# EXPLORING NEW APPROACHES TOWARDS DESIGN OF BLOCK CIPHER TESTING ALGORITHMS

A Thesis Submitted to the University of North Bengal

for the Award of

**Doctor of Philosophy**

in

Computer Science and Application

BY

**Avijit Datta**

SUPERVISOR

Dr. Sharad Sinha

Department of Computer Science and Application

University of North Bengal

AUGUST, 2019

Dedicated to my Elder Brother and my Wife

**Mr. Surajit Dutta**

**and**

**Mrs. Ranjita Roy**

# Declaration

I hereby declare that the thesis titled EXPLORING NEW APPROACHES TOWARDS DESIGN OF BLOCK CIPHER ALGORITHMS has been prepared by me under the supervision of Dr. Sharad Sinha, Assistant Professor, Department of Computer Science and Application, University of North Bengal. No part of this thesis has formed the basis for the award of any degree or fellowship previously.

Avijit Datta

Research Scholar
Department of Computer Science and Application,
University of North Bengal

# Acknowledgement

I am beholden to a large number of people for their constant support and inspiration towards my motivation to the research work and writing of this thesis.

I am very much thankful Authorities of University of North Bengal for allowing me to pursue my Ph.D. in this University

I take this opportunity to express my respect and gratitude to my supervisor Dr. Sharad Sinha, Assistant Professor, Department of Computer Science and Application, University of North Bengal for his huge support, invaluable guidance and extreme encouragement throughout the research work and preparation of the thesis.

I am very much thankful to my elder brother Mr. Surajit Dutta and my better-half Mrs. Ranjita Roy for their enormous support and keeping faith on me to complete the research work. I am also grateful to my parents, in-laws, friends and colleagues for their enormous encouragement.

I would like to express my deep sense of gratitude to the Head, all faculty members and staff of the Department of Computer Science and Application, University of North Bengal for providing facilities, co-operation and valuable guidance throughout my research work.

I am thankful to various Journals for accepting and publishing our papers, which encouraged me a lot and gave me the direction to move forward in my research work.

Last but not the least, my heartfelt thanks goes to Mr. Dipanjan Bhowmick, my fellow researcher and UGC SRF, Department of Computer Science and Application, University of North Bengal for his knowledge sharing and his significant role in the team work which always uplifted my motivation towards this research work.

Avijit Datta

Research Scholar

Department of Computer Science and Application,

University of North Bengal

# DEPARTMENT OF COMPUTER SCIENCE AND APPLICATION

# UNIVERSITY OF NORTH BENGAL

**Dr. Sharad Sinha**
**Assistant Professor**

P.O. University of North Bengal
Pin- 734013
Phone (O): +91-353-2776344
Fax: +91-353-2699001
Email:
ssinha.nbu@gmail.com


***Ref. No.***

**Date:** 13/08/2019

## To whom it may concern

I certify that the thesis titled EXPLORING NEW APPROACHES TOWARDS DESIGN OF BLOCK CIPHER TESTING ALGORITHMS is a genuine piece of research work by Mr. Avijit Datta under my supervision and is being submitted for the award of Ph.D. degree of the University of North Bengal. He has carried out his work at the Department of Computer Science and Application, University of North Bengal. I further declare that no part of this thesis has been submitted anywhere for any kind of degree, whatsoever.

13/8/2019

Dr. Sharad Sinha
Assistant Professor
Department of Computer Science and Application
University of North Bengal

# Abstract

Analysis of block ciphers has been done efficiently on the vulnerability of well known block ciphers such as DES and AES and reviewed. There are lots of excellent research work on the -

a) Statistical testing on block ciphers,
b) The role of key schedules in attack on iterated ciphers,
c) Automated cryptanalysis on substitution cipher,
d) SAC randomness test including SPAC and SKAC,
e) Pattern recognition approach to block cipher identification, and
f) Many distinct ways of randomness tests.

After a thorough study of existing works of on cryptanalysis, it is found that though a lot of research has been done on cryptanalysis of block ciphers as a whole, but very little attention has been given towards the strength analysis of S-boxes of different cryptographic algorithms.

In this proposed research work, S-boxes are analyzed with some novel approaches to measure their strength against various types of attacks. Different statistical methods have been used with the proposed algorithms to arrive at the conclusions. The major aim of the proposed research work is to establish a testing suite for the existing S-boxes with the help of novel algorithms.

Every statistical approach has concentrated on computing *p-value* and its randomness. This research work also targets the *p-values* but with a different approach. It looks forward to make it feasible to cover both linear and differential cryptanalysis approaches. The bit level block cipher diffusion and confusion analyses are the key areas covered in this research work. The core intension is to establish a standard algorithmic test suite on block ciphers to test most of the internationally recognized encryption methods. Both statistical and randomness tests have been taken into consideration to develop the suite.

The approach adopted in this study is first to implement all the standard block ciphers to operate on all possible modes of operation, then each of these block ciphers are subjected to all possible tests available along with the newly proposed tests. The bit-level block cipher diffusion and S-box confusion have been thoroughly analyzed in this research work.

Initially, a randomly selected $n$ bit of block of plaintext ($say\ P$) has been used, which is then encrypted using the standard encryption methods to produce the corresponding cipher block ($say\ C$).

Then, a matrix of size $n \times n$ is produced, where each row of the matrix is say $P_i$, a new plaintext block in itself derived from the original block by flipping the bit at the $i^{th}$ position

i.e. $P_i[i] = P \oplus e_i$ where $e_i$ is a zero vector containing 1 at $i^{th}$ position.

Each row of the $P_i$ matrix is then fed as input to the underlying cipher to produce the corresponding cipher text, which is stored as the $i^{th}$ row of the $C_i$ matrix of size $n \times n$.

i.e. $C_i[i] = E(P_i[i])$ where *E()* denotes encryption using the underlying block cipher.

At this point, the scheme proceeds to produce the SAC (Strict Avalanche Criterion) matrix (say $X$) of size $n \times n$, where $i^{th}$ row of the matrix is obtained by bit wise addition (modulo 2 additions) of $C_i$ vector with $C$ vector,

$X[i] = C_i[i] \oplus C$.

Then the diffusion-factor is obtained by scanning each column of the $X$ matrix.

Subsequently, all obtained data in connection with the underlying encryption method is subjected to statistical analysis. From the generated SAC matrix, $1's$ of each column have been calculated to check the vulnerability of the cipher. The appearance of $1's$ has been analyzed statistically with the threshold value $n/2$ and the vulnerability factor computed accordingly. This algorithm has been named as BLDAT test in this research.

In another approach, the SAC matrix has been used to analyze the confusion of standard S-boxes. SAC matrices for each of the 8 S-boxes of size $4 \times 16$ of DES and the lone $16 \times 16$ S-box of AES have been implemented and analyzed.

For the analysis of confusion, SAC matrix includes the original set of input bits and all sets of input with every 1-bit alteration of original set of input bits. Individual SAC matrices have been generated for every S-box of DES and AES and the occurrences of 1's in the output have been calculated for each column of every S-box.

Further, the SAC matrix includes the original set of input bits and all sets of input with every 2-bit alteration of original set of input bits. Individual SAC matrices have been generated for every S-box of DES and AES and the occurrences of 1's in the output bit have been calculated for each column of every S-box.

In the above two methods, the SAC matrices being generated using the set of output bits are subsequently subjected to analysis of frequencies of various avalanche effects, analysis of Hamming weights according to the bit position and analysis of Coefficient of Variance.

Further in the research work, two more approaches have been introduced with the truncated differential cryptanalysis and boomerang-style attack on S-boxes of DES and AES.

In the truncated differential cryptanalysis approach, after generating the outputs of the original inputs to the S-boxes, the inputs are divided into two parts of same bit size, say $P(x_1, x_2)$. For the both parts of the truncated inputs, new $P(x_1', x_2')$ was generated by one bit alteration in each part and then combined. For the SAC matrix with original output and outputs corresponding to the new inputs, the occurrences of 1's in the output bit has been calculated and analyzed for each column of every S-box.

The Boomerang Attack is one of the trending attacks of the recent times. Boomerang attack has been used against well known encryption algorithms. In this research work, boomerang-style attack has been implemented with the confusion analysis of standard S-boxes of DES and AES. In this approach SAC matrix has been generated with output of all possible pair combination of original input.

Finally, the conclusion and recommendation has been drawn with respect to the statistical comparison on the confusion and diffusion of cryptographic algorithms. A comparative study has been done at the end of the thesis to get a clear view of whole work. It helps to draw the conclusion on the security levels of S-boxes with respect to confusion.

A comparative study of all experimental data for DES and AES has been included here to draw the conclusion.

# Abstract

Analysis of block ciphers has been done efficiently on the vulnerability of well known block ciphers such as DES and AES and reviewed. There are lots of excellent research work on the -

a) Statistical testing on block ciphers,
b) The role of key schedules in attack on iterated ciphers,
c) Automated cryptanalysis on substitution cipher,
d) SAC randomness test including SPAC and SKAC,
e) Pattern recognition approach to block cipher identification, and
f) Many distinct ways of randomness tests.

After a thorough study of existing works of on cryptanalysis, it is found that though a lot of research has been done on cryptanalysis of block ciphers as a whole, but very little attention has been given towards the strength analysis of S-boxes of different cryptographic algorithms.

In this proposed research work, S-boxes are analyzed with some novel approaches to measure their strength against various types of attacks. Different statistical methods have been used with the proposed algorithms to arrive at the conclusions. The major aim of the proposed research work is to establish a testing suite for the existing S-boxes with the help of novel algorithms.

Every statistical approach has concentrated on computing *p-value* and its randomness. This research work also targets the *p-values* but with a different approach. It looks forward to make it feasible to cover both linear and differential cryptanalysis approaches. The bit level block cipher diffusion and confusion analyses are the key areas covered in this research work. The core intension is to establish a standard algorithmic test suite on block ciphers to test most of the internationally recognized encryption methods. Both statistical and randomness tests have been taken into consideration to develop the suite.

The approach adopted in this study is first to implement all the standard block ciphers to operate on all possible modes of operation, then each of these block ciphers are subjected to all possible tests available along with the newly proposed tests. The bit-level block cipher diffusion and S-box confusion have been thoroughly analyzed in this research work.

Initially, a randomly selected $n$ bit of block of plaintext ($say\ P$) has been used, which is then encrypted using the standard encryption methods to produce the corresponding cipher block ($say\ C$).

Then, a matrix of size $n \times n$ is produced, where each row of the matrix is say $P_i$, a new plaintext block in itself derived from the original block by flipping the bit at the $i^{th}$ position

i.e. $P_i[i] = P \oplus e_i$ where $e_i$ is a zero vector containing 1 at $i^{th}$ position.

Each row of the $P_i$ matrix is then fed as input to the underlying cipher to produce the corresponding cipher text, which is stored as the $i^{th}$ row of the $C_i$ matrix of size $n \times n$.

i.e. $C_i[i] = E(P_i[i])$ where *E()* denotes encryption using the underlying block cipher.

At this point, the scheme proceeds to produce the SAC (Strict Avalanche Criterion) matrix (say $X$) of size $n \times n$, where $i^{th}$ row of the matrix is obtained by bit wise addition (modulo 2 additions) of $C_i$ vector with $C$ vector,

$X[i] = C_i[i] \oplus C$.

Then the diffusion-factor is obtained by scanning each column of the $X$ matrix.

Subsequently, all obtained data in connection with the underlying encryption method is subjected to statistical analysis. From the generated SAC matrix, $1's$ of each column have been calculated to check the vulnerability of the cipher. The appearance of $1's$ has been analyzed statistically with the threshold value $n/2$ and the vulnerability factor computed accordingly. This algorithm has been named as BLDAT test in this research.

