IMPROVING AES-128 USING MULTIPLE CIPHER KEYS AND KEY DEPENDENT S-BOXES

A Thesis Submitted in Fulfilment of the Requirement for the Award of the Degree of

Master of Engineering

in

Electronics and Communication

Submitted by

SHIVANI SACHDEVA

Roll No. 801661023

Under Supervision of

Dr. Ajay Kakkar

Assistant Professor

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB
JUNE, 2018

DECLARATION

I hereby declare that the thesis work entitled "**Improving AES-128 using Multiple Cipher Keys and Key Dependent S-Boxes**" is an authentic record of my research work carried out as requirement for the award of degree of ME (Electronics and Communication Engineering) at Thapar Institute of Engineering and Technology, Patiala, under the guidance of **Dr. Ajay Kakkar**, Assistant Professor, Department of Electronics and Communication Engineering.

The contents of this thesis work have not been submitted to any other University/Institute for the award of any other degree.

Date: 7-6-18

Shivani Sachdeva
Roll Number 801661023

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

Date: 07/06/18

Dr. Ajay Kakkar
Assistant Professor, ECED

ACKNOWLEDGEMENT

Firstly, I would like to express my heartfelt gratitude to **Dr. Ajay Kakkar, Assistant Professor**, Department of Electronics and Communication Engineering, Thapar Institute of Engineering and Technology, Patiala for his constant guidance and support that I received throughout the completion of my thesis work. I consider myself privileged to have the opportunity to work with him. Also, I would like to thank our Head of Department **Dr. Alpana Agarwal** for providing me with the necessary resources that helped greatly in the completion of my research work. I am also thankful to the faculty and staff of Department of Electronics and Communication and Engineering and then friends who supported and helped me in direct or indirect ways throughout the completion of this work.

Lastly, I would like to thank my parents for their unyielding love, support and inspiration that has helped me in my endeavours all my life.

Shivani Sachdeva
ME-ECE
801661023

ABSTRACT

Privacy and Security are the two major advantages that every user strives to achieve in today's era of digital communication. Sharing and downloading of millions of files by people all over the globe has raised many privacy and security concerns. The sharing of critical data among large corporations, departments and governments is always under the radar of third parties called the adversaries, attackers or hackers. These parties try to exploit the form of communication in order to retrieve the confidential information. This can be avoided using encryption schemes. Encryption schemes are basically of two types- symmetric encryption and asymmetric encryption. AES-128 is one such symmetric encryption scheme, which was standardized in 2001, and is in widespread use for data encryption in today's world. It uses same key to encrypt and decrypt the data. However, the static nature of keys and Substitution boxes used in the algorithm for AES poses vulnerability for many cryptanalytic attacks. In the proposed work, Chapter 1 aims at discussing the basic concepts in cryptography. Chapter 2 lists the work done by various researchers in order to improve the cryptographic strength of AES-128 and on the basis of observations, objectives are drawn. The detailed discussion on algorithmic implementation of AES-128 is presented in Chapter 3. The proposed methodology, as described in Chapter 4 aims at increasing the cryptographic security by introducing dynamic keys and key-dependent S-Boxes in the conventional AES-128 approach. Chapter 5 presents the results in form of advantages or improvements in quality tested by analyzing the execution time, avalanche effect, and strict avalanche criteria, which tests the randomization of output when input is changed only slightly. Finally, in Chapter 6, the conclusion and future scope has also been mentioned.

TABLE OF CONTENTS

S.NUMBER	Name of the Chapters	Page Number
	Declaration	ii
	Acknowledgement	iii
	Abstract	iv
	Table of Contents	v
	List of Figures	viii
	List of Tables	ix
	List of Abbreviations	x
Chapter 1:	**Introduction**	**1-11**
1.1	Fundamental Concepts about Cryptography	1
1.2	Goals of Cryptography	2
1.3	Basic Terminologies	3
1.4	Types of Cryptography Schemes	3
1.5	Symmetric Encryption Schemes	4
1.6	Asymmetric Encryption Schemes	5
1.7	Comparing symmetric and asymmetric encryption schemes	6
1.8	Cryptanalysis	7
1.9	Cryptanalysis Techniques and Attacks	8
1.10	Models for evaluating Security	10
1.11	Outline of Thesis	10
Chapter 2:	**Literature Review**	**12-24**
2.1	Data Encryption and Improvement in Encryption Technique	12
2.2	Observations	23
2.3	Motivation and Problem Formulation	23
2.4	Objectives	24
Chapter 3:	**Advanced Encryption Standard (AES)**	**25-38**
3.1	History of AES	25
3.2	Description of Cipher	25
3.3	Byte	26
3.4	State of the Algorithm	27
3.5	Galois Field Arithmetic	27
3.5.1	Addition and Subtraction	28

3.5.2	Multiplication	29
3.5.3	Inversion	29
3.6	Internal Structure of AES encryption	30
3.6.1	Byte Substitution	32
3.6.2	Diffusion Layer	33
3.6.3	Key Addition Layer	33
3.6.4	Key Schedule for AES-128	34
3.7	AES Decryption	36
3.7.1	Inverse Mixcolumn Sublayer	37
3.7.2	Inverse ShiftRows Sublayer	37
3.7.3	Inverse Byte Substitution	38
3.7.4	Key Schedule for AES-128 Decryption	38
Chapter 4:	**Proposed Methodology**	**39-45**
4.1	Proposed Approach	39
4.2	First Encryption using Conventional AES Approach	40
4.3	Second Encryption using Random key Generation and Key-Dependent S-Box	40
4.3.1	Random key generation	40
4.3.2	Key-Dependent S-Box Generation	40
4.3.2.1	Pseudo-Expansion of the key	41
4.3.2.2	Generation of Key-Dependent S-Box and Inverse S-Box	41
4.4	Avalanche Effect in Encryption	43
4.5	Strict Avalanche Criterion	43
4.6	Methodology	43
4.6.1	First Encryption using static key	43
4.6.2	Second Encryption using Dynamic key and Key-Dependent S-Box generation	44
4.6.3	Avalanche effect calculation	45
4.6.4	Strict Avalanche Criteria	45
Chapter 5:	**Results and Discussion**	**46-53**
5.1	Key Dependent S-Box	46
5.2	Comparison based on Execution Time	47
5.2.1	Simple AES and AES with key-dependent S-Box	47
5.2.2	AES-128 using single key and Multiple keys	47
5.2.3	Conventional AES and AES using Proposed Methodology	48
5.3	Avalanche Effect Test	49

5.3.1 Simple AES and AES with key-dependent S-Box 49
5.3.2 AES-128 using single key and Multiple keys 50
5.3.3 Conventional AES and AES using Proposed Methodology 50
5.4 Strict Avalanche Criteria 51
5.4.1 Simple AES and AES with key-dependent S-Box 51
5.4.2 AES-128 using single key and Multiple keys 52
5.4.3 Conventional AES and AES using Proposed Methodology 53

Chapter 6: **Conclusion and Future Scope of Research** **54**

References 55-59
List of Publications 60

LIST OF FIGURES

Figure Number	Figure Details	Page number
1.1	*Goals of Cryptography*	2
1.2	*Basic Cryptography Model*	3
1.3	*Symmetric Key Cryptography*	3
1.4	*Asymmetric Key Cryptography*	4
1.5	*Various Cryptography Schemes*	4
3.1	*AES input/output parameters*	26
3.2	*Arithmetic operations in GF(5)*	28
3.3	*Block Diagram of AES-128 Encryption*	30
3.4	*Structure of Single AES Round*	31
3.5	*Arrangement of data bytes in state and key bytes in 4x4 matrix*	31
3.6	*Substitution Box used for encryption in AES*	32
3.7	*Key-Schedule for AES-128*	34
3.8	*Constituent Operations in non-linear function g()*	35
3.9	*Block Diagram for AES-128 Decryption*	36
3.10	*Inverse S-Box for AES-128 Decryption*	38
4.1	*Block diagram for proposed Approach*	39
4.2	*Flowchart depicting Proposed methodology*	44
5.1	*Key-dependent S-Box using key in equation 7.1*	46
5.2	*Execution time comparison between AES-128 and AES-128 with key-dependent S-Box*	47
5.3	*Execution time comparison of AES-128 using single & multiple key*	48
5.4	*Execution time comparison of Conventional AES and AES using proposed methodology*	48
5.5	*Comparison between Avalanche effect shown by simple AES and AES with key-dependent S-Box*	49
5.6	*Comparison between Avalanche effect shown by AES with single key and AES with multiple keys*	50
5.7	*Comparison between Avalanche effect shown by conventional AES and AES with proposed methodology.*	51
5.8	*Comparison between simple AES and AES with key dependent S-Box based on Strict Avalanche Criteria*	52
5.9	*Comparison between Single key AES and multiple key AES based on Strict Avalanche Criteria*	52
5.10	*Comparison between Conventional AES and AES with proposed methodology based on Strict Avalanche Criteria*	53

LIST OF TABLES

Table Number	Table Details	Page Number
1.1	*Comparison of Symmetric Encryption Standards*	5
3.1	*Key Lengths and Round Functions*	26
3.2	*Representation of various patterns of bits in Hex form*	27
3.3	*Arrangement of elements in State Matrix*	27

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
GF	Galois Field
S-Box	Substitution Box
DES	Data Encryption Standard
3DES	Triple Data Encryption Standard
RSA	Rivest Shamir Adleman
DSA	Digital Signature Authentication
NBS	National Bureau of Standards
NIST	National Institute of Standards and Technology
FIPS	Federal Information Processing Standard
RC	Round Constant
MPEG	Motion Picture Expert Group
IDEA	International Data Encryption Algorithm
FPGA	Field Programmable Gate Array
LSB	Least Significant Bit
SAC	Strict Avalanche Criteria
LUT	Look-Up Table
LAT	Linear Approximation Table
ECDHE	Elliptic Curve Diffie-Hellman Exchange
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Greatest Common Multiple
SHA	Secure Hash Algorithm
ROM	Read Only Memory
BIC	Bit Independence Criteria

CHAPTER 1
INTRODUCTION

This chapter introduces the basic concepts in cryptography and its related terminologies. The basic definitions, goals, types of cryptography, especially AES-128 are described at length among the sections and subsections that follow.

1.1 FUNDAMENTAL CONCEPTS ABOUT CRYPTOGRAPHY

Cryptography refers to the technique which is used to convert secret information to a form which appears unintelligible to a random user. This imparts security to the process of communication because the information in random form can be stored or transmitted across inherently insecure communication channels in a safe and secure way. The random information cannot be correctly read by a random person trying to access the system [1].

Contemporary cryptography derives its principles and logical basis from an amalgamation of the various subjects like mathematics, computer sciences, electronic sciences, communication sciences etc. Cryptography finds its use in most of the domains in today's scenario like the e-commerce, crypto-currencies, secure log-ins, in authentication purposes, social media platforms, security purposes like passwords in door locks, safe locks etc., military applications, computer networking and others [2].

Cryptanalysis is the technique which has its focus to defeat the cryptography. The main aim of the technique of cryptanalysis is to catch the inherent flaws, shortcomings and lapses of an encryption strategy, so that security can be breached. Ethically used, cryptanalysis can help to find vulnerabilities in the scheme which can be overcome to make the system secure without any underlying weakness in algorithmic implementation.

With the exponential growth in the internet usage in the recent years, people are communicating and sharing data among themselves more often than ever before. The data that exchanged ranges from textual documents to images, audio and video. Huge amounts of data sets are collected, analyzed and are shared by various governmental agencies and private sector organizations. There are always groups or individuals present who try to illegally access the secret communication between two parties, and may cause harm to both, in the form of information loss, financial loss, leakage of secret information etc. Phishing attacks, hacking attempts have raised security concerns all over the world. The communication of confidential data, financial transactions and other similar activities require the communication system to be highly secure and ensure the privacy and integrity of users [2].

While designing a new cryptographic scheme, researchers have to keep in mind the future developments which may render the system unsecure. Advancements in processing power, quantum computers can make many existing secure systems vulnerable to cryptanalytic attacks like brute-force attacks. Thus, researchers are devoted to develop new techniques to cope up with these attacks. New techniques are being devised which are highly secured, reliable and computationally efficient. Existing communication techniques are being updated to minimise their vulnerability to attacks.

1.2 GOALS OF CRYPTOGRAPHY

Cryptography aims at achieving a certain set of goals, which impart the user with a desirably secure communication system.

- Confidentiality: The information communicated between two parties must be known and be accessible by them only. No adversary or third party should be able to read or know the contents of communication even if he/she has access to it [2].
- Data Integrity: The information transmitted by a sender in the communication channel must reach the intended recipient in its original form and should not be modified before reaching the destination. Illegal access should not be able to modify the contents of data [2].
- Authenticity: This is related to identification. Only the users registered with the service provider should be able to access the system. The unauthorized users should be denied service as they may damage the system with mischievous activities [3].
- Non-Repudiation: In the event of a conflict in which a party denies about the data transferred, some plan of action must be put in place to efficiently resolve the issue. As an analogy, consider that first party had accredited the acquisition of a commodity by second party and then denied about having done such kind of act. This may lead to the intervention of a trusted third party so that the dispute can be resolved [3].

  The above goals are depicted in figure 1.1.

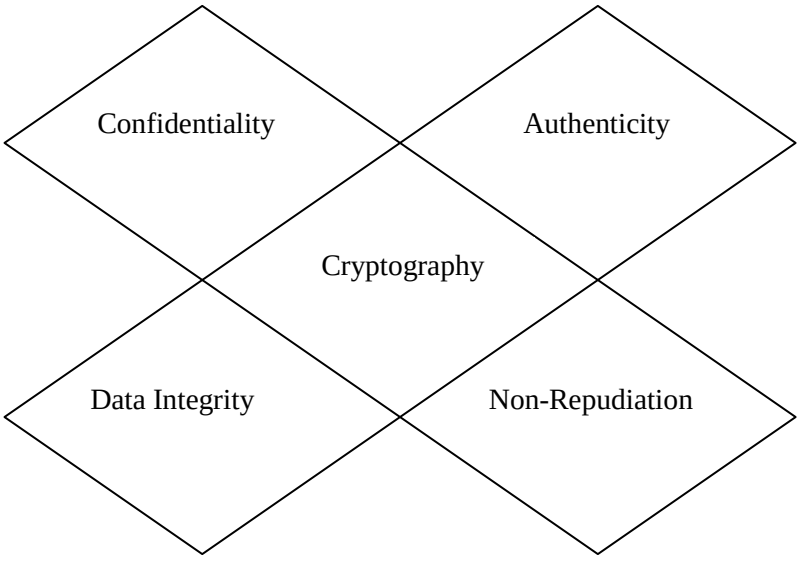

Figure 1.1 Goals of Cryptography

1.3 BASIC TERMINOLOGIES

The basic definitions used in describing a basic cryptographic model, are described as follows:

- Plaintext: It refers to the data/information in its original form. This is the form of data generated by a sender which it intends to communicate with another party.
- Encryption: It is defined as the process of converting the plaintext to a scrambled form which becomes unintelligible to an adversary. It follows certain mathematical operations on the plaintext to convert it to ciphertext. [4]
- Ciphertext: The scrambled form of the plaintext after encryption is called ciphertext. This appears as a totally random collection of alphanumeric characters and symbols so that it is unintelligible to people other than sender and intended recipient.
- Key: The key is the logic using which plaintext is converted into ciphertext. The output or the ciphertext generated from the encryption algorithm will change according to the key employed at that specific instant. The value of the key which is used does not depend on either the plain data or the algorithm [4].
- Decryption: It refers to the process using which we can convert the encrypted data into its simple original form. The basic cryptography model is depicted in figure 1.2.

Figure 1.2 Basic Cryptography Model

1.4 TYPES OF CRYPTOGRAPHY SCHEMES

Cryptographic schemes can be classified depending upon the method which is followed for the generation or usage of the keys in the system.

- Private Key Encryption: This scheme utilizes same cipher key for encrypting and decrypting the data. Sender and intended recipient have the firsthand knowledge about the key and associated mechanisms. Figure 1.3 describes the basic concept of private/symmetric key cryptography.

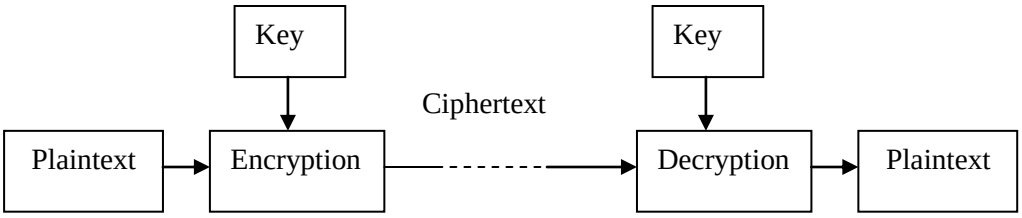

Figure 1.3 Symmetric Key Cryptography

- Public Key Encryption – This scheme uses a public key and a private key. If A wants to communicate some data securely to B, A encrypts the data using B's public key. To decrypt the data, B uses its own private key. This technique in encryption is utilized to securely exchange the cryptographic keys over the network [3]. Figure 1.4 describes the basic concept of asymmetric cryptography.

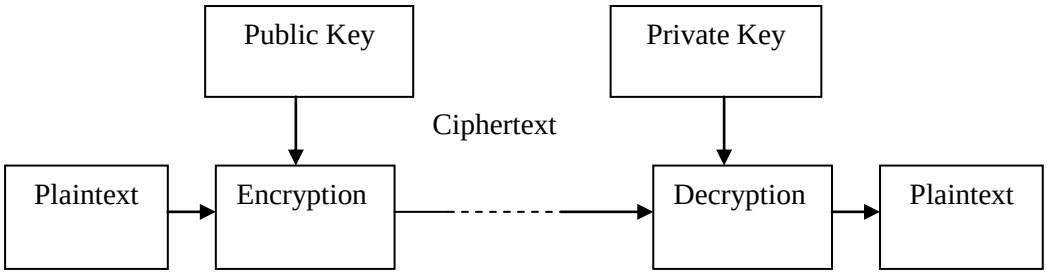

Figure 1.4 Asymmetric Key Cryptography

Figure 1.5 lists some of the common standards employed in both the schemes.

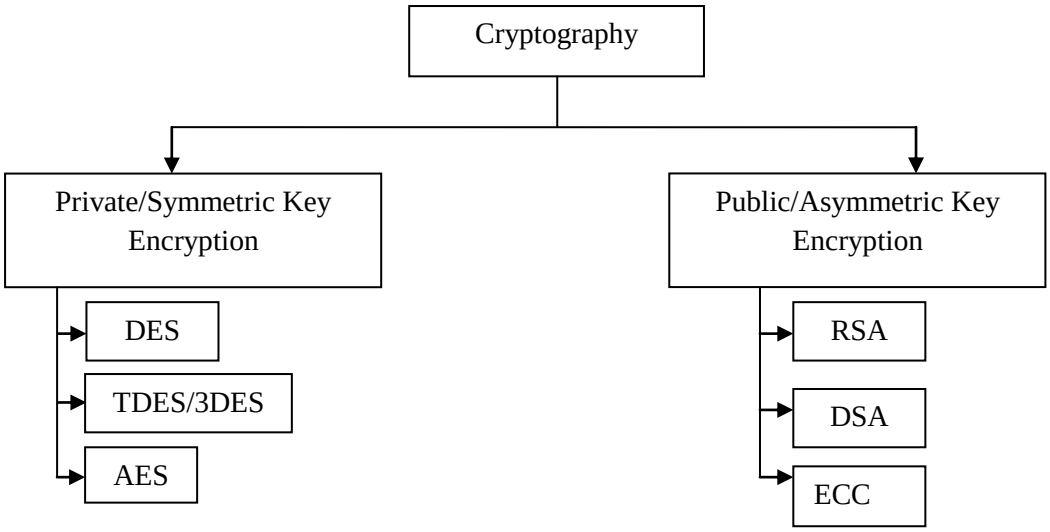

Figure 1.5 Various Cryptography Schemes

1.5 SYMMETRIC ENCRYPTION SCHEMES

The symmetric encryption schemes are described as follows:

- DES (Data Encryption Standard): This standard was incepted at the IBM in 1974 and was approved as a federal standard by NBS (now NIST) in 1977. It was published as FIPS PUB 46. It utilized a key having 56 bits and a data block having 64 bits. DES became victim to cryptanalytic attacks because the key length used to encrypt the data in the algorithm was very small. In January 1999, a group of organizations collaborated to publicly break the security of DES in a short span of time. Later, it was withdrawn as a standard by the NIST [5].

- TDES/3DES (Triple Data Encryption Standard): TDES or 3DES attempted to increase the security of DES by encrypting each data block three times according to the technique described in DES. As the key in DES was too short to provide sufficient security, it simply increased the key size, thus eliminating the need of designing an altogether new cipher algorithm. The key bundle of three keys made it more secure but using one key bundle for more than 2^{20} 64-bit data blocks, the security was affected [6].

