
	AES	MASK	AES	MASK	AES	MASK
1	78.6	80.3	318.1	321.99	1265.9	1286.7
2	79.68	81.18	318.25	321.45	1267.8	1284.3
3	79.8	79.15	316.77	317.73	1268.6	1262.5
4	79.58	79.84	317.72	317.19	1265.9	1269.5
5	80.07	79.17	317.86	316.95	1268.8	1269.3
6	78.64	78.99	316.21	317.0	1271.0	1266.6
7	79	79.40	316.87	318.95	1268.4	1265.3
8	79.67	79	317.17	318.09	1266.4	1268.6
9	78.69	79.3	317.65	317.2	1270.3	1269.0
10	79.11	79.06	318	316.36	1269.1	1266.2
Average	79.284	79.539	317.46	318.291	1268.22	1270.8

From the results tabulated above, for different images of size 512 × 512 pixels, the average encryption quality of MASK is found to be 967.72 as against the encryption quality of AES which is only 956.82. Figure 4.23 shows the encryption quality averaged over all 10 rounds for three different images of size 512 × 512 pixels obtained from AES and MASK. Figure 4.24 shows the encryption quality averaged over all 10 rounds for the same image with 3 different sizes obtained from AES and MASK.

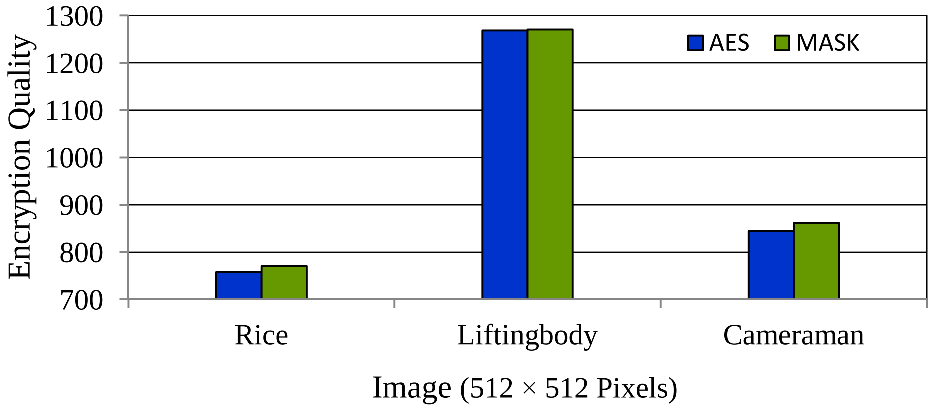


Figure 4.23. Encryption quality of AES and MASK with 3 different images of size 512 × 512 pixels.

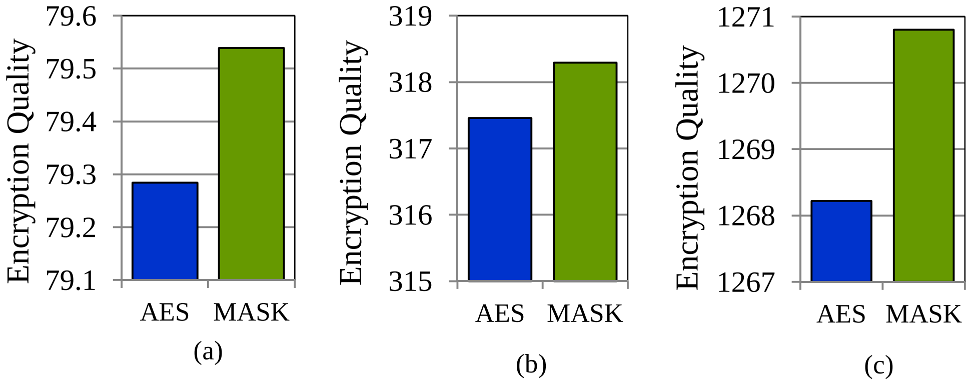


Figure 4.24. Encryption quality of AES and MASK using same image having three different sizes. (a) Size 128 × 128 pixels, (b) Size 256 × 256 pixels and (c) Size 512 × 512 pixels.

The average encryption quality, obtained from the Table 4.5, of AES and MASK using same image with different sizes, indicates that MASK has 556.21 as against 554.98 in the case of AES. These measurements show that the encryption quality of MASK is better than that of AES and the encryption quality increases with image size. This is because, a large size image contain more number of pixels. As the number of pixels increase, difference in number of pixels having same gray level increases giving a higher encryption quality value. The encryption quality with different encrypted images of same size shows different values because the image contents are different for these images even though the image sizes are same.

4.1.5 Measurement of Encryption Speed

Encryption speed of MASK algorithm has been measured in Bytes/second and compared with that of AES. The tests have been conducted using Matlab-7 in an Intel Core Duo CPU @ 2.00 GHz with Windows-XP operating system. In the first test, three separate images having sizes 128 × 128 pixels, 256 × 256 pixels and 512 × 512 pixels have been used to measure the encryption speed. In the second test, same image having three different sizes (128 × 128 pixels, 256 × 256 pixels and 512 × 512 pixels) have been used. The time taken for encryption for each round has been measured using Matlab commands. The encryption speed is then calculated by taking the ratio of the number of pixels (Bytes) in the image to the time taken for encryption. The encryption speed obtained using these images are given in Tables 4.6 to 4.9. The average encryption speed achieved by AES and MASK while encrypting different images of different sizes are respectively 1489.49 bytes/second and 11536.74 bytes/second. This shows MASK encryption is 7.75 times faster than AES encryption.

Table 4.6 Comparison of Encryption Speeds of AES and MASK with Different Images of Dimension 128 × 128 pixels.

Encryption Speed (Bytes/Sec.) of AES and MASK with Images of Size 128 × 128 Pixels						
Ciphering rounds	Image name					
	Rice		Liftingbody		Cameraman	
	Algorithm type					
	AE S	MASK	AES	MASK	AES	MASK
1	7656.07	80709.36	7839.23	74472.73	7953.40	70017.09
2	2540.16	15031.193	2608.92	15603.81	2608.92	6023.53
3	1325.57	8359.1837	1349.59	8904.35	1351.82	4641.36
4	815.94	5461.3333	827.06	5830.60	829.15	3624.78
5	553.70	3873.2861	558.99	4106.27	560.71	2879.44
6	400.59	2899.823	403.25	3068.16	404.74	2320.68
7	302.90	2256.7493	304.71	2377.94	306.41	1900.70
8	237.17	1806.3947	238.49	1896.30	239.29	1579.94
9	190.87	1481.3743	191.69	1548.58	192.35	1328.79
10	156.87	1235.5958	157.51	1290.08	158.04	1132.27
Average	1417.98	12311.429	1447.943	11909.88	1460.48	9544.86

Table 4.7 Comparison of Encryption Speeds of AES and MASK with Different Images of Dimension 256 × 256 pixels.

Encryption Speed (Bytes/Sec.) of AES and MASK with Images of Size 256 × 256 Pixels						
Ciphering rounds	Image name					
	Rice		Liftingbody		Cameraman	
	Algorithm type					
	AE S	MASK	AES	MASK	AES	MASK
1	7972.75	48188.24	8021.54	81920.00	8100.87	48907.46
2	2624.59	11497.54	2672.76	11872.46	2696.95	16718.37
3	1355.17	5985.02	1373.92	7728.30	1369.33	9077.01
4	829.25	4314.42	838.49	5363.01	834.22	5549.20
5	560.28	3013.15	565.75	3873.29	562.78	3830.27
6	403.10	2123.66	407.49	2953.40	404.67	2526.45
7	303.96	1587.98	306.85	2323.97	305.57	1817.42
8	237.83	1252.84	239.85	1863.94	238.77	1379.71
9	191.04	1014.33	192.84	1533.72	191.88	1088.64
10	156.88	830.83	158.36	1283.01	157.53	883.23
Average	1463.48	7980.80	1477.78	12071.51	1486.26	9177.77

Table 4.8 Comparison of Encryption Speeds of AES and MASK with Different Images of Dimension 512 × 512 Pixels.

Encryption Speed (Bytes/Sec.) of AES and MASK with Images of Size 512 × 512 Pixels						
Ciphering rounds	Image name					
	Rice		Liftingbody		Cameraman	
	Algorithm type					
	AE S	MASK	AES	MASK	AES	MASK
1	7975.17	81843.272	8128.50	82176.80	7891.15	82176.80
2	2635.67	20384.448	2740.95	21790.86	2745.83	21399.51
3	1346.75	10349.151	1394.98	11079.63	1390.24	10890.90
4	821.61	6415.6632	843.91	6842.70	845.00	6711.32
5	553.05	4408.004	569.05	4679.47	568.70	4594.18
6	398.15	3230.7616	408.87	3412.00	408.92	3353.08
7	300.27	2474.9245	308.77	2597.03	308.74	2558.75
8	234.70	1959.5156	241.16	2051.04	2406.32	2022.40
9	188.55	1591.0658	193.66	1661.77	194.00	1639.22
10	154.56	1319.1626	158.85	1374.50	159.07	1356.08
Average	1460.85	13397.597	1498.869	13766.58	1691.80	13670.22

Table 4.9 Comparison of Encryption Speeds of AES and MASK with Identical Images of Different Dimensions.

Encryption Speed (Bytes/Sec.) of AES and MASK for Same Image with Different Sizes								
Ciphering rounds	Image: Liftingbody							
	Size 128 × 128		Size 256 × 256		Size 512 × 512		Size 1024 × 1024	
	Algorithm type							
	AE S	MASK	AES	MASK	AES	MASK	AES	MASK
1	7839.23	74472.73	8021.54	81920.00	8128.50	82176.80	8186.24	81984.05
2	2608.92	15603.81	2672.76	11872.46	2740.95	21790.86	2697.30	22024.28
3	1349.59	8904.35	1373.92	7728.30	1394.98	11079.63	1375.18	11096.04
4	827.06	5830.60	838.49	5363.01	843.91	6842.70	837.26	6823.56
5	558.99	4106.27	565.75	3873.29	569.05	4679.47	565.12	4651.24
6	403.25	3068.16	407.49	2953.40	408.87	3412.00	407.06	3385.45
7	304.71	2377.94	306.85	2323.97	308.77	2597.03	307.87	2585.18
8	238.49	1896.30	239.85	1863.94	241.16	2051.04	240.79	2035.95
9	191.69	1548.58	192.84	1533.72	193.66	1661.77	193.60	1649.40
10	157.51	1290.08	158.36	1283.01	158.85	1374.50	159.00	1363.50
Average	1447.943	11909.88	1477.784	12071.51	1498.869	13766.58	1496.94	13759.87

The average encryption speeds of AES and MASK for encrypting the same image with three different sizes are 1474.87 bytes/second and 12582.66 bytes/second. This shows that MASK is 8.53 times faster than AES. Figure 4.25 shows the plot of average encryption speed of AES and MASK with three different images of size 256 × 256 pixels in different diffusion rounds.