In another approach, the SAC matrix has been used to analyze the confusion of standard S-boxes. SAC matrices for each of the 8 S-boxes of size $4 \times 16$ of DES and the lone $16 \times 16$ S-box of AES have been implemented and analyzed.

For the analysis of confusion, SAC matrix includes the original set of input bits and all sets of input with every 1-bit alteration of original set of input bits. Individual SAC matrices have been generated for every S-box of DES and AES and the occurrences of 1's in the output have been calculated for each column of every S-box.

Further, the SAC matrix includes the original set of input bits and all sets of input with every 2-bit alteration of original set of input bits. Individual SAC matrices have been generated for every S-box of DES and AES and the occurrences of 1's in the output bit have been calculated for each column of every S-box.

In the above two methods, the SAC matrices being generated using the set of output bits are subsequently subjected to analysis of frequencies of various avalanche effects, analysis of Hamming weights according to the bit position and analysis of Coefficient of Variance.

Further in the research work, two more approaches have been introduced with the truncated differential cryptanalysis and boomerang-style attack on S-boxes of DES and AES.

In the truncated differential cryptanalysis approach, after generating the outputs of the original inputs to the S-boxes, the inputs are divided into two parts of same bit size, say $P(x_1, x_2)$. For the both parts of the truncated inputs, new $P(x_1', x_2')$ was generated by one bit alteration in each part and then combined. For the SAC matrix with original output and outputs corresponding to the new inputs, the occurrences of 1's in the output bit has been calculated and analyzed for each column of every S-box.

The Boomerang Attack is one of the trending attacks of the recent times. Boomerang attack has been used against well known encryption algorithms. In this research work, boomerang-style attack has been implemented with the confusion analysis of standard S-boxes of DES and AES. In this approach SAC matrix has been generated with output of all possible pair combination of original input.

Finally, the conclusion and recommendation has been drawn with respect to the statistical comparison on the confusion and diffusion of cryptographic algorithms. A comparative study has been done at the end of the thesis to get a clear view of whole work. It helps to draw the conclusion on the security levels of S-boxes with respect to confusion.

A comparative study of all experimental data for DES and AES has been included here to draw the conclusion.

# Table of Content

# List of Tables

# List of Figures

# List of Algorithms

# Chapter 1: Overview

## 1.1 Security of Cryptography

In the year of 1949 *Claude Shannon* published a paper entitled "*Communication Theory of Secrecy Systems*" in the *Bell Systems Technical Journal*. This paper had a great influence on the scientific study of cryptography. Some of the various approaches for evaluating the security of cryptosystems are considered here.

**Computational Security:** This measure concerns the computational effort required to break a cryptosystem. A cryptosystem is conceptually secure if the best algorithm for breaking it requires at least ***N*** operations, where ***N*** is a specified and very large integer. The problem is that no known practical cryptosystem can be proved to be secure under this definition. In practice, people often study the computational security of a cryptosystem with respect to certain specific type of attacks like *exhaustive key search*. Security against one specific type of attack does not ensure security against some other type of attack.

**Provable Security:** This approach is to provide evidence of security by means of a reduction. It shows that if the cryptosystem can be *broken* in some specific way, then it would be possible to efficiently solve some well-studied problem that is thought to be difficult. It may be possible to prove a statement of the type "*a given cryptosystem is secure if a given integer n cannot be factored*". Cryptosystem of this type are sometimes termed as provably secure.

**Unconditional Security:** This measure concern the security of cryptosystems when there is no bound placed on the amount of computation. A cryptosystem is defined to be unconditionally secure if it cannot be broken, even with infinite computational resources.

A cryptosystem has perfect secrecy if $P_r[X|Y] = P_r[X]$. For all $X \in P,\ Y \in C$.

That is, posteriori probability that the plaintext is $X$, given that the ciphertext $Y$ is observed is identical to a priori: probability that the plaintext is $X$.

## 1.2 Protocol of Cryptography

The whole point of cryptography is to solve problems. Cryptography solves problems that involve secrecy, authentication, integrity and dishonest people. The characteristics of protocol of cryptography are **[1]**:

- Everyone involved in the protocol must know the protocol and all the steps to follow.

- Everyone involved in the protocol must agree to follow it.
- The protocol must be unambiguous, each step must be well defined and there must be no chance of misunderstanding.
- The protocol must be complete; there must be a specified action for every possible situation.

A cryptographic protocol is a protocol that uses cryptography. It involves some cryptographic algorithms, but generally the goal of the protocol is something beyond simple secrecy. The whole point of using cryptography is a protocol is to prevent or detect eaves dropping and cheating.

**Arbitrated Protocols:** An *arbitrator* is a disinterested third party trusted to complete a protocol. Arbitrators can help to complete protocols between two mutually distrustful parties.

**Adjucated Protocols:** Because of the high cost of hiring arbitrators, arbitrated protocols can be sub-divided into two lower level sub-protocols. One is non-arbitrated protocol, and other is an arbitrated sub-protocol, executed only in exceptional circumstances – where there is a dispute. This special type of arbitrator is called an adjucator. An adjucator is also a disinterested and trusted third party. Unlike an arbitrator, it is not directly involved in every protocol.

**Self-enforcing Protocols:** A self-enforcing protocol is the best type of protocol. No arbitrator is required to complete the protocol. No adjucator is required to resolve the dispute. The protocol is constructed in a way so that there cannot be any dispute. If one of the parties try to cheat, the other party immediately detects the cheating and the protocol stops functioning.

## 1.3 Cryptographic Techniques

The cryptographic techniques are broadly discussed as followings **[22]**

**1.3.1 Key Length**: Key length of any cryptosystem may be measured in two ways:

- **Symmetric Key Length:** The security of symmetric cryptosystem is a combined function of two things: the strength of the algorithm and the length of the key. Assuming that the strength of the algorithm is perfect, there is no better way to break the cryptosystem other than trying every possible key in a brute-force attack.
  Calculating the complexity of a brute-force attack is easy. If the key is 8 bit long, there are $2^8$ or $256$ possible keys, and then it will take $256$ attempts to find the correct key with a 50% chance of finding the key after half of the attempt. The security of cryptosystem should rest in the key, not in the details of the algorithm.
- **Public-key Length:** Public-key cryptography uses the idea to make a trap-door-one way function. Actually that is a lie, factoring is conjectured to be a

hard problem. Today's dominant public-key encryption algorithms are based on the difficulty of factoring large numbers that are the product of two large primes. These algorithms are also susceptible to a brute-force attack, but of a different type. Breaking these algorithms does not involve trying every possible key, breaking these algorithms involve trying to factor a large number. If the number is too small then there is no security, if the number is large enough, it has security against all the computing power.

- **How long should a key be?** There is no single answer to this question. To determine how much security you need, you must ask yourself some question. The key length must be such that there is a probability of not more than 1 in $2^{32}$.
- **Key Generation:** Every security of an algorithm depends upon the key. If we use cryptographically weak process to generate key, the whole system will become weak. The generation of key may be handled by following means.
    a. **Reduced Key space.**
    b. **Random Keys.**
    c. **Pass Phrases.**
    d. **X9.17 Key Generation.**
    e. **DoD Key Generation.**

**1.3.2 Key Exchanges**: Although the asymmetric encryption algorithms are more secure than the symmetric types, they are also much slower and it is not feasible to use them to secure large quantities of data, as the consequent increase in transmission times would be excessive. Similarly, although chained mode of symmetric algorithms can process large quantities of plaintext at speed, they do not offer the requisite level of security because the key is a shared secret, that must be exchanged over the insecure medium prior to the transmission of the cipher-text. This paradox may be resolved as follows. A random secret, known as the *Session Key*, is generated and an asymmetric cipher secures this small piece of data for exchange over the Internet. A fast symmetric cipher then uses the *Session Key*, known only to the two security peers, to encrypt their exchanges of bulk data.

## 1.4 Objective of Work

The prime objective of this research work is to identify the demerits of the existing algorithms for testing the vulnerabilities of the encryption algorithms, especially S-boxes, and suggest improvement(s) or propose totally new test(s) for the same. To summarize, following are the main areas of investigation:

i. To study the already existing testing algorithms and analyze their relative strengths and weaknesses, and also to identify areas where there is scope of improvement.
ii. To suggest modifications, if possible, wherever the scope for improvement have been identified, so as to increase the strength of the underlying test.
iii. To propose new test wherever there is a scope.

### 1.4.1 Studying Block Ciphers: The Significance

Some public key algorithms such as RSA are capable of encrypting a block of as many bits as the size of the modulus - commonly 1024, 2048, or 4096 bits, dependent on the key. The problem is that most public key algorithms require a very large amount of CPU cycles to encrypt one block of data. In the case of RSA, the larger the modulus (the block size), the greater CPU cycles are required. RSA can take thousands or millions of times as many CPU cycles as a block cipher to encrypt the same amount of data. The slowdown is so significant that public key cryptography is often posed as the limiting factor of a system such as a web server. If encrypting every block of data requires that amount of CPU, the computer requirements for encrypting a stream of data would be prohibitive.

In comparison, symmetric key block ciphers are much more efficient in terms of speed. Given the amount of CPU it takes to encrypt 512 bytes of data with RSA, a symmetric block cipher such as AES would encrypt megabytes of data. The problem with symmetric algorithms is that of storing the keys securely, and the difficulty of exchanging keys with other people without the risk of interception. The public key algorithm is used only one time to encrypt a symmetric algorithm's key and the symmetric algorithm is then used to encrypt the data. The performance problem of public key cryptography suffers only once to exchange the keys, and the volumes of data are efficiently encrypted with a symmetric block cipher. Reasons for studying block ciphers may be summarized as:

- One Time Pads are believed to be the most secure cipher till date as long as the pad is used to encrypt to a single message. One Time Pads which are special purpose Stream Cipher are practically implemented less often as because the length of the key is as long as the message, but still it is considered to the most secure cipher. Block Ciphers may be implemented in a way to realize the powers of OTP.
- Though Asymmetric Ciphers are believed to be much stronger as compared to the Block Ciphers, they are mathematically very intense, which is the reason behind them being inherently slower as compared to the Block Ciphers, so much so that Asymmetric Ciphers are primarily used for key exchanges and not for actual data encryption [1].
- Kerckhoffs stated that for a secure cipher, "*it's key must be communicable and retainable without the help of written notes, and changeable and modifiable at the will of the correspondents"* [1]. But in case of Public Key Cryptography, the keys are of much greater length as compared to Block Ciphers. A comparison between the length of the keys required to achieve same level of security is listed in Table 1.1:

| Security Level (in bits) | Asymmetric Ciphers (RSA, Elgamal) | Block Ciphers (RSA, 3DES) |
|---|---|---|
| 80 | 1024 | 80 |
| 128 | 3072 | 128 |
| 192 | 7680 | 192 |
| 256 | 15380 | 256 |

Table 1.1: Security Levels of Various Ciphers

From the above discussion, it is justified why Block Ciphers are considered as the most fundamental building blocks for any modern cryptosystem, and, that is specifically the reason why the security of block ciphers is of great interest.

When a block cipher is used in a given mode of operation, the resulting algorithm should ideally be about as secure as the block cipher itself. ECB (*discussed in Sec. 1.4.4*) emphatically lacks this property: regardless of how secure the underlying block cipher is, ECB mode can easily be attacked. On the other hand, CBC mode can be proven to be secure under the assumption that the underlying block cipher is likewise secure. However, making statements like this require formal mathematical definitions for what it means for an encryption algorithm or a block cipher to "be secure". This section describes two common notions for what properties a block cipher should have. Each corresponds to a mathematical model that can be used to prove properties of higher level algorithms, such as CBC.