- AES (Advanced Encryption Standard): Conceptualized by Vincent Rijmen and John Daemen, who were Belgian cryptographers, AES was originally known as Rijndael, it was adopted as the standard by NIST in 2001. It was published as FIPS PUB 197 in 2001. Key length varies according to the choice of transformation rounds which can be 10, 12 or 14. A data block with length 128 bits is encrypted using keys with varying length of 128, 192 and 256 bits. Among all three symmetric encryption schemes, AES is considered to be the most strongest and is in widespread use. AES manipulate the data by using substitution and permutation. This has linear and non-linear operations combined which increases the security of the algorithm. The subkeys are derived from a key schedule which is different for different key lengths. The number of keys used in the algorithm is always one more than the number of transformation rounds [7].

Table 1.1 compares the above three standards on basis of their technical implementations.

Parameters	**DES (1974)**	**3DES (1998)**	**AES (2001)**
Number of keys	1	3	3
Key Length (bits)	56	56,112, 168	128,192, 256
Data Block (bits)	64	64	128
Round Function	16	48-DES equivalent	10,12,14 Depending on key length

Table 1.1 Comparison of Symmetric Encryption Standards

1.6 ASYMMETRIC ENCRYPTION SCHEMES

The asymmetric encryption schemes are explained as follows.

- RSA (Rivest-Shamir-Adleman) Encryption: It is a type of public key cryptography scheme. In this, encryption key is public and is completely different from the key used for decryption, which is kept private. Ron Rivest, Adi Shamir and Leonard Adleman devised this scheme in 1978. Its key size varies from 1024 bits to 4096 bits. RSA is comparatively a slower approach for encryption of data. Therefore, it is not directly used to encrypt bulk amounts of data.

Rather, it is used to encrypt the shared keys used in symmetric encryption schemes which offer high speed encryption [8].

- DSA (Digital Signature Algorithm): This scheme was published in FIPS in 1993. Digital Signatures are used as a means to prove the authenticity of the documents shared online between different parties. It has two phases- choice of algorithm parameters i.e. to choose hash functions and generation of public and private keys which are to be used during encryption [9].

- ECC (Elliptic Curve Cryptography): This type of asymmetric cryptography harnesses the algebraic properties of elliptical curves in Galois fields. Using a relatively shorter key length, it can provide an equivalent security as other algorithms. They can be used for authentication purposes or random number generations. It is used as ECDH (Elliptic curve Diffie Hellman) protocol for exchanging the keys and ECDSA (Elliptic Curve Digital Signature Algorithm) for authentication purposes [10].

1.7 COMPARING SYMMETRIC AND ASYMMETRIC ENCRYPTION SCHEMES

- Advantages of Symmetric Key Encryption

  These have high throughput rates in encrypting the data of the range of hundreds of megabytes per second. Key length for symmetric key encryption schemes is comparatively shorter in length. They can be employed as basic techniques in order to construct various useful mechanisms like pseudo-random number generation, hash functions, digital signature schemes etc [11].
- Disadvantages of Symmetric Key Encryption

  The key involved in encrypting and decrypting the data must remain secret at both ends of sender and receiver. In a large network, it becomes difficult to manage keys and simultaneously maintaining their secrecy. Utilizing a single constant key for entire encryption operation can prove harmful. For efficient and more secure communication, keys must change with every session, which involves more focus on key management [11]. Digital signature mechanism using encryption often requires the use of large keys.
- Advantages of Asymmetric Key Encryption

  The private key is required to be kept a secret while encrypting the data using asymmetric encryption. Relying on the method of using, the set of private and the public keys can be kept unchanged in considerable periods of time. This can be used to encrypt keys which are to be shared over the network. Digital signature mechanism using asymmetric encryption requires shorter keys than their symmetric counterparts.

- Disadvantages of Asymmetric Key Encryption

  Throughput rates for asymmetric encryption schemes are usually very low in comparison to symmetric encryption schemes. This is because of the larger keys which are used to encrypt the data using asymmetric encryption scheme, owing to the great number of substitution and permutation – combination approaches used in encryption of the data [11].

The comparison of symmetric and asymmetric encryption schemes gives two useful insights:

- Symmetric key encryption schemes are useful for encrypting bulk amounts of data at high throughput rates.
- Asymmetric key encryption schemes are useful for key management and authentication services.

1.8 CRYPTANALYSIS

Cryptanalysis is the science or technique of studying and analysing various cryptographic schemes in order to find out the vulnerabilities to breach the security provided by them or in other words, cracking them. Adversaries or third-party persons often try to intrude or listen to the secret communication going on, and try to snoop in, to break the security and read the message being transmitted. Cryptanalysis can also be used by military or some big organizations, who want to test the security of their communication securities, so they can find the loopholes in the implementation and strengthen them before any adversary could take illegal advantage of that [1].

Because the cryptography algorithms employ lots of permutations, combinations, substitutions and mathematical manipulations, thus, to perform a successful cryptanalysis on a cryptographic scheme requires that the attacker must have first-hand in-depth knowledge of mathematics and computer science and systems with very high computational power, in order to launch an attack and successfully crack or decode the system.

The practice of cryptanalysis is exercised by a large number of organizations, involving national authorities targeting to decrypt other country's secret communications; firms and companies implementing security procedures and employ cryptanalysts for checking and enhancing the security details; the analysts, attackers, amateur researchers, scholars and other people who inspect the encryption techniques and approaches to find loopholes in security. It is a constant ongoing struggle or hunt between cryptographers aiming at enhancing the security of information and cryptanalysts which aim to successfully decode or break or breach the security of cryptosystems [2].

An attacker generally uses the information available about the plaintext or the ciphertext and harnesses it in order to launch attacks and decrypt the data.

1.9 CRYPTANALYSIS TECHNIQUES AND ATTACKS

The types of cryptanalytic attacks depend upon the kind of cryptography scheme employed and the information available with the hacker or attacker using which he/she starts to decipher the system. It usually involves the attacker having some knowledge about the plaintext, ciphertext, encryption algorithm. Some of the cryptanalytic attacks are briefly described as follows:

- Ciphertext-only Attack (COA): For this kind of attack, the intruder tries to breach the security of ciphering algorithm by having at hand the knowledge of cipher texts. Also if the intruder is competent to obtain any kind of information about plaintext or the key, from using only the knowledge of ciphertext set, the attack is considered to be successful.

- Known Plaintext Attack (KPA): As is clear from the name, the attacker acquires the first-hand knowledge about the plaintexts and its encrypted ciphertext versions. Using the knowledge about both, the intruder tries to obtain the knowledge about the key or logic which is being used to encrypt the data. If the known plaintext constantly gives out the same ciphertext, the hacker can analyze and retrieve the key combinations from this [2].

- Chosen Plaintext Attack (CPA): For CPA type attack, it is assumed that attacker has the access to the device that is performing the encryption, and the encryption algorithm. Using this algorithm, the attacker can feed certain plaintexts to the device and obtain the corresponding ciphertext. This can lead to obtaining some important information about the key which is involved in converting the known plaintext into ciphertext [3].

- Chosen Ciphertext Attack (CCA): Similar to CPA type attacks, in Chosen Ciphertext Attacks, the attacker has the information about the ciphertext only. Feeding ciphertext to the encrypting device, hacker could retrieve varying plaintext by the use of variable un-noted keys, and thus can obtain some estimates about the real encryption key [3].

- Man in the Middle Attack: For this type of cryptanalytic effort, the intruder places itself between the centre of sender and receiver and assumes the alias of receiver for the sender, and that of sender for the receiver. By doing so, the attacker can intercept the communication from both sides whereas the sender and receiver are under the illusion of communicating through a secure channel. This type of attack can be prevented by taking essential authentication before the start of any communication session or also in between the ongoing communication sessions. The authentication can be done using hashing or digital signatures [3].

- Side Channel Attack: These kinds of attacks can take place when the attacker has some knowledge about or access to the device performing the encryption. Side-channel attacks, which are successful, use information which does not equals ciphertext obtained from the encryption. Also, it is not equal to the plaintext which is yet to go under encryption. It could depend on various factors like the time taken to execute a certain operation, rate at which the system consumes the power, or EM emissions which originate from system performing the encryption [3].

- Brute-Force Attack: For this kind of intrusion, the hacker tries all possible combinations of key in order to successfully break the system. For example, in case of AES-128, as the length of the key is of 128 bits, therefore, there are 2^{128} possible combinations of this key and for a worst case environment, attacker has to try all of them for guessing the true cipher key. It has the requirement of availability of systems with very high computational power because this attack is very time consuming. [4].

- Dictionary Attack: This type of attack is generally based on guessing the passwords. In this kind of attack, the human tendency, to have password which are simple and easy to remember, is exploited, because of which they become very vulnerable to attacks. Due to this, many websites or organizations ask the individuals who are setting up accounts for authentication and other purposes, to have stronger passwords based on amalgamations of alphabetic letters in both upper and lower case, numerical and the various characters etc. with certain specific key lengths. Larger key space with variation in characters is often considered a stronger password [4].

- Linear Cryptanalysis: This type of cryptanalytic attack is generally widely employed on block ciphers. These are based on finding linear approximations to the operation of cipher. Attacker can generate the secret key harnessing the knowledge of sufficient data and encrypted ciphertext pairs. Linear cryptanalysis has two parts, in which the first part is responsible for generating linear equations related to plain text, secret key and cipher text and the other part utilizes linear equations generated by first part with established plaintext and ciphertext to generate secret logic [4].

- Differential Cryptanalysis: These kinds of techniques are also generally relevant to block ciphers. In general terms, it can be defined as the study of how the differences in input combinations result in variation in output results. For case of block ciphers, differential

cryptanalysis is used to study the behaviour of encryption algorithm, where and how the algorithm exhibits non-linear behaviour, and synthesizing a pattern from the above data that how the key logic operates for the purpose of encryption of the data. It was first applied to the DES in 1980s [5].

1.10 MODELS FOR EVALUATING THE SECURITY

The amount of security that a cryptographic system does provide can be evaluated under different models [11].

- Unconditional Security:
  An encryption technique is deemed for being unconditionally secure granted that the security it provides being independent of the computing power or the computational time an attacker can employ. In other words, an unconditionally secure scheme provides the perfect secrecy. For a symmetric encryption scheme, a necessary condition to provide unconditional secrecy is length of logic key should be equal to message.

- Provable Security:
  An encryption technique is deemed for being provably secure in case that the difficulty which arises in breaking it, is comparable to the effort required to solve some other seemingly hard problem such as factoring of suitably large composite integers or computation of discrete logarithms.

- Computational Security:
  An encryption technique is deemed for being computationally secure on the basis that the computational power required to break the security of code far exceeds the computational power of the resources available with the attacker. Most of the known encryption schemes, whether symmetric or asymmetric in nature, belongs to this categories. This condition is also sometimes called practical security.

1.11 OUTLINE OF THE THESIS

In this thesis report, beginning from the basic concepts in cryptography, symmetric cryptography with special attention to AES-128 has been discussed in detail. The aim of the thesis work is to devise a technique to increase the security of AES technique. There is a proposed improvement in encryption scheme which is the basic driver of the thesis and is discussed in detail in chapter 3. Following is the outline of the thesis and the main contents of each chapter:

- In Chapter 1, the basic foundations in the field of cryptography, types of cryptographic schemes and cryptanalytic attacks are discussed in detail. Models for cryptographic security are also presented in brief.

- In Chapter 2, the research work done by various people in the field of cryptography is analysed, so as to find the gaps in the study and drawing some observations which will form the basis of the thesis work.

- In Chapter 3, detailed discussion on the history, basic concept and algorithmic implementation of AES-128 is described in detail. The various layers and sublayers which perform the manipulation of the data while encryption is discussed in detail along with the necessary flowcharts and diagrams.

- In Chapter 4, the proposed methodology to improve the security of the AES-128 scheme is taken into account. Various criteria based on which the performance can be judged, are discussed.

- In Chapter 5, the result of the simulations, and comparison with traditional schemes are presented.

- In Chapter 6, the concluding remarks for the proposed work and scope for the future work is discussed.

CHAPTER 2

LITERATURE REVIEW

Research efforts of numerous researchers regarding the area of cryptography are presented in this chapter. The continuous hunt for more secure, better, and efficiently implementable algorithms in software and hardware are continuously on. Observations have been drawn from the literature review. Based on the gaps in the study, problem formulation has been done. Finally, the objectives for this thesis are determined.

2.1 DATA ENCRYPTION AND IMPROVEMENT IN ENCRYPTION TECHNIQUES

This section involves the research efforts in the area regarding data encryption and improvements in encryption techniques as suggested by the researchers.

Diffie *et al.* [12] discussed the various problems of key distribution, confidentiality, trap doors and others faced by the previously employed cryptographic schemes. Basic fundamentals and concepts in conventional cryptography models and information flow were discussed. Security models and types of attacks on various cryptographic techniques were listed. Detailed discussion on the public key and private key cryptosystems, computational complexity and authentication using digital signatures was presented.

D. Coppersmith [13] presented basic fundamentals of two prospects of cryptography- Data Encryption Standard and set of protocols for exchanging keys over the internet given by Diffie-Hellman. Key passing and Prime integer selection in fixed and finite fields, and the computational constraints were discussed. Also, elliptic curves were introduced to solve discrete logarithm problems. The basic working and algorithmic specifications for DES were presented.

G. J. Simmons [14] described at length the problems in information authentication. Authentication was described as the determination procedure that an authorized recipient may use, such as a selected acknowledged message was in fact actually initiated by authorized sender and has its integrity intact. In early days, Acme codes were used to scramble information. Then, cryptographic standards for symmetric and asymmetric encryption. The principles for computationally secure, provably secure, unconditionally secure were discussed.

Ehrsam *et al.* [15] recounted the basic concepts in cryptographic models and end-to-end encryption in cryptography through the use of key management. In the approach presented, the master key was stored in a secure cryptographic facility. Every new session key was retrieved from the basic master cipher key. Thus, the problem of providing a secure and safe handling of multiple number of keys was reduced to securely managing a key.

C. Boyd [16] described the basic principles and techniques employed for the area of cryptography. Various types of cipher schemes, feistal and non-feistal structures were described. The strength of DES against cryptanalytic attacks was discussed as it was the prominent encryption standard of that time. Among the asymmetric algorithms for encryption algorithms, RSA algorithm was discussed at length. Also, digital signatures were an upcoming remarkable technique for authentication purposes. Application and implementations of various algorithms were presented.

D. Coppersmith [17] explained the development of Data Encryption Standard (DES) in IBM in 1974, which later became a national federal standard in 1977 through NBS (now NIST). The algorithm in encrypting a data block with length of 64-bits using key logic with length of 56-bits to generate ciphertext block of length 64-bits was discussed in length. Also, an overview about a known cryptanalytic attack called differential cryptanalysis was presented which basically attacks block ciphers. The criteria for selection of S-Box and permutations was presented which thwart off such attacks.

S. Garfinkel [18] referenced the problem of securing financial transactions and other critical applications through the use of cryptography. The general proposal for this asymmetric key encryption was developed by Diffie and Hellman. Researchers Rivest, Shamir and Adleman devised a suitable algorithm called RSA algorithm relying on principle of asymmetric key encryption. Asymmetric key cryptography is the technique in which data is converted into ciphertext using a public key and decrypted by the use of private key. The reasons behind the delay of RSA coming into widespread use were discussed at length.

J. Omura [19] discussed at length the applications of cryptography in the domain of digital communication. Techniques had to be devised for the purpose of ensuring the authenticity and confidentiality in the data exchanged. The numerous ways advised were public-key encryption systems like RSA algorithm, Hash functions, Digital signatures etc. The process of creation and verification of digital signatures was presented and importance of key management was justified.

M. Agrawal [20] discussed the basics in cryptology and cryptanalysis. Several types of cryptanalytic attacks were listed. The issues in designing a cryptosystem were discussed. Private key cryptosystems DES and System IDEA along with Brute-force attack and differential cryptanalytic attacks were discussed. Public key cryptography scheme RSA developed at MIT was discussed. Other applications like authentication, digital signatures and secret sharing were mentioned.

H. M. Hayes [21] discussed the most common and significant attacks applicable on block ciphers-linear and differential cryptanalysis. A basic cipher including a substitution-permutation network was taken as example and its algorithm for encryption was presented in detail. Linear cryptanalysis is defined as the type of known-plaintext attack which tries to take advantage of the linearity present in patterns of bits of plaintext, key and ciphertext. Differential cryptanalysis applied on block ciphers

using plaintext. Thus, intruder has the facility of selecting certain set of plaintexts and then closely examining outputs for purpose of retrieving the key which was involved for encrypting the plaintext.

M. Yang *et al.* [22] described the general theoretical ideas, algorithms, and standards for encryption of data, images and MPEG video like DES, AES, RSA and IDEA. The general model of cryptography system for symmetric key and asymmetric key cryptography were discussed. Comparisons were drawn between the various cryptographic algorithms based on various parameters like complexity, speed, memory requirement, key length, key space size, security level etc. Some other type of algorithms like affine transformation algorithms, chaotic based algorithms, joint compression encryption etc. were discussed.

Colin Tankard [23] discussed the advantage security achieved by encryption and key management for big data sets, which refer to the huge data sets that have large quantities of data gathered, evaluated and communicated between organisations. It is impossible to locate all of the critical information and tracking the users which have the access to sensitive data. All critical information is to be secured involving those of data bases, spread sheet, archives etc. For moving of the data or purposes of cloud storage of the data, it was to be noted that the keys were kept with the organisation for avoiding the unauthorized access to sensitive data.

B. L. Tomhave [24] introduced the basic concepts of key management. It was described that key management lifecycle went through 8 stages: creation, backup, deployment, monitoring, rotation, expiration, archival and destruction. Also, it must be considered for the organizations that require to implement guidelines, specifications, and principles. To cope up with lost keys, it might seem correct to establish the escrow key which could be utilized to restore information during an emergent condition.

J. R. Vacca [25] highlighted the difficulty in expertly positioning the encryption key logics in huge volumes of information. In asymmetric key encryption, it was sufficient to find encryption key, as the hacker could itself develop the requisite plaintext and ciphertext data. Central problem for the application of this logic in RSA technique involved that every exponenting operation was pretty costly. Also, time complexity of the technique increases cubically.

R. A. Gove [26] discussed the basic ideas and principles in the field of cryptography. The detailed theory about block ciphers, stream ciphers, DES were presented. The importance of key management was stressed upon. Also, some cryptanalytic attacks and computational constraints governing them were discussed. The major client issue involves the length of key, efficiency involved in computation and robust application. The issue related to key management was indirectly solved by involving asymmetric key encryption. This system was known to be liable to man-in-middle attack although this could be averted using signatures and hashing.

Madhvan and Saxena [27] discussed a brief history of application of cryptology in International and Indian roots. Classical methods for cryptography like Caeser cipher, mono-alphabetic, poly-alphabetic and polygraphic substitution ciphers and others were mentioned. Private key cryptography involving block, stream and code-book ciphers was detailed. Public key cryptography taking an example of famous RSA algorithm was discussed. Also Speech cryptography, coding, speech secrecy and steganography were mentioned.

D. B. Parker [28] stressed upon the judicious usage of cryptography in various applications. The dangers and vulnerabilities posed to the system by using cryptography were highlighted. Transmission errors, lost keys etc. could lead to critical information loss and harm an organisation greatly. Creating several back-ups also poses the problem of exposure to sensitive information. A cryptographic algorithm with maximum strength should be used for enciphering crucial information as one is always unknown to eaves-dropper's intentions and capabilities.

P. Patil *et al.* [29] discussed the basic standards for symmetric and asymmetric cryptography. They implemented the techniques in Java platform and compared then on the basis of various parameters like encrypting period, decrypting period, storage space required, avalanche effect, randomness and number of bits involved in optimal encoding. The comparison basis for avalanche effect shows that AES exhibits the maximum avalanche effect i.e. the little change in input affects the output greatly.

Nadeem and Javed [30] considered the symmetric data encryption standards like DES, 3DES, AES and Blowfish algorithms for implementation and comparison. The algorithmic basics and specifications for all are discussed. The various standards were implemented on platforms like Java. The different standards were compared on the basis of execution times in different modes and different machines. The performance versus security trade-off was presented. Employing more rounds offered high security but also degraded the performance.

W. J. Buchanan *et al.* [31] researched about the varying servers used across various industry sectors in today's scenario. Computer industries like Apple, Microsoft used highly secured connection but industries like news and sports used the weaker connections. Server should select the strongest possible key exchange, encryption and hashing method from the ones available during client hello. They also discussed and listed the work done by other researchers which highlighted the weakness in certain Transport Layer Security (TLS) protocols. ECDHE-RSAAES256-GCM-SHA384 (Elliptic Diffie-Hellman Key Exchange using RSA, Encryption using AES-256, hashing using SHA384) was described to be the most popular cipher suite.

Daemen and Rijmen [32] were Belgian cryptographers, who presented the key concepts and algorithmic details of Rijndael cipher which was the winner of international contest organised by NIST for the adoption of an algorithm as the Advanced Encryption Standard. The detailed step-by-step explanation of AES along with essential mathematical requisites was presented. The essentials in

differential and linear cryptanalysis were described. Following concepts inspect the design philosophy and cryptanalytic attacks involving Rijndael algorithm and describe the application and developmental issues. Also, other encryption techniques relevant to Rijndael were described.