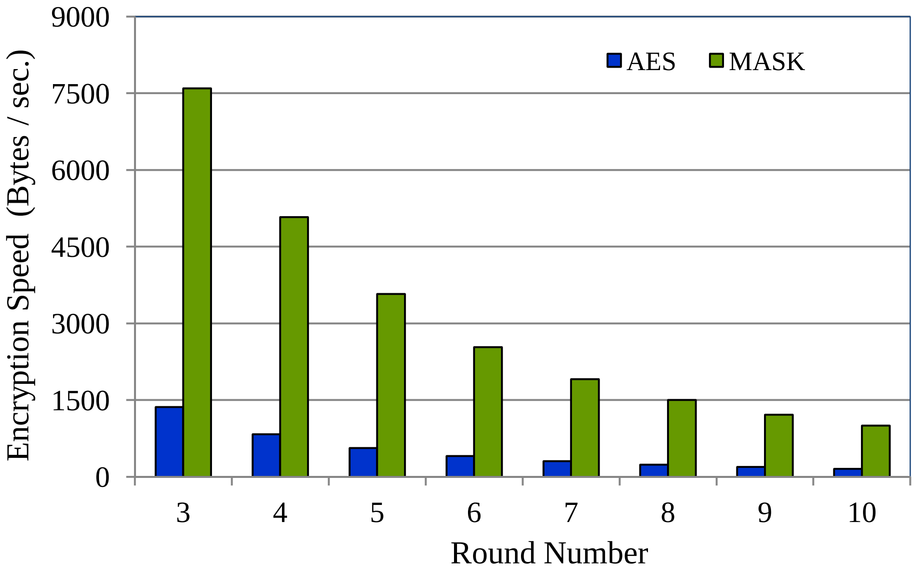


Figure 4.25. Average encryption speed of AES and MASK with 3 different images of size 256 × 256 pixels in different diffusion rounds.

The Figure indicates that the encryption speed decreases with increasing number of rounds, as expected, both in AES and MASK encryption. However, the performance of MASK is superior to that of AES. Figure 4.26 shows the plot of encryption speed of AES and MASK for an image of size 512 × 512 pixels for different diffusion rounds. This Figure also indicates that the performance of MASK is superior to that of AES. Figure 4.27 shows encryption speed of AES and MASK averaged over 10 diffusion rounds for three different images of size 256 × 256 pixels.

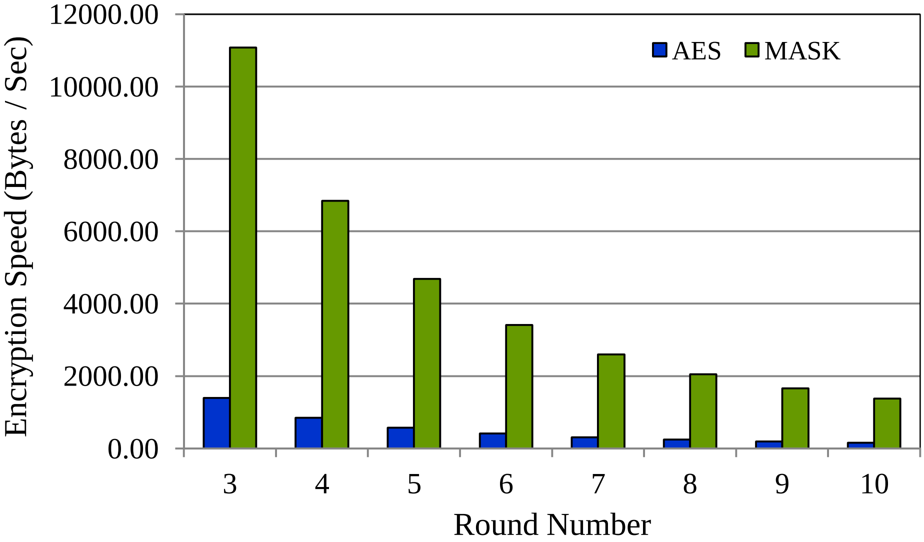


Figure 4.26. Encryption speed of AES and MASK with an image of size 512 × 512 pixels in different diffusion rounds.

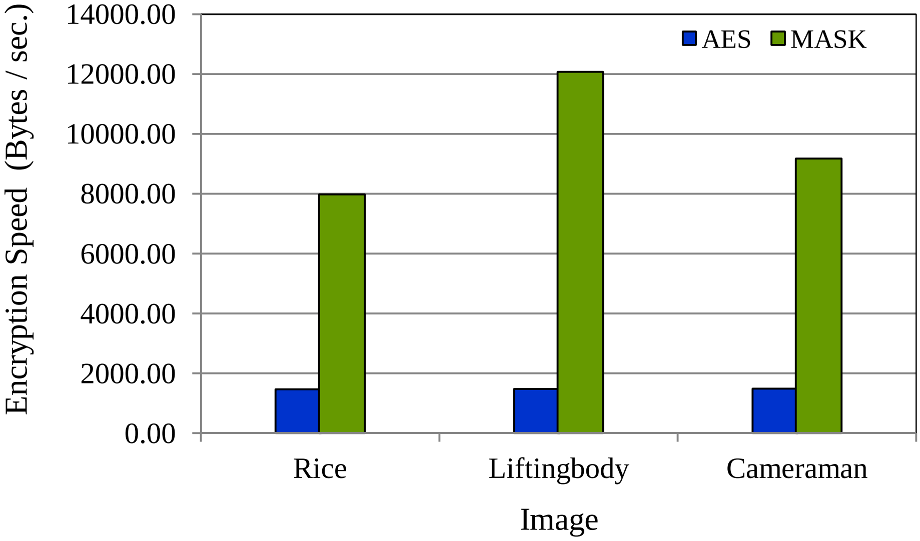


Figure 4.27. Encryption speed of AES and MASK averaged over 10 diffusion rounds for 3 images of size 256 × 256 pixels.

Figure 4.28 shows encryption speed of AES and MASK averaged over 10 diffusion rounds for the same image ‘Liftingbody’ having sizes 128 × 128 pixels, 256 × 256 pixels, 512× 512 pixels and 1024 × 1024 pixels.

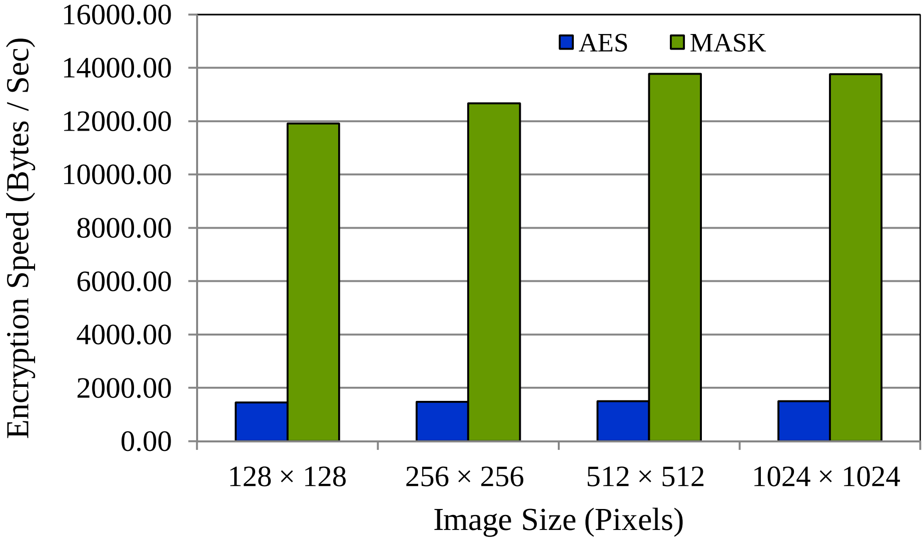


Figure 4.28. Encryption speed of AES and MASK averaged over 10 diffusion rounds for same image of different sizes.

In the case of AES, the encryption speed measured is consistent for different images of same size. But in the case of MASK, the encryption speed measured shows variation with different images of the same size. It has been observed that for the three images ‘Rice’, ‘Liftingbody’ and ‘Cameraman’ having same size 256 × 256 pixels, the average encryption speeds achieved by MASK are respectively 7980.80 bytes per second, 12071.51 bytes per second and 9177.77 bytes per second. This is because of the fact that AES encryption does not have data dependant operations in the diffusion rounds and MASK incorporates data dependant operations in its diffusion rounds. In the diffusion rounds of MASK, right half of data block is rotated number of times equal to a value calculated from the left half data block and vice-versa. Therefore, even though the image size is same, the encryption time varies with different image texture as the data in an image depends on the texture of the image. This is true for all encryption schemes incorporating data dependant operations. It may be noted that data dependant operations introduced in diffusion round operations of a cipher enhances the security of the cipher.

4.2 Summary of Results

The summary of observations from the test results and the analysis carried out on MASK and AES using images are given below:

1) Encrypted images of MASK do not reveal any texture of original image.
2) Histograms of encrypted images of MASK exhibit uniform distribution of pixel gray levels over the entire range. This indicates effectiveness of MASK encryption.
3) Adjacent pixel correlation in the encrypted images of MASK is very low. This shows that the pixels in the MASK encrypted images are statistically independent.
4) Mean value plots of encrypted images of MASK show that the mean value of pixels across the encrypted image is uniform compared to that of the original image. This also shows MASK encryption is effective.
5) Key sensitivity analysis of encrypted image of MASK indicates that one bit change in secret key brings 33% change in the encrypted image.
6) The encryption speed measurement shows that MASK encryption is eight times faster than AES. Thus MASK is efficient in converting plaintext data and images into ciphertext data and cipher images.
7) The average encryption quality is more in MASK compared to AES. Encryption quality of MASK is 967.72 and that of AES is 956.82 for an image of size 512× 512 pixels.

Chapter 5

Security Analysis on MASK

In this chapter a basic security analysis of the Matrix Array Symmetric Key [MASK] encryption, is offered. Security attacks such as statistical attack, ciphertext only attack, known plaintext attack, chosen plaintext attack, non-linear attack and linear attack are considered. Statistical data using images and plaintext are obtained and presented. Results obtained from AES are also shown for comparison.