This general approach to cryptography---proving higher-level algorithms (such as CBC) are secure under explicitly stated assumptions regarding their components (such as a block cipher) --- is known as *provable security*.

**1.4.2 Stream Cipher vs. Block Cipher**

While both are symmetric ciphers, stream ciphers are based on generating an "infinite" cryptographic key stream, and using that to encrypt one bit or byte at a time (similar to the one-time pad), whereas block ciphers work on larger chunks of data (i.e. blocks) at a time, often combining blocks for additional security (e.g. AES in CBC mode).

- Stream ciphers are more difficult to implement correctly, and prone to weaknesses based on usage - since the principles are similar to one-time pad, the key stream has very strict requirements. On the other hand, that is usually the tricky part, they can be offloaded to external box.
- Because block ciphers encrypt a whole block at a time (and furthermore have "feedback" modes which are most recommended), they are more susceptible to noise in transmission, that is, if someone messes up one part, the rest of the data that helps to protect files and data in any web server, is provably unrecoverable. Whereas in stream ciphers there are bytes that are individually encrypted with no connection to other chunks of data (in most ciphers/modes), and often have support for interruptions on the line.

- Stream ciphers do not provide integrity protection or authentication also, but some block ciphers (depending on mode) can provide integrity protection, in addition to confidentiality.
- Because of all the above, stream ciphers are usually best for cases where the amount of data is either unknown, or continuous - such as network streams. Block ciphers, on the other hand are more useful when the amount of data is pre-known - such as a file, data fields, or request/response protocols, such as HTTP where the length of the total message is already known from the beginning and also used in most of the hand hold devices.

**1.4.3 Some Standard Cryptographic Algorithms**

In this section a few popular algorithms are discussed from different perspectives, clearly indicating the evolution in minimizing the chance of breaking ciphers. After all, these algorithms laid the foundation for efficient ciphers. Some of these algorithms have been even used before the advent of the computers.

**Data Encryption Standard (DES):** Up until recently, the main standard for encrypting data was a symmetric algorithm known as the *Data Encryption Standard (DES)* **[1]**. However, this has now been replaced by a new standard known as the *Advanced Encryption Standard (AES)*.

DES is a 64 bit block cipher which means that it encrypts data 64 bit at a time. This contrasted to a stream cipher in which only one bit at a time is encrypted. DES was the result of research project set up by IBM Corporation in the year of late 1960's which resulted in a cipher known as *LUCIFER*.

DES of course is not only symmetric cipher. There are many others, each with varying levels of complexity. Such ciphers include: IDEA, RC4, RC5, RC6 etc.

DES is based on a cipher known as Feistal block cipher. It consists of a number of round where each round contains bit shuffling, non-linear substitution (*S-box*) and *exclusive OR* operation. DES accepts two inputs – the plaintext to be encrypted and the secret key.

**Advanced Encryption Standard (AES):** All of the cryptographic algorithms discussed earlier have some problem. The earlier cipher can be broken with ease on modern computation system [1]. The DES algorithm was broken in 1998 using a system that cost about $250,000. Triple DES has three times, as many rounds as DES and is correspondingly slower. Moreover the 64 bit block size of triple DES & DES is not very efficient and is questionable when it comes to security. Then the requirement for any brand new algorithm was the resistant to all known attacks. The National Institute of Standards and Technology, United States (NIST) wanted to help in the creation of a new standard.

Like DES, AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. The block and key can be chosen independently from

128, 160, 192, 224, 224, 256 bits. The AES standard states that the algorithm can only accept a block size of 128 bit and any choice of three keys – 128, 192, 256 bits. As well as these differences AES differs from DES is that it is not feistal structure.

### 1.4.4 Modes of Operation

In cryptography, a mode of operation is an algorithm that uses a block cipher to provide an information service such as confidentiality or authenticity. A block cipher by itself is only suitable for the secure cryptographic transformation one fixed length group of bits called *block*. A mode of operation describes how to repeatedly apply a cipher's single block operation to securely transform data that are larger than a block. Most modes require a unique binary sequence, often called *Initialization Vector (IV)* of each encryption operation **[21]**.

**Electronic Codebook (ECB):** The simplest of the encryption modes is the *electronic codebook (ECB)* mode. The message is divided into blocks and each block is encrypted separately.

The disadvantage of this method is that identical plaintext blocks are encrypted into identical Ciphertext blocks; thus it does not hide the data pattern well. It does not provide serious message confidentiality and it is not at all recommended for use in cryptographic protocol.

**Cipher-block Chaining (CBC):** IBM invented the *cipher-block chaining (CBC)* mode of operation in 1976. In CBC mode each block of plaintext is *XOR-ed* with the previous Ciphertext block before being encrypted. This way each Ciphertext block depends on all plaintext blocks processes up to the point. To make each message unique, an initialization vector must be used in the first block. If the first block has index 1, the mathematical formula for CBC encryption is:

$$C_i = E_k(P_i \oplus C_{i-1}), C_0 = IV$$

While the mathematical formula for CBC decryption is:

$$P_i = D_k(C_i) \oplus C_{i-1}, C_0 = IV$$

CBC has been the most commonly used mode of operation.

**Propagating Cipher-block Chaining (PCBC):** The propagating or plaintext cipher-block chaining mode was designed to cause small changes in the Ciphertext to propagate indefinitely when decrypting, as well as when encrypting. Formulas for encryption and decryption algorithms are as follows:

$$C_i = E_k(P_i \oplus P_{i-1} \oplus C_{i-1}), P_0 \oplus C_0 = IV$$
$$P_i = D_k(C_i \oplus P_{i-1} \oplus C_{i-1}), P_0 \oplus C_0 = IV$$

**Cipher Feedback (CFB):** The cipher feedback (CFB) mode, a close relative of CBC, is a mode block cipher. Operation is very similar; in particular, CFB decryption is almost identical to CBC encryption performed in reverse.

$$C_i = E_k(C_{i-1}) \oplus P_i$$
$$P_i = E_k(C_{i-1}) \oplus C_i$$
$$C_0 = IV$$

**Output Feedback (OFB):** The output feedback (OFB) mode makes a block cipher into a synchronous stream cipher. It generates key stream blocks, which are then *XORed* with the plaintext blocks to get the Ciphertext. Just as with other stream ciphers, flipping a bit in the Ciphertext produce flipped bit in the plaintext at the same location. This property allows many error correcting codes to function normally even when applied before encryption.

$$C_j = P_j \oplus O_j$$
$$P_j = C_j \oplus O_j$$
$$O_j = E_k(I_j)$$
$$I_j = O_{j-1}, I_0 = IV$$

**Counter (CTR):** Like OFB counter mode turns a block cipher into a stream cipher. It generates the next key stream block by encrypting successive values of a counter. CTR mode is widely accepted, and the problems resulting out of the input function are recognized as weaknesses of the underlying block cipher. CTR mode has similar characteristics to CFB, but also allows a random access property during decryption.

## 1.5 Review of Existing Works

Going with one of the prime objectives of this research work, i.e., findings the merits and demerits of existing and well accepted cryptanalytic models, a narrative review has been performed. During the review work of related scholarly paper, two key feature of literature review has been followed, that are, review article and systematic review. The thorough review of scholarly papers including the subject of this research topic, helped to find the theoretical and methodological knowledge and contribution to this research topic.

Cristof Paar et. al. has defined **[1]** that two very important security principles for block ciphers are diffusion and confusion. *Diffusion* is an encryption operation where the influence of one plaintext symbol is spread over many ciphertext symbols with the goal of hiding statistical properties of the plaintext. A simple diffusion element is the bit permutation, which is used frequently within DES. On the other hand, *confusion* is an encryption operation where the relationship between key and ciphertext is obscured. Today, a common element for achieving confusion is substitution which found is in both DES and AES. Ciphers which only perform confusion or diffusion are not secure. By using the both operation, a strong cipher can be built and such ciphers are known as product ciphers.

In the research paper titled "On the statistical testing of Block Cipher" **[2]**, it was shown that how a cryptanalyst can use algorithms of a certain kind to attack block cipher and it has been established when a cryptanalyst cannot break the given block cipher. There are two basic problems that a cryptanalyst could attempt to solve and if cryptanalyst cannot solve at least one of these for a given block cipher, they cannot break this block cipher. These two basic problems are:

*a. To find an algorithm that is distinguishing for given block cipher.*
*b. To find an algorithm that is key-subset distinguishing for a given block cipher and for a given decomposition of the key space.*

In statistical hypothesis testing, probabilities will be unknown is almost a universal assumption.

In Kerckhoffs's principle, it is assumed that the cryptanalyst knows the entire mechanism of encipherment, except for the value of the secret key. During the course of attack, some questions may arise in front of cryptanalyst that for chosen plaintext block what will be the corresponding ciphertext and vice versa. For all question, the cryptanalyst may try to solve in the following way:

a. *Ciphertext block chosen uniformly at random for decryption.*
b. *Plaintext block chosen uniformly at random for encryption.*
c. *Finding of additional entry in function table.*
d. *Find the secret key*

An invertible function $f$ should be analyzed to solve the above mentioned 4 problems with any black box device that can compute an invertible function $f$ and its inverse $f^{-1}$. The deterministic algorithms help to entry in the function table of $f$ and additionally deterministic algorithm has access to random table that provides all the 'randomness' in the probabilistic algorithm. A random table can be loaded with a random string $R$ chosen according to a specified probability distribution $P_R$. A probabilistic algorithm for analyzing an invertible function can be applied to a randomly chosen encryption function of a block cipher $e$ where random variable $F$ and random string R are the inputs and random variable $A$ is the output. The encryption and decryption time do not change against the probabilistic algorithm for analyzing an invertible function. So, as concluded, the encryption and decryption time may be neglected and slow block cipher may have no advantage over a fast block cipher.

A cryptanalyst cannot break the block cipher $e$ if no computationally feasible probabilistic algorithm for analyzing an invertible function is known which solves for the block cipher $e$ the problem of:

- decrypting a ciphertext block chosen uniformly at random without asking the black box to encrypt it,
- finding an additional entry in the function table of the encryption function,
- finding the secret key.

The cryptanalyst can break the block cipher $e$ if he knows a computationally feasible probabilistic algorithm for analyzing an invertible function that solves the block cipher $e$ at least for one of these three problems.

Statistical testing of block ciphers is intended to provide tests that are capable of analyzing any practical block cipher, no matter what the internal structure of the block cipher may be. Therefore such tests should analyze a block cipher based only on the input-output-behavior for its bivariate function *e*.

A cryptanalyst can use statistical testing of a block cipher as a first step towards breaking a block cipher. After using several tests on block cipher, if some of these tests show a non-ideal behavior of the block cipher to see what caused the non-ideal behavior. This might give him ideas how he could break the block cipher. Statistical testing helps the cryptographer to ensure that the designed block cipher that does not have any weaknesses.

Several models have been proposed including the "*model for a probabilistic algorithm for extracting a feature from a sequence of invertible functions*", "*a model for independent execution of a probabilistic algorithm for extracting a feature from a sequence of invertible function and analysis of extracted features for randomly chosen encryption functions of a block cipher e*", "*a model for statistical testing of a block cipher e*".

The block cipher $e$ being tested with block length $N$ and key space $Z_e$, and the block cipher $e^{\perp}$ with block length $N$ and key space $Z_{e^{\perp}} = Z_e$, are duals if the encryption function $e_{z^{\perp}}$ of the block cipher $e^{\perp}$ is identical to the decryption $e_{z^{\perp}}$ of the block cipher $e$ for every secret key $Z$ in the key space $Z_e$.