A. Kakkar *et al.* [33] described that cryptography is an indispensable tool in today's era, but the security of cipher keys degrades over time. A single key provides only limited security. The optimized usage of cipher keys, their length, number and mutual arrangement was what would provide with optimum secured model. At the same time, trade off between security and processing time was still an issue. This was because multiple keys with long bit lengths take longer time to process but provide better security. The cryptographic key pair has to be maintained for the purpose of keeping up with huge processing power feasible in destroying security of keys which should be replaced periodically.

R. R. Rachh *et al.* [34] proposed two efficient implementations in hardware for AES encryption technique. The primary architecture in encrypting was established on S-Box which was optimized using three cascaded blocks. Then the layers of MixColumns and key addition were bit-wise implemented. For decryption purposes, an optimized inverse S-Box was proposed. This was followed by applying layers of Inverse MixColumns and key addition for the purpose of decryption of data. For the second implementation, the third block of the optimized S-Box was combined with the layers of MixColumns and Key Addition which formed an integrated unit for encryption. FPGA implementations of above architectures were described.

Osvik *et al.* [35] analyzed the performance speeds of AES-128 encryption for both high performing and low performing architectures. Low performing architectures included AVR ASICs with data lines of 8 bits and ARM microprocessors 32-bit. High performing architectures included broadband devices and GPUs. For AES-128 implementation on both the architectures, the platform specific approaches were applied to enhance the performing speeds. The first ever approach of performing AES encryption and decryption on a GPU was also presented.

Jingmei *et al.* [36] discussed that the simple S-Box of AES with only 9 terms could pose a vulnerability of attack to AES. To rectify this problem, a new S-Box with 255 terms was presented which follows strict avalanche criterion (SAC). It also increased the security against attacks like linear and differential cryptanalysis. The structure and basic algebraic properties of S-Box were described. The algebraic properties of the improved S-Box were also discussed in detail. Also, it was theoretically analysed that the new proposed approach was immune to differential cryptanalysis.

J. Gong *et al.* [37] discussed the implementation of AES algorithm and S-Box using multiple LUTs. The proposal was to implement the AES using five LUTs developed using S-Box. The pseudo-codes for simple AES encryption and the one using look-up tables were described. The main advantage was to reduce the encryption time. The mathematical preliminaries for generating the look-up tables were

discussed. It was also proposed that idea had increased implementation efficiency and could be applicable easily on processors with word length above 32 bit.

M. Dara *et al.* [38] discussed that substitution boxes (S-Boxes) in Advanced Encryption Standard are the major important elements of symmetric cryptography systems which bring nonlinearity and increase the encryption strength. The S-Box utilized by conventional AES had static nature, if it could be tried to generate the S-Box by dynamic way, strength of AES system against some attacks would be increased. Generation of S-Boxes for AES encryption scheme using RC4 technique was proposed. Tests like Avalanche Effect, Strict avalanche criterion, key sensitivity test were applied and the idea passed all of them.

D. Artz [39] presented the idea of hiding the secure data using the process called steganography. As the obscure text secured using encryption could attract the attention of an attacker, a better idea would be to transmit it in hindsight by concealing it from general eye view. This was done by hiding data file within another file. Basic concepts and notions of steganography and process of data flow using encryption and steganography were discussed.

Wahaballa *et al.* [40] discussed that merely hiding the data using steganography was not fully secure because certain attacks could prove to be fatal to security attempts. Thus, it would prove better to provide multiple layers of security to the data. This can be done by encrypting the data file prior to hiding it using steganography. The different techniques for cryptography and steganography were discussed and comparisons were drawn between them. LSB Steganography and AES was combined.

Musa *et al*. [41] proposed a simplified approach to implement symmetric key encryption using AES. The mathematical basic concepts of finite fields, substitution boxes and keys were presented. Instead of 128 bit plaintext and ciphertext blocks of the conventional AES Scheme, instead 16 bit blocks of plaintext data were employed to simplify the implementation of AES. Instead of bytes, nibbles of bits were substituted using S-Boxes. Only two rounds or iterations were performed in encrypting the data. Also, the classic approach to standard AES is presented and some cryptanalytic attacks were discussed.

S. Simmons [42] presented an approach to apply algebraic cryptanalytic attack on simplified AES algorithm was presented. For this type of cryptanalytic attack, the attacker models out some polynomial equations from the cryptographic systems and attempts to solve those equations. If the equations were solvable, then the attack was considered to be a success. As the linear cryptanalysis aims to find linearities in the system and solve them, which was comparatively easier to do, but non-linear equations in algebraic attack require powerful software to solve them.

Kazlauskas *et al.* [43] discussed that the substitution box was the main element which adds non-linearity to the system in AES. Known S-Boxes as presented conventionally were vulnerable for

linear and differential cryptanalysis. A scheme for generating dynamic key dependent substitution boxes was presented which thwarts off the vulnerability to cryptanalytic attacks. Algorithm for generating key dependent S-Box is presented as pseudo-code. Generated S-Boxes pass the ideal independency measure ration and thus are considered secure.

T. Ao et al. [44] presented an approach to generate good, high quality and more secure S-Boxes. Previous work done in generating key dependent S-Boxes was analyzed and it was noted that previous approaches generated weaker S-Boxes which could pose threats to the security of cryptosystems. Proposed approach generates good quality highly secure affine key dependent S-Boxes. Mathematical concepts for this approach were presented and analyzing criteria of algebraic degree, resistance to algebraic attacks and strict avalanche criterion were discussed.

Singh *et al.* [45] highlighted the inherent weakness of S-Box used in AES encryption scheme due to its static nature because of which it poses the vulnerability to attack. In the paper, a review of the various approaches to generate dynamic S-Boxes was presented. AES encryption and decryption processes were discussed. Also, the essential properties of S-Box like hamming weight, hamming distance, completeness, balancedness, non-linearity, SAC criteria and others were discussed. Various approaches to design dynamic S-Boxes were compared based on these properties.

D. Lambic [46] discussed that the approach to generate dynamic S-Boxes for block based cipher based on tent map had high security problems related to huge amounts in fragile keys, complications in dissolving property and lesser sensitive regarding the variation of data. Other cryptanalytic attacks like weak key attack and attack based on weak dissolving were proposed on that scheme. Also, some improvements in the scheme were suggested that could further improve and strengthen the security of system against previous attacks.

D. Lambic [47] proposed an approach to obtain random substitution boxes using discrete chaotic maps. Mathematical preliminaries and algorithmic implementation for the approach was presented. The generated S-Box was tested based on many criteria like bijection, non-linearity, SAC, output bit independence, equiprobable XOR profile, and maximum expected linear probability. It was observed that it satisfies all the criteria and at the same time a larger key space could be achieved which could give the advantage of generation of larger n x n S-Boxes.

Partheeban *et al.* [48] described that the non-linear and bijective nature of S-Boxes, utilized in substitution of data bytes was a very important step in AES. However, because of unchanging or static nature of S-Box in conventional AES, these were always vulnerable to attacks. The proposed approach tried to design S-Boxes with high non-linearity and low-correlation. A dynamic subkey dependent S-Box was proposed. These dynamic subkeys were derived from the data. Therefore, this approach depended on information or data to achieve better results. Algorithm for approach was discussed and noted that proposed approach had less execution time compared to similar approaches.

A. Bedri [49] discussed the importance of S-Box in AES system design. An approach was proposed to design S-Box based on random selection basis. Algorithmic implementation for the above proposed approach was presented. The resulting S-Box was analyzed based on many criteria like bit independence, non-linearity, information integrity, avalanche criteria and XOR Table. It was observed that the method achieves satisfactory results which are better in comparison to other contemporary approaches.

F. Ozkaynak [50] highlighted the importance of S-Boxes in cryptographic systems. An approach for S-Box with chaotic maps was presented. The approach was applicable to all the chaotic system classes. Details about S-Box used for substitution, relevant efforts accomplished to this area and the properties of the S-Box designed according to the proposed approach were presented. Analysis metrics chosen were bijective property, non-linearity, SAC, bit independence criteria, and XOR profile. It was observed that the proposed method achieved better results of simulation in comparison to other comparable methods.

Guesmi *et al.* [51] described that S-Boxes are a very important step to introduce non-linearity in the implementation of cryptosystems like AES. A scheme in designing S-Boxes using chaos function technique was proposed. The analytical metrics are bit independence criteria, SAC, non-linearity, hamming distance, equiprobable XOR profile etc. were discussed. Stepwise algorithmic implementation of proposed approach was discussed. It was observed that generated S-Boxes using proposed approach satisfied all criteria and achieved high immunity against differential cryptanalysis.

Farah *et al.* [52] reviewed the work done previously in the approach to design high quality S-Boxes. Performance criteria of S-Boxes were discussed in detail. An approach to generate S-Boxes using chaos function and teaching-learning relying optimisation was discussed. Algorithm and flowcharts for the same were discussed in detail. It was observed that implemented technique satisfied all criterias. Also, the S-Boxes generated from proposed approach could be changed to maximise a certain property other than non-linearity, for S-Boxes to attain high cryptographic performance.

Ozkaynak *et al.* [53] highlighted the non-linearity property induced by S-Box in many well known cryptographic systems. An approach for designing S-Boxes relying on fractional-order chaos chen sytem was presented. Work done by various researchers previously in this field and the algorithmic implementation of the approach were discussed in detail. The resultant S-Box was analyzed against various metrics like non-linearity, bijective property, output bits independence criterion, SAC, XOR profile etc. It was observed that the proposed design satisfies all criteria and achieved best XOR profile among all the referenced works.

Roslan *et al.* [54] reviewed the algebraic and heuristic approach in designing S-Boxes for various cryptosystems. Mathematical details and algorithmic specifications of Advanced Encryption Standard were discussed in detail. Algebraic and heuristic approaches to design S-Boxes were discussed at

length in the paper. To compare the security features, metrics were non-linearity, differential uniformity. It was observed that S-Box from heuristic approach was at par with the original one which is utilized to design S-Boxes.

Kazlauskas *et al.* [55] described the importance of S-Boxes as non-linear components in cryptosystems, which provided the strength to sustain the threat of linear and differential cryptanalysis. Following this, various approaches in designing S-Boxes were proposed. The distance metrics to measure the quality of the generated S-Boxes were hamming distance, squared spearman's distance, T-distance, Kendall distance, Correlation Coefficient Distance, Pearson's Correlation Coefficient Distance, Longest Common Subsequence Distance were discussed in detail. It was observed that generated S-Boxes exhibit good randomness property.

Alamsyah *et al.* [56] described the importance of S-Box for substitution of data bytes in AES encryption system. An approach to construct S-Boxes for substitution purposes using elemental polynomial equation with a static 8-bit irreducible polynomial varying from AES, was presented. The analytical metrics like Strict Avalanche Criterion, bit independence criteria, bijectivity i.e. mapping of a symbol to a unique symbol, non-linearity etc. were discussed in detail. Analyzing the results of the simulation, it was observed that the cryptographic strength of the substitution S-Boxes constructed using proposed approach was far better than the other contemporary approaches.

A. Joshi *et al.* [57] proposed the implementation of Substitution box of AES relevant to field arithmetic by deploying combinational logic. This approach overcomes some inherent delays which occur due to ROM based implementation of look-up tables. Basic theoretical concept of AES and composite field arithmetic were discussed at length. Results showed improved implementation of this technique over other techniques. Due to the advantage of pipelining, delays due to look-up tables were avoided. The design was implemented on Xilinx FPGA.

M. Retiner [58] described that the strength against cryptanalytic attacks provided by the non-linearity in AES was majorly due to manipulation of data involving permutations or set of permutations. An approach for a novel permutation generation algorithm was presented. The mathematical theorems and properties to describe the permutation generation algorithm were discussed at length in the paper. It was described that the S-Box resultant from above proposed approach did not require much memory and were inherently key dependent.

S. Farwa *et al.* [59] highlighted the importance of S-Box in cryptosystems. A novel approach to construct S-Box using fractional linear transformation was presented. Efforts of numerous researchers previously in area of constructing S-Boxes was studied at length. Algorithm to generate S-Box using proposed approach was presented using mathematical requisites. The analytical metrics to measure the results were non-linearity, linear approximation expectation, differential approximation, SAC and

independent bits criteria. It was observed that the S-Box generated using proposed approach was at par with S-Box of conventional AES in BIC criteria.

D. Canright [60] presented an approach to introduce a compact form of S-Box for AES depending on sub-field arithmetic as suggested by Rijmen. Subfield arithmetic means that 8- bit calculations are reduced to 4-bit and 2-bit ones. Previous work done in the same area was discussed. In the presented approach, optimization was done to minimize the number of logic gates necessary for implementation. Optimizations were done at both the high and low levels of hierarchy in implementations. The gate operations based on XOR and AND were compactly implemented using XOR and OR resulting in a 20 % reduction in size.

Moradi *et al.* [61] presented an approach for a compact and efficient implementation for AES. The research work done previously to implement low power consuming and area optimised approaches for implementation of AES were presented. Countermeasures to Differential Power Analysis in order to thwart off side channel attacks were described. For hardware implementations, scan flip flops were used to reduce area requirement. The resulting implementation had a 23% smaller size than any other implementation. It was noted that implementation was still susceptible to certain sophisticated attacks.

M. T. Tran *et al.* [62] presented an approach to design gray code based S-Boxes for AES implementations. A pre-processing step in the form of binary gray code transformation was applied to original S-Box implementation. It generated a polynomial with 255 items in comparison to only 9 terms generated by the S-Box of conventional AES Approach. This approach highly increased the security against the linear and differential cryptanalytic attacks. The original S-Box and the S-Box from the proposed approach were compared based on metrics like differential uniformity, non-linearity and Strict Avalanche criteria. It was observed that the increase in the number of polynomial terms due to proposed approach provided better security.

Rebeiro *et al.* [63] proposed an approach to implement AES using Bitslice implementation on three different microprocessors. Algorithm for bitslice implementation and technical details of the microprocessor implementation platforms was discussed in detail. The results were analyzed based on the type of instructions fed to the microprocessor and number of clock cycles a platform consumed in order to encrypt one input cycle. It was observed that different machines differed in performance by changing the type of instructions fed to them.

Oswald *et al.* [64] proposed an approach for designing of S-Box which was resistant to side-channel attacks. This was achieved by changing the basis of mathematical operations in AES from GF(2) to GF(4). Due to this, the inversion operation in AES was converted to a linear one which was easy to mask. The masking also combined the properties of multiplicative and additive masking due to which resistance to side-channel attacks was increased.

A. Satoh *et al.* [65] proposed a high-speed and compact architecture for implementation of Advanced Encryption Standard. Whole of the encrypting process flow in algorithm was discussed. Assets available were shared among encrypting and decrypting procedures for the purpose of optimum utilization and for minimizing space requirements. Using composite field arithmetic, GF(2) calculations were converted to GF(4), which led to simpler S-Box implementation. Combining the above approaches, a 25% reduction in size as compared to the conventional AES was observed.

D. Canright *et al.* [66] proposed to reduce the vulnerability to side channel attacks by adding random masks to the data being encrypted using AES S-Boxes. The inversion properties with and without using masking are discussed and compared. Masking induced randomness into calculations necessary for side-channel attacks. Gate-level and logic-level optimizations were done wherever necessary, to achieve the smallest masked S-Box. Implementing this countermeasure gave high degree of strength to avoid differential side-channel attacks and also minimized the area requirement.

O. Billet *et al.* [67] proposed an efficient cryptanalytic attack on white box based AES implementation. White box implementation refered to an environment in which the implementation of cryptographic algorithm, keys and other critical details were exposed to an unsafe environment. This presented an analogue to the encrypting atmosphere present today where accessing the system offering encryption by any random user was very easy. The analysis of look-up tables used for S-Boxes in AES offer useful insights about the logic of programming and hence the attack became easier. Using the above approach, whole of the secret key used in encryption could be extracted.

N. Ahmed *et al.* [68] described a scheme for optimizing the performance of AES using combinational circuit logic design. The main delay occured in calculation and substitution due to S-Box. Logic optimizations based on truth table were involved. The results indicated an efficient area consumption using the proposed approach. Also, the delay involved in the design of S-Box using the proposed approach was found to be 73% lesser than the other conventional approaches and that of using composite field arithmetic.

Murphy and Robshaw [69] proposed a new cipher BES (Big Encryption System) which offered the advantage of simple operations in $GF(2^8)$, over the conventional AES where conflict in operations arised due to $GF(2^8)$ and GF(2). The mathematical preliminaries related to both AES and BES were discussed at length. The round function for BES included simple Algebraic operations although the security features of AES may not be fully attained by this.

Hossienkhani and Javadi [70] proposed an approach to construct the S-Box in AES dynamically using the cipher key, for increasing the encryption strength and thwart off the vulnerability to linear and differential cryptanalysis posed due to static S-Boxes. AES algorithm was presented in detail with special attention to S-Box. The detailed algorithm for the proposed algorithm was also discussed at

length. The results exhibit better BIC criteria and better implementation which could be achieved due to dynamic S-Box implementation as compared to the conventional approach.

2.2 OBSERVATIONS

From the literature review, following observations have been drawn for data encryption and improvement in encryption techniques:

- The symmetric key encryption schemes are widely utilized for data encryption purposes while the asymmetric key encryption schemes are often utilized for key management.
- Advanced Encryption Standard (AES), standardized in 2001, has been deployed widely to encrypt the data.
- Despite its wide use, AES is under scrutiny due to the static nature of its S-Boxes. Therefore, most of the work reviewed focussed on the generation of better S-Boxes.
- The increase in key space for an encryption algorithm results in better security as attacker has to try huge number of combinations and colossal efforts in guessing or approximating the key.
- Lower key space can also lead to increased vulnerability to attacks.
- Static nature of S-Boxes and keys used in encryption makes the encryption scheme highly vulnerable to cryptanalytic attacks.

2.3 MOTIVATION AND PROBLEM FORMULATION

With the help of the observations it has been found that AES is in wide use across the industries to encrypt huge amounts of data. The static nature of S-Boxes in conventional AES technique makes it highly vulnerable to linear and differential cryptanalysis. Also, the static nature of keys utilized for encryption in AES can also lead to certain side-channel attacks. These types of issues point to the fact that there is a need to introduce dynamic effect and randomness in the conventional nature of AES. There must be dynamic S-Boxes and some randomness in keys in order to avoid certain attacks. This will increase the security of AES against attacks like linear, differential cryptanalysis and side-channel schemes. It must also be kept in mind that introducing dynamism and randomness to the AES must not increase the computational complexity by a huge factor.

2.4 OBJECTIVES

From observations, following objectives have been drawn:

- To analyze the algorithmic implementation of Advanced Encryption Standard (AES) in detail.
- To develop a scheme to generate dynamic S-Boxes in place of the static ones as in the conventional AES approach. This S-Box must satisfy certain evaluation criteria to test the quality of S-Boxes.
- A scheme to increase the key size must be introduced as it will require greater effort on the part of hacker in order to breach the security.

- Some randomness must be included in the ways that keys are included in the system, for avoiding the side-channel attacks.
- Comparisons must be made between the execution complexity and quality of the proposed schemes with the conventional approach of encryption in AES.

CHAPTER 3

ADVANCED ENCRYPTION STANDARD (AES)

In this chapter, the history, mathematical analysis and algorithmic implementation of AES (Advanced Encryption Standard) are presented.

3.1 HISTORY OF AES

As the computational power and performance capacity of machines increased over the years, DES became susceptible to different cryptanalytic attacks and was withdrawn as an encryption standard. In 1997, United States Institute of Standards and Technology invited proposals for the new encryption standard AES [71]. All the proposals had to follow the following the necessary requisites and standards for all AES applicants:

- cipher which encrypts data blocks with block length 128-bits
- it must support keys with lengths: 128, 192, 256 bit
- cipher should be more secure compared to other algorithms
- efficient implementation and performance in software and hardware

In 1999, five proposals, out of the fifteen submitted, were further shortlisted for performance analysis. The five selected finalists were:

- MARS
- RC6
- Rijndael Cipher
- Serpent
- Twofish

In 2000, NIST was announced that Rijndael algorithm was selected as AES. It was properly selected as US federal standard in 2001. This got presented as FIPS 197 in 2001. Advanced Encryption Standard is derived from Rijndael Cipher, which was designed by two Belgian cryptographers - Jon Daemen and Vincent Rijmen. In original form, Rijndael technique involves schemes with varying key and data lengths [71].

3.2 DESCRIPTION OF THE CIPHER

AES is a symmetric/private encryption scheme i.e. this employs similar cipher key both for encrypting and decrypting purposes. It encrypts blocks of 16 bytes i.e. 128 bits in single iteration. It can use keys with different bit lengths- 128, 192, 256. Therefore, depending upon length of key, different versions of AES are present. Depending on the key size, the number of round functions are 10, 12 and 14 respectively [71]. Basic parameters for AES are described as follows.

Figure 3.1 describes the basic input and output parameters for AES encryption.