5.1 Introduction

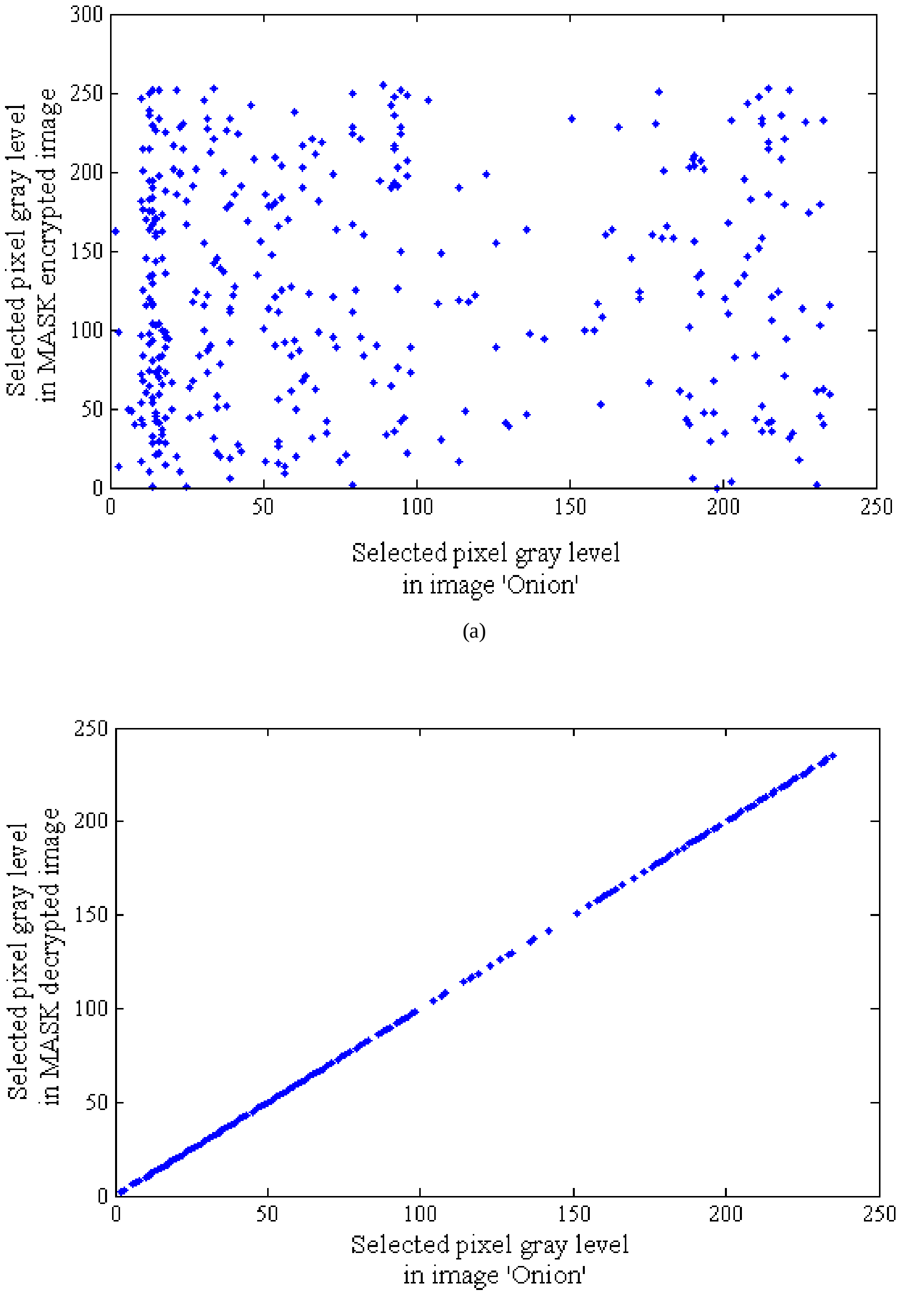
Security analysis is a very important aspect of any encryption scheme. The cipher should be capable of resisting all known security attacks. After the design and development of a cipher the designer should establish that the cipher is capable of dealing with various security attacks. In the following section security analysis of MASK encryption algorithm is presented.

5.2 Attacks on Cipher

The common attacks on symmetric cipher are 1) Statistical attack 2) Ciphertext-only attack (Brute force attack) 3) Known plaintext attack 4) Chosen-plaintext attack 5) Linear attack and 6) Differential attack. The following sub-sections discuss how the cipher MASK, is capable of resisting these common attacks.

5.2.1 Statistical Attacks

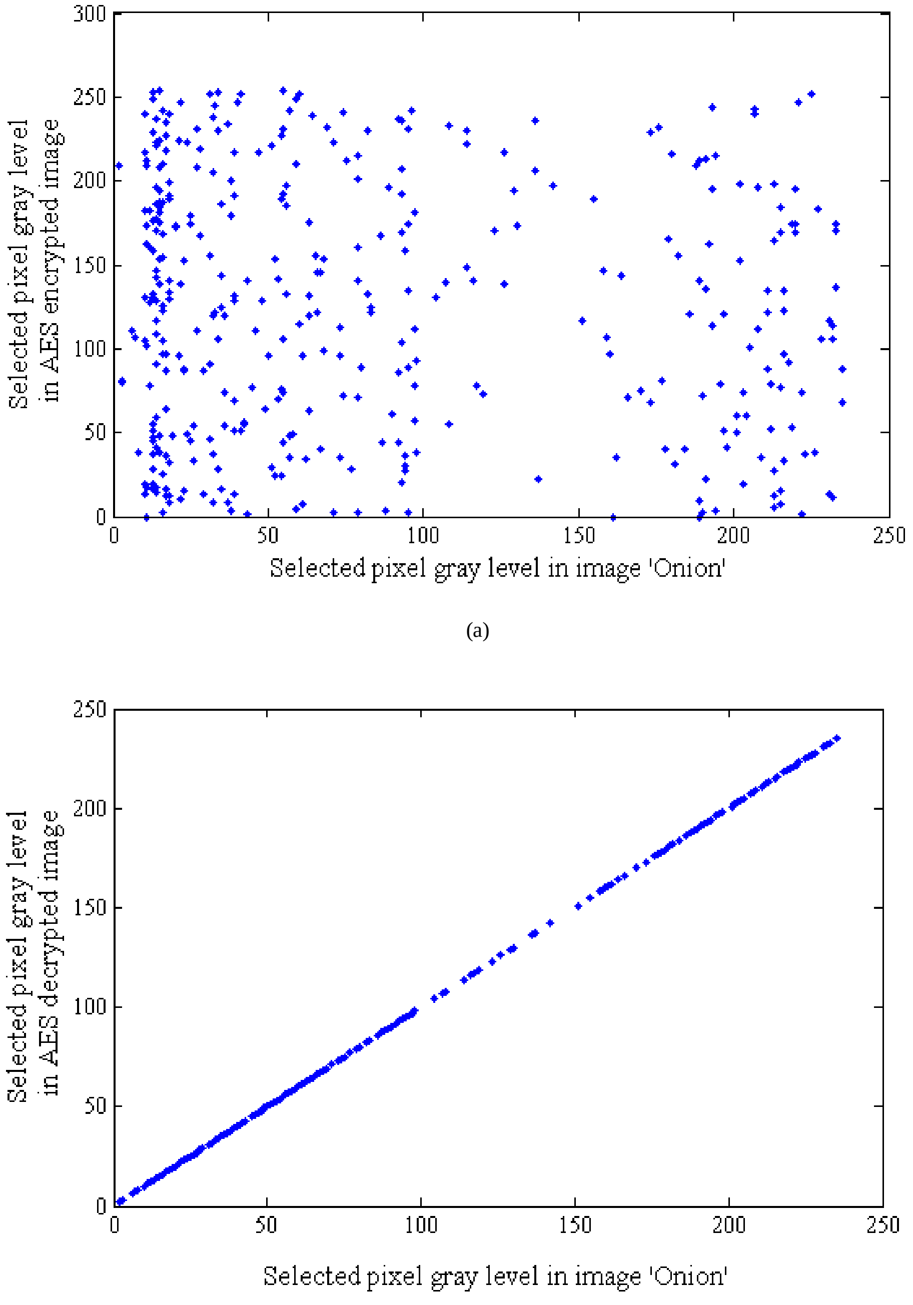
The most basic requirement of a good block cipher is that the input plaintext and ciphertext generated should be statistically independent, if not attacks against the cipher are most probable. A cross correlation analysis using images can indicate the degree of statistical independence between plaintext and ciphertext generated by a cipher. In this test, cross correlation between selected vertical segments of the input image and corresponding vertical segments in the encrypted and decrypted images of MASK and AES have been obtained. Figures 5.1 to 5.4 show cross correlation plots of images ‘Onion’ and ‘Lena’ with MASK and AES encrypted and decrypted images. Table 5.1 shows cross correlation coefficients from MASK and AES encryptions and decryptions with different images.



(a)

(b)

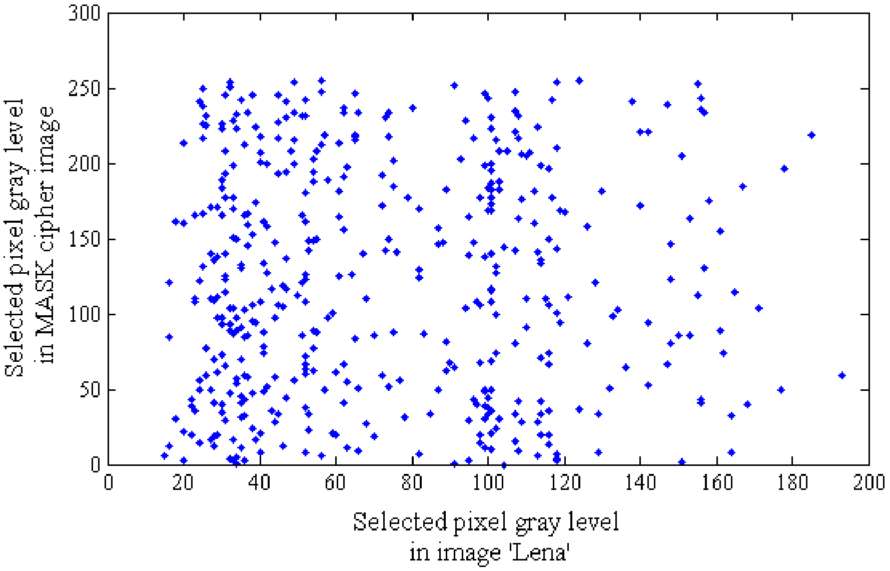
Figure 5.1. Cross correlation between original image ‘Onion’ and MASK encrypted and decrypted images. (a) Cross correlation between original image and MASK cipher image and (b) Cross correlation between original image and MASK decrypted image.