The derivation of block cipher can help to analyze the models that have been proposed. Those models show that instead of looking directly for a probabilistic algorithm for analyzing an invertible function that is key-subset distinguishing for the given block cipher *e*, it is preferred to look for a probabilistic algorithm for analyzing an invertible function that is distinguishing for some of the reduced-key-space versions of the given block cipher *e*.

Knudsen and Mathiassen **[3]** have considered iterated ciphers and their resistance against linear and differential cryptanalysis. It is shown by experiments that cipher with complex key schedules resists both the attack better than ciphers with more straightforward key schedules. It is presented in experiment to illustrate that some iterated ciphers with very simple key schedules will never reach this uniform distribution. It is also shown that cipher with well-designed, complex key schedules reach the uniform distribution faster using fewer rounds than ciphers with poorly designed key schedules.

The author believes that there exists cipher for which the differential of the highest probability for one fixed key is also the differential of the highest probability for any other key and it shown as a side result. The experimental results showed that the

uniform distribution is reached faster for the 10-bit and 12-bit block ciphers than for the 8-bit block ciphers. A good and complex key schedule therefore help to make a cipher more resistant to differential and linear attack.

According to W.S. Forsyth and R. Safavi-Naini **[4]**, in a ciphertext attack it is always possible to test every possible key. This is called exhaustive key search. For any alphabet of size *N* there are $N! > N^{n/2}$ possible substitutions and hence the size of the substitution space increases exponentially in proportion to the size of plaintext alphabet. An algorithm for finding the affine mapping from plaintext to the ciphertext would include:

- Finding the *N*-gram relative frequencies of the sample ciphertext.
- Matching the frequencies against those of the plaintext language and suggesting a key.
- Verifying the decryption obtained by using the suggested key.

Testing likely keys is a non-trivial test by itself and has traditionally required dictionary search, pattern matching and human assistance to except/reject a cryptogram decrypted under a suggested key.

Forsyth and Safavi-Naini formulate the cryptanalysis of the substitution cipher as a combinatorial optimization problem and use simulated analysis to find the optimal solution which corresponds to the affine mapping used encrypt the plaintext alphabet. This approach appealing as it completely eliminates human intervention and does not require any sophisticated pattern matching technique. It also provides an elegant way of solving substitution cipher which is also promising for block cipher algorithm.

G. Piret and F.-X. Standaert **[5]** showed their concern with the security of block cipher against the linear cryptanalysis and discussed the distance between the so-called practical security approach and the actual theoretical security provided by a given cipher. The comparison has been performed between the linear probability of the best linear characteristic and the actual best linear probability. A test is also done for the key equivalence hypothesis. An experiment to evaluate the relevance of the *use of characteristics for arguing the security of a construction* as defined by Knudsen et.al.**[3]** These experiment highlight another aspect of the practical security approach: if the best linear approximation of a given cipher is key-dependent, it can hardly be exploited by an actual adversary. All these have been discussed based on the (im)possibility to derive practical *design criteria* for block cipher.

Their experiments only considered Substitution Permutation Network (SPN), but similar investigation can be considered on the Feistal ciphers.

In order to evaluate the extent to which the practical security approach is meaningful for actual block ciphers, they first computed following quantities:

$$max_{char} := max_{\Omega} ELCP(\Omega)$$

$$max_{hull} := E_{Ě} max_{a,b} LP(a, b; Ě),$$

for various SPNs and as result the following facts are observed:

- After sufficient number of rounds, the average best approximation of a given cipher only depends on its block size *n* and suggested that the average best linear probability of 16-bit and 12-bit ciphers are $6.30 \times 10^{-4}$ and $7.44 \times 10^{-3}$, respectively.
- The probability of the best characteristic goes on decreasing with the number of rounds *R*.
- The value of $E_{Ě}$ max LP is faster achieved with 8-bit S-box than 4-bit ones.
- Linear probability increases when one more round is added.

A new statistical test for randomness, the *strict avalanche criterion* (SAC) test, is there, together with its result over some well known generators in the literature is given and analyzed by J.C.H. Castro et.al **[6]**. The avalanche effect was originally proposed for s-boxes by Webster and Tavares in 1986 **[7]**.

The SAC was further generalized by R. Forre **[8]**. The SAC act as a generalization of the avalanche effect, but not formulated in concrete terms, in early works in the field of cryptography. The avalanche effect tries to reflect, to some extent, the intuitive idea of high-nonlinearity: a very small difference in the input producing a high change in the output, thus an avalanche of changes. Mathematically:

$$\forall x, y \mid H(x, y) = 1, average\ \left(H\big(F(x),\ F(y)\big)\right) = {}^{n}/_{2}$$

So, if *F* is to have the avalanche effect, the Hamming distance between the outputs of a random input vector and are generated by randomly flipping one of its bits should be, on average ${}^{n}/_{2}$. Forre presented a result obtained with the SAC test over a number of well-known pseudo-random number generators using different lengths, from 8 to 128 bits, and marked the results that have corresponding *p*-values less than 0.01 and thus proved a failure for the generator to pass the test.

Applying tests of randomness to block ciphers the cipher will be viewed as a black box such that the actual algorithm used is unknown, the only information being the block size for both the message and key **[9]**. Plaintext that appears random will generally produce ciphertext which also random. Therefore, the application of randomness measures need to indicate that there is no relationship between the plaintext and ciphertext block, i.e. the plaintext is independent of the ciphertext. In order to test this hypothesis, a large number of blocks of length *n* were examined for randomness. Two different ways to generate such set of blocks are:

- Non-random (patterned) plaintext as input and then corresponding ciphertext are tested for randomness.

- Purely random plaintext blocks are combined with the corresponding ciphertext blocks under bitwise modulo-two addition and then tested for randomness.

The Hamming weight, runs, linear complexity, sequence complexity are derived during randomness measures on whole block.

For randomness measures on block differences, for each ciphertext bit position, the strength of the Boolean function may be investigated by measuring the change in the ciphertext bit when subsets of input bits are complemented. This may be expressed as $F(P,K) \oplus F(P \oplus H_i, K)$where P is randomly chosen from the set of *n*-tuples, K is the constant key and $H_i$ is an n-bit vector having hamming weight of *i*, i.e. $W(H_i) = i$.

The property of SAC may be applied to the complementation of plaintext bits as the *strict plaintext avalanche criterion (SPAC)*. For a fixed key, each bit of the ciphertext block changes with the probability of one half whenever a single bit of plaintext block is complemented. This property is applied to key changes where a block cipher satisfies the *strict key avalanche criterion (SKAC)*.

For the independent measures on sub-blocks, a block cipher algorithm aims to combine the elements of the plaintext and key using confusion and diffusion techniques. The subset of plaintext bit will be concatenated with the subset of ciphertext bits to give a combined subset of $l = p + c$ bits for testing. When $l$ is small the classical test of uniformity is applied and for larger l a new test, involving the *Poisson* approximation to the classical occupancy problem distribution and following tests are introduced:

- Small sub-blocks and uniformity test.
- Large sub-blocks and repetition test.

In the paper titled '*Statistical Analysis of Block Cipher*' **[10]** it is stated that diffusion and confusion are the two important principle of security for block cipher. For diffusion, a little change in plaintext or key should result in massive change to the ciphertext i.e. each ciphertext bit depends on each bit of plaintext. Completeness and avalanche criterion are the measures of diffusion and both have been combined to define strict avalanche criterion (SAC). In SAC a change in a single bit of plaintext result in the change of each output with probability $1/2$ over all possible key and plaintext combination. Moreover, satisfying SAC property for encryption does not imply that it is satisfied for decryption.

The paper also enlists few tests under Distinguishing Properties Tests, all of which use fixed key. The Distinguishing Properties tests include:

1) Frequency test that examines the effect of weight of plaintext blocks in the weight of cipher test blocks or vice versa. In this test it is tested whether input with low or high density affect the output weight. The Chi-square test is applied to analyze the result.

2) Run test, here as input large/small numbers of runs are picked and its effect on the number of run in the output are analyzed again using chi-square technique.

3) Alphabetic character test, where, in the frequency of the alphabets (both upper and lower cases) are observed in the ciphertext generated from the plaintext. For this purpose, the ciphertext is broken into group of 8 bits in both overlapping and non-overlapping fashion.

The test detail of SAC states that it is impossible to test the security of the cipher for each key value but it is possible, with a very low probability, to identify weak key classes. The SAC test can be summarized that the hypothesis is plaintext and ciphertext blocks are not correlated when different types of keys in terms of their weight are used. The test for SAC is enough to test the diffusion principle, and can be concluded if a block cipher satisfies SAC, this means that it is also satisfies the completeness and the avalanche criteria.

In the thesis by Sreenivasulu Nagireddy **[11],** it was stated that for the cryptanalysis of DES than any other block cipher, the most practical attack is still a brute-force approach. There are three attacks known that can break the full sixteen rounds of DES with less complexity than a brute-force attack: differential cryptanalysis (DC), linear cryptanalysis (LC) and Devies' attack. However, these attacks are theoretical and are not feasible to mount in practice.

The best attack known on 3key TDES requires around $2^{32}$ known plaintext, $2^{113}$ steps, $2^{90}$ single DES encryption and $2^{88}$ bit memory **[12]**. This is not practically feasible at present. TDES is slowly disappearing from use, largely replaced by the Advance Encryption Standard (AES). TDES suffers from slow performance in software and AES tends to be around 6 times faster than the earlier.

The AES, also known as Rijndeal is a substitution-permutation network, not feistal network and fast in both software and hardware. So far, the only attack against AES implementations have been side channel attacks. By this attack, it is not possible to attack the underlying cipher, but attack implementations of the cipher on systems which inadvertently leak data.

According to Xuejia Lai and James L. Massey **[13]**, they considered the encryption of pair of distinct plaintext by an r-round iterated cipher, where the round function $Y = f(X, Z)$ is such that, for every round sub key $Z$, $f(., Z)$ establishes a one-to-one correspondence between the round input $X$ and round output $Y$. They assumed the difference $\triangle X$ between two plaintext (or two ciphertext) $X$ and $X^*$ is defined as $\triangle X = X \otimes X^{*-1}$, where $\otimes$ denotes a specified group operation on the set of plaintext and $X^{*-1}$ denotes the inverse of the element $X^*$ in the group. The round function $Y = f(X, Z)$ is said to be cryptographically weak if, given a few triples$(\Delta X, Y, Y^*)$, it is feasible to determine the sub key $Z$.

It has been summarized the basic procedure of differential cryptanalysis attack on an r-round iterated cipher as:

1) Find an (r-1)-round differential $(\alpha, \beta)$ such that $P(\Delta Y(r-1)) = \beta \mid \Delta X = \alpha$ has maximum or nearly maximum probability.

2) Choose a plaintext $X$ uniformly at random and compute $X^*$ so that the difference $\triangle X$ between $X$ and$X^*$ is$\alpha$. Submit $X$ and $X^*$ for encryption under the actual key $Z$. From the resultant ciphertexts $Y(r)$ and $Y^*(r)$, find every possible value of the sub key $Z^{(r)}$ of the last round corresponding to the anticipated difference is $\Delta Y(r-1) = \beta$. Add one to the count of the number of appearances of each such value of the sub key is $Z^{(r)}$.

3) Repeat (2) until one or more values of the sub key $Z^{(r)}$is counted significantly more often than the others. Take this more-often-counted sub key as the cryptanalyst's decision for the actual sub key is $Z^{(r)}$.

It is noted that, in a differential cryptanalysis attack, all the sub keys are fixed and only the plaintext can be randomly chosen.