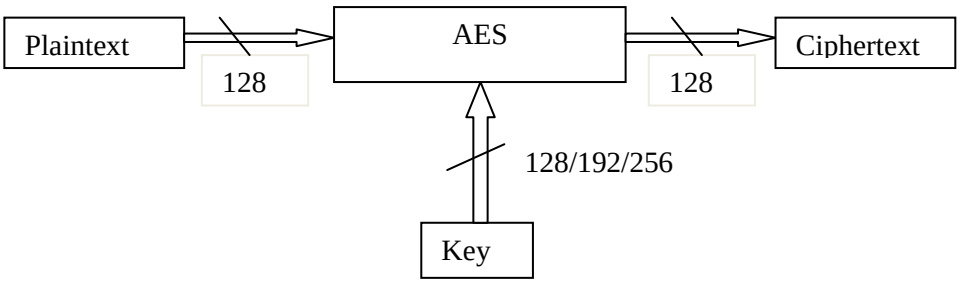

Figure 3.1 AES input/output parameters

Table 3.1 lists the lengths of keys and related round functions for encryption in AES.

Key Length (bits)	**Number of Round Functions**
128	10
192	12
256	14

Table 3.1 Key Lengths and Round Functions

AES is a not a feistel scheme. Said schemes don't process a full block of data in one go. For example, in Data Encryption Standard, 32 bits of data get encrypted for one turn. Thus, DES has a feistel structure. AES is non-feistel in the sense that it processes a full single block of 128 bits in single iteration [73].

3.3 BYTE

The byte is the main entity of data which is manipulated in AES. A grouping in 8 bit is considered to be the basic unit. Plaintext, Cipher key and Ciphertext are dealt with as a collection of bytes [71].

Block length= 128 bits (16 byte x 8)

Key Size=128 bits(16 byte x 8)

Key Size=192 bits(24 byte x 8)

Key Size=256 bits(32 byte x 8)

Here, the discussion is limited to AES-128 only as the proposed technique attempts at modifying it to increase its security.

Bytes in the AES design are symbolized as the grouping of its constituent bits 0 or 1 as a set in the order $\{g_7, g_6, g_5, g_4, g_3, g_2, g_1, g_0\}$. The individual bytes are represented as elements which belong to a finite field with representation as follows in equation 3.1:

$$g_7x^7 + g_6x^6 + g_5x^5 + g_4x^4 + g_3x^3 + g_2x^2 + g_1x + g_0 = \sum_{i=0}^{7} g_i\ x^i \qquad (3.1)$$

Consider byte denoted as {11100000} is equal to the specific finite field element $x^7+x^6+x^5$ [9]. Polynomial Arithmetic is used to operate upon bytes.

Byte values can also be specified using hexadecimal notation as described in table 3.2.

Pattern	Representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Pattern	Representation
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Table 3.2 Representation of various patterns of bits in Hex form

3.4 STATE OF THE ALGORITHM

The different manipulations of AES scheme are operated upon a 4x4 matrix where each element represents a byte. This 4x4 array is called state of the algorithm. Initially, all the data bytes are input into the state matrix as shown in table 3.3.

X0	X4	X8	X12
X1	X5	X9	X13
X2	X6	X10	X14
X3	X7	X11	X15

Table 3.3 Arrangement of elements in State Matrix

Constituent bytes in every column of the array form 32 bit words. Therefore, state is signified as 1x4 matrix of four 32-bit words [71].

3.5 GALOIS FIELD ARITHMETIC

Galois field refers to a set with finite number of elements. The operations like the addition, subtraction, multiplication and inversion are applicable to the elements of Galois fields. Smallest Galois field is a field with only two of the elements i.e. 0 and 1. This is regarded as GF(2). The operations of addition and multiplication in GF(2) are equivalent to logical XOR and logical AND [72]. For the purpose of encryption in AES, a field having 256 elements denoted as $GF(2^8)$ was chosen. This was done because each element in this field is represented by 8 bits and 8 bits together constitute one byte. Byte is therefore the basic entity of operations in the manipulation of data through AES. Since this field is derived from the basic Galois field with two elements, therefore all the operations in AES follow the basic rules of GF(2). Consider a field having 5 elements, denoted as GF(5)={0,1,2,3,4}. The operations of addition, multiplication and inversion are described in figure 3.2 [73].

Addition in GF(5)

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

Multiplication in GF(5)

x	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Additive Inverse

-0 = 0

-1 = 4

-2 = 3

-3 = 2

-4 = 1

Multiplicative Inverse

0^{-1} doesn't exist

1^{-1} = 1

2^{-1} = 3

3^{-1} = 2

4^{-1} = 4

Figure 3.2 Arithmetic operations in GF(5)

For GF(2^n) fields, components are represented as polynomials having coefficients of GF(2), which have the highest degree of n−1. This refers to total n number of coefficients for each component. For GF(2^8), selected as the GF for AES, every component G ϵ GF(2^8) therefore denoted like:

$$G(x)= g_7 x^7 + g_6 x^6 + ...+ g_1 x + g_0 , g_i \epsilon GF(2)=\{0,1\} \quad (3.2)$$

For GF(2^8), there are 256 polynomials. The coefficients of these lie in GF(2). Each polynomial in GF(2^8) can be saved in memory like group of 8 bits or a vector G= (g_7, g_6, g_5, g_4, g_3, g_2, g_1, g_0).

3.5.1 Addition and Subtraction

The sum or result of adding the two components of galois field is obtained through addition corresponding coefficients of polynomials in two of the components. The addition being accomplished using the XOR (denoted by ⊕). Subtracting the polynomials is equivalent to the addition of polynomials in galois field [71, 72].

Consider the bytes {x_7 x_6 x_5 x_4 x_3 x_2 x_1 x_0} and {y_7 y_6 y_5 y_4 y_3 y_2 y_1 y_0}, result of addition being {z_7 z_6 z_5 z_4 z_3 z_2 z_1 z_0}, where each $z_i = x_i \oplus y_i$.

Following above discussions, following terms are analogous:

$(a^6 + a^4 + a^2 + a + 1) + (a^7 + a + 1) = a^7 + a^6 + a^4 + a^2$ (In polynomial form)

{01010111} ⊕ {10000011} = {11010100} (In binary form)

{57} ⊕ {83} = {d4} (In hexadecimal form)

Subtraction gives the same result as for sum because in GF(2), the subtraction has the same result as addition. Therefore, the result in addition and subtraction are same [71].

3.5.2 Multiplication

In GF(2^n), multiplication being done as follows. Firstly, the components from the field are multiplied using basic multiplication principles. Generally, the resulting product would have degree greater than n-1, therefore it should be reduced. Result of multiplication is divided using specific element. Only the remainder is considered after performing the division [71, 72]. Certain polynomials which are irreducible are required for this kind of operation. Irreducible polynomials have their factors as only 1 and the polynomial itself. For AES, the irreducible polynomial used is as follows:

$$G(g) = g^8 + g^4 + g^3 + g + 1 \quad (3.3)$$

This is specified in AES. Following discussion explains the process of polynomial multiplication [71].

X={57} (Hexadecimal Notation) Y ={83} (Hexadecimal Notation)

X={01010111} (Binary Notation) Y={10000011}(Binary Notation)

$X(g)=\{g^6 + g^4 + g^2 + g + 1\}$ (Polynomial Notation)

$Y(g) =\{g^7 + g + 1\}$ (Polynomial Notation)

The product is computed as follows.

$$Z(g) = X(g) \bullet Y(g) = \{g^6 + g^4 + g^2 + g + 1\} \bullet \{g^7 + g + 1\} = g^{13} + g^7 + g^6 + g^{11} + g^5 + g^4 + g^9 + g^3 + g^2 + g^8 + g^2 + g + g^7 + g + 1$$

$$= g^{13} + g^{11} + g^9 + g^8 + g^6 + g^5 + g^4 + g^3 + 1$$

Continuing,

$(g^{13} + g^{11} + g^9 + g^8 + g^6 + g^5 + g^4 + g^3 + 1)$ modulo $(g^8 + g^4 + g^3 + g + 1) = g^7 + g^6 + 1$

$Z(g) =\{ g^7 + g^6 + 1\}$ (Polynomial Notation)

Z= {11000001} (Binary Notation)

Z= {c1} (Hexadecimal Notation)

Thus, {57}•{83} = {c1}

3.5.3 Inversion

For a galois field expressed as GF(2^n) and the polynomial used as irreducible polynomial G(g), X^{-1} belonging to the component $X \in GF(2^n)$ being described as: $X^{-1}(g) \cdot X(g) = 1 \bmod G(g)$. In smaller ones i.e. field with lesser elements, LUTs containing pre-computed inverse components for every component in the field are often used [74]. For 0, an inverse does not exist. However, cryptographers have mapped input 0 to output 0. Fields like 2^{16} or lower have pre-defined tables to compute inverse values of corresponding bytes.

3.6 INTERNAL STRUCTURE OF AES ENCRYPTION

AES consists of layers which manipulate the 128-bit block of data. These layers are briefly described as follows. Figure 3.3 describes the basic block diagram for AES-128 encryption .

- Key Addition(AddKey): The subkey of length 128 bits, that is sequentially extracted out of original key using key schedule, and the state matrix are added.
- Byte Substitution(SubByte): Every component in state array is altered in a non-linear fashion with the help of tables called substitution boxes or S-Boxes [73].
- Diffusion layer**:** This layer helps in diffusing the information of one bit on all other bits. It comprises of two sublayers. ShiftRows sublayer operates upon the data in byte form. The MixColumn sublayer helps to mix the columns in state matrix each of which has four bytes [74].

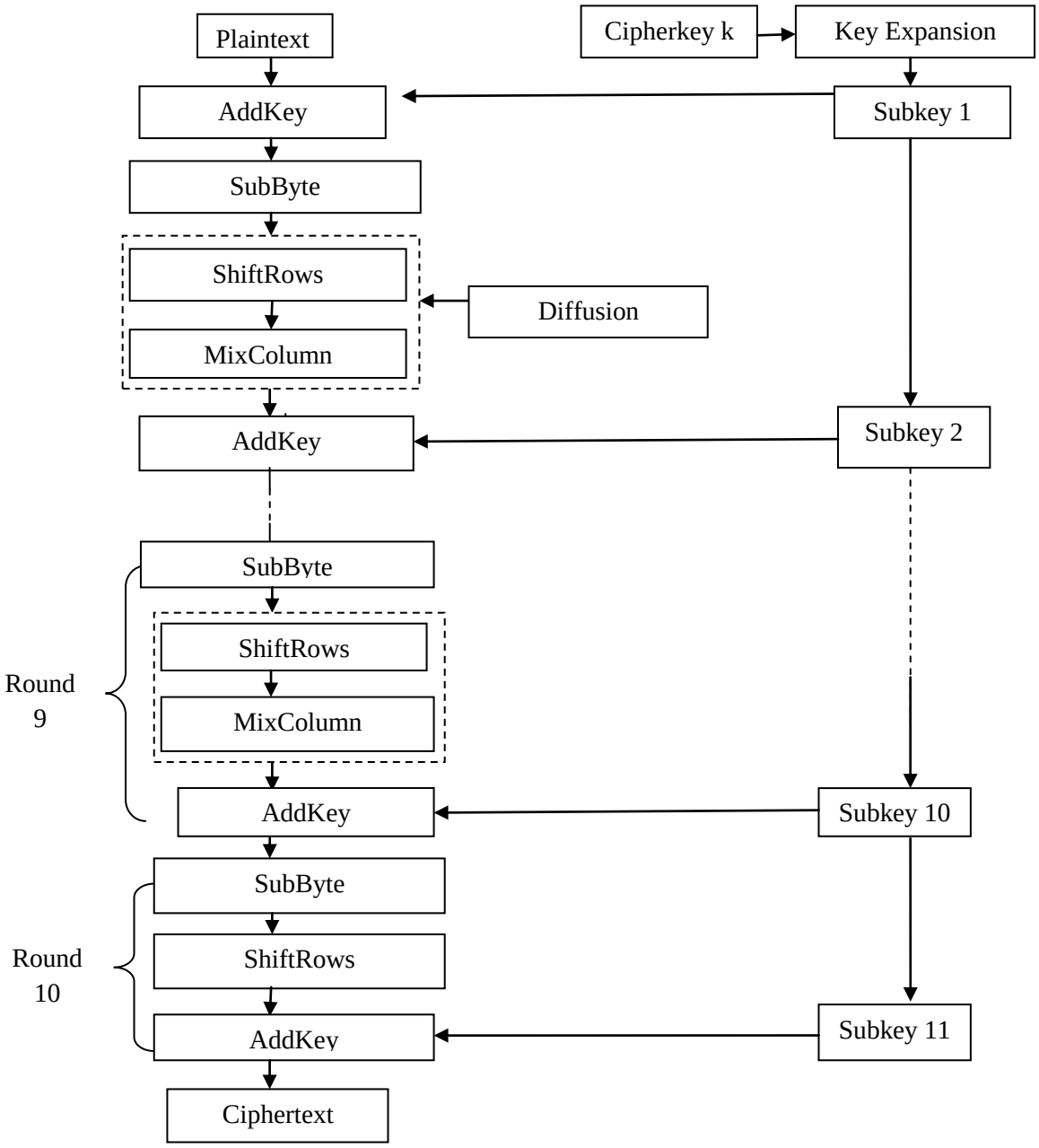

Figure 3.3 Block Diagram for AES-128 encryption

Figure 3.4 describes operations in an AES round. The value of individual bytes in the data keep on changing as the data passes through the various layers in AES algorithm. It can be noted that AES is a byte-oriented cipher [73].

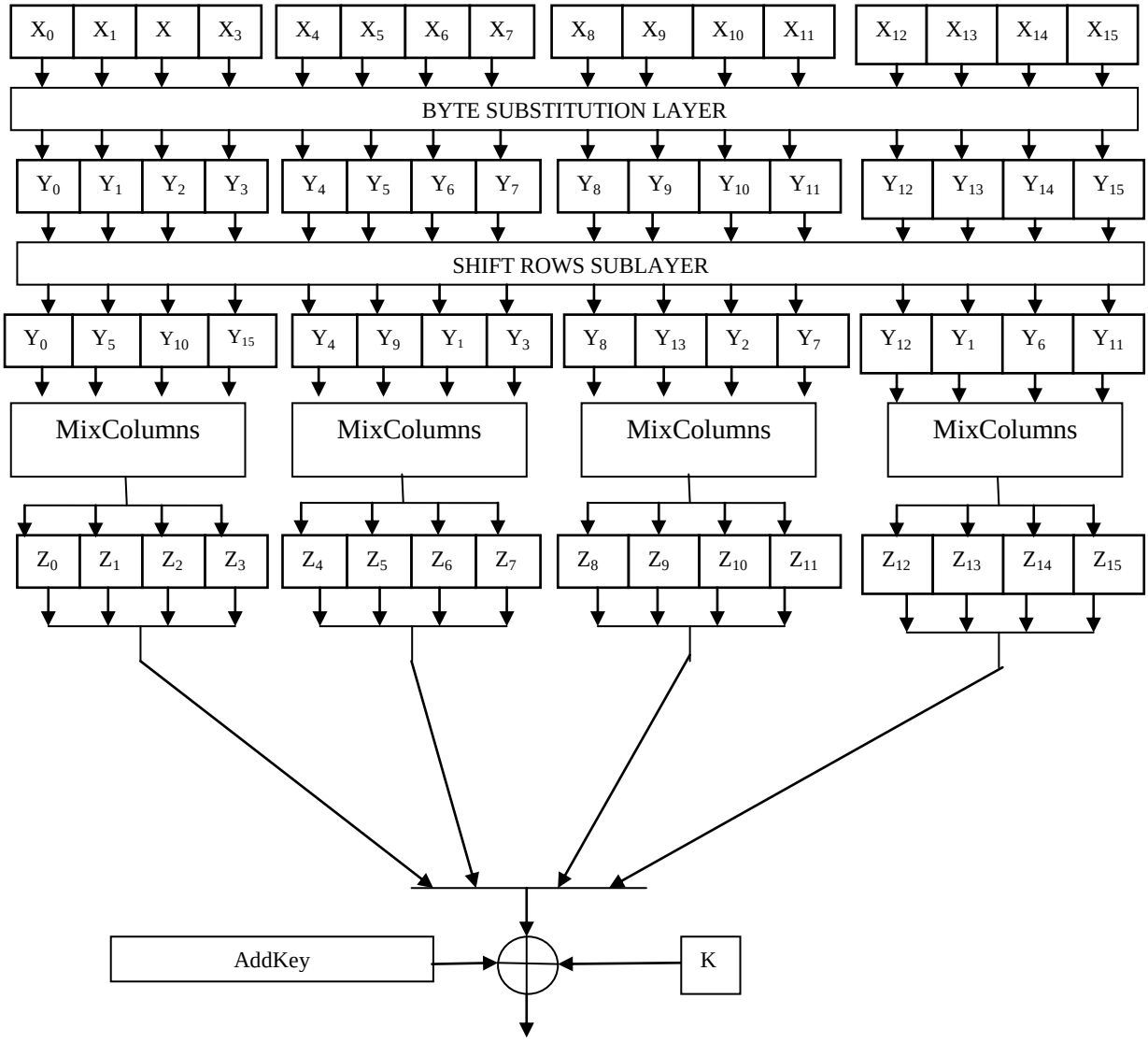

Figure 3.4 Structure of a single AES Round

Consider state X that has 16 bytes X_0, X_1, . . . ,X_{15} which arranged as 4x4 array. AES manipulates the components in columns and rows of current state array. In a similar way, bytes of key Y_0, Y_1, ... , Y_{15} which arranged as 4x4 array as shown in figure 3.5.

X_0	X_4	X_8	X_{12}
X_1	X_5	X_9	X_{13}
X_2	X_6	X_{10}	X_{14}
X_3	X_7	X_{11}	X_{15}

Y_0	Y_4	Y_8	Y_{12}
Y_1	Y_5	Y_9	Y_{13}
Y_2	Y_6	Y_{10}	Y_{14}
Y_3	Y_7	Y_{11}	Y_{15}

Figure 3.5 Data bytes arranged in State matrix and Key bytes in a 4x4 matrix

The following discussion lists in detail the manipulations performed on the data during AES Encryption.

3.6.1 Byte Substitution

The initial operation in every round is byte substitution layer. It operates as a non-linear fashion independently on every element of the State matrix utilizing Substitution-box. Every element of the state matrix substituted by a component from the substitution-box. This is the only step or manipulation which is non-linear in AES [73]. This means that, for two states X and Y, SubByte(A+B) != SubByte(A) + SubByte(B). Each element of the state array is uniquely charted to single element in substitution box. S-Box for AES is a 16x16 table. The S-Box is generated by following transformation in equation 3.4.

$$\begin{bmatrix} x0' \\ x1' \\ x2' \\ x3' \\ x4' \\ x5' \\ x6' \\ x7' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x0 \\ x1 \\ x2 \\ x3 \\ x4 \\ x5 \\ x6 \\ x7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (3.4)$$

Figure 3.6 depicts the S-Box used for encryption.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	99	124	119	123	242	107	111	197	48	1	103	43	254	215	171	118
1	202	130	201	125	250	89	71	240	173	212	162	175	156	164	114	192
2	183	253	147	38	54	63	247	204	52	165	229	241	113	216	49	21
3	4	199	35	195	24	150	5	154	07	18	128	226	235	39	178	117
4	9	131	44	26	27	110	90	160	82	59	214	179	41	227	47	132
5	83	209	0	237	32	252	177	91	106	203	190	57	74	76	88	207
6	208	239	170	251	67	77	51	133	69	249	2	127	80	60	159	168
7	81	163	64	143	146	157	56	245	188	182	218	33	16	255	243	210
8	205	12	19	236	95	151	68	23	196	167	126	61	100	93	25	115
9	96	129	79	220	34	42	144	136	70	238	184	20	222	94	11	219
A	224	50	58	10	73	6	36	92	194	211	172	98	145	149	228	121
B	231	200	55	109	141	213	78	169	108	86	244	234	101	122	174	8
C	186	120	37	46	28	166	180	198	232	221	116	31	75	189	139	138
D	112	62	181	102	72	3	246	14	97	53	87	185	134	193	29	158
E	225	248	152	17	105	217	142	148	155	30	135	233	206	85	40	223
F	140	161	137	13	191	230	66	104	65	153	45	15	176	84	187	22

Figure 3.6 Substitution Box used for Encryption

3.6.2 Diffusion Layer

Diffusion is regarded as spreading the effect of each singular bit on the full matrix. This layer comprises two sublayers- ShiftRows and MixColumns sublayers. These are described as follows.

Shift Rows Sublayer

This operation performs shifting on the second row in state matrix cyclically by single position in left, left shift by two position the third row and three positions to the left for the fourth row. No change done to the first row. The main motive behind the operation of ShiftRows is to diffuse each element over the whole matrix [73]. Let us assume that the input for this operation is state matrix S = (S_0, S_1, . . . , S_{15}) :

S_0	S_4	S_8	S_{12}
S_1	S_5	S_9	S_{13}
S_2	S_6	S_{10}	S_{14}
S_3	S_7	S_{11}	S_{15}

The result of the operation as a new state matrix is shown below:

S_0	S_4	S_8	S_{12}	← Elements not shifted
S_5	S_9	S_{13}	S_1	← Cyclic left Shift with 1 byte
S_{10}	S_{14}	S_2	S_6	← Cyclic left shift with 2 bytes
S_{15}	S_3	S_7	S_{11}	← Cyclic left shift with 3 bytes

Mix Columns Sublayer

This operation is a linear operation which operates by mixing every column in the state matrix. MixColumn affects the AES in such a way that every input element changes the properties of four other bytes, therefore, most of the diffusion in AES occurs by this step. A column in state matrix considered for vector which has four bytes. Multiplication is performed on this vector using a static 4-by-4 matrix. GF(2^8) influences the rules followed in every subsequent operation [74]. For example, consider the calculation of first four output bytes as follows in equation 3.5.

$$\begin{bmatrix} X0 \\ X1 \\ X2 \\ X3 \end{bmatrix} = \begin{bmatrix} 02\ 03\ 01\ 01 \\ 01\ 02\ 03\ 01 \\ 01\ 01\ 02\ 03 \\ 03\ 01\ 01\ 02 \end{bmatrix} \begin{bmatrix} Y0 \\ Y5 \\ Y10 \\ Y15 \end{bmatrix} \tag{3.5}$$

The next four bytes for the next column can be calculated using similar matrix given above.