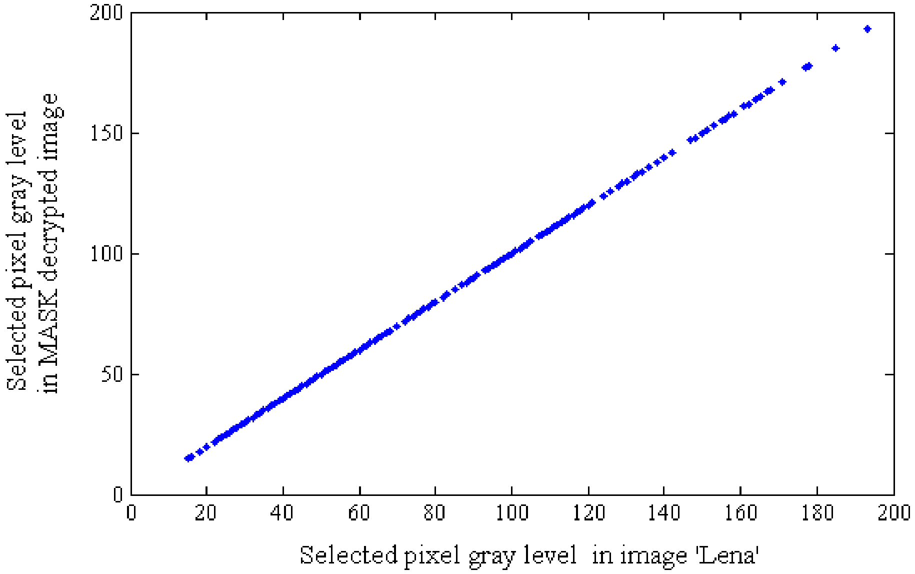
(a)

(b)

Figure 5.2. Cross correlation between original image ‘Onion’ and AES encrypted and decrypted images. (a) Cross correlation between original image and AES encrypted image and (b) Cross correlation between original image and AES decrypted image.

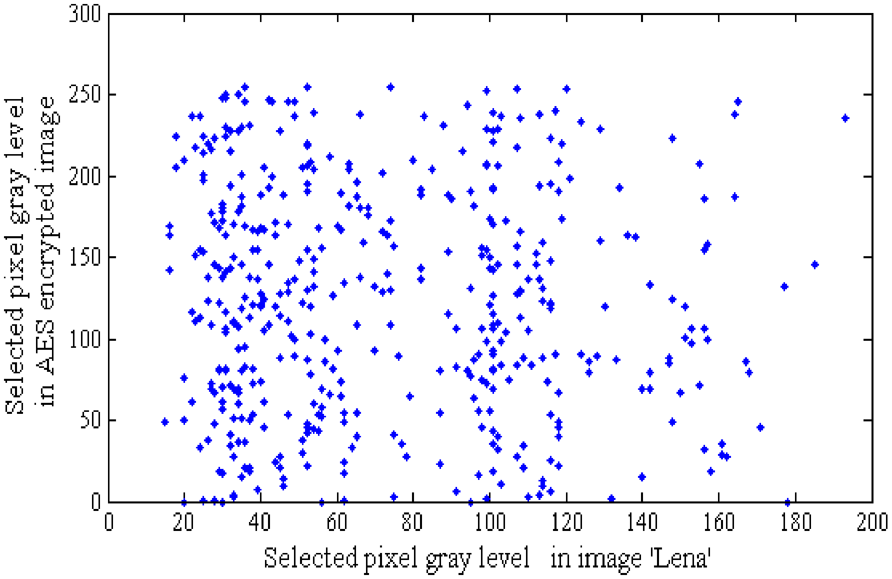


(a)

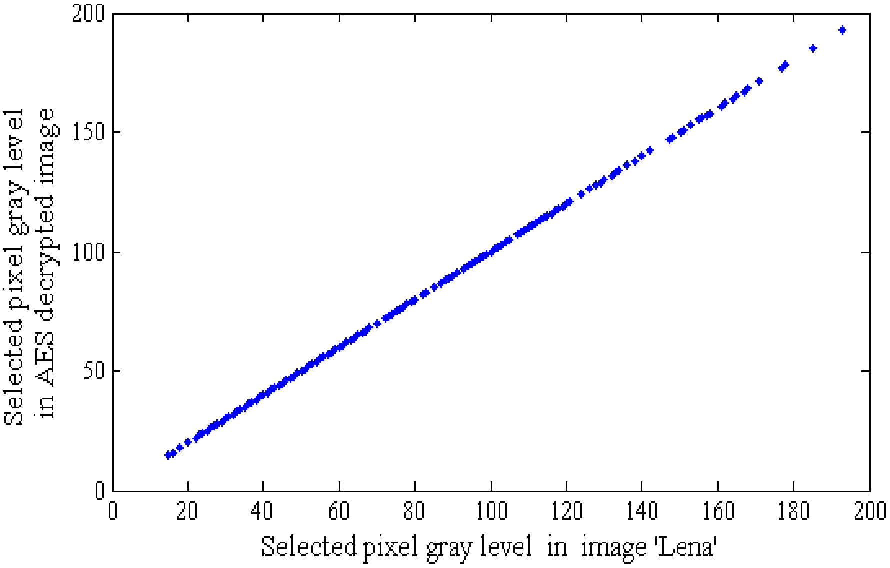


(b)

Figure 5.3. Cross correlation between original image ‘Lena’ and MASK encrypted and decrypted images. (a) Cross correlation between original image and MASK encrypted image and (b) Cross correlation between original image and MASK decrypted image.



(a)



(b)

Figure 5.4. Cross correlation between original image ‘Lena’ and AES encrypted and decrypted images. (a) Cross correlation between original image and AES encrypted image and (b) Cross correlation between original image and AES decrypted image.

Table 5.1. Cross Correlation Coefficients of MASK and AES Encryptions and Decryptions with Images.

Image	Cross correlation coefficient			
	MASK Encryption	AES Encryption	MASK Decryption	AES Decryption
Onion	0.010	0.079	1.0	1.0
Lena	0.032	0.054	1.0	1.0
Rice	0.013	0.025	1.0	1.0
Saturn	0.003	0.007	1.0	1.0

In the cross correlation plots between original images and ciphered images, the points spread over the entire x-y plane indicating weak correlation. Whereas, in the cross correlation plots between original images and decrypted images the points are along the diagonal in the x-y plane indicating 100% correlation. The cross correlation coefficients of MASK cipher images with the original images are lower than that of AES as indicated in the above table. The cross correlation coefficients of decrypted images with the original image are unity indicating perfect decryption. With these selected images, MASK encryption presents low cross correlation coefficients between original images and the ciphered images. On an average the cross correlation coefficient is 0.0145 for MASK encryptions and 0.04125 for AES encryptions. The weak correlation between original images and ciphered images indicates that MASK encryptions facilitate better statistical independence between the original images and the ciphered images than AES.

5.2.2 Ciphertext-only Attack

In this type of attack the opponent will have only ciphertext messages and proceed to decrypt the message by trying all key values. Although K is user specified, it can be generated as a random number or a set of random numbers each independent of the other. The key size is defined as 128 bits in the MASK encryption algorithm. This means that K has 2^{128} possible combinations. Even if an adversary employs a 1000 MIPS computer to find K, the computational load is 2^{128} / [(1000 × 10^6) × 60 × 60 × 24 × 365] or over 100 years! Like any other cipher having 128 bit secret key, MASK cipher is also free from ciphertext-only attack. Figure 5.5 shows MASK decrypted images using decryption keys that are closer to the key (K_e) used for encrypting the original image. The decryption keys K_1, K_2, K_3, K_d, K_4, K_5 and K_6 are keys with values that differ by one count from each other centered about K_d. Whereas, the key K_d is same as the key (K_e) used to encrypt the original image. It may be noted, here, that the decrypted images using closest keys do not reveal any intelligence contained in the original image. This shows that only with the correct secret key (that was used for encrypting the image) it is possible to decrypt the encrypted image and get back the original image. This shows that limited trials with few guessed keys will not lead the search towards the secret key. The adversary is left with no option other than trying almost all the key values to decrypt the message.

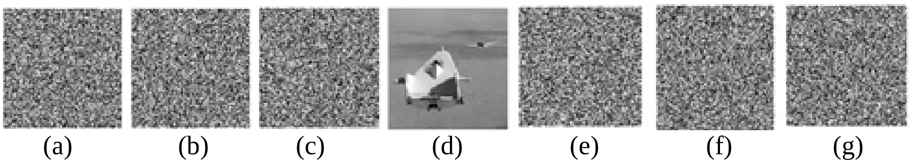


Figure 5.5. Decryption with closest keys. (a) decrypted image using key K_3 (b) decrypted image using key $K_{2,}$ (c) decrypted image using key $K_{1,}$ (d) decrypted image using key K_d, the right key, (e) decrypted image using key $K_{4,}$ (f) decrypted image using key K_5 and (g) decrypted image using key $K_{6.}$

5.2.3 Known Plaintext Attack

A more common assumption in modern cryptography is to assume that an attacker may have several pairs of messages and the corresponding ciphertexts from which to try to recover the key (a known plaintext attack).

The adversary may even be able to pick messages or ciphertexts to have encrypted or decrypted (chosen plaintext or chosen ciphertext attack, or, if both are allowed, a chosen text attack). By using previously known plaintext message and corresponding ciphertext message pairs an adversary can launch an attack on the cipher to deduce the secret key being used for secure communications. This is not possible when the encryption produce high diffusion of data and key changes into the entire cipher data block in the cipher. The complex key schedule and data based rotations in the diffusion rounds of MASK facilitate diffusion characteristic to the cipher. This feature of the algorithm thus protects MASK from known plaintext attacks. The following encryptions illustrate this feature inherent in the cipher.

MASK encryption using secret key ‘Godiseternalyes!’ for the message ‘You are not here’ produced ciphertext (in decimal format) as given below:
45 203 72 32 178 185 54 80 215 6 175 110 102 127 217 4.

For the message ‘You are not herd’ the encryption produced ciphertext (in decimal format), for the same key, as given below:
165 252 78 215 237 167 116 97 199 176 193 113 142 198 149 53.

AES encryption using the same secret key ‘Godiseternalyes!’ and message ‘You are not here’, has generated ciphertext as given below:
125 196 134 215 204 244 16 200 212 38 20 203 155 187 194 106.

For the message ‘You are not herd’ and using the same secret key ‘Godiseternalyes!’ AES encryption generated ciphertext as given below:
13 105 10 50 139 18 144 60 134 6 53 95 70 99 0 135.

From the above results, it can be seen that the encryptions produced for two seemingly same text messages, are distinctly different. The data diffusion effect produced in MASK and AES for the given plaintext data pair and same secret key are shown in Figures 5.6 and 5.7.