The terminology "Markov Cipher" is been explained by the theorem as:

*If an r-round iterated cipher is a Markov cipher and the r-round keys are independent and uniformly random, then the sequence of difference* $\Delta \boldsymbol{X} = \Delta \boldsymbol{Y}(\mathbf{0})$, $\Delta \boldsymbol{Y}(\mathbf{1}), \ldots., \Delta \boldsymbol{Y}(\boldsymbol{r})$ *is a homogeneous Markov chain. Moreover, this Markov chain is stationary and* $\Delta \boldsymbol{X}$ *is uniformly distributed over the non- neutral elements of the group.*

In the research article "Cryptographic Randomness Testing of Block Cipher and Hash Function" **[14]** another version of SAC test was proposed along with three other tests namely 'linear span test', 'correlation test' and 'coverage test'. In this version of SAC test, a SAC matrix is prepared which is similar to the matrix prepared in **[10]** but, in this case unlike **[10]** $2^{20}$ randomly chosen plaintext are used instead of an arbitrary $m$ number of plaintext. Since $2^{20}$ numbers of randomly chosen plaintext are used, each of the entries in SAC matrix is expected to have a value close to $2^{19}$ in order to have a probability of ½. After preparing the matrix, chi-square goodness of fit test is applied to evaluate the distribution of the values of the entire matrix, if the matrix produces a $p - value$ less than $0.01$, then it is considered as non-random. Next to catch any correlation between a particular input bit and a particular out bit entry outside a particular range $2^{20} - 5009, 2^{20} + 5009$ are flagged and the test is applied once again. If the flagged entries deviate from the expected value once more significantly, it indicates that a specific input bit and a specific bit are correlated, which in term indicate a weakness in the underlying cipher.

In the linear span test, non-linearity of the underlying block cipher is tested by producing an input set of size $m = 2^t$ obtained from $t$ independent plaintext. After obtaining the input set, a $m \times m$ matrix is obtained from the corresponding ciphertext

and the rank of matrix is calculated and compared with the rank of a random binary matrix. After rank is being obtained, the corresponding binary value is incremented by one. This process is repeated as many times as possible and the resulting binary values are then put through the chi-square goodness of fit test to produce $p - value$. If the $p - value$ is less than $0.01$, it indicates a non-random mapping.

A block cipher should produce random looking outputs, the outputs randomness is generally evaluated using a pseudo-random number generator (PRNG). One of the most commonly used randomness test suites for PRNG is the Diehard test suite **[15]** which was considered to be the best test suite for PRNG until the authors came up with the concept in "Some difficult-to-pass tests for randomness" **[16]**. On the other hand, another statistical test suite was designed and used by NIST **[17]** in order to evaluate the randomness criteria of the AES finalist. Although the NIST test suite was designed to test the randomness characteristics of the block cipher, in fact it is a general purpose test suite for evaluating the randomness of a binary string which may come from any source, the NIST test suite describes 15 statistical tests and it also describes the implementation details for each of these tests. Generally there are two issues which are needed to be addressed while evaluating a block cipher. The first issue which needs to be addressed is that block ciphers are not PRNGs by itself and they do not generate arbitrary long binary strings. In this regard, NIST test suite is not well suited for evaluating block ciphers, it would be better if some statistical test directly evaluate the randomness of the block cipher mappings, without aiming it to turn the block cipher to behave as a PRNG.

The second issue with most statistical tests is that they take a frequentist approach to statistical testing. In other words, it can be said that while analyzing a binary string, each test computes a statistic and a $p - value$ . Now even if the binary string being analyzed is actually random, if the $p - value$ falls below a threshold, it is assumed that the string is not random. It is assumed if the underlying PRNG is random, $p - value$ should be uniformly distributed between 0 to 1, so, the NIST test suite applies a *second level* statistical test to the $p - value$ for each *first level* test to determine whether the test $p - values$ are randomly distributed. From the $p - values$, the analyst has to decide whether the binary string and the underlying block cipher that produced the string holds good randomness property or not. The frequentist approach does not specify a procedure for combining multiple $p - values$ into a single number that yields an overall random/non random decision.

## 1.6 Justification of Proposed Research Work

After a thorough study of existing works of various cryptanalysis, it is found that there is very limited work of cryptanalysis has been done towards the strength analysis of S-boxes of different cryptographic algorithms.

In this proposed research work, S-boxes are analyzed with some novel approaches to measure the strength against the attacks. Different type of statistical methods has been used with the proposed algorithms to establish the conclusion.

The major aim of the proposed research work is to establish a testing suite for the existing S-boxes with the help of novel algorithms.

## 1.7 Cryptographic Standards

There are number of standards related to cryptography are available. Standard algorithms and protocols provide a lot of help for research study and attract to large amount of cryptanalysis. Some of the well known standards are being reviewed during the preparation of this research work.

### 1.6.1 Encryption Standards

#### *1.6.1.1 Data Encryption Standard*

Data Encryption Standard (DES) is a symmetric algorithm for encrypting data of recent time. But DES now has been replacing by another well known new standard algorithm called Advance Encryption Standard (AES) **[43]**. DES is the result of changes, made by IBM, requested by NBS (now NIST), on popular cipher LUCIFER. The notable change in DES on LUCIFER is the key size, which is reduced from 128-bits to 56-bits. DES actually accepts 64-bits key as input and remaining 8-bits for parity checking. Biham and Shamir **[44]** publicly discovered the concept of S-boxes which is appeared secure against an attack called Differential Cryptanalysis.

DES is based on Feistal Block Cipher, Developed by IBM researcher Horst Feistal. It consists of a number of rounds where each round contains bit-shuffling, non-linear substitutions (S-boxes) and XOR operations. As DES is 64-bits block cipher, if the number of bits in the message is not evenly divided by 64 then the last block will be padded. To increase the difficulty of cryptanalysis, multiple permutations and substitutions are there in DES. The sequence of events that occur during an encryption operation using DES is shown in Figure 1.1.

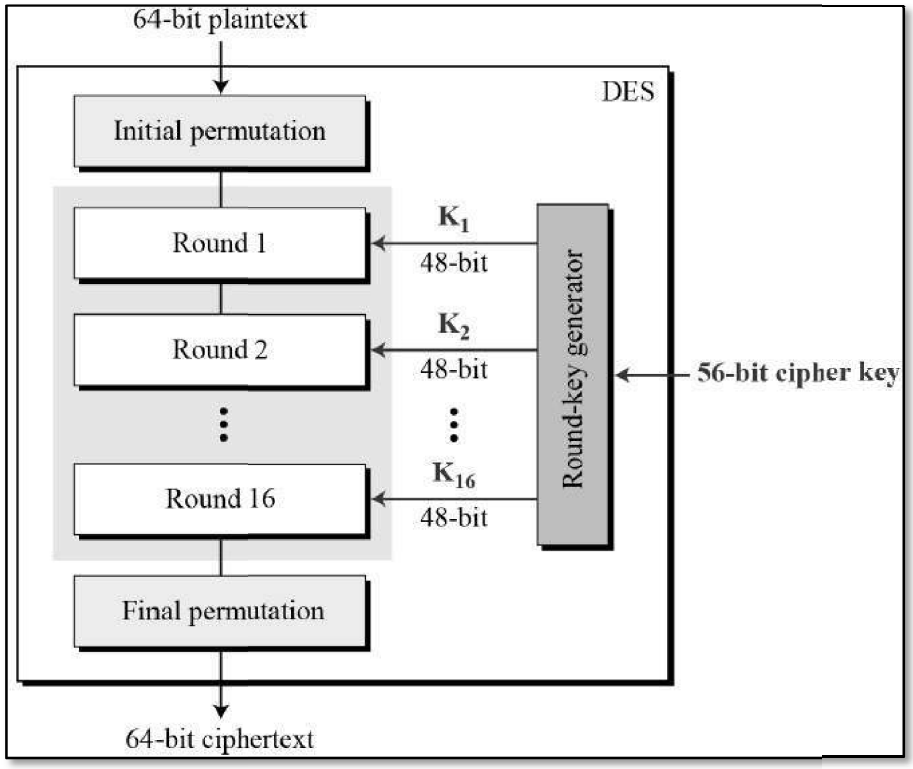


***Figure 1.1: Flow Diagram of DES for Encryption Data***

The Figure 1.2 shows the initial and final permutations (P-boxes) of DES. Each permutation takes 64-bit input and permutes them according to predefined rule. The permutation rules of these P-boxes are shown in Table 1.2.

The S-boxes of DES does the real confusion. DES uses 8 S-boxes, each with 6-bit input and a 4-bit output.

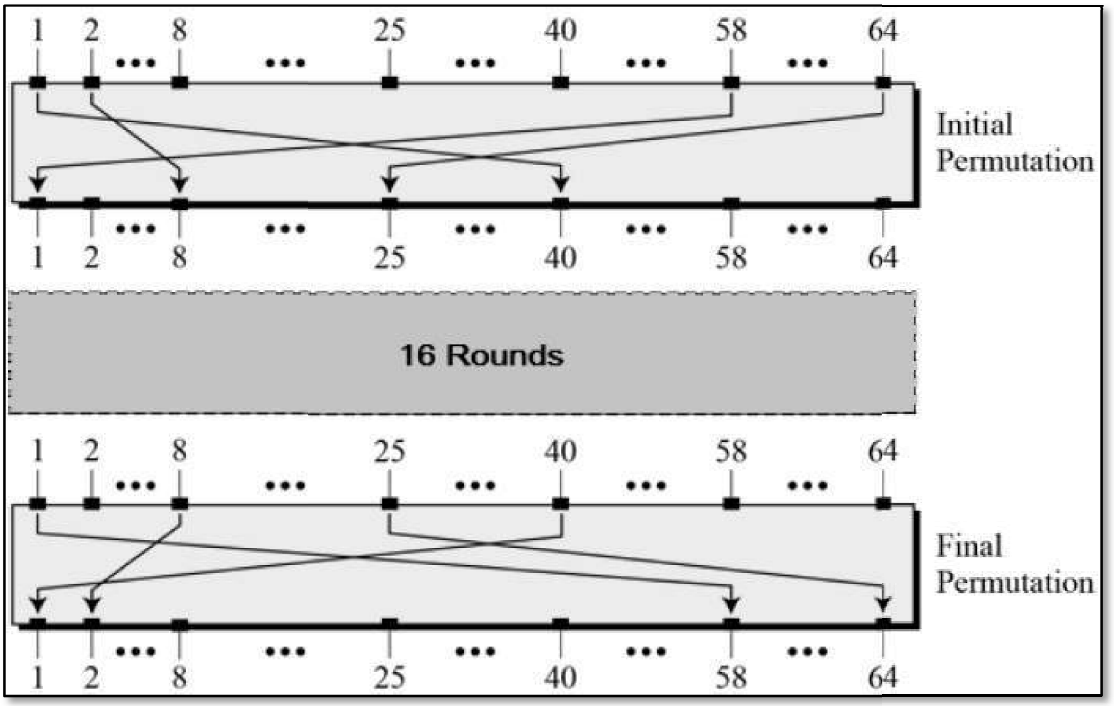


***Figure 1.2: Initial and final permutation steps in DES***

| Initial Permutation | | | | | | | | Final Permutation | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 02 | 40 | 08 | 48 | 16 | 56 | 24 | 64 | 32 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 04 | 39 | 07 | 47 | 15 | 55 | 23 | 63 | 31 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 06 | 38 | 06 | 46 | 14 | 54 | 22 | 62 | 30 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 08 | 37 | 05 | 45 | 13 | 53 | 21 | 61 | 29 |
| 57 | 49 | 41 | 33 | 25 | 17 | 09 | 01 | 36 | 04 | 44 | 12 | 52 | 20 | 60 | 28 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 03 | 35 | 03 | 43 | 11 | 51 | 19 | 59 | 27 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 05 | 34 | 02 | 42 | 10 | 50 | 18 | 58 | 26 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 07 | 33 | 01 | 41 | 09 | 49 | 17 | 57 | 25 |

**Table 1.2. Initial and Final Permutation Table**

The decryption of DES is same as encryption process; it uses the ciphertext as input to DES algorithm and use the key $K_i$ in reverse order.