3.6.3 Key Addition Layer

This layer in AES has two inputs – the state matrix which has 16 elements or 16 bytes, and the cipher key which also has byte length of 16 bytes. The inputs are added, which is equal to a logical XOR in Galois field having two elements. The subkeys for subsequent rounds are extracted by iterating the same operations in key schedule [73]. In AES-128, for 10 rounds, 11 subkeys are needed.

3.6.4 Key Schedule for AES-128

Key Schedule extracts the subkeys from cipher key. The number of subkeys required are equal to one more than the total number of rounds. Therefore, in case of AES-128, the required number of are 11 since there are total 10 rounds. The subkeys are derived recursively in AES i.e. in order to extract 2^{nd} key, 1^{st} key must be known and for 3^{rd} subkey, 2^{nd} subkey must be known and so on [73]. Key Schedule for AES uses words to derive subkeys from the cipher key. Here in AES, 1 word = 32 bits. An array Z, the key expansion array, is made to hold all the subkeys. The 11 subkeys saved in the key expansion matrix Z as Z[0], ...,Z[43]. The key schedule to compute subkeys for AES-128 is depicted as in figure 3.7.

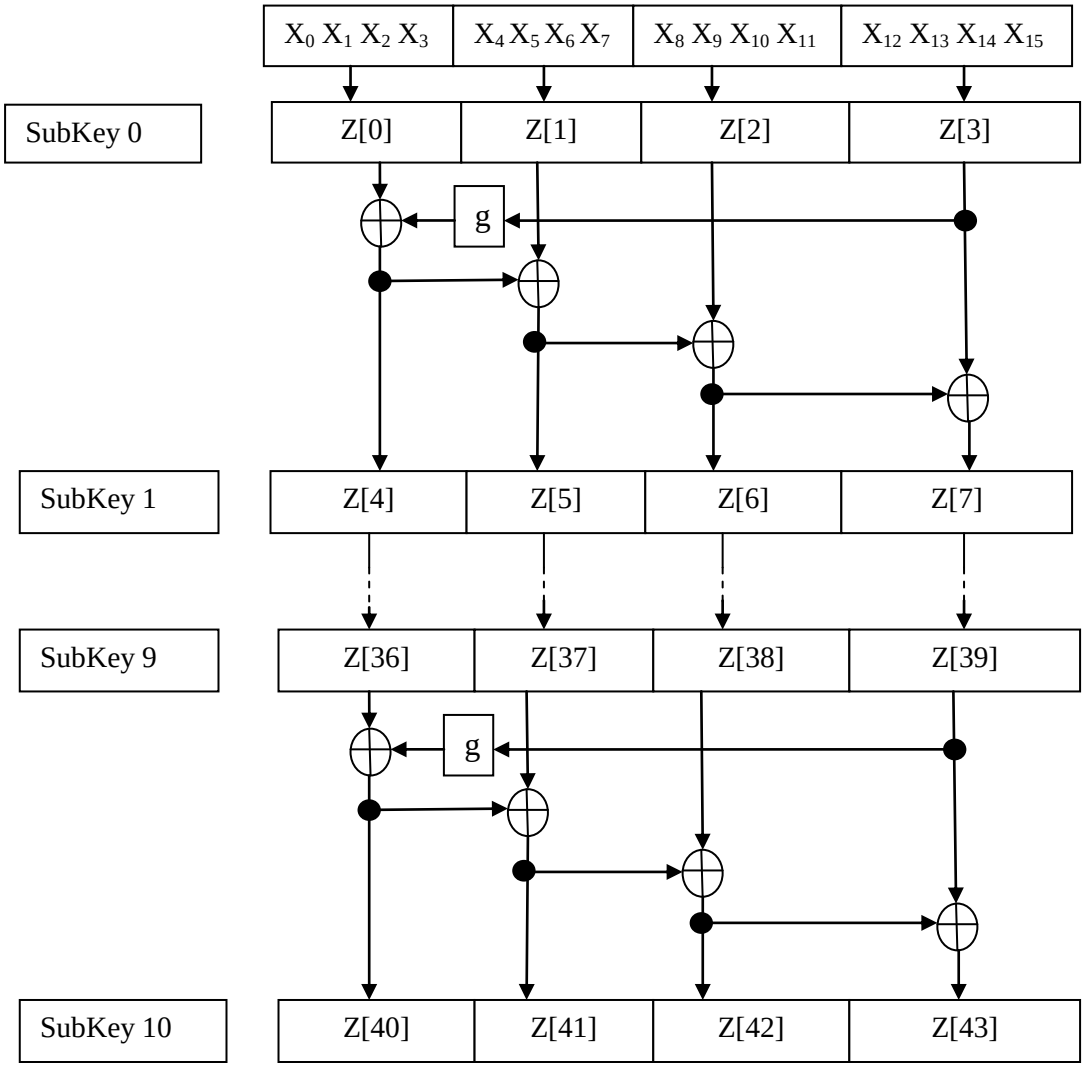

Figure 3.7 Key Schedule for AES-128

From the figure 3.7, it is noted that initial session sub key is the original cipher key and this is duplicated in the first four components of the key expansion array Z. Each element of the array Z is made of words i.e. the combination of four bytes [73]. Also, as it is observed from the figure 3.7, the leftmost words of other subkeys are calculated using the formula in equation 3.6:

$$Z[4i] = Z[4(i-1)] + g(Z[4i-1]) \quad (3.6)$$

The remaining all components of subkey are calculated recursively utilizing the following formula:

$$Z[4i+j] = Z[4i+j-1] + Z[4(i-1)+j] \quad (3.7)$$

Here i=1, 2,....., 10 and j=1,2,3 [73].

Function g() operates in way which is not linear as it takes four bytes as input, gives a same number of bytes as output. This is depicted as follows in figure 3.8.

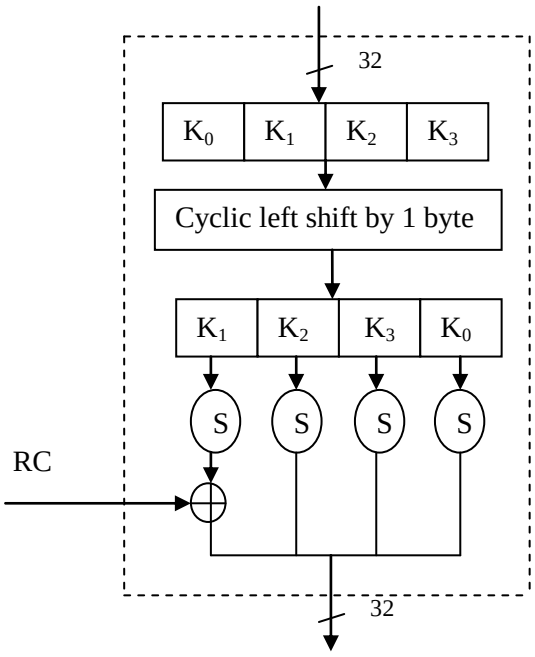

Figure 3.8 Constituent operations in non-linear function g()

Function g() first gives a cyclic left rotation by 1 position to its input bytes. Next it substitutes all its constituent bytes using the elements of S-Box. It then performs addition of round constant RC with leftmost byte of the input. Round coefficients for AES-128 is an element of Galois field GF(2^8). These form a 10x4 matrix with [01, 02, 04, 08, 10, 20, 40, 80, 1B, 36] as the first column. Rest all the entries in matrix are zero. Different round coefficients are calculated for different key addition rounds using the equations 3.8 and 3.9:

$$RC(1) = 0x01 \quad (3.8)$$

$$RC(j) = (0x02) \times RC(j-1) \quad (3.9)$$

The non-linear operation g() serves two goals [14]:

- Addition of non-linearity in key schedule.
- Removal of symmetry from the encryption technique.

The above goals serve important functions because they are really necessary in order to make AES resistant to certain cryptanalytic attacks like linear and differential cryptanalysis and many attacks discussed previously. For key length of 192 and 256 bits, key schedule is more complex. For 192 bit key, above procedure is followed twice, and for 256-bit key, the procedure is followed thrice. Pipelining could be done to expand the key as and when required while performing the encryption on the data. On the other side, for the decryption, it would require advance calculation of the keys.

3.7 AES DECRYPTION

For the purpose of decryption in AES, all the rounds of encryption should be inverted. An inverse S-Box is used to perform inverse replacement. InvShiftRows and InvMixColumns are the other two operations to remove the diffusion from AES. The block diagram depicting AES-128 decryption is depicted in figure 3.9 [73].

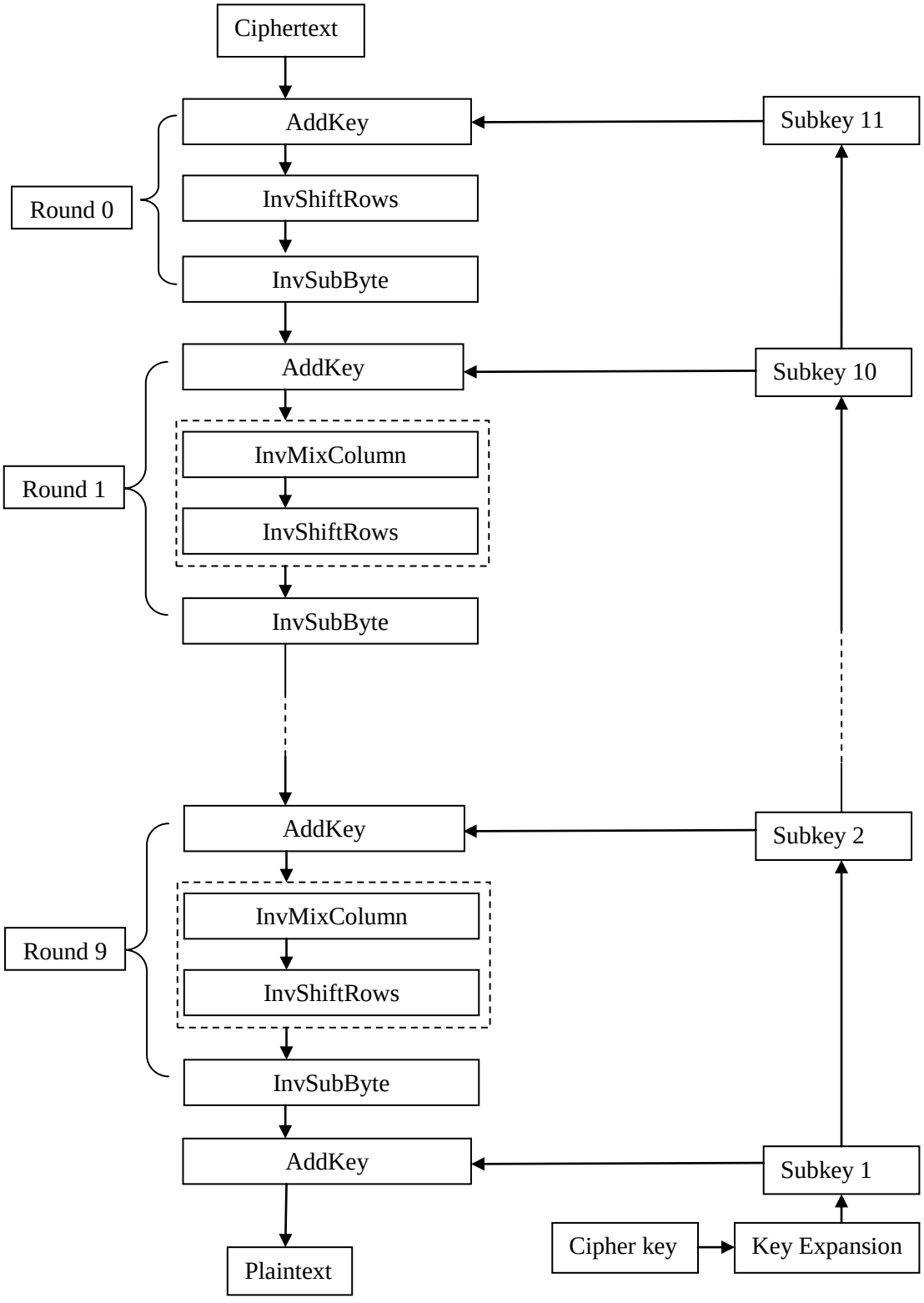

Figure 3.9 Block diagram for AES-128 Decryption

The number of rounds in AES decryption remain the same as in encryption, i.e. 10. For 10 rounds, 11 subkeys are needed. Only the order of subkeys are inverted. First round should have last round key, second round should have last-but-one round key. It is done by calculating the all subkeys initially and then storing them and retrieving as and when required. This aspect leads to a small delay in performing the decryption. Because the last round of encryption didn't have the MixColumn function, therefore, first round in decryption doesn't have the inverse operation. Every other decryption rotation, contain all AES layers as described previously.

3.7.1 Inverse Mix Columns Sublayer

For inversing effect of MixColumn function, inverse of the array utilized in MixColumn has to be used. The input to this sublayer is a 4x1 matrix having 4 bytes that are multiplied with a 4x4 inverse matrix. Multiplication and addition is done according to the rules of GF(2^8).

$$\begin{bmatrix} X0 \\ X1 \\ X2 \\ X3 \end{bmatrix} = \begin{bmatrix} 0E\ 0B\ 0D\ 09 \\ 09\ 0E\ 0B\ 0D \\ 0D\ 09\ 0E\ 0B \\ 0B\ 0D\ 09\ 0E \end{bmatrix} \begin{bmatrix} C0 \\ C1 \\ C2 \\ C3 \end{bmatrix} \tag{3.10}$$

Bytes of second column [X4, X5, X6, X7] are calculated using multiplying next set of bytes by the same inverse matrix [73].

3.7.2 Inverse Shift Rows Sublayer

For inversing the effect of ShiftRows function on state matrix from the encryption algorithm, the rows of state matrix must be shifted in an opposite direction for the decryption purpose. It is done with keeping first row in state array unchanged, second row is shifted with one byte in right, third is shifted with two bytes to in right, shifting fourth row with three bytes in right. Let us assume that the state matrix S = (S_0, S_1, . . . , S_{15}) [73]:

S_0	S_4	S_8	S_{12}
S_1	S_5	S_9	S_{13}
S_2	S_6	S_{10}	S_{14}
S_3	S_7	S_{11}	S_{15}

The resultant state array after the inverse shift rows operation is as follows.

S_0	S_4	S_8	S_{12}	← Elements not shifted
S_{13}	S_1	S_5	S_9	← Cyclic right shift with 1 byte
S_{10}	S_{14}	S_2	S_6	← Cyclic right shift with 2 byte
S_7	S_{11}	S_{15}	S_3	← Cyclic right shift with 3 byte

The inverse of the matrix used in MixColumns, is utilized in InvMixCoulumns. Similarly, for InvShiftRows, inverse operation of that of ShiftRows is applied.

3.7.3 Inverse Byte Substitution

Substitution box in inverse utilized for decryption of the text. Similar as S-Box in encryption, there is a one-to-one mapping between the elements of the inverse substitution box and the state matrix bytes. Figure 3.10 depicts the inverse S-Box in decryption.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	82	9	106	213	48	54	165	56	191	64	163	158	129	243	215	251
1	124	227	57	130	155	47	25	135	52	142	67	68	196	222	233	203
2	84	123	148	50	166	194	35	61	238	76	149	11	66	250	195	78
3	8	46	161	102	40	217	36	178	118	91	162	73	109	139	209	37
4	114	248	246	100	134	104	152	22	212	164	92	204	93	101	182	146
5	108	112	72	80	253	237	185	218	94	21	70	87	167	141	157	132
6	144	216	171	0	140	188	211	10	247	228	88	5	184	179	69	6
7	208	44	30	143	202	63	15	2	93	175	189	3	1	19	138	107
8	58	145	17	65	79	103	220	234	151	242	207	206	240	180	230	115
9	150	172	116	34	231	173	53	133	226	249	55	232	28	117	223	110
A	71	241	26	113	29	41	197	137	111	183	98	14	170	24	190	27
B	252	86	62	75	198	210	121	32	154	219	192	254	120	205	90	244
C	31	221	168	51	136	7	199	49	177	18	16	89	39	128	236	95
D	96	81	127	169	25	181	74	13	45	229	122	159	147	201	156	239
E	160	224	59	77	174	42	245	176	200	235	187	60	131	83	153	97
F	23	43	4	126	186	119	214	38	25	105	20	99	85	33	1	125

Figure 3.10 Substitution Table used in AES Decryption

3.7.4 Key Schedule for AES-128 Decryption

Decryption key schedule for AES requires that all the subkeys have to be added in a reverse order of that of encryption. Therefore, first round requires the last round key, next round requires last-but-one round key. Because of this, it has to calculate all the subkeys in advance before performing the actual decryption. This pre-requisite adds a little delay to the decryption procedure.

CHAPTER 4

PROPOSED METHODOLOGY

After reviewing the various improvements in encryption approach presented in the literature, there is a requirement to develop a technique which ensures almost perfect security to the data meant to be encrypted and communicated over the secure channels. Proposed scheme ensures that encryption algorithm is safe from different cryptanalytic attacks for example, linear cryptanalysis, differential cryptanalysis and side-channel attacks etc. The proposed scheme ensures that the maximum efforts and almost infinitely high computational power is required at the end of the attacker in order to make an attempt to breach the data security. The static key in the algorithm provides one layer of security. Use of a random key along with key dependent S-Box is what gives the algorithm its best security feature. Random key is changed with every session which provides us with the advantage that same plaintext will never result into same ciphertext.

4.1 PROPOSED APPROACH

Here, a block diagram relevant to the proposed approach is presented in figure 4.1, which provides a clear precise view of the working principle. The block diagram provides the methodology used to achieve the desired results. The detailed working about the block diagram is described as follows:

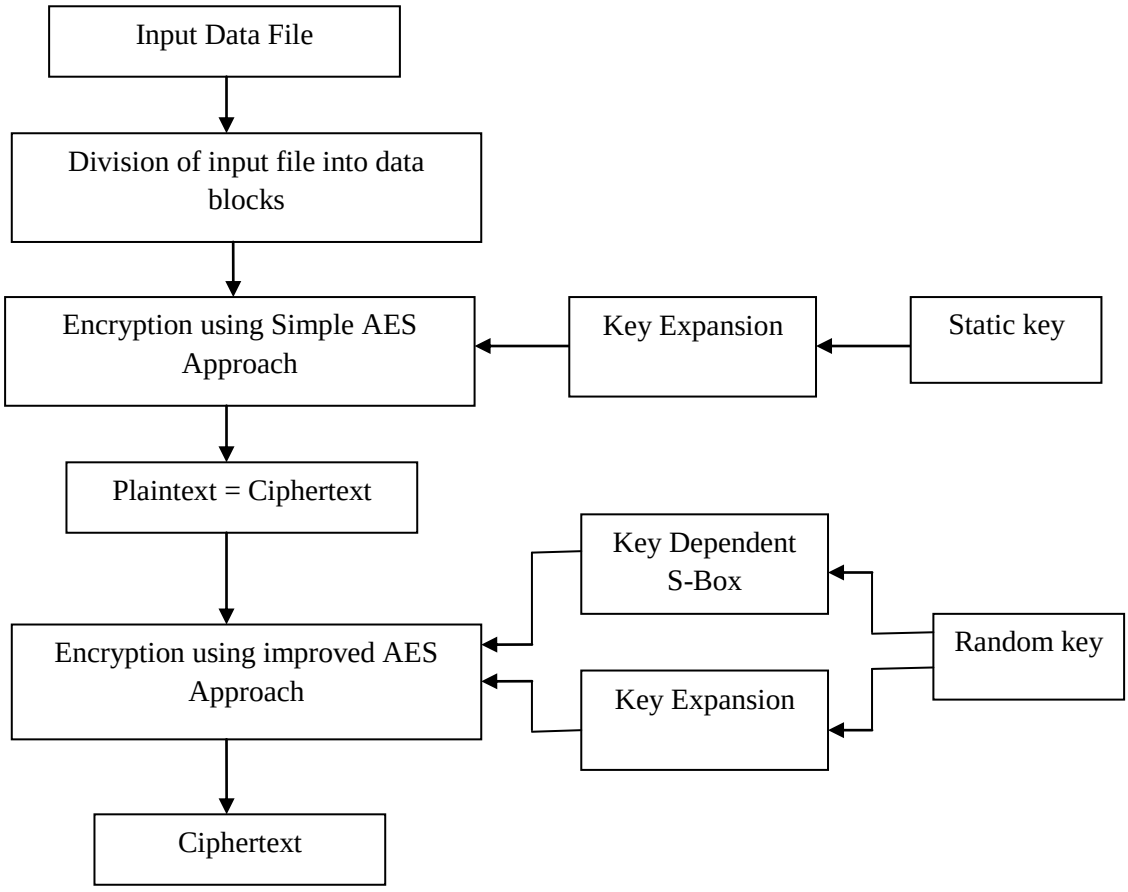

Figure 4.1 Block Diagram representing the proposed Approach

4.2 FIRST ENCRYPTION USING CONVENTIONAL AES APPROACH

Here, for the first encryption, a static key of 16 bytes is used to encrypt the data block of 16 bytes. This static key is once sent over the channel and could be distributed using RSA method in conjunction with DSA or hashing technique to provide authentication. The approach for encryption using this static key is the conventional AES Approach as described in chapter 3. For the first encryption, total of 10 rounds of iterations are made to manipulate the data according to the AES-128 algorithm. The ciphertext resulting from this round is treated as plaintext for the second round of encryption.