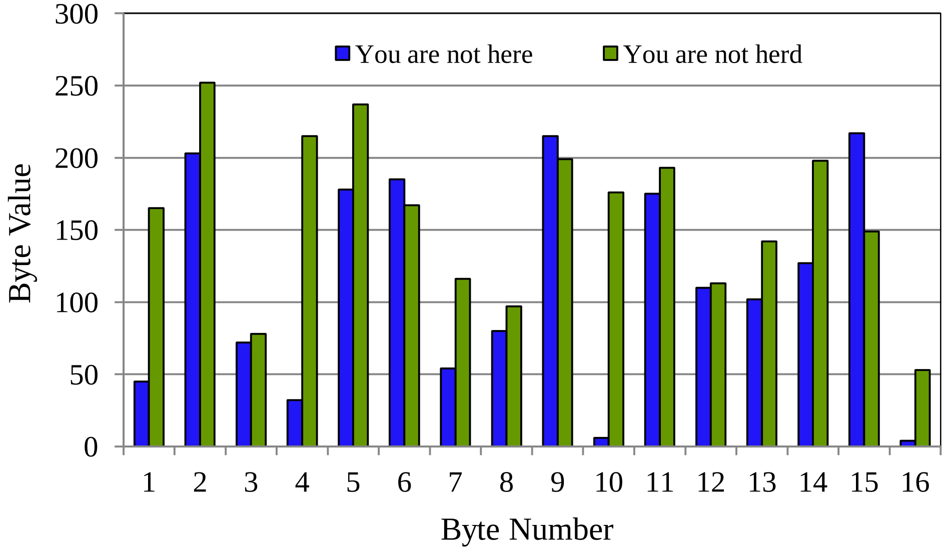


Figure 5.6. Encryptions of closest Data in MASK.

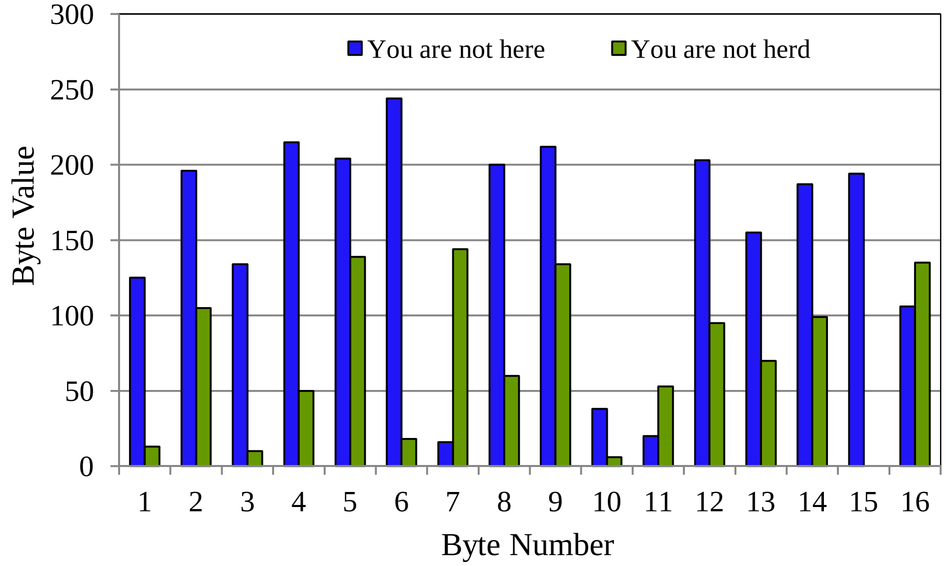


Figure 5.7. Encryptions of closest Data in AES.

From these data, average change occurred in ciphertext output block for one bit change in the input block has been computed and found to be 94.31 for MASK and 106.12 for AES. This shows AES produces higher data diffusion than MASK. As the data diffusion is more in AES known plaintext attacks on AES is harder.

5.2.4 Chosen Plaintext Attack

In this kind of attack the adversary, by using his own secret key and pairs of chosen plaintexts having certain relational characteristics tries to analyze the encryption algorithm and proceeds to obtain the relationship between the plaintext and the secret key for a given encryption session. Using this knowledge, the adversary can perform an attack on a cipher with minimum effort and try to deduce the secret key value used for the communication sessions. Even for this attempt to become successful, the cipher should have weak key diffusion characteristics. Consider a cipher that generates a ciphertext block C_1, from a plaintext block P, with a given secret key K_1. Then generates another ciphertext block C_2, with another secret key K_2, which is closer to the first key $K_{1.}$ Then, if C_1 and C_2 does not differ much, then an adversary will have to make only minimum number of trails to deduce the secret key. In MASK encryption it has been shown that even with a key value that differ by only one bit, the encryption of the plaintext block generates a totally different ciphertext block. The following encryptions illustrate this feature inherent in the cipher.

MASK encryption using secret key K_1=‘Godiseternalyes!’ for the message M=‘You are not here’ produced ciphertext as given below:
45 203 72 32 178 185 54 80 215 6 175 110 102 127 217 4.

MASK encryption using secret key K_2='Hodiseternalyes!' for the message M='You are not here' produced ciphertext (in decimal format) as given below:

161 30 22 85 170 153 105 175 203 213 168 146 137 244 71 121.

In the above two MASK encryptions, the plaintext is same and the secret key differ only by one bit. But the ciphertext generated by MASK are totally different (all bytes in the ciphertext have been changed).

AES encryption using the same secret key K_1='Godiseternalyes!' and message 'You are not here', has generated ciphertext (in decimal format) as given below:

125 196 134 215 204 244 16 200 212 38 20 203 155 187 194 106.

AES encryption using secret key K_2='Hodiseternalyes!' for the message M='You are not here' generated ciphertext (in decimal format) as given below:

84 140 95 248 246 9 47 204 76 232 55 161 150 183 101 33.

In the above two AES encryptions, the plaintext is same and the secret key differ only by one bit. But the ciphertext generated by AES are totally different (all bytes in the ciphertext have been changed here too).

From the above results, it can be seen that the encryptions of a given plaintext messages with two closer secret keys, are distinctly different. The key diffusion effect produced in MASK and AES for the given plaintext and closer secret keys are shown in Figures 5.6 and 5.7.

From these data, average change occurred in ciphertext output blocks for one bit change in the secret key has been computed and found to be 78.44 for MASK and 66.43 for AES. This shows that MASK produces higher key diffusion than AES. As the key diffusion is more in MASK, chosen plaintext attacks on MASK is harder. Figure 5.8 shows the key diffusion characteristics of MASK and Figure 5.9 shows the key diffusion characteristics of AES.

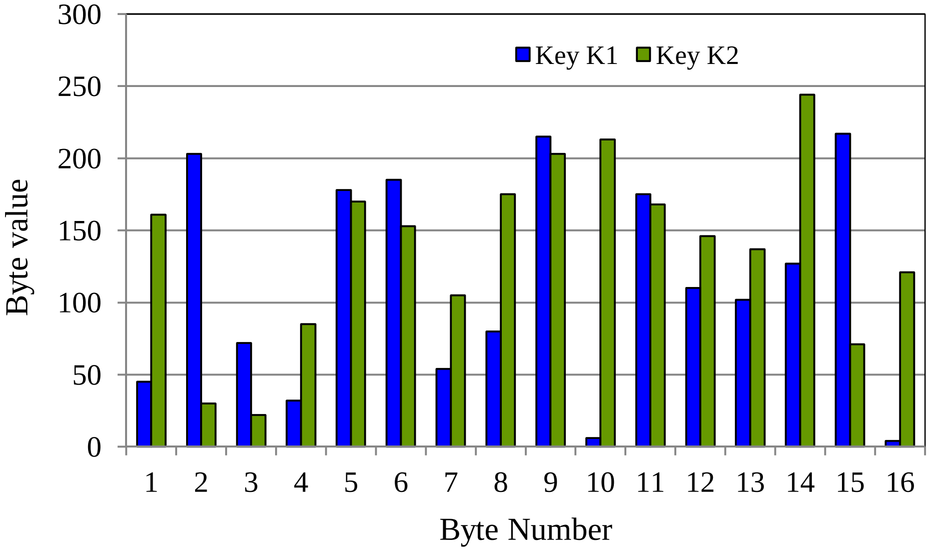


Figure 5.8. Encryptions with closest key in MASK.

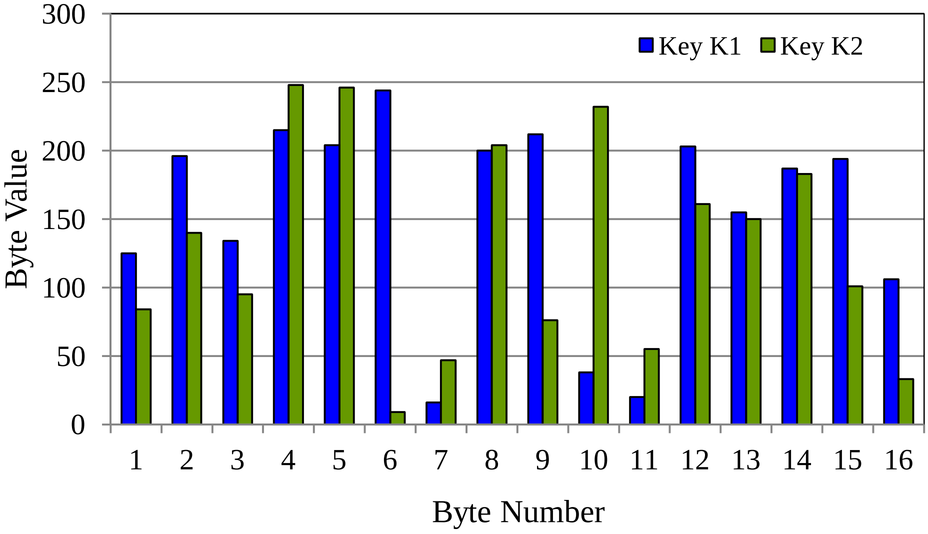


Figure 5.9. Encryptions with closest key in AES.

5.2.5 Linear Attack

If the relation between plaintext data / key and ciphertext data in a cipher is linear, then an adversary can launch a linear attack on the cipher with much ease. To resist the linear attack, the cipher should exhibit non-linearity in the encryption transformation. MASK encryption achieves input-output non-linearity by the use of data based rotations incorporated in the diffusion rounds and the key-output non-linearity by the complex key schedule procedure. Tests have been conducted to study the non-linearity in the encryption transformation in the case of MASK and AES. Ciphertext data block byte sum values obtained with linearly changing block data values applied at the input of MASK and AES ciphers have been obtained. The relationship between input data block byte sum values and output ciphertext data block values in MASK and AES are shown in Figure 5.10. From this above plot, it can be seen that the input-output relationship in both MASK and AES is non-linear.