#### *1.6.1.2 Triple DES*

The modified scheme Triple DES came in practice as a result of discomfort of users against the exhaustive key searching in DES. There are two variants of Triple DES known as 3-key Triple DES (3TDES) and 2-key DES (2TDES).

1. 3-Key Triple DES (3TDES): The key ($K$) of 3TDES is consists of three different DES keys $K_1$, $K_2$, $K_3$. The key length of $K$ is $3 \times 56 = 168$ bits.
2. 2-Key Triple DES (2TDES): The 2TDES scheme is mostly same as 3TDES except that $K_3$ is replaced by $K_1$, therefore the key length for 2TDES is $2 \times 56 = 112$ bits.

#### *1.6.1.3 Advanced Encryption Standard*

The 64-bit DES and triple DES is not very efficient and questionable when it comes to security. NIST chose the algorithm known as Rijndeal [44] which is then named as AES as standard block cipher cryptographic model. The salient features of AES are as follows:

- AES is a block cipher with a block length of 128 bits.
- AES allows 3 different key lengths: 128, 192 and 256 bits.
- Encryption consists of 10 rounds for 128 bits key, 12 rounds for 192 bits key and 14 rounds for 256 bits key.
- Except the last round in each case, all other rounds are identical.
- Each round of processing includes one single-byte based substitution step, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. The order in which these four steps are executed is different for encryption and decryption.
- Like DES, AES is an iterated block cipher in which plaintext is subject to multiple rounds of processing, with each round applying the same overall transformation function to the incoming block.
- Unlike DES, AES is an example of key-alternating block ciphers. In such ciphers, each round first applies a diffusion-achieving transformation operation — which may be a combination of linear and nonlinear steps — to the entire incoming block, which is then followed by the application of the round key to the entire block. As you'll recall, DES is based on the Feistel structure in which, for each round, one-half of the block passes through un-changed and the other half goes through a transformation that depends on the S-boxes and the round key. Key alternating ciphers lend themselves well to theoretical analysis of the security of the ciphers.

## 1.8 Action Plan of the Research Work

Analysis of block ciphers has been done efficiently, encrypted with well-known encryption methods like DES and AES, till date, and reviewed minutely in *1.5*. There are lots of excellent research works on the:

a) Statistical testing on block ciphers,

b) On the role of key schedules in attack on iterated ciphers,

c) Automated cryptanalysis on substitution cipher,

d) SAC randomness test including SPAC and SKAC,

e) A pattern recognition approach to block cipher identification, and

f) Many distinct ways of randomness tests.

All statistical approach has concentrated on computing $p - value$ and its randomness. This approach has been identified as one of the scopes of this research work.

The research work looks forward to make it feasible to cover both linear and differential cryptanalysis approaches. The bit level block cipher diffusion and confusion analysis are the key areas to be covered in this research work. The core intension of this research work is to establish a standard algorithmic test suite on block ciphers to test most of the internationally recognized encryption methods. Both statistical and randomness tests are taken into consideration to develop the suite.

# Chapter 2: A Look into Cryptanalysis

## 2.1 Security Goals

In this informative age, there is a need to keep information about every aspect of our life. Information is extremely valuable like any other asset. Moreover, it is obvious, like other assets, information need to be secured from *attacks*. Until a few decades ago, the information collected by an organization was stored as physical files. The confidentiality of the file are taken care of by restricting the access among few authorized and trusted people within the organization.

With the advent of computers, information storage became electronic. Instead of being stored in the physical form, it is stored in computers and related devices. To make information secure, it should be protected and kept hidden from unauthorized access during both storage and transmission. To keep information secure, following three requirements need to be maintained:

**Confidentiality:** Confidentiality is the concealment of information or resources. During the last two decades, computer network created a revolution in the use of information. The need for keeping information secrecy arises from the use of computers in sensitive areas like government and corporate sectors etc.

**Integrity:** Integrity refers to the trustworthiness of data or resources and it usually deals with in terms of preventing unwanted and unauthorized changes. Integrity includes *content integrity* and *source integrity*. Working with integrity is absolutely different from working with confidentiality. With confidentiality, the data is either compromised or not, but integrity includes both the correctness and trustworthiness of data.

**Availability:** Availability is the ability to use the desired information or resource. It is an important aspect of reliability and system design. The aspect of availability that is relevant to security is that someone may deliberately arrange to deny access to some data or to a service by making it unavailable. Attempts to block availability are termed *denial of service attack* that can be most difficult to detect.

Cryptographic algorithms are designed to meet the above security goals.

## 2.2 Cryptanalysis

The goal of cryptanalysis is to find the weaknesses or insecurity in a cryptographic scheme, thus permitting its subversion or evasion. It is common misconception that every encryption method can be broken. In his *WWII* work at *Bell Labs, Claude Shannon* proved that one-time-pad cipher is unbreakable, provided the key material is

truly random, never reused, kept secret from all possible attackers and of equal or greater length than the message. *Cryptanalysis is the science of recovering the plaintext of a message without access to the key. Successful cryptanalysis may recover the plaintext or the key* **[1]**.

An attempt of cryptanalysis is called an *attack*. Attacks can be divided into two broad categories:

**Cryptanalytic Attack:** There are three general types of cryptanalytic attack **[41]**.

i. Ciphertext-only Attack: In this attack the attacker knows only the ciphertext to be decoded. The attacker will try to find the key or decrypt one or more pieces of ciphertext (only relatively weak algorithms fail to withstand a ciphertext-only attack).
ii. Known-plaintext Attack: The attacker has a collection of plaintext-ciphertext pairs and is trying to find the key or to decrypt some other ciphertext that has been encrypted with the same key.
iii. Chosen-plaintext Attack: This is a known plaintext attack in which the attacker can choose the plaintext to encrypted and read the corresponding ciphertext.

*There are at least two other types of cryptanalytic attack.*

i. Chosen-ciphertext Attack: The attacker has the able to select any ciphertext and study the plaintext produced by decrypting them.
ii. Chosen-key Attack: The attacker has the abilities required in the Chosen-plaintext and Chosen-ciphertext attacks.

An encryption scheme is *unconditionally secure* if the generated ciphertext does not contain enough information to determine uniquely the corresponding plaintext. Except onetime pad, no cipher is unconditionally secure.

The security of a *conditionally secure* algorithm depends on the difficulty in reversing the underlying cryptographic problem. Other than the one-time pad, all other ciphers fall into this category.

An encryption scheme is said to be *computationally secure* if:

a. The breaking of cipher is costly than the cost of the encrypted information.
b. The breaking of cipher is more time consuming than the useful lifetime of the information.

**Non-cryptanalytic Attack:** The non-cryptanalytic attacks do not exploit the mathematical weaknesses of the cryptographic algorithm. However, the goals of security can very much threatened by this class of attack. Figure 2.1 shows the taxonomy.

### 2.2.1 Differential Cryptanalysis

One of the most significant advances in cryptanalysis in recent years is differential cryptanalysis. Although this appears to have been discovered at least 30 years ago it was not reported in the open literature until 1990. The first published effort appears to have been the cryptanalysis of a block cipher called FEAL. This was followed by a number of papers by Biham and Shamir.

Differential cryptanalysis is the first popular attack that is capable of breaking DES in less than $2^{55}$ complexity.

Differential cryptanalysis exploits the high probability of certain occurrences of plaintext differences and differences into the last round of the cipher **[26]**. For example, consider a system with input $X = [X_1\, X_2 \ldots\ldots X_n]$ and output $Y = [Y_1\, Y_2 \ldots\ldots Y_n]$. The input difference is given by $\triangle X = X' \oplus X''$ where $\oplus$ represents a bit-wise exclusive-OR of $n - bit$ vectors. Hence, $\triangle X = [\triangle X_1\ \triangle X_2 \ldots.\triangle X_n]$. Similarly $\triangle Y = Y' \oplus Y''$ is the output difference and $\triangle Y = [\triangle Y_1\ \triangle Y_2 \ldots.\triangle Y_n]$. Differential cryptanalysis seeks to exploit a scenario where a particular $\triangle Y$ occurs given a particular input difference $\triangle X$ with a very high probability $p_D$. The pair $\triangle X$ & $\triangle Y$ is referred to as a differential cryptanalysis.

Differential cryptanalysis is a chosen plaintext attack, meaning that the attacker is able to select inputs and examine outputs in an attempt to derive the key.

### 2.2.2 Linear Cryptanalysis

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits (actually bits from the 2nd last round output are used), and subkey bits. It is a known plaintext attack: that is, it is premised on the attacker having information on a set of plaintexts and the corresponding ciphertexts. However, the attacker has no way to select which plaintexts (and corresponding ciphertexts) are available. In many applications and scenarios, it is reasonable to assume that the attacker has knowledge of a random set of plaintexts and the corresponding ciphertexts. The basic idea is to approximate the operation of a portion of the cipher with an expression that is linear where the linearity refers to a mod-2 bit-wise operation (i.e., exclusive-OR denoted by "⊕"). Such an expression is of the form:

$$X_{i_1} \oplus X_{i_2} \oplus \ldots \oplus X_{i_u} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \ldots \oplus Y_{j_v} = 0$$

where $X_i$ represents the $i^{th}$ bit of the input $X = [X_1, X_2, \ldots]$ and $Y_j$ represent the $j^{th}$ bit of the output $Y = [Y_1, Y_2, \ldots]$. This equation is representing the exclusive-OR "sum" of $u$

input bits and $v$ output bits. The approach in linear cryptanalysis is to determine expressions of the form above which have a high or low probability of occurrence.