4.3 SECOND ENCRYPTION USING RANDOM KEY GENERATION AND KEY DEPENDENT S-BOX

In second round of encrypting, two different approaches are used in conjunction for the second encryption of data- random key and key dependent S-Box.

4.3.1 Random Key Generation

The subkey which is random has been developed by utilizing the inbuilt functions for pseudo-random number generation functions in MATLAB. The random key generated is made to have byte values between the range of 1 to 255 which are the values of ASCII characters. The key has 16 byte values according to the implementation of AES. The 16 byte key thus generated has individual byte values between 1 and 255. The key is then sent to the key expansion routine in order to generate 11 subkeys which are to be further used for encryption in the second round. The advantage with this random key is that it will change with every session. Therefore, same plaintext will never result in same ciphertext. This provides enhanced security against some cryptanalytic attacks.

4.3.2 Key Dependent S-Box Generation

Static nature in S-Boxes or the substitution boxes used in implementation of AES Algorithm pose a threat as they are highly vulnerable to linear and differential cryptanalysis. In advance knowledge of static S-Boxes and then observing the patterns between plaintexts and ciphertexts, the attacker can retrieve some knowledge about the encryption key or the key itself. This could prove fatal to the whole of the communication process. Therefore, it will be very favourable to devise a way by which the S-Boxes can be generated dynamically. As the substitution boxes will be unknown to the attacker, i.e. the attacker will not have any previous knowledge about substitution boxes, therefore, it will give a very useful advantage of making AES secure against the previously mentioned attacks.

Key dependent S-Box development method utilized for proposed work is followed from [75]. The work from this paper generates S-Boxes with ideal independency ratios which is a very favourable quality for the S-Boxes. The algorithm utilized for the development of S-Boxes is as follows.

The scheme followed in the development of S-Box is structured in two parts and is described as follows [75].

4.3.2.1 Pseudo-Expansion of the Key

This transformation takes the key and round constants as the input and generates 44 words or 176 bytes (44 words x 4 bytes=176 bytes) stored in an array K. The input key is arranged in a 4x4 matrix. RCon implies the Round Constants. This is given the name pseudo-expansion because this is not equivalent to the way in which keys are expanded in conventional AES-128.

Algorithm: Pseudo-Expansion of the Key
Input: Key, RCon
Output: Array K containing 44 words or 176 in bytes
Step 1: K= reshaping of key as 4x4 matrix
Step 2: For x=5,....., 44
Follow
Step 3: Prevrow=K(x-1, :)
Step 4: If x mod 4 = 1
Step 5: Prevrow=Prevrow([2 3 4 1])
Step 6: Y=RCon((x-1)/4,:)
Step 7: Prevrow= XOR(Prevrow,Y)
Step 8: End
Step 9: K(x, :) = XOR(K(x-4, :), Prevrow)
Step 10: End

The 16 bytes of the cipher key are treated as the first subkey i.e. words 1 to 4 in matrix K is the first subkey. Prevrow consists of the previous row. If row index is such that on dividing by 4 the result is 1, for ex. 5, 9 etc. such that the index mod 4=1, then the Prevrow is cyclically rotated and is added with Round Coefficient as suggested in step 7. Round Constants RCon consists of a matrix having hexadecimal numbers [01, 02, 04, 08, 10, 20, 40, 80, 1b, 36] as the first column. The remaining three of the columns of the round constant, each having 10 entries, contain 00 as the entries. Here, it is observed that the pseudo-expansion of the key does not make use of the S-Box or the byte substitution as is done in the conventional AES approach. This brings about a certain speed to the implementation of key-schedule.

4.3.2.2 Generation Of Key-Dependent S-Box

Result from previous scheme is the key pseudo-schedule that derived from input cipher key. Key-pseudo schedule K of 176 bytes in length, calculated in the previous step, is input parameter to this step. The output is S-Box and the inverse S-Box which depend on the key which is currently being utilized for the purpose of encryption.

Algorithm: Generation of Key-Dependent S-Box

Input: The Pseudo-Schedule K.

Result: S-Box depending upon key

Step 1: Initialization

a=0

b=1

c=1

Step 2 : Computation of the first total mod 256:

S(1)=(K(1) + K(2)) modulo 256

S_Box(1)= S(1)

Step 3: Till b<256

a=a+1

m=1+(b+a*c) mod 176

S(a+1)= (S(a)+K(m))mod 256

c=0

Step 4: for y=1,.....,b follow

Compare the subtotal S(a+1) with the elements S_Box(y) and count the number c of the S-Box elements which are not equal to S(a+1)

end for

Step 5: In case that c=y

S_Box(b+1)=S(a+1)

b=b+1

end if

end while

Step 6: for b=1,.....,256 do

Inv_S_Box(S_Box(b)+1)=b-1

End for

Thus, the above algorithm provides the S-Box which has been developed and are based on the input key. Because random key will keep on changing with in every session, similarly the key dependent S-Boxes will also change, which will bring about the security from the differential and linear cryptanalysis. The S-Boxes which are generated using the above approach follow nearly an ideal independency ratio [75] which means that correlation amongst the elements of the S-Box is the least possible and hence finding patterns among the data becomes a near impossible task. This also provides security against the side-channel attacks.

4.4 AVALANCHE EFFECT FOR ENCRYPTION

This effect is highly covetable characteristic in encryption and hashing algorithms. Avalanche effect can be broadly defined as the change observed in the output when the input is changes slightly. For encryption algorithms, this effect can be calculated by calculating the number of bits that change in the resulting ciphertext, when the input plaintext is changed by 1 bit or when 1 bit of the plain text is flipped.

If tb= total number of bits in the output

And nb=number of bits that change when 1 bit of input is flipped

Then, Avalanche effect = (nb/tb)*100%

An avalanche effect of more than 50% is often required by a cryptographic scheme as it exhibits great randomness in the output even when input changed only slightly. A high value of avalanche effect indicates more randomness in the output due to which, it will be impossible to observe any patterns in the resulting ciphertext data and hence security will be increased. If a high value of avalanche effect is not observed, it means the attacker can make certain assumptions about the input based on the output result only and hence can launch an attack.

4.5 STRICT AVALANCHE CRITERION (SAC)

SAC is based on or is a formalisation of avalanche effect. An encryption technique assumed to satisfy strict avalanche criteria if at least 50% of the output bits change when input is changed by 1 bit. In other words, it symbolizes that change of one bit in the input plaintext and the resultant bits in the output ciphertext change by minimum 50%, then it can be said that ciphering algorithm exhibits Strict Avalanche Criteria.

4.6 METHODOLOGY

The methodology followed to implement the proposed approach is described as follows.

4.6.1 First Encryption Using a Static Key

The data to be encrypted is taken as the plaintext input. The input plaintext is divided into blocks of 16 bytes (16x8=128 bits). The 16 bytes static key is also taken as the input. The initializations concerning the substitution or S-Boxes and also polynomial matrix are done. Using initializations, subkeys are expanded using scheduling in order to generate 11 subkeys from the main cipher key. Then the keys and plaintext blocks are given as input to the AES encryption scheme. SubBytes() implements the byte substitution from the substitution boxes. ShiftRows() does the operation of shifting of rows. MixColumn() is another main diffusing element which spreads the effect of one byte to whole of the state matrix. Total 10 rounds utilizing 11 subkeys are performed.

4.6.2 Second Encryption Using Dynamic Key with Key-Dependent S-Box Development

Dynamic random key is taken as the input and is expanded using key pseudo-expansion schedule, as discussed within section 4.3.2.1. Using pseudo-expansion result, the key-dependent S-Box is generated utilizing method discussed within section 4.3.2.2. Using this dynamic expanded key and dynamic S-Box, encryption is performed on similar grounds. Figure 4.2 depicts the above described methodology.

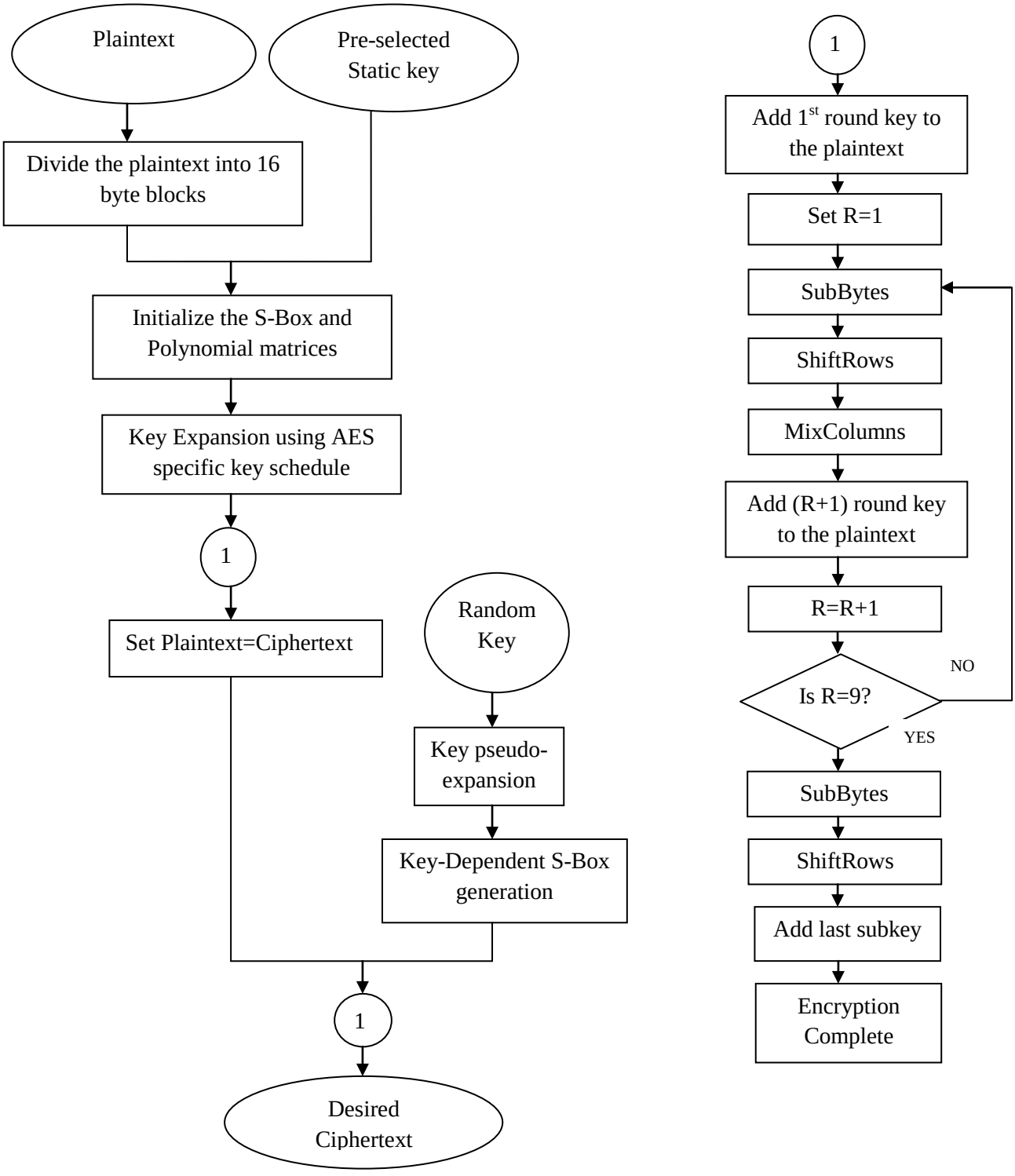

Figure 4.2 Flowchart depicting the proposed methodology

4.6.3 Avalanche Effect Calculation

The calculation for avalanche effect is done as follows. Firstly, the ciphertext using simple technique is observed in binary form. Then, one bit is flipped in the input plaintext or the subkey. Total bits which are flipped within resultant ciphertext as result of change in plaintext or key are noted. Then the avalanche effect is calculated as follows:

$$\text{Avalanche Effect} = \frac{No.of\ bits\ Flipped\ in\ Ciphertext}{Total\ no.of\ bits\ in\ Ciphertext} * 100\ \%$$

4.6.4 Strict Avalanche Criterion

The bits in ciphertext are observed closely. Then 1 bit in plaintext/subkey inverted. If this change results that minimum 50 % of the bits in ciphertext also gets flipped, then it can be said that the algorithm is exhibiting Strict Avalanche Criteria (SAC).

CHAPTER 5

RESULTS AND DISCUSSIONS

The following section focuses on comparing the proposed schemes with the conventional AES-128 approach for encrypting data. The comparison is based upon the metrics like execution time, avalanche effect test, strict avalanche criterion test. The graphs and the tables depicting the comparison are computed from the simulations and presented in this chapter.

5.1 KEY-DEPENDENT S-BOX

Consider a 16 byte key K given as follows in equation 5.1 [75]:

K= {5f, 6e, fd, 8c, cb, dc, 79, f8, c7, 56, e5, bf, d3, c2, e3, 2f} (5.1)

Following the algorithms for Substitution-Box depending on key discussed from previous chapter, S-Box generated using the above key is depicted in figure 5.1.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	238	111	194	228	165	51	54	157	216	18	202	92	74	129	188	250
1	230	154	0	72	208	151	125	24	83	146	114	218	121	227	21	2
2	185	37	17	8	214	134	220	80	39	181	6	128	207	77	191	231
3	10	118	96	79	187	226	50	247	110	15	245	171	86	81	126	224
4	234	3	53	108	65	192	35	26	150	14	57	144	167	38	106	91
5	236	89	205	88	155	55	104	178	85	215	123	193	76	229	112	56
6	75	44	163	97	174	52	199	243	30	133	143	232	137	61	87	211
7	179	235	241	223	4	59	201	42	90	225	162	47	22	139	253	13
8	136	120	82	170	101	180	102	189	23	78	45	64	63	131	198	153
9	183	210	169	251	43	115	173	46	244	164	168	116	20	212	9	33
A	200	93	100	36	49	130	109	132	66	248	246	41	124	221	254	141
B	28	95	204	252	69	27	209	1	71	242	122	103	213	196	190	67
C	182	34	135	219	140	48	147	98	60	177	142	148	237	195	175	99
D	5	186	70	166	184	117	7	107	239	217	62	176	159	138	11	113
E	68	249	105	149	127	58	73	145	32	19	84	197	94	156	161	160
F	172	40	25	158	222	31	16	206	12	255	152	233	29	240	119	203

Figure 5.1 Key-Dependent S-Box generated using Key in Equation (5.1)

As depicted in [75], the measure of independency ratio for the elements of above key-generated S-Box elements is almost equal to the ideal ratio. Thus, the above S-Box provides a certain degree of randomness and also increases strength against the attacks like linear and differential cryptanalysis.

5.2 COMPARISON BASED ON EXECUTION TIME

In the following result, comparing among conventional AES-128 Encryption and the proposed modified methodology for AES-128 in three sub-sections, applying the methodology in steps to the classic approach.

5.2.1 Simple AES-128 and AES-128 with Key-Dependent S-Box

Figure 5.2 represents a comparison between encrypting the data file using the simple AES-128 approach and the AES-128 using key-dependent S-Box.

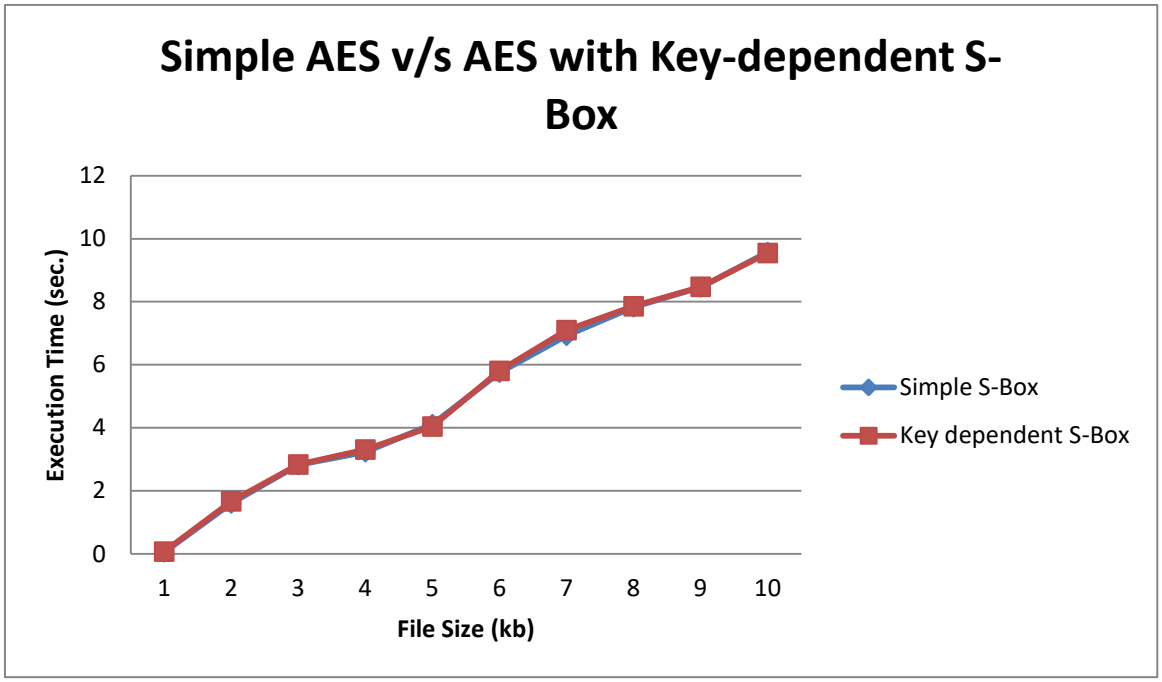

Figure 5.2 Execution time comparison between simple AES-128 and AES using key dependent S-Box approach

As is apparent from the figure, the execution time for generating key-dependent S-Box followed by encryption, and the time to perform encryption using S-Box from standard look-up table is almost the same. The red and blue curves almost overlap each other. Different sizes of data files ranging from 1 kb to 10 kb are input to the encryption algorithms and the comparison shows that the algorithm followed to generate key-dependent S-Box is fast enough.

5.2.2 AES-128 with Single Key and Multiple Keys

The comparison between the execution times of encryption using AES-128 with single keys and three independent keys is depicted in figure 5.2. Observing the execution period using multiple keys is significantly more than with single key. This is as expected because when using a single subkey in encrypting purpose, number of rounds in encrypting a single block of data is 10 i.e. combining the various operations SubBytes, ShiftRows, MixColumns and KeyAddition.

Figure 5.3 depicts the comparison between the execution time of AES-128 with single key and using three independent keys.

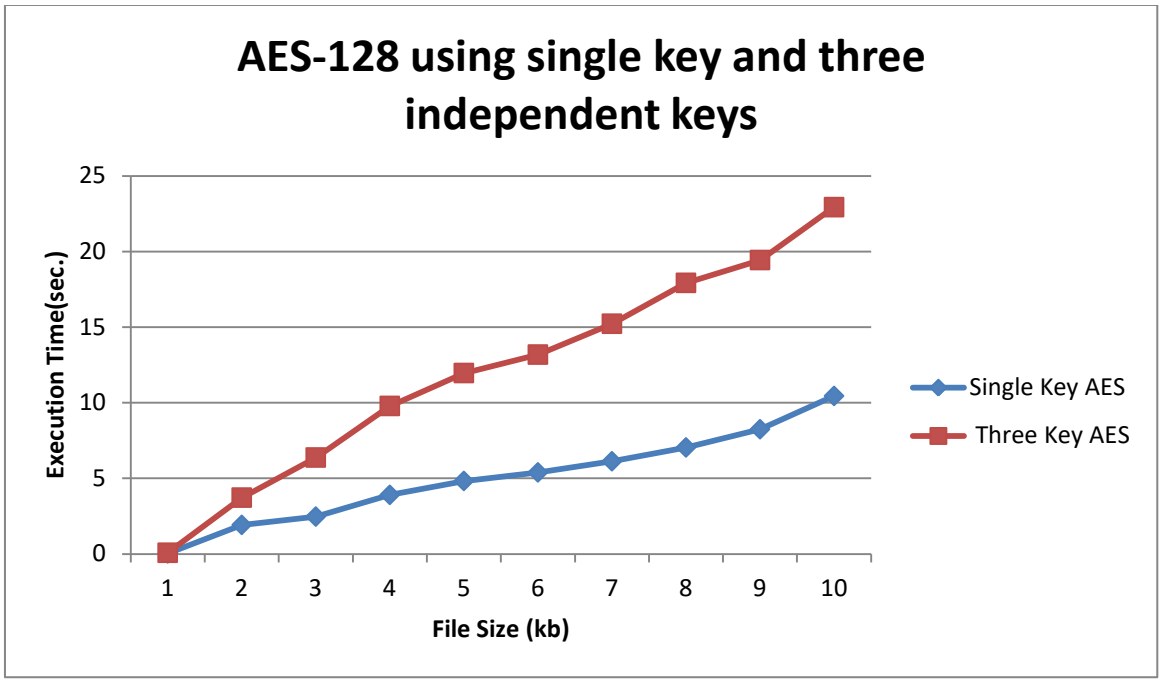

Figure 5.3 Execution time comparison between AES-128 using single key and three independent keys

5.2.3 Conventional AES-128 & AES-128 with Two Keys and Key-Dependent S-Box

Figure 5.4 represents comparison between execution time of Conventional AES and AES using proposed methodology.