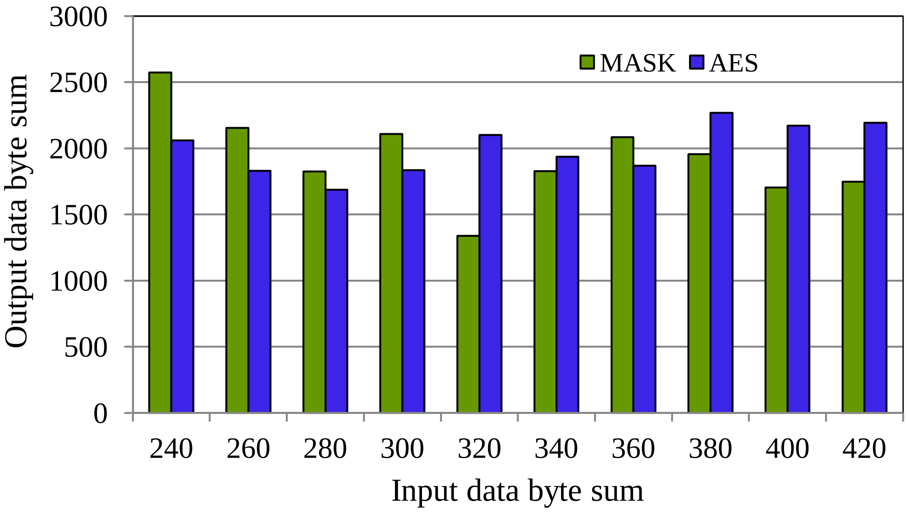


Figure 5.10. Input – output relationship in MASK and AES.

From this data difference between successive output data byte sum values are computed. Average values of these differences are calculated for MASK and AES. The average difference for MASK is 330 and for AES the value is 170. This shows that the input-output relationship is more non-linear in MASK compared to AES.

Similarly, ciphertext data block byte sum values with linearly changing key values have been obtained from MASK and AES ciphers and is plotted. Figure 5.11 shows the non-linear relationship between secret key value and output ciphertext data block. It can be seen that the key-output relationship in both MASK and AES is non-linear.

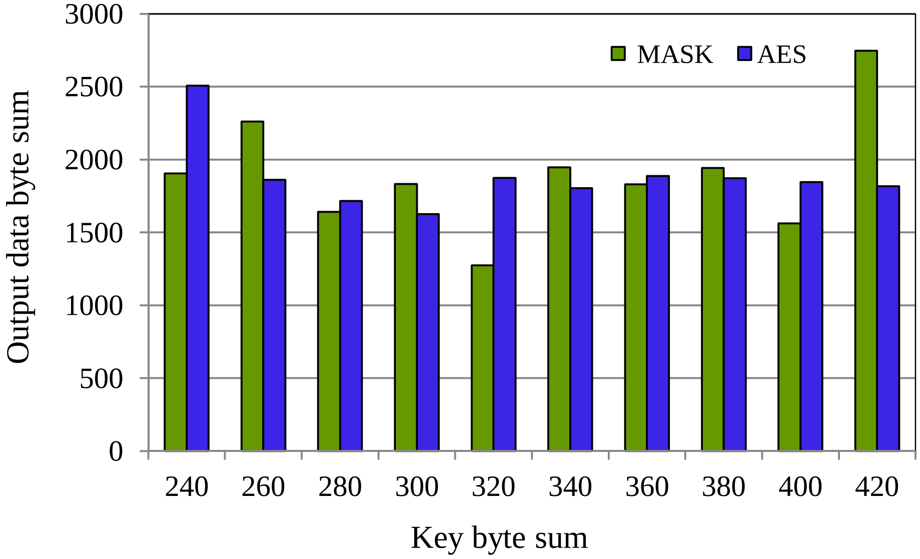


Figure 5.11. Key - Output relationship in MASK and AES.

The difference between successive output data byte sum values are computed in the case of MASK and AES. Average values of these differences are calculated for MASK and AES. The average difference for MASK is 466 and for AES the value is 155. This shows that the key-output relationship is more non-linear in MASK compared to AES. Since there is non-linearity in input-output relationship and key-output relationship, linear attack on MASK is hard to perform.

5.2.6 Differential Attack

If the differential data or key propagation through diffusion rounds in a block cipher is consistent, an adversary can launch a differential attack and determine the sub keys and the secret key without much difficulty. Therefore a good cipher should exhibit large variations in the differential data and key propagation through the diffusion rounds. Tests have been conducted to determine the differential data and key propagation through the diffusion rounds of MASK and AES. Figure 5.12 shows the differential data propagation observed in MASK and AES through their diffusion rounds.

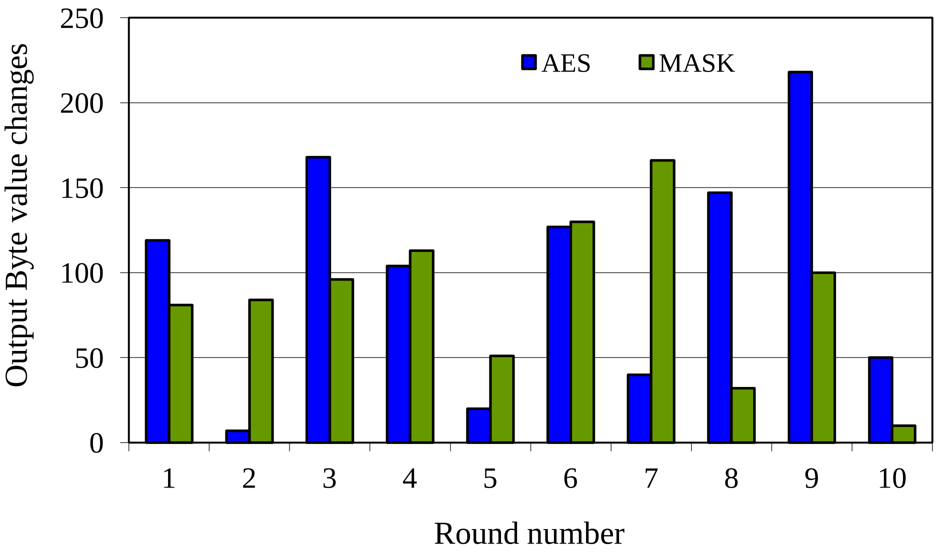


Figure 5.12. Differential Data Propagation in MASK and AES.

It can be seen from the plot that the differential data propagation in both MASK and AES are random through the diffusion rounds. Average of the differential data byte value propagated in diffusion rounds of MASK and AES have been calculated. The average change in data byte value over ten diffusion rounds obtained in MASK is 86.3 and in AES it is 100. This shows that the differential data propagation is better in AES compared to MASK. A higher average value of differential data propagation through diffusion rounds is able to provide better resistance to differential data attack on AES cipher. Figure 5.13 shows the differential key propagation observed in MASK and AES through their diffusion rounds.

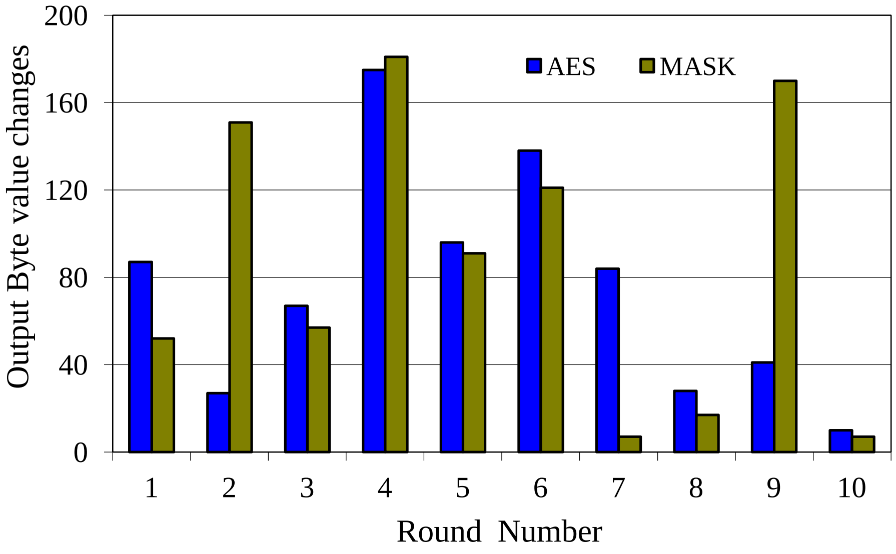


Figure 5.13. Differential key propagation in MASK and AES.

It can be seen from the plot that the differential key propagation in both MASK and AES are random through the diffusion rounds. The average of differential key byte value propagated in diffusion rounds of MASK and AES have been calculated. The average change in data byte value over ten diffusion rounds obtained in MASK is 85.4 and in AES it is 75.3. This shows that the differential key propagation is better in MASK compared to AES. A higher average value of differential key propagation through diffusion rounds is able to provide better resistance to differential key attack on MASK cipher.

5.3 Summary of Security Analysis

Various known attacks on symmetric ciphers have been examined. The features inherent in MASK encryption scheme that resist such attacks have been explained with selected data and images and performance comparison of MASK and AES have been made. The summary is given below:

1) Cross correlation analysis carried out on MASK and AES with selected images indicates that the cross correlation between original images and cipher images of MASK is small. On an average, the cross correlation coefficient observed in MASK encryption is 0.0145 and in AES encryption is 0.04125. The lower cross correlation between original images and ciphered images makes statistical attacks harder on MASK.

2) The key size used in MASK encryption is 128 bits. This gives a very large key space which demands very large time (many years!) for exhaustive key search. Like in any other encryption scheme using 128 bit secret key, ciphertext-only attack on MASK encryption is not practical.

3) MASK encryption exhibit good data diffusion and key diffusion properties. This makes known plaintext attacks and chosen plaintext attacks on MASK harder.

4) MASK encryption exhibits better non-linear relationship between input plaintext data blocks and output ciphertext data blocks as well as key and

output ciphertext data blocks than AES. This feature makes linear attacks on MASK encryption harder.

5) The differential data and key propagation through diffusion rounds in MASK are random. This makes differential attacks on MASK encryption harder.

Chapter 6

Conclusion and Scope for Further Work

An efficient cryptographic algorithm has been developed for text messages and images using matrix and array manipulations. In this chapter the conclusions and scope for further research work on the topic are indicated. The advantages of MASK over AES are highlighted. Other features that facilitate security of the cipher are also indicated in comparison with AES. The work that could be pursued in the future for improvement and to support diversity of applications is also projected.

6.1 Conclusion

The cryptographic scheme presented here, using matrix and array manipulations, is a new concept in symmetric key cryptography. It is a simple algorithm using matrix-based substitution, matrix based key scheduling and array based diffusion operations using data and key values facilitating high conversion speed. This makes the cipher suitable for high-speed encryption applications.

The matrix based substitution and matrix based complex key scheduling offer the following advantages over the conventional schemes.