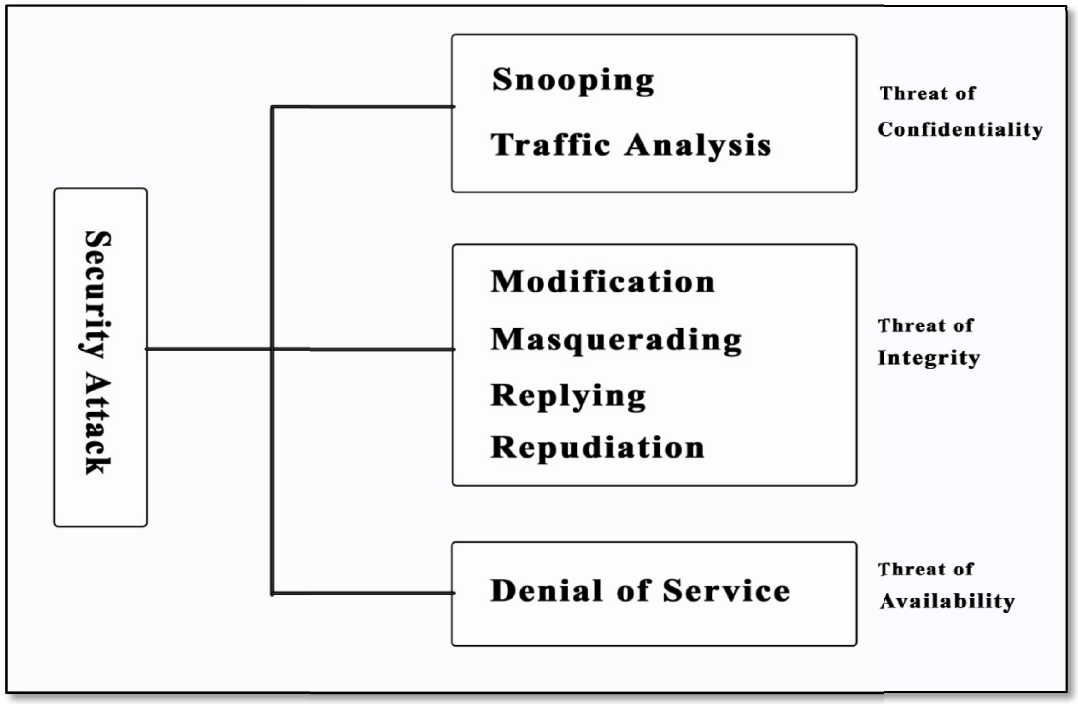


*Figure 2.1: Taxonomy of Non-cryptanalytical Attack*

## 2.3 More Attacks on Cryptosystem

Some more methods of attack on symmetric ciphers like DES and AES are discussed below:

### 2.3.1 Birthday Attack

The Birthday attack is a use of Linear Cryptanalysis, where it tries to attack cryptographic hash function by using Birthday paradox. The Birthday paradox can be stated as **[41]**:

What is the minimum value of $k$ such that the probability is greater than $0.5$ that at least two people in a group of $k$ people have same birthday? The answer is $23$ which quite a surprising result is. If there are $100$ people $(i.e. k = 100)$ then the probability is $.9999997$ and the graph of the probabilities against the value of $k$ is shown in Figure 2.2

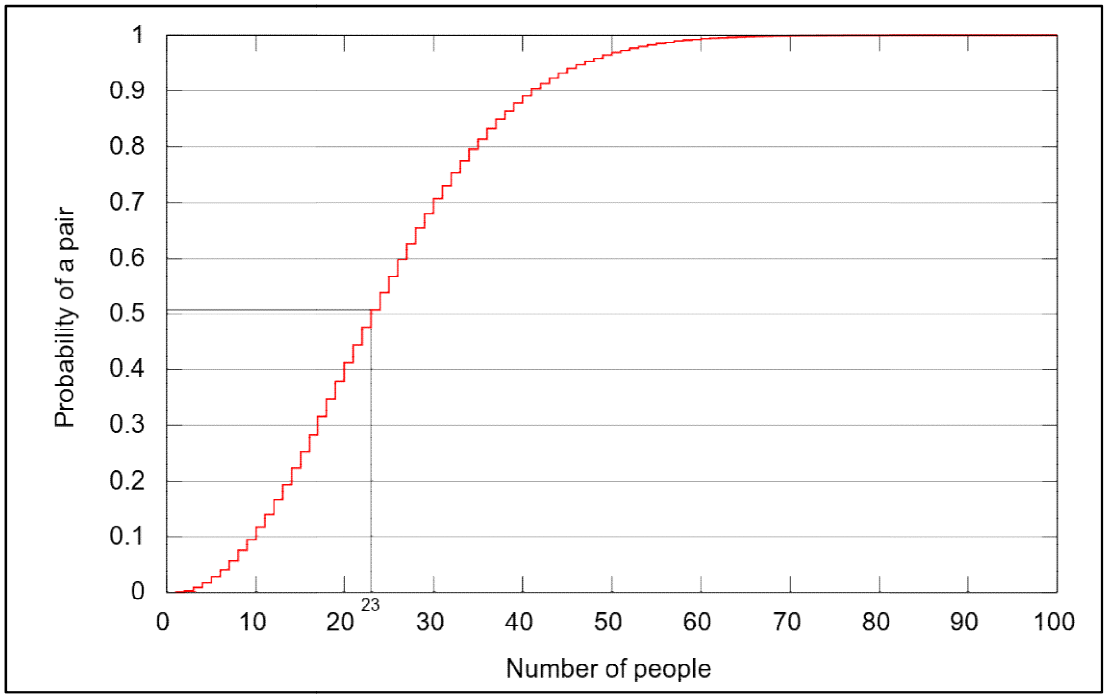


***Figure 2.2: The Birthday Paradox***

### 2.3.2 Implementation Attack

In comparison to others, the Implementation attack is a different approach to reveal the secret key. This method of attack searches the advantage of physical characteristics that occurs when a cryptographic algorithm is implemented in hardware. It never approaches the mathematical properties of the algorithm like as side channel attack. The approach of this attack deals with security requirements of cryptographic module like *Power Analysis*, *Timing Analysis*, *Fault Induction* and *TEMPEST*.

#### *2.3.2.1 Power Analysis*

The attack based on the analysis of power consumption can be divided into two types: Simple Power Analysis (SPA) and Differential Power Analysis (DPA). SPA involves direct analysis of electrical power consumption patterns whereas DPA has the same approach but utilizes advance statistical methods or other techniques to analyze the variations of electrical power consumption of a cryptographic module.

#### *2.3.2.2 Timing Analysis*

Measuring the time required by the cryptographic module to perform a mathematical operation of a cryptographic algorithm or process is the pathway of Timing analysis attack.

#### *2.3.2.3 Fault Induction*

Fault Induction attack utilizes external forces like microwaves, temperature, voltage to cause processing errors in the cryptographic module. Proper selection of physical security features may be used to reduce the risk of this attack.

#### *2.3.1.4 TEMPEST*

TEMPEST attack deals with detection and collection of electromagnetic signals emitted from a cryptographic module and associated equipment during processing.

### 2.3.3 Timing Attacks

A timing attack is analogous to a wild guessing the combination by observing how long it takes for someone to turn the dial from number to number. Timing attack is a serious threat and there are simple counter measures that can be used including the followings:

- Constant Exponentiation Time
- Random Delay
- Blinding

### 2.3.4 Boomerang Attack

Differential analysis has been used to break many published ciphers then that block cipher designers are thoughtful to ensure security against differential style attacks **[42]**. The algorithm designer obtains somehow an upper bound $p$ on the probability of any differential characteristic for the cipher. Then the designer invokes an oft-repeated "folk theorem" to justify that any successful differential attack will require at least $1/p$ texts to break the cipher.

Unfortunately, according to David Wagner **[42]** this folk theorem is wrong and exhibits an attack which is Boomerang Attack.

The Boomerang attack is a differential attack that attempts to generate a quartet structure at an intermediate value half-way through the cipher. The attack considers four plaintexts $P, P', Q, Q'$along with their ciphertexts $C, C', D, D'$. Let $E$ represents the encryption operation, and decompose the cipher into$E = E_1 \circ E_0$, where $E_0$ represents the first half of the cipher and $E_1$ represents the last half and the differential characteristics used call $\Delta \rightarrow \Delta^*$, for $E_0$ and $\nabla \rightarrow \nabla^*$ for $E_1^{-1}$. The graphical structure is shown in Figure 2.3 below.

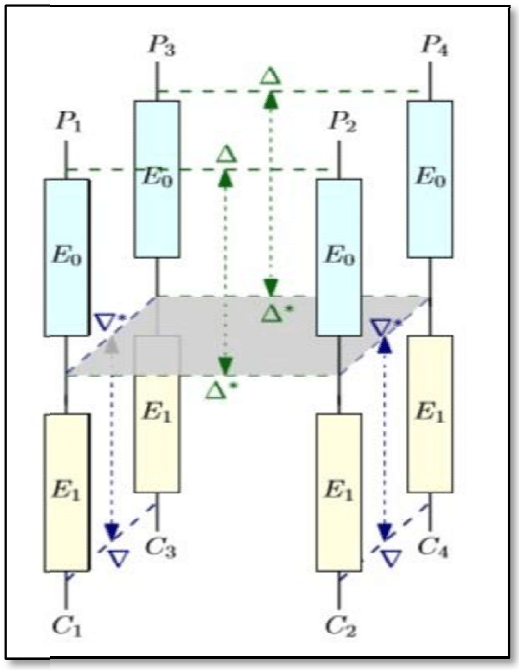


***Figure 2.3: Boomerang Attack***

## 2.4 Truncated Differential Analysis

The concept of truncated differential analysis was introduced by Lars R. Knudsen in **[49]**. Truncated differential is a type of differential where only a part of the difference in the ciphertexts can be predicted. The block ciphers which are secure against differential attack those are vulnerable against truncated differential or other higher order differentials. Traditional differentials used to predicts $n$ bits of $2n$ bits block ciphers. So a differential that predict only a part of $n$ bits value is called truncated differential.

# Chapter 3: Basics and Terminology

This chapter includes the terminology used throughout this research work. The input-output structure of S-boxes of DES and the sole S-box of AES are also discussed here for reference purpose.

## 3.1 Preliminaries

### 3.1.1 Terminology

**Truly Random Sequence**
An $n$ bit sequence is a truly random sequence if each bit is independent from every other bit in the sequence.

Informally, it can be stated that the probability of regeneration of a truly random sequence is very low, though we cannot guarantee the non-regeneration of such a sequence **[1]**.

**Relatively Random Sequence**
Two *n* bit sequences are relatively random if the number of bit-by-bit successful matches between the two sequences is *n/2* **[1]**.

### 3.1.2 Notations Used

| | |
|---|---|
| $\mathbb{Z}$ | The set of integers |
| $\mathbb{Z}_2^n$ | The *n* dimensional vector space over the finite field $\mathbb{Z}_2 - GF(2)$ |
| $\oplus$ | The addition over $\mathbb{Z}_2^n$, or, the bitwise exclusive-OR (XOR) |
| H | Hamming distance |
| wt (..) | Hamming weight function |
| $C_2^n$ | *N* dimensional vector with Hamming weight 1 at the $i^{th}$ position |
| $N_F$ | Nonlinearity of an S-box |
| $L_F$ | Linearity of an S-box |

Table 3.1 Used Notation

### 3.1.3 Substitution Box (S-box)

The S-box of a block cipher, as shown in Figure 3.1, can be represented as an $m \times n$ mapping $S: \{0,1\}^m \rightarrow \{0,1\}^n$, and is designed using the principle laid down by Shannon (1949) **[23]**.

Thus there are $n$ component functions, each being a map from $m$ bits to 1. S-box may express security against particular a class of attack.

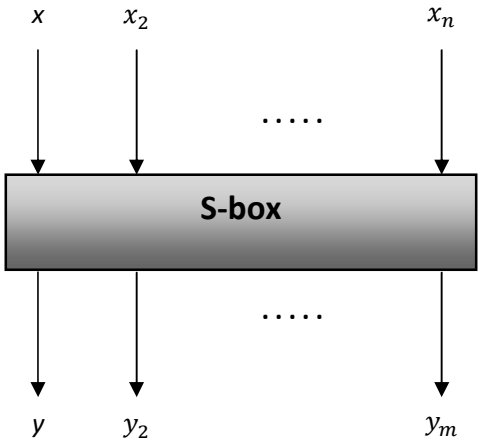


***Figure 3.1: Substitution Box (S-Box)***

Introduction of correlation attack **[24]** results the study of Boolean function and S-boxes in cryptography. Nonlinearity $N_F$ and linearity $L_F$ of an S-box is defined as

$$N_F = \min_{b \in F_2^m / \{0\}} N_{b.F} \text{ and } L_F = \max_{b \in F_2^m / \{0\}} L_{b.F}$$

The study of cryptographic properties of an S-box related to the linearity needs to consider all non-zero linear combination of S-box components **[25]**. So, an S-box can be represented by a vector$(f_0, f_1, \dots, f_{m-1})$, where $f_i$ is one of the Boolean functions from $\{0,1\}^m$ to $\{0,1\}^n$.

#### *3.1.3.1 Properties of an Ideal S-Box*

In the "Practical S-Box Design" **[46]** the following properties has been identified that an ideal S-Box would possess:

- All linear combinations of S-Box columns are bent.
- All entries in the S-Box XOR table are 0 or 1.
- The S-Box satisfies MOSAC **[46]** [Minimum Order Strict Avalanche Criterion].
- The S-Box satisfies MOBIC **[46]** [Minimum Order Bit Independence Criterion].
- The set of weights of rows has a binomial distribution with mean $^n/_2$.
- The set of weights of all pairs of rows has a binomial distribution with mean $^n/_2$.

Substitution process ensures the security of data because it is a non-linear transformation which performs confusion of bits. Non-linear transformation is very essential in modern encryption algorithms and it is proved to be a strong cryptographic primitive against linear and differential cryptanalysis **[47]**.

There are several properties available for S-box as listed below **[48]**:

- Robustness: If $F = (f_1, f_2 \ldots . f_n)$ be an $n \times n$ S-box then $F$ must be robust against differential cryptanalysis.
- Balancing: $S: \{0,1\}^n \longrightarrow \{0,1\}^m$ is balanced, if $HW(f) = 2^{n-1}$.
- Strict Avalanche Criterion (SAC).
- Non-linearity.
- Differential Uniformity: The smaller the differential uniformity, the better is the S-box resistance against differential cryptanalysis.
- Linear Approximation: The smaller the linear approximation, the better is the S-box resistance against linear cryptanalysis.
- Algebraic Complexity: The S-Box should be able to resist interpolation and algebraic attacks.
- Bit Independence Criterion

### 3.1.4 Structure of S-box of DES

The number of S-boxes of DES is 8 and every S-box contains 4 rows and 16 columns matrix form and each row consists of values ranging from $(0)_{10}$ to $(15)_{10}$. The structure of an S-box is illustrated in Figure 3.1. A 6 bit input to the S-box of DES generates 4 bit output as below:

$S = matrix\ 4 \times 16, values\ from\ 0\ to\ 15$
$B\ (6\ bit\ input) = \boldsymbol{b_1} b_2 b_3 b_4 b_5 \boldsymbol{b_6}$
$\boldsymbol{b_1 b_6} \rightarrow r = row\ of\ the\ matrix\ (2\ bits: 0,1,2,3)$
$b_2 b_3 b_4 b_5 \rightarrow c = column\ of\ matrix\ (4\ bits: 0,1, \ldots . ,15)$
$C\ (4\ bit\ output) = Binary\ representation\ S(r, c)$

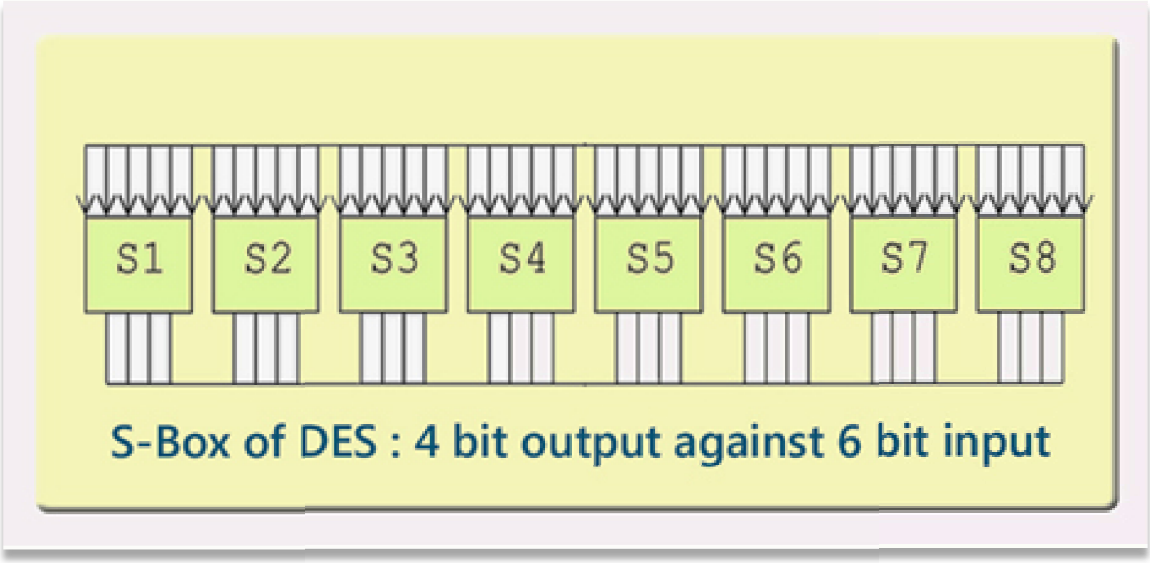


***Figure 3.2: Structure of an S-box of DES***

**3.1.5 Structure of S-box of AES**

There is a single non-linear S-box in AES with matrix structure of $16 \times 16$ where every individual element is a HEX value. Every 8-bit block of input generates 8-bit block of output. The structure of the S-box of AES is shown Figure 3.2. The evaluation method of output is as follows:

$S = matrix\ 16 \times 16, in\ HEX, values\ from\ 0\ to\ F$
$B\ (8\ bit\ input) = b_1 b_2 b_3 b_4 \boldsymbol{b_5 b_6 b_7 b_8}$
$b_1 b_2 b_3 b_4 \rightarrow r = row\ of\ the\ matrix\ for\ output$
$\boldsymbol{b_5 b_6 b_7 b_8} \rightarrow c = column\ of\ the\ matrix\ for\ output$
$C\ (8\ bit\ output) = Binary\ representation\ of\ hex\ value\ S(r, c)$

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 7C | 77 | 7B | F2 | 6B | 6F | C5 | 30 | 01 | 67 | 2B | FE | D7 | AB | 76 |
| 1 | CA | 82 | C9 | 7D | FA | 59 | 47 | F0 | AD | D4 | A2 | AF | 9C | A4 | 72 | C0 |
| 2 | B7 | FD | 93 | 26 | 36 | 3F | F7 | CC | 34 | A5 | E5 | F1 | 71 | D8 | 31 | 15 |
| 3 | 04 | C7 | 23 | C3 | 18 | 96 | 05 | 9A | 07 | 12 | 80 | E2 | EB | 27 | B2 | 75 |
| 4 | 09 | 83 | 2C | 1A | 1B | 6E | 5A | A0 | 52 | 3B | D6 | B3 | 29 | E3 | 2F | 84 |
| 5 | 53 | D1 | 00 | ED | 20 | FC | B1 | 5B | 6A | CB | BE | 39 | 4A | 4C | 58 | CF |
| 6 | D0 | EF | AA | FB | 43 | 4D | 33 | 85 | 45 | F9 | 02 | 7F | 50 | 3C | 9F | A8 |
| 7 | 51 | A3 | 40 | 8F | 92 | 9D | 38 | F5 | BC | B6 | DA | 21 | 10 | FF | F3 | D2 |
| 8 | CD | 0C | 13 | EC | 5F | 97 | 44 | 17 | C4 | A7 | 7E | 3D | 64 | 5D | 19 | 73 |
| 9 | 60 | 81 | 4F | DC | 22 | 2A | 90 | 88 | 46 | EE | B8 | 14 | DE | 5E | 0B | DB |
| A | E0 | 32 | 3A | 0A | 49 | 06 | 24 | 5C | C2 | D3 | AC | 62 | 91 | 95 | E4 | 79 |
| B | E7 | C8 | 37 | 6D | 8D | D5 | 4E | A9 | 6C | 56 | F4 | EA | 65 | 7A | AE | 08 |
| C | BA | 78 | 25 | 2E | 1C | A6 | B4 | C6 | E8 | DD | 74 | 1F | 4B | BD | 8B | 8A |
| D | 70 | 3E | B5 | 66 | 48 | 03 | F6 | 0E | 61 | 35 | 57 | B9 | 86 | C1 | 1D | 9E |
| E | E1 | F8 | 98 | 11 | 69 | D9 | 8E | 94 | 9B | 1E | 87 | E9 | CE | 55 | 28 | DF |
| F | 8C | A1 | 89 | 0D | BF | E6 | 42 | 68 | 41 | 99 | 2D | 0F | B0 | 54 | BB | 16 |

*Figure 3.3: Structure of S-box of AES*

# Chapter 4: Diffusion Analysis in Block Cipher using SAC †

This chapter describes a scheme aimed at measuring the diffusion characteristic of a block cipher. The cryptographic strength of a cipher is directly proportional to the extent to which diffusion is achieved. The scheme described in this chapter is used to measure the above mentioned characteristic of a cipher and the test results are analyzed to come to a conclusion. Potentially, the scheme can be added as a part of an already existing test suite to act as a distinguisher based on the diffusion characteristic of the underlying cipher.

## 4.1 Introduction

Many test suites have been designed to test the extent of randomness approximated by a block cipher **[10].** Most of these tests measure the degree of randomness of change at block level by changing a bit in the original block **[18]**. However, while operating at block level, a situation may arise where the $i^{th}$ bit of the block changes with a very high frequency whereas some other $j^{th}$ bit hardly changes. This gives a false impression that all the bits are changing with a probability of 0.5.

In this chapter, a scheme is being proposed which is not significantly different in nature with some of the existing test sets but is rather different in terms of how it is implemented to measure the diffusion characteristic of the concerned cipher.

The scheme named "Bit-level Block Cipher Diffusion Analysis Test (BLDAT)" **[56]** is aimed at how vulnerable the underlying block cipher is with regards to a particular bit.

Like most tests in this field, the proposed scheme also treats the underlying block cipher as a black box and the results of analysis are based solely on the input to and output from the cipher under test.

## 4.2 Bit-Level Diffusion Analysis Test (BLDAT)

The scheme uses a randomly selected $n$ bit block of plaintext (say $P$), which is then encrypted using the underlying cipher to produce the corresponding cipher block (say $C$) **[46]**. Then, a matrix of size $n \times n$ is produced, where each row of the matrix is $P_i$, a new plaintext block in itself derived from the original block by flipping the bit at the $i^{th}$ position i.e. $P_i[i] = P \oplus e_i$, where $e_i$ is a zero vector containing 1 at $i^{th}$ position.

† This chapter is referenced from the published research paper: " A Bit-Level Block Cipher Diffusion Analysis Test, Springer International Publishing Switzerland 2015, Vol. 1, Advances in Intelligent Systems and Computing 327, DOI: 10.1007/978-3-319-11933-5_75".

$$P_i = \begin{pmatrix} p[0,0] & \cdots & p[0,n-1] \\ \vdots & \ddots & \vdots \\ p[n-1,0] & \cdots & p[n-1,n-1] \end{pmatrix}$$

Next, each row of the $P_i$ matrix is fed as input to the underlying cipher to produce the corresponding ciphertext, which is stored as the $i^{th}$ row of the $C_i$matrix of size $n \times n$.

$$C_i = \begin{pmatrix} c[0,0] & \cdots & c[0,n-1] \\ \vdots & \ddots & \vdots \\ c[n-1,0] & \cdots & c[n-1,n-1] \end{pmatrix}$$

i.e. $C_i[i] = E(P_i[i])$ where E() denotes encryption using the underlying block cipher.

At this point, the scheme kicks in to produce another matrix (say $X$) of size $n \times n$, where $i^{th}$ row obtained by bitwise modulo 2 addition of $C_i$ vector with $C$ vector **[19]**, $X[i] = C_i[i] \oplus C$.

$$X = \begin{pmatrix} x[0,0] & \cdots & x[0,n-1] \\ \vdots & \ddots & \vdots \\ x[n-1,0] & \cdots & x[n-1,n-1] \end{pmatrix}$$

The scheme further produces the diffusion