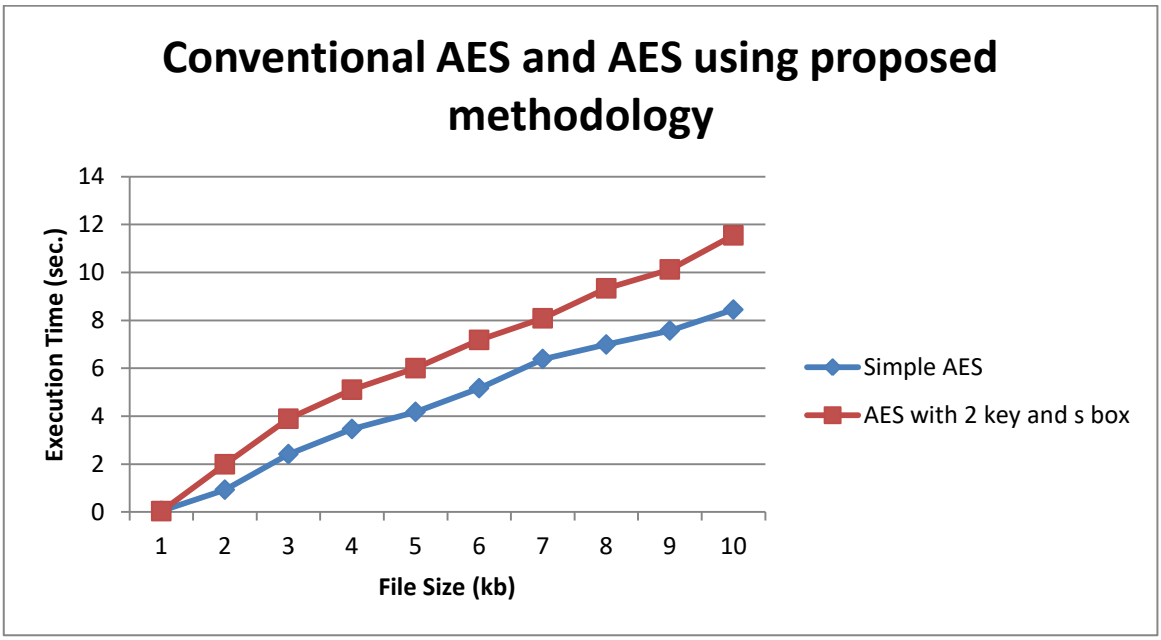

Figure 5.4 Execution time comparison of Conventional AES and AES using proposed methodology

From figure 5.4, it has been observed that an increment of execution period is observed between proposed approach and the conventional approach. While this may appear a disadvantage at first, we must also keep in mind the advantages achieved using the proposed methodology. As the increment of number of keys increases effective key space, random key which will be renewed with each session will reduce the vulnerability to the side-channel attacks also the key-dependent S-Box will strengthen the security avoiding vulnerability to linear and differential cryptanalysis.

5.3 AVALANCHE EFFECT TEST

To test the conventional and proposed methodology, on the basis of avalanche effect, different number of samples were fed to the schemes for encryption at different times. Out of the total number of samples, the number of times an algorithm gives better avalanche effect than the other is taken as criteria for better performance.

5.3.1 Simple AES and AES Having Key-Dependent S-Box

Avalanche effect comparison between different approaches is depicted in figure 5.5.

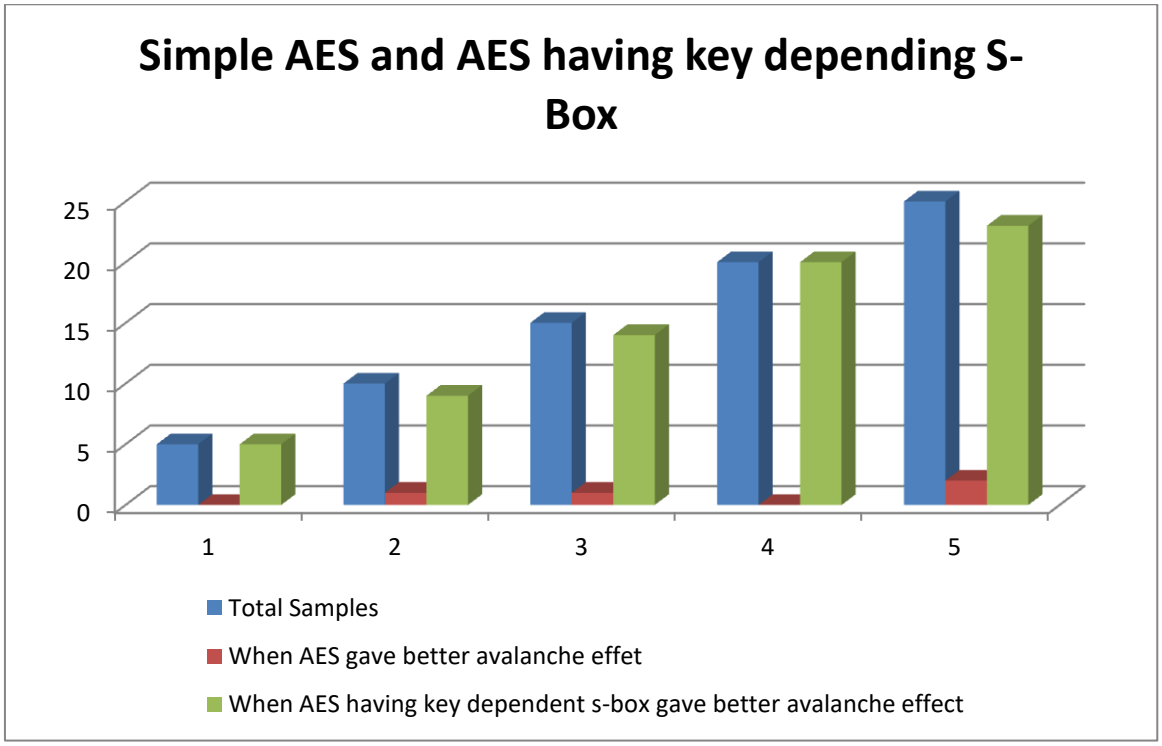

Figure 5.5 Comparison of avalanche effect shown by simple AES and AES having key-dependent S-Box

In figure 5.5,it has been clearly depicted that key-dependent S-Box will have a better avalanche effect than the static one. This is due to the high independency measure ratio of the key-dependent S-Box as is depicted in [75]. More randomness occurs in the data when one bit in subkey is inverted because key-dependent S-Box generates a completely different result.

5.3.2 AES with Single Key and AES with Multiple Keys

Here, the comparison between the avalanche effect exhibited by the AES using single key and multiple keys has been depicted in figure 5.6. It has been observed from figure 5.6 that the static nature of S-Box gives slightly better results for avalanche effect when multiple keys are utilized. Multiple keys increase the effective key space which provides with better security against the attacks.

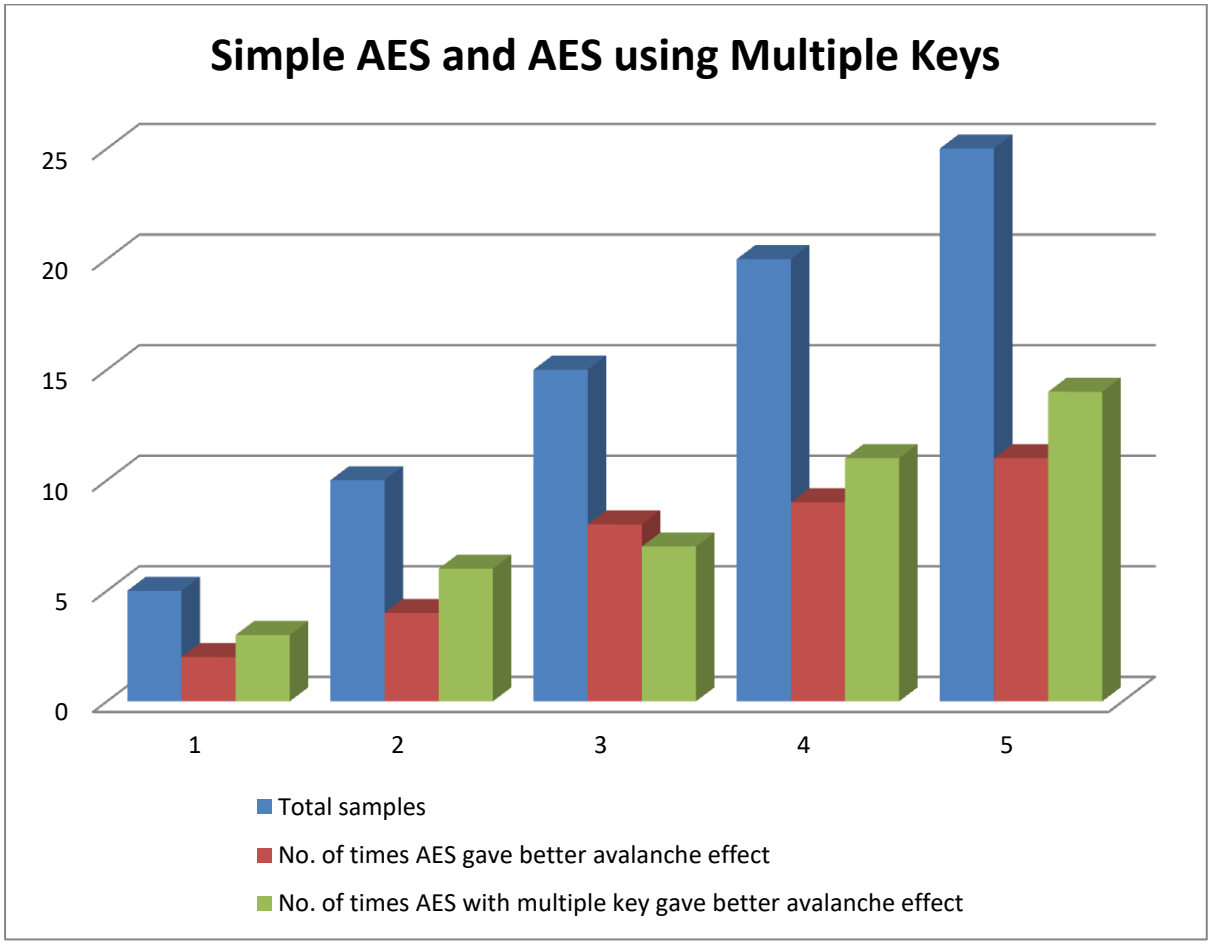

Figure 5.6 Comparison between Avalanche effect shown by AES with single key and AES with multiple keys

5.3.3 Conventional AES and AES Using Proposed Methodology

Proposed methodology has the advantage of increased key space and more randomness in the form of random key which changes with every session and the key-dependent S-Box. The basic advantages which comes by adopting the proposed methodology is that the small difference in random key would bring about large difference in ciphertext, due to which it will become highly difficult for the hacker to keep track change in patterns of ciphertext. This will bring about security from the side-channel attacks because the random and rapid changes in the features which will be monitored closely, will require immense effort from the sides of the attacker. Also, the key dependent S-Box will provide strength against certain attacks.

Figure 5.7 provides comparison between avalanche effect shown by conventional AES and AES with proposed methodology. It has been easily observed that the proposed methodology gives better results for avalanche effect test than the conventional approach.

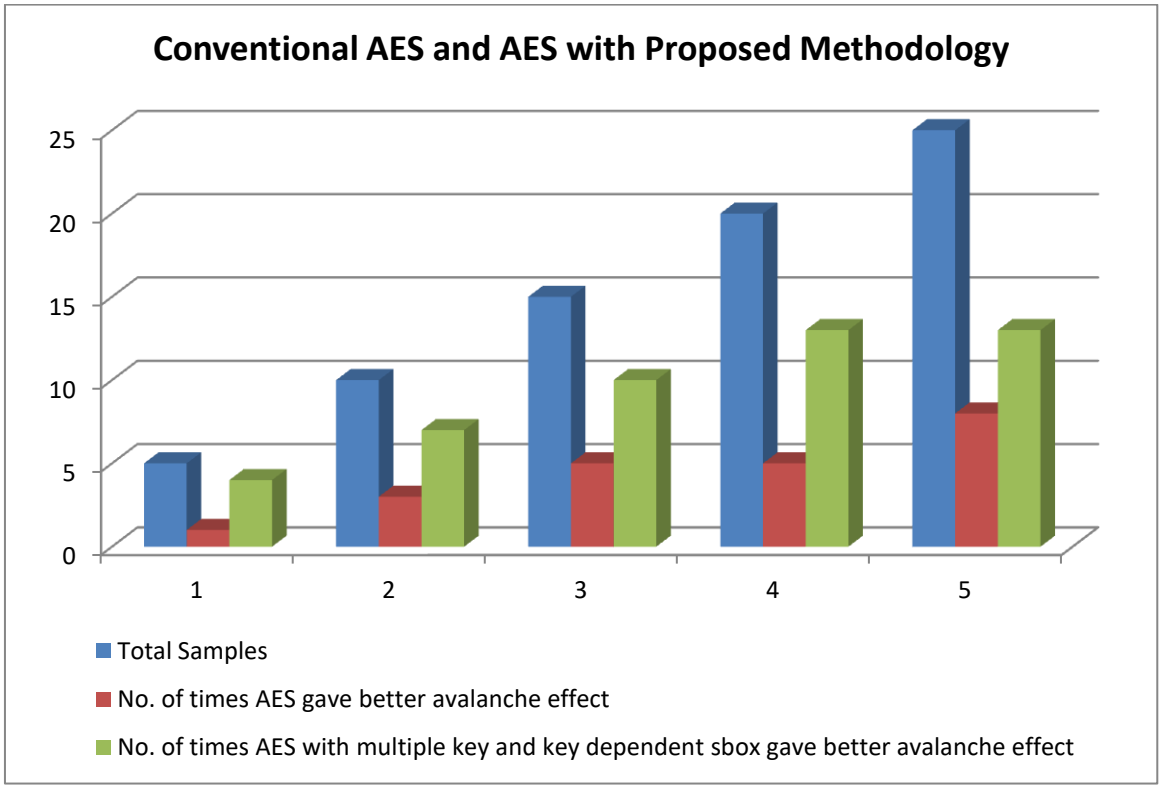

Figure 5.7 Comparison between Avalanche effect for conventional AES and AES with proposed methodology.

5.4 STRICT AVALANCHE CRITERION (SAC)

SAC is an important quality measurement statistic for cryptographic algorithms. SAC is exhibited by a cryptographic algorithm when, on changing or flipping single bit of input data or key, the output changes by minimum 50%. The SAC comparison between conventional and proposed approaches is shown as follows.

5.4.1 Simple AES and AES having Key-Dependent S-Box

Because conventional AES scheme has a static S-Box, a single bit difference in input in the plaintext or the key will not bring about remarkable difference in the result. Thus amount of avalanche effect exhibited by the AES with the static S-Box will always be low. On the other hand, AES having key-dependent S-Box will have a huge change in the ciphertext when one bit is flipped in the key. Because a little change in the key, will affect all the elements of the key-dependent S-Box, which in turn will exhibit a large change in ciphertext when one bit of the input is flipped. Thus, AES cryptographic scheme which includes the key-dependent S-Box will always exhibit better avalanche effect than theconventional approach of AES encryption.

Figure 5.8 depicts that the key-dependent S-Box will give better SAC than conventional SAC all the time. SAC exhibited by conventional AES always lies below 50%, whereas large avalanche effect exhibited by key-dependent S-Box AES will give high SAC.

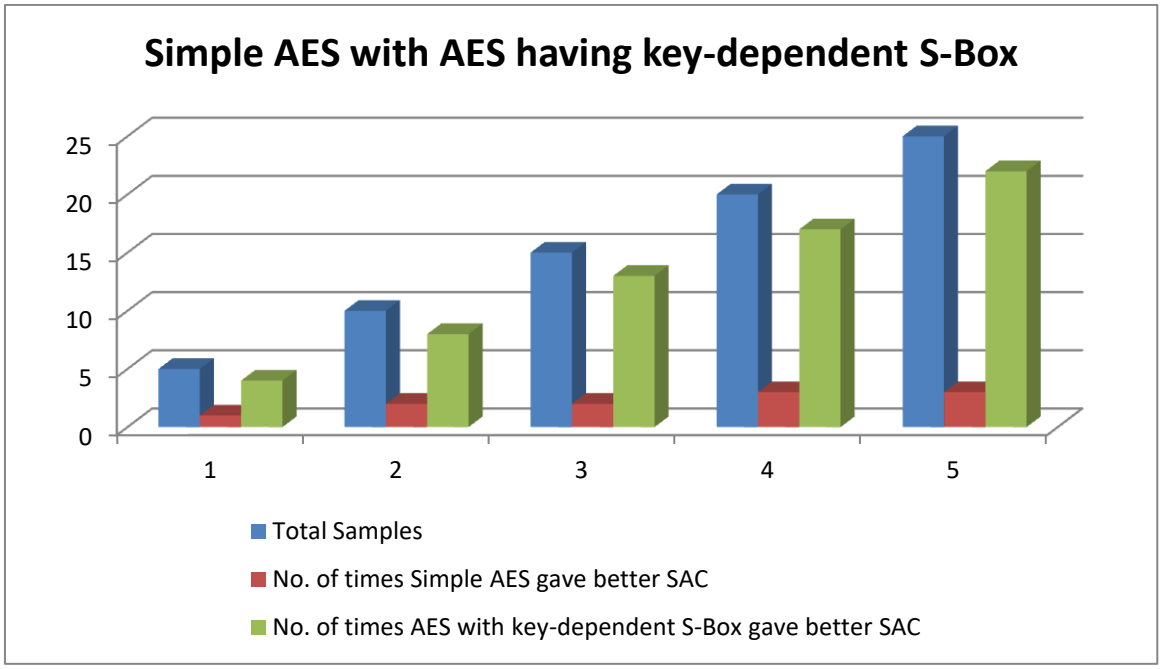

Figure 5.8 Comparison between Simple AES with AES having key dependent S-Box based on Strict Avalanche Criteria

5.4.2 AES with Single Key and AES with Multiple Key

Figure 5.9 gives the SAC based comparison between single key AES and multiple key AES.

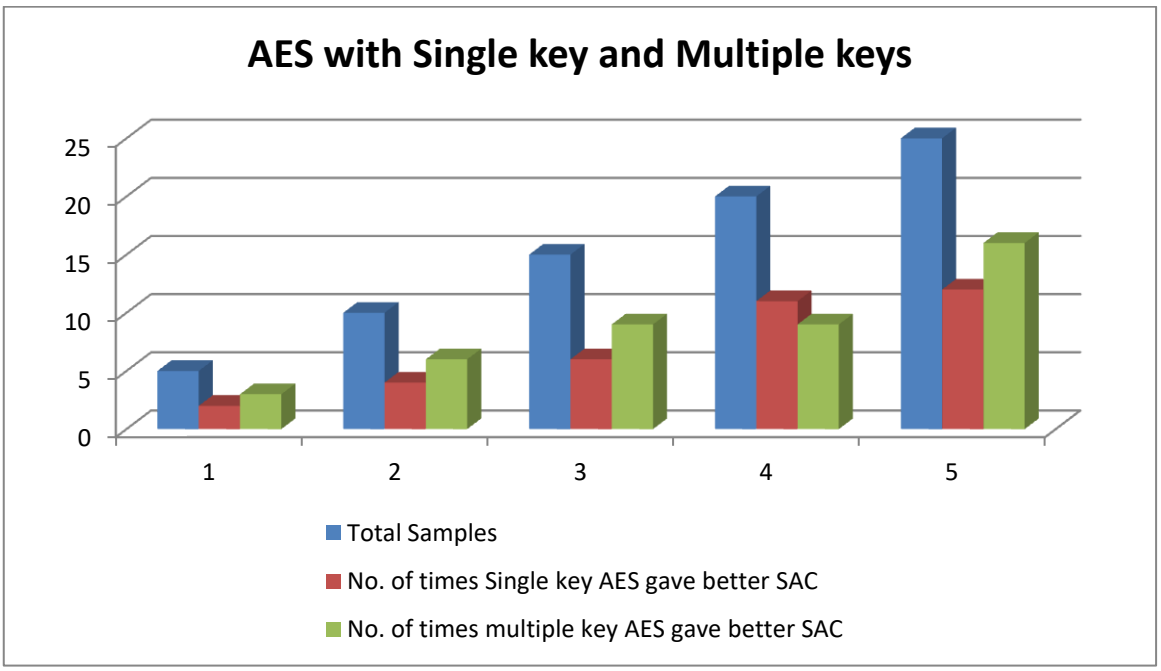

Figure 5.9 Comparison between Single key AES and multiple key AES base on Strict Avalanche Criteria

From figure 5.9, it has been observed that the SAC exhibited by multiple key AES is slightly better than the SAC exhibited by the single key AES. Thus, adding multiple keys works a little in favour of the proposed methodology. Also increase in effective key space gives out better cryptographic strength.