1) The matrix-based substitution facilitates poly-alphabetic encoding. Poly-alphabetic encoding presents high degree of confusion to an adversary during crypt analysis. It is a desirable feature of a good symmetric block cipher.
2) The complex key generation procedure that generate sub keys for diffusion round operations facilitates strong key avalanche making many bits in the sub keys to change with one bit change in the secret key. This feature brings strong key diffusion in the output of the cipher presenting difficulty in crypt analysis to the adversary. Thus tracing sub keys during crypt analysis becomes very hard.
3) The data based rotations, introduced in the diffusion rounds, on right half data and left half data, separately, using rotation values derived from left half data and right half data respectively facilitate strong data avalanche in the output of the cipher. Many bits in the output data block of cipher change when one bit is changed in the input data block. This feature brings strong data diffusion and renders crypt analysis very difficult.

4) The cipher text generated by this algorithm does not have one to one correspondence in terms of position of the characters in plaintext and cipher text within a block. This is due to the poly-alphabetic substitution and circular shift operations performed in the round function. This feature also makes decryption extremely difficult by crypt analysts. Decryption of cipher text message using a secret key, which is off by only one bit (closest key) from the original key used for encryption, produced a totally unintelligible plaintext. This feature makes a ciphertext-only attack ineffective.

The results obtained and analysis carried out reveal the following:

1) Cipher images generated by the algorithm do not exhibit any texture of the original image.

2) The histogram analysis performed on the encrypted images show that the gray level distribution is more or less uniform so that the encryption is effective.

3) The mean value analysis indicates that the encrypted image mean value along the horizontal direction is more or less consistent. This also shows that the encryption is effective.

4) Adjacent pixel correlation analysis indicates that the correlation between adjacent pixels along horizontal, vertical and diagonal directions are weak in MASK encrypted images. This ensures statistical independence between data bytes within the ciphertext generated by the cipher. Statistical attack on the cipher thus becomes infeasible.

5) Cross correlation between the input image and the encrypted image is weak, indicating that the input and output of encryption are statistically independent. This also makes the cipher immune to statistical attack.

Measurements taken on MASK and AES ciphers show that:

1) The average encryption quality value in MASK (556.21) is better than that of AES (554.98).
2) The average encryption speed of MASK is 8 times faster than AES in the case of text encryption and image encryption.
3) Key avalanche effect produced on the ciphertext blocks is better in MASK (78.44) than in AES (66.43).
4) Data avalanche produced on the ciphertext blocks is less in MASK (94.31) than in AES (106.12).
5) Input-output relationship is more non-linear in MASK than in AES.
6) Key-output relationship is also more non-linear in MASK than in AES.
7) The average value of differential data propagation in MASK (86.3) is less than that of AES(100).
8) The average value of differential key propagation through diffusion rounds in MASK (85.4) is better than that of AES (75.3).

6.2 Suggestions for Further Work

The research work presented here may not be complete in all respects. Due to limitations (availability of crypt analysis tools and expertise) a detailed crypt analysis could not be undertaken. The present algorithm addresses only plaintext data and static images (gray scale and colour) that represent only a part of the information spectrum. Information has diversity and in today’s transactions many data formats such as plaintext, rich text, voice, video, images etc. are used.

All these data formats are to be handled by the encryption service. These aspects need to be addressed in the cryptographic transformation process in order to support the needs of today’s complete secure communications.

As a continuation of the work, following activities could be pursued that may lead to the development of a better and flexible algorithm facilitating enhanced security and diversity of applications.

1) The algorithm can be subject to a detailed cryptanalysis, using tools designed for this purpose, to determine its strengths and weaknesses. This will help in finding the areas where improvements could be attempted.

2) Circular shift operations and Ex-OR operations using non-linear data such as square of a number extracted from a data block can be introduced that can make linear crypt analysis still harder.

3) Substitution mapping using separate matrices initialized using sub keys separately in all diffusion rounds could be attempted. This will make crypt analysis still harder there by making the cipher stronger.

4) Modifications on the algorithm could be tried to handle rich text, video and voice. This will allow full multimedia encryption capability to the cipher facilitating encryption of present day web pages.

References

[1] WILLIAM STALLINGS, “Network Security Essentials (Applications and Standards)”, Pearson Education, pp. 2–80, 2004.

[2] CHARLS P. PFLEEGER, SHARI LAWRENCE PFLEEGER, “Security in computing”, Pearson Education, pp. 66-120, 2004.

[3] DATA ENCRYPTION STANDARD: http://csrc.nist.gov/publications /fips/fips 46-3/fips- 46- 3.pdf

[4] JU YOUNG O.H et.al, “A Selective Encryption Algorithm based on AES for Medical Information”, Health Informatic Research, Vol. 16, No. 1, March 2010, pp. 22–29.

[5] ADVANCED ENCRYPTION STANDARD: http://csrc.nist.gov /publications/fips/fips197/ fips- 197.pdf.

[6] ESCROWED ENCRYPTION STANDARD: http://csrc.nist.gov/ publications/fips/fips1185/ fips-185.txt.

[7] JOSE J. AMADOR, ROBERT W. GREEN, “Symmetric-Key Block Ciphers for Image and Text Cryptography”, International Journal of Imaging System Technology, Vol. 15, pp. 178-188, 2005.

[8] AAMEER NADEEM, M. YOUNUS JAVED, “A Performance Comparison of Data Encryption Algorithms”, 0-7803-9421-6 /2005 IEEE, 2005.

[9] DRAGOS TRINCA, “Sequential and Parallel Cascaded Convolution Encryption with Local Propagation: Toward Future Directions in Cryptography”, Proceedings of The third International Conference on Information Technology-New Generations. (ITNG’06), 0-7695-2497- 4 / 2006, IEEE Computer Society, (2006).

[10] ADAM J. ELBIRT, CHRISTOF PAAR “An Instruction-Level Distributed Processor for Symmetric-Key Cryptography”, IEEE Transactions on Parallel and Distributed Systems, Vol. 16, No. 5, May 2005.

[11] MIN-KYU JOO, JIN-HYUNG KIM and YOON-HWACHOI, "A Fault tolerant architecture for Symmetric Block Ciphers", Proceedings of the 11th Asian Test Symposium (ATS'02), 1081-7735/2002, IEEE Computer Society.

[12] CHRISTOPHE DE CANNIERE, ALEX BIRYUKOV and BART PRENEEL, "An introduction to Block Cipher Crypt Analysis". Proceedings of the IEEE, Vol. 94, No. 2, February, 2006.

[13] DIFFIE W., HELMAN M.E., Multi User Cryptographic Techniques, Proc. of AFIPS National Computer Conf., pp.109-112, 1976.

[14] MERKLE R.C., Secrecy, Authentication and Public Key Systems, UMI Research Pres, Michigan, 1979.

[15] BRASSARD G., Modern Cryptology: a Tutorial, LNCS 325, Springer-Verlag, 1988.

[16] SLOMA A., Public Key Cryptography, Springer-Verlag, Berlin, 1990.

[17] STINSON D.R., Cryptography: Theory and Practice, CRC Press, Boca Raton, 1995.

[18] RIVEST R.L., Cryptograhy, Handbook of Theoretical Computer Science, pp. 719-755, Elsevier Science Publishers, 1990.

[19] DIFFIE W., The First Ten Years of Public Key Cryptology, The Science of Information Integrity. IEEE Press, pp.135 – 175, 1992.

[20] KERCHOFFS A., La Cryptographic Miltaire, Journal des Science Miltaires, 9th Series, pp. 161-191, 1883.

[21] KHAN D., The Codebreakers, Macmillian Publishing Company, New York, 1967.

[22] SHANNON C.E., Communication Theory of Secrecy Systems, Bell Systems Technical Journal, Vol. 28, pp. 656-715, 1949.

[23] VERNAM G.S., Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communication, Journal of The American Institute for Electrical Engineers, 1926.

[24] GUNTER C.G., A Universal Algorithm for Homophonic Coding, Advances in Cryptology-EUROCRYPT 88, pp. 405-414, 1988.

[25] JENDAL H.N., KUHN Y.J.B, and MASSEY J.L, An Information-theoretic Treatment of Homophonic Substitution, EUROCRYPT 89, pp.382-394, 1990.

[26] BEKER H, PIPER F., Cipher Systems: The Protection of Communications, John Wiley & Sons, New York, 1982.

[27] HILL L.S., Cryptography in an Algebraic Alphabet, American Mathematical Monthly, 36, pp. 306-312, 1929.

[28] DIFFIE W. and HELMAN M.E., Privacy and Authentication: Proceedings of the IEEE, Vol. 67, pp. 397-427, 1979.

[29] DAVIES D. and PRICE W.L., Security for Computer Networks, John Wiley & Sons, New York, 2nd Edition, 1989.

[30] FIPS 81, DES Modes of Operation, Federal Inf. Processing Standards Publication 112, US Department of Commerce / N B S, National Technical Information Service, 1981.

[31] COPPERSMITH D., KRAWCZYK H. and MANSUR Y., The Shrinking Generator, Advances in Cryptology – CRYPTO 93, pp. 22-39, 1994.

[32] MATSUI M., On Correlation Between the Order of S Boxes and the Strength of DES, Advances in Cryptology – CRYPTO 87, pp. 185-193, 1988.

[33] MATSUI M. and YAMAGISHI A., A New Method of Known Plain Text Attack of FEAL Cipher, Proceedings of international conference on Advances in Cryptology-EUROCRYPT 92, pp. 81-91, 1993.

[34] Federal Information Processing Standards Publication 197 (FIPS197), http:// csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[35] SHIMIZU A. and MIYAGUCHI S., Fast Data Encipherment Algorithm FEAL, Advances in Cryptology – EUROCRYPT 87, pp. 267-278, 1988.

[36] MIYAGUCHI S., SHIRAISHI A. and SHIMUZU A., Fast Data Encipherment Algorithm FEAL-8, Review of the Electrical Communication Laboratories, Vol. 36, pp. 433-437, 1988.

[37] LANGFORD S.K. and HELLMAN M.E., Differential Linear Crypt Analysis, Advances in Cryptology, Proc. of CRYPTO 94, pp. 17-25, 1994.

[38] OHTA K. and AOKI K., linear Crypt analysis of the Fast Data Encryption algorithm, Advances in Cryptology- CRYPTO 94, pp. 12-16, 1994.

[39] LAI X., On the Design and Security of Block Ciphers, ETH Series in Information Processing, Technische Hochschule, Zurich, 1992.

[40] LAI X. and Massey J.L., A proposal for a New Encryption Standard, Advances in Cryptology, Proc, of EUROCRYPT 90, pp. 389-404, 1991.

[41] MEIER W., On the Security of the IDEA Block Cipher, Advances in Cryptology – EUROCRYPT 93, pp. 371-385, 1994.

[42] MASSEY J.L. and SAFER, K-64.: One Year Later, Fast Software Encryption, Intl. Workshop, Springer-Verlag, pp. 212-241, 1995.