5.4.3 Conventional AES and AES using Proposed Methodology

Figure 5.10 serves the purpose of successfully depicting that the proposed methodology exhibits better SAC than the conventional approach. This is based on the advantages achieved by using random keys which change with every session and the key-dependent S-Boxes.

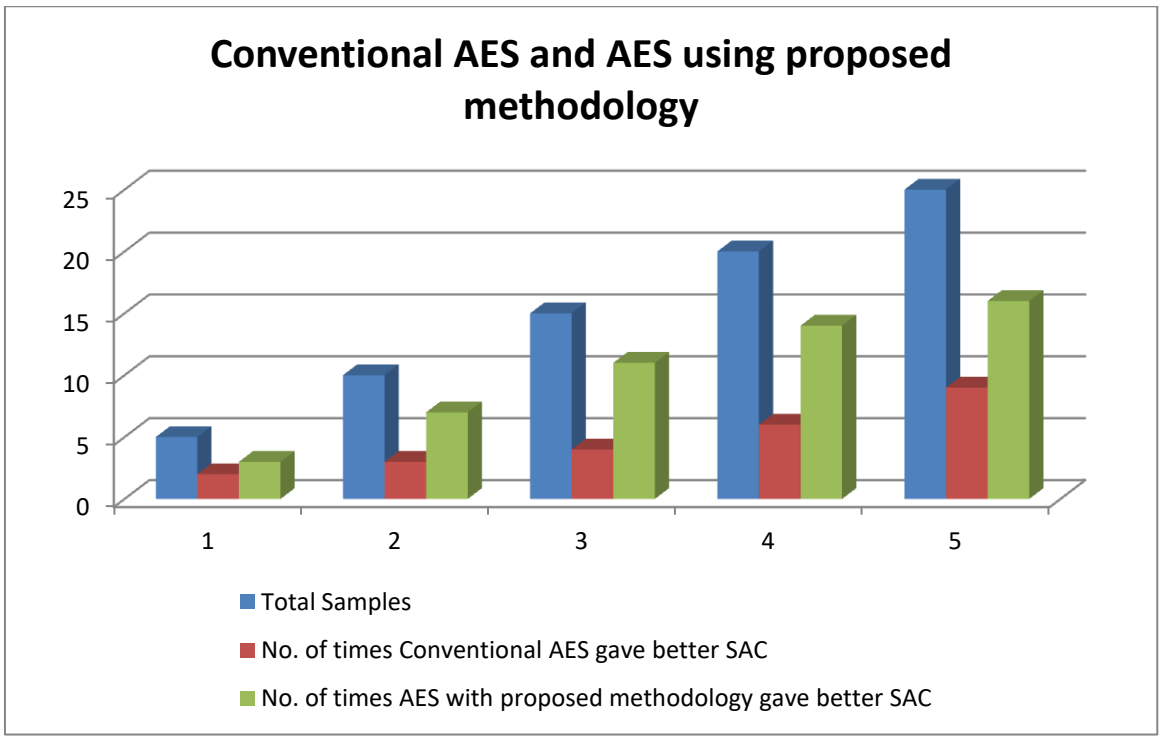

Figure 5.10 Comparison between Conventional AES and AES with proposed methodology based on Strict Avalanche Criteria

CHAPTER 6

CONCLUSION AND FUTURE SCOPE OF RESEARCH

Privacy and security are the most crucial important things that a consumer demands and the firms and companies strive to provide in this world of digital communication. Cryptography is an indispensable tool in today's era of internet where millions of files are shared among the users all over the globe with every passing second. The e-commerce applications, social media communications, calls and messaging etc., everything includes cryptography in one way or the other. End-to-end encryption has also become a new normal for messaging platforms operating on internet. Thus, cryptography is studied in detail in this thesis work. Among the various data encryption schemes, AES, which was standardized in 2001, is a widely utilized tool to encrypt data today. Therefore, the literature review in chapter 2 majorly focuses on improvements that are brought about by various researchers in the AES algorithm. Studying AES in detail in chapter 3, it was found that the static nature of keys and S-Boxes in AES provides an attacker with some in hand knowledge about the patterns of ciphertext and the key and will make the scheme vulnerable to attacks for example linear and differential cryptanalysis and some other techniques. Thus, in proposed work, an approach is followed to introduce some dynamic nature and random nature in the conventional AES approach. This has been done by using multiple keys, out of which one is static and the other ones is randomly generated and changes with every session. This will reduce the vulnerability to side-channel attacks. Also, an algorithm is proposed based on usage of key-dependent S-Boxes which will show high change in output when a small change is brought in the input. This dependency of S-Boxes on input random changing keys will provide strength against cryptanalytic attacks. The analysis of resulting approaches with the conventional ones has been made on the basis of execution time, avalanche effect and strict avalanche criteria. It was found that the proposed approach exhibits better avalanche effect and SAC than the classic approach. Thus, proposed approach is a definite improvement over the conventional one. Also, it was noticed that an increment of execution time due to increased complexity in proposed approach. Therefore, for the future work, attempts can be made in order to minimize the execution complexity and time for the proposed approach.

REFERENCES

[1] *An Introduction to Cryptography* : PGP Corporation, 2002.

[2] Forouzan, B. A. and Mukhopadhyay, Debdeep. *Cryptography and Network Security,* 2nd ed., McGraw Hill Publication, 2013.

[3] Stallings, W. *Cryptography and Network Security: Principles and practice*, 4th ed., Prentice Hall Publications, 2005.

[4] Denning, Dorothy Elizabeth R., *Cryptography and Data Security*, Addison Wesley Publishing Company, 1982.

[5] Biham, E. and Shamir, A., *Differential Cryptanalysis of Data Encryption Standard*, 1st ed., Springer-Verlag Publication, 1993.

[6] Buchanan, W. J., *3DES Encryption and Decryption in Microsoft*, 2010.

[7] Kak, A., *Lecture Notes on Computer and Network Security, Purdue University,* 2015.

[8] Zhou, X. and Tang, X., *Research and implementation of RSA algorithm for encryption and decryption*, Proceedings of 6th International Forum on Strategic Technology, Harbin, Heilongjiang, pp. 1118-1121, 2011.

[9] Ganley, M. J., *Digital signatures and their uses*, Computers & Security, Volume 13, Issue 5, Pages 385-391, 1994.

[10] Hankerson D., Menezes, A. and Vanstone, S., *Guide to Elliptic Curve Cryptography*, Springer Publications, 2004.

[11] Menezes, A. J., Oorschot, P. C. and Vanstone, S., *Handbook of Applied Cryptography,* 3rd ed., CRC Press, 1997.

[12] Diffie W. And Hellman, M.E., *New Directions in cryptography*, IEEE Transactions on Information Theory, Volume 22, Issue 7, pp. 644-654, 1976.

[13] Coppersmith, D., *Cryptography*, IBM Journal of Research and Development, Volume 13, Issue 2, 1987.

[14] G.J. Simmons, *A Survey of Information Authentication*, Proceedings of the IEEE, Volume 76, Issue 5, 1988.

[15] Ehrsam, W.F., Matyas, S. M., Meyer, H.C. and Tuchman, W. L., *A cryptographic key management scheme for Data Encryption Standard*, IBM Systems Journal, Volume 17, Issue 2, 1978.

[16] Boyd, C., *Modern Data Encryption*, Electronics and Communication Engineering Journal, Volume 5, Issue 5, 1993.

[17] Coppersmith, D., *The Data Encryption Standard and its strength against attacks*, IBM Journal of Research and Development, Volume 38, Issue 3, 1994.

[18] Garfinkel, S. L., *Public Key Cryptography*, Computer, IEEE, Volume 29, Issue 6, 1996.

[19] Omura, J. K., *Novel Applications of Cryptography in Digital Communications*, IEEE Communications Magazine, 1990.

[20] Agrawal, M., *Cryptography: A survey*, IETE Technical Review, Volume 16, Issue 3-4, pp.287-296, 1999.

[21] Hayes, H. M., *A Tutorial on Linear and Differential Cryptanalysis*, Technical Report, Electrical and Computer Engineering, University of Newfoundland, St. John's, Newfoundland, Canada, 2004.

[22] Yang, M., Bourbakis, N., Li, S., *Data-Image-Video Encryption*, IEEE Potentials, Volume 23, Issue 3, 2004.

[23] Tankard, C., *Encryption as the corner stone of big data security*, Elsevier Journal of Network Security, Volume 2017, Issue 3, pp. 5-7 , 2017.

[24] Tomhave, B. L., *Key Management: The key to encryption*, EDPACS Journal, Volume 38, Issue 4, pp. 12-19, 2009.

[25] Vacca, J. R., *Encryption keys: Randomness is Key to their undoing*, Taylor and Francis Journal of information Systems Security, Volume 8, Issue 4, pp. 1-6, 2006.

[26] Gove, R. A., *An Overview of Modern Cryptography*, Taylor and Francis Journal of Information Systems Security, Volume 6, Issue 3, pp. 55-68, 2008.

[27] Madhavan, C. E., Saxena, P. K., *Recent Trends in Applied Cryptology*, IETE Technical Review, Volume 20, Issue 2, 2015.

[28] Parker, D. B., *Cryptographic Threat Analysis*, Taylor and Francis Journal of Information Systems Security, Volume 2, Issue 3, pp. 13-17, 2008.

[29] Patila, P., Narayankar, P., Narayan D G, Meena S M, *A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish*, International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, India, pp. 617-624, 2015.

[30] Nadeem, A., Younus, J. M., *A performance comparison of data encryption algorithms*, First International Conference on Information and Communication Technologies, *ICICT 2005*, pp. 84-89, 2005.

[31] Buchanan, W. J., Woodward A., Helme, S., *Cryptography across industry sectors*, Taylor and Francis Journal of Cyber Security Technology, pp. 1-18, 2017.

[32] Daemen, J., Rijmen, V., *The design of Rijndael Cipher*, Springer Publication, ISBN 978-3-662-04722-4, 2002.

[33] Kakkar , A., Singh, M. L. and Bansal, P. K., *Mathematical analysis and simulation of multiple keys and S-Boxes in a multinode network for secure transmission*, Taylor and Francis International Journal of Computers and Mathematics, Volume 89, Issue 16, pp. 2123-2142, 2012.

[34] Rachh, R. R., Ananda Mohan, P. V., Anami, B. S., *Efficient Implementations for AES Encryption and Decryption*, Taylor and Francis Journal of Circuits, Systems and Signal Processing, Volume 31, Issue 5, pp. 1765-1785, 2012.

[35] Osvik, D. A., Bos, J. W., Stefan D., and Canright, D., *Fast Software AES encryption*, Foundations of Software Engineering (FSE), 2010.

[36] Jingmei, L., Baodian, W. & Xinmei, W., *One AES S-Box to increase complexity and its cryptanalysis*, Elsevier Journal of Systems Engineering and electronics, Volume 18, Issue 2, pp.427-433, 2007.

[37] Gong, J., Liu, W., Zhang, H., *Multiple Lookup Table-Based AES Encryption Algorithm Implementation,* International Conference on Solid State Devices and Materials Science 2012, Physics Procedia, Volume 25, pp. 842-827, 2012.

[38] Dara, M. & Manochehri, K., *Using RC4 and AES Key Schedule to Generate Dynamic S-Box in AES*, Taylor and Francis Information Security Journal: A Global Perspective, Volume 23, Issue 1-2, 2014.

[39] Artz, D, *Digital Steganography: Hiding Data within data*, IEEE Internet Computing, Volume 5, Issue 3, pp. 75-80, 2001.

[40] Wahaballa, A., Wahballa, O., Li, F., Ramadan M., & Qin, Z., *Multiple-Layered Securities Using Steganography and Cryptography,* Taylor and Francis International Journal of Computers and Applications, Volume 36, Issue 3, pp. 93-100, 2015.

[41] Musa, M. A., Schaefer, E. F. & Wedig, S., *A Simplified AES Algorithm and its Linear and Differential Cryptanalysis*, Cryptologia, Volume 27, Issue 2, pp. 148-177, 2003.

[42] Simmons, S., *Algebraic Cryptanalysis of Simplified AES*, Cryptologia, Volume 33, Issue 4, pp. 305-314, 2009.

[43] Kaslauskas, K., Vaicekauskas, G., Smaliukas, R., *An Algorithm for Key Dependent S-Box generation in Block Cipher System,* Informatica, Volume 26, Issue 1, pp. 51-65, 2015.

[44] Ao, T., Rao, J., Dai, K., and Zou, X., *Construction of high quality key dependent S-Boxes,* International Journal of Computer science, Volume 44, Issue 3, 2017.

[45] Singh, A., Aggarwal, P., Chand, M., *Analysis of Development of S-Box Generation,* Journal of Computer Science and Information Technology, Volume 5, Issue 5, pp. 154-163, 2017.

[46] Lambic, D., *Security analysis and improvement of a block cipher with dynamic S-Boxes based on tent map,* Springer Journal of Non-Linear Dynamics, Volume 79, Issue 4, pp. 2531-2539, 2015.

[47] Lambic, D., *A novel method of S-Box design based on discrete chaotic map*, Springer Journal of Non-Linear Dynamics, Volume 87, Issue 4, pp. 2407-2413, 2017.

[48] Partheeban, P., Kavitha, V., *Dynamic key dependent AES S-Box generation with optimized quality analysis,* Springer Journal of Cluster Computing, pp.1-11, 2018.

[49] Ozer, A. B., *An alternative S-Box design method based on random selection*, European Journal of Technique, Volume 17, Issue 2, pp. 229-236, 2017.

[50] Ozkaynak, F., *Construction of robust substitution boxes based on chaotic systems*, Springer Journal of Neural Computing and Applications, pp. 1-10, 2017.

[51] Guesmi, R., Farah, M. A. B., Kachouri, A. and Samet, M., *A novel design of Chaos based S-Boxes using genetic algorithm techniques*, IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), Doha, pp. 678-684, 2014.

[52] Farah, T., Rhouma, R., Belghith, S., *A novel method for designing S-Box based on chaotic map and Teaching–Learning-Based Optimization*, Springer Journal of Non-Linear Dynamics, Volume 88, pp. 1059-1074, 2017.

[53] Ozkaynak, F., Celik, V., Ozer, A. B., *A new S-Box construction method based on the fractional-order chaotic Chen system*, Springer Journal of Signal, Image and Video Processing, Volume 11, Issue 4, pp. 659-664, 2017.

[54] Roslan, M. F., Seman, K., Halim, A. H. A., Sayuti, M. N. S. M., *Current Implementations of Advance Encryption Standard (AES) S-Box*, Journal of Fundamental and Applied Sciences, Volume 9, Issue 4S, pp. 518-542, 2017.

[55] Kaslauskas, K., Vaicekauskas, G., Smaliukas, R., *A Novel Method to Design S-Boxes Based on Key- Dependent Permutation Schemes and its Quality Analysis*, International Journal of Advanced Computer Sciences and Applications, Volume 7, Issue 4, pp. 93-99, 2016.

[56] Alamsyah, Bejo, A., Adji, T. B., *AES S-Box Construction Using Different Irreducible Polynomial and Constant 8-bit Vector*, 2017 IEEE Conference on Dependable and Secure Computing, Taipei, pp. 366-369 2017.

[57] Joshi, A., Dakhole, P. K., and Thatere, A., *Implementation of S-Box for Advanced Encryption Standard*, IEEE International Conference on Engineering and Technology (ICETECH), *Coimbatore*, pp. 1-5, 2015.

[58] Ratiner, M., *The method of S-Box construction*, Journal of Discrete Mathematical Sciences and Cryptography, Volume 8, Issue 2, pp. 203-215, 2005.

[59] Farwa, S., Shah, T., Idrees, L., *A highly nonlinear S-Box based on a fractional linear transformation*, SpringerPlus, Volume 5, Issue 1, pp. 1658, 2016.

[60] Canright, D., *A Very Compact S-Box for AES*. International workshop on Cryptographic Hardware and Embedded Systems – CHES 2005. CHES 2005. Lecture Notes in Computer Science, Volume 3659, Springer, Berlin, Heidelberg, 2005.

[61] Moradi, A., Poschmann, A., Ling, S., Paar, C., Wang, H., *Pushing the Limits: A Very Compact and a Threshold Implementation of AES*, International conference on theory and applications of Cryptographic Techniques .Advances in Cryptology - EUROCRYPT. Lecture Notes in Computer Science, Volume 6632. Springer, Berlin, Heidelberg, 2011.

[62] Tran, M. T., Bui, D. K., and Duong, A. D., *Gray S-Box for Advanced Encryption Standard*, 2008 International Conference on Computational Intelligence and Security, Suzhou, pp. 253-258, 2008.

[63] Rebeiro,C., Selvakumar, D., Devi, A. S. L., *Bitslice Implementation of AES*, International Conference on Cryptology and Network Security, Cryptology and Network Security. CANS 2006. *Lecture Notes in Computer Science*, Volume 4301. Springer, Berlin, Heidelberg, 2006.

[64] Oswald, E., Mangard, S., Pramstaller, N., Rijmen, V., *A Side-Channel Analysis Resistant Description of the AES S-Box*, International Workshop on Fast Software Encryption. FSE 2005. Lecture Notes in Computer Science, Volume 3557. Springer, Berlin, Heidelberg, 2005.

[65] Satoh, A., Morioka, S., Takano, K., Munetoh, S., *A Compact Rijndael Hardware Architecture with S-Box Optimization.* International Conference on the Theory and Application of Cryptology and Information Security, Advances in Cryptology - ASIACRYPT 2001, Lecture Notes in Computer Science, Volume 2248. Springer, Berlin, Heidelberg, 2001.

[66] Canright, D., Batina, L., *A Very Compact "Perfectly Masked" S-Box for AES*, International Conference on Applied Cryptography and Network Security. ACNS 2008. Applied Cryptography and Network Security. Lecture Notes in Computer Science, Volume 5037, Springer, Berlin, Heidelberg, 2008.

[67] Billet, O., Gilbert, H., Ech-Chatbi, C., *Cryptanalysis of a White Box AES Implementation*, International Workshop on Selected Areas in Cryptography, SAC 2004. Lecture Notes in Computer Science, Volume 3357, Springer, Berlin, Heidelberg, 2004.

[68] Ahmad, N., Hasan, R. and Jubadi, W. M., *Design of AES S-Box using combinational logic optimization*, IEEE Symposium on Industrial Electronics and Applications *(ISIEA 2010)*, Penang, pp. 696-699, 2010.

[69] Murphy, S., Robshaw, M. J., *Essential Algebraic Structure within the AES*, International Cryptology Conference. Advances in Cryptology — CRYPTO 2002. CRYPTO 2002. Lecture Notes in Computer Science, Volume 2442. Springer, Berlin, Heidelberg, 2002.

[70] Hosseinkhani, R., Javadi, S. H. H. S., *Using cipher key to generate dynamic S-Box in AES Cipher System*, International Journal of Computer Science and Security, Volume 6, Issue 1, pp. 19-28, 2012.

[71] *Announcing the Advanced Encryption Standard (AES)*, FIPS 197, November 26, 2001.

[72] Benvenuto, C. J., *Galois Field in Cryptography*, Mathematics Notes, University of Washington, 2012.

[73] Paar, C., Pelzl, J., *Understanding Cryptography: A textbook for students and practitioners*, Springer Publication, 2010.

[74] http://www.facweb.iitkgp.ernet.in/~sourav/AES.pdf

[75] Kazlauskas, K., Kazlauskas, J., *Key-Dependent S-Box generation in AES Block Cipher System*, Journal Informatica, Volume 20, Issue 1, pp. 23-34, January 2009.

LIST OF PUBLICATIONS

1. Sachdeva, S., Kakkar, A., *Implementation of AES-128 using Multiple Cipher Keys*, International conference on Futuristic Trends in Network and Communication Technologies (FTNCT-2018), 9th-10th Feb. 2018, JUIT, Waknaghat, Solan. (Presented)

Thesis_Shivani

ORIGINALITY REPORT

6%	4%	3%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	2%
2	www.mstarlabs.com Internet Source	<1%
3	www.ee.unt.edu Internet Source	<1%
4	web.fuip.fukuyama-u.ac.jp Internet Source	<1%
5	kwangja.tistory.com Internet Source	<1%
6	Submitted to Visvesvaraya Technological University Student Paper	<1%
7	eprints.utm.my Internet Source	<1%
8	www.bun23.com Internet Source	<1%