[43] RIVEST R.L., The RC5 Encryption Algorithm, Fast Software Encryption, Second International Workshop, pp. 86-96, 1995.

[44] BROWN L., PIEPRZYK J., SEBERRY J. and LOKI A., Cryptographic Primitive for Authentication and Secrecy Applications. AUSCRYPT 90, pp. 229-236, 1990.

[45] BROWN L., KWAN M. and PIEPRZK J., Improving Resistance to Differential Crypt Analysis and the Redesign of LOKI, ASIACRYPT 91, pp. 36-50, 1993.

[46] ADAMS C., TAVARES S.E., “Designing S-Boxes for CiphersResistant to Differential Crypt Analysis”, Pro. 3rd Symposium on State and Progress of Research in Cryptography, pp. 181-190, 1993.

[47] SCHNEIER B., Description of a New Variable Length Key, 64 Bit Block Cipher, Fast Software Encryption, Cambridge Security Workshop, pp. 191-204, 1994.

[48] VAUDENAY S., On the Weak Keys of Blowfish, Fast Software Encryption, Second International Workshop, pp. 27-32, 1996.

[49] DAEMEN J., Cipher and Hash Function Design, Katholicke Universiteit Press, 1995.

[50] DAEMEN J. and GOVAERTS R., A New approach to Block Cipher Design, Fast Software Encryption, Cambridge Security Workshop, pp.18-32, 1994.

[51] RIJMEN V. and PRENEEL B., The Cipher SHARK, Fast Software Encryption, Third International Workshop, pp. 99-111, 1996.

[52] ANDERSON R. and BIHAM E., Two Practical and Provably Secure Block Ciphers: BEAR and LION, Fast Software Encryption, Third International Workshop, pp.113-130, 1995.

[53] LUBY M., RACKOFF C., How to Construct Pseudorandom Permutations from Pseudorandom functions, SIAM Journal on Computing, Vol. 17, pp. 373-386, 1988.

[54] FIPS 185, Escrowed Encryption Standard, US Dept. of Commerce / NIST National Technical Information Service, Virginia, 1994.

[55] ROE M., How to Reverse Engineer an EES Device, Fast Software Encryption, Second Intl. Workshop, pp. 3305-328, 1995.

[56] CHARNES C., O’CONNOR L. and PIEPRZYK J., Comments on Soviet Encryption Algorithm, EUROCRYPT 94, pp.433-438, 1995.

[57] WHEELER D.J., A Bulk Data Encryption Algorithm, Fast Software Encryption, Cambridge Security Workshop, pp. 127-134, 1994.

[58] WHEELER D.J., TEA: A tiny Encryption Algorithm, Fast Software Encryption, Second Intl. Workshop, pp. 363-366, 1995.

[59] ZHANG YUN-PENG, ZHAI ZHENG-JUN and LIU WEI, “Digital Image Encryption Algorithm Based on Chaos and Improved DES”, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA – October 2009, pp. 474-479.

[60] SAI CHARAN KODURU and V. CHANDRASEKARAN, “Integrated Confusion-Diffusion Mechanisms for Chaos Based Image Encryption”, IEEE 8th International Conference on Computer and Information Technology Workshops, IEEE Computer Society, pp. 260-263, 2008.

[61] DANG P.P. and CHAN P.M., Image encryption for secure Internet multimedia applications [J]. IEEE Transactions on Consumer Electronics, 2000, 46(3): pp. 395-443.

[62] HUA ZHONG. “Based on the chaotic image encryption technology research”, Changsha Polytechnic University Master's Thesis, Hunan, Changsha, pp.5-6.

[63] ALIREZA JOLFAEI and ABDOLRASOUL MIRGHADRI, “An Image Encryption approach using Chaos and Stream cipher**”,** Journal of Theoretical and Applied Information Technology, 2005-2010.

[64] E. DAWSON, H. GUSTAFSON and A.N. PETTITT, “Strict Key Avalanche Criterion”, Australasian Journal of Combinatorics, 6 (1992), pp.147-153.

[65**]** ADITEE GAUTAM, MEENAKSHI PANWAR and DR.P.R GUPTA, “A New Image Encryption Approach Using Block Based Transformation Algorithm “, International Journal Of Advanced Engineering Sciences And Technologies, Vol. 8, Issue 1, pp. 90 – 96, 2011.

[66] ZHANG YUN-PENG et. al, “Digital Image Encryption Algorithm Based on Chaos and Improved DES”, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics San Antonio, TX, USA - October 2009, pp. 474-479.

[67] FRIEDMAN W., Cryptanalysis, US Government Printing Office, Washington, 1994.

[68] GAINES H., Cryptanalysis: A Study of Ciphers and Their Solutions, Dover Publications, New York, 1956.

[69] SINKOV A., Elementary Crypt Analysis: a Mathematical Approach, Random house, New York, 1968.

[70] STINSON D.R., Cryptography: Theory and Practice, CRC Press, Boca Raton, 1995.

[71] KALISKI B.S. and YIN Y.L., On Differential and Linear Crypt analysis of RC5 Encryption Algorithm, proc. of Advances in Cryptology-CRYPTO 95, pp. 171-184, 1995.

[72] COPPERSMITH D. and JOHNSON D.B. and MATYAS S.M., Proposed Mode for Triple-DES Encryption, IBM Journal of Reasearch and Devel. Vol. 40, No. 2, pp. 253-262, 1996.

[73] WIENER M.J., Efficient DES Key Search, Technical Report TR-244, School of Computer Sciences, Carleton University, Ottawa, 1993.

[74] EBERLE H., A High Speed DES Implementation for Network Applications, Advances in Cryptology, CRYPTO 92, pp. 521-539, 1993.

[75] WAYNER P.C., Content Addressable Search Engines and DES like Systems, Advances in Cryptology-CRYPTO 92, pp. 575-586, 1993.

[76] ADAMS C., "Constructing Symmetric Ciphers Using the CAST Design Procedure", in: Designs, Codes and Cryptography, v.12, no. 3, Nov 1997, pp. 71-104.

[77] ADAMS C., "RFC2144: The CAST-128 Encryption Algorithm", May 1997.

[78] BIHAM E., SHAMIR A., Differential Crypt Analysis of Data Encryption Standard, springer, New York, 1993

[79] MATSUI M, Linear Crypt Analysis Method for DES Cipher, Advances in Cryptology, proc. of EUROCRYPT 93, pp. 386-397, 1994.

[80] JEAN-LUC BEUCHAT, “FPGA Implementations of the RC6 Block Cipher”, FPL 2003, LNCS 2778, pp. 101–110, 2003. Springer-Verlag Berlin Heidelberg 2003.

[81] SUSAN K. LANGFORD and MARTIN E. HELLMAN, “Differential-Linear Cryptanalysis”, Advances in Cryptography-Crypto94-Springer, pp. 17-25. 1998.

[82] NYBERG K. and KNUDSEU L., “Provable Security Against Differential Cryptanalysis”, Journal of Cryptography, Vol. 8, No. 1 (1995).

Publications of the Author

1. Paul A.J., P. Mythili, K. Poulose Jacob, “Matrix based Cryptographic Procedure for Efficient Image Encryption”.

    - IEEE Explore digital library, *ISBN: 978-1-4244-9478-1*: Recent Advances in Intelligent Computational Systems (RAICS),2011IEEE,pp.173–177. http://ieeexplore.ieee.org/xpl/freeabsall.jsp?arnumber=606929.

    - Proceedings of International Conference on Recent Advances in Intelligent Computational Systems, RAICS IEEE, Sept-2011, pp. 173-177.

2. Paul A.J., P. Mythili, K. Poulose Jacob, “Matrix based Key Generation to Enhance Key Avalanche in Advanced Encryption Standard”.

    - International Journal of Computer Applications, No. 2, article 1, pp. 31–34, 2011. Published by Foundation of Computer Science (USA).*ISSN:0975-8887*. http://www.ijcaonline.org/proceedings/icvci/number2/2638-1152.

    - Proceedings of the International Conference on VLSI, Communication and Instrumentation, April 7-9, 2011, pp. 67-70.

3. Paul A.J., P. Mythili, K. Poulose Jacob, “Matrix based key Generation for Enhanced Key Avalanche Effect in Symmetric Block ciphers”.

    - IETECH Journal of Electrical Analysis, Vol. 4, No. 3, 2010, pp. 94-100. *ISSN: 0973-8088.*

4. Paul A.J., P. Mythili, Varghese Paul, “Matrix based Substitution for Cryptographic Transformations”.

    - International Journal of Computer Sciences, Software Engineering and Electrical Communication Engineering, Vol. 1, No. 1, January-June, 2010, pp. 9-14, 2010. *ISSN: 2229-3175*.

5. Paul A.J., P. Mythili, Varghese Paul, “Fast Symmetric Cryptography using Key and Data based Masking Operations”.

    - International Journal of Computational Intelligence - Research & applications, Vol. 3, No. 1, January – June 2009, pp. 5-10. *ISSN: 0973-6794.*

    - Proceedings of International Conference on VLSI and Communication Engineering, April 16-18, 2009, pp. 845-850.

6. Paul A.J., Varghese Paul, P. Mythili “A Fast and Secure Encryption Algorithm for Message Communication”.

    - IETECH international Journal of Communication Techniques, Vol. 2, No. 3, 2008, pp. 104-109. *ISSN: 0973-8053.*

    - Proceedings of International Conference on Information and Communication Technology in Electrical Sciences (ICTES), 20-22 Dec. 2007, Vol. 2, No. 2, pp. 629-634.

    - IET Digital library**,** 629 (2007), *ISBN: 978 0 86341 937 9.* http://dx.doi.org/10.1049/ic:20070688.

    - IEEE Explore digital library, *ISSN : 0537-9989,* http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4735872&isnumber=4735760.

7. Paul A.J., Varghese Paul, P. Mythili, “Matrix Array Symmetric Key Encryption”.

    - Journal of Computer Society of India, Vol. 37, Issue 1, Jan – Mar 2007, pp. 48 – 53. *ISSN 0254-7813.*

8. Paul A.J., Varghese Paul, P. Mythili, “A Matrix Based Cryptographic Algorithm for High Speed Encryption Applications”.

    - Proceedings of National Conference on VLSI and Communication, 14-15 March 2008, pp. 97-100.

9. Paul A.J., P. Mythili, “Poly-alphabetic Substitution Mapping for Cryptographic Transformations”.

    - Proceedings of National Conference on Recent Innovations in Technology (NC-RIT), March 26-28, 2009, pp. 32-36.

10. Paul A.J., P. Mythili, “Fast Encryption using Key based Substitution and Data based Rotations”.

    - Proceedings of National Conference on Research in Engineering, March 27, 2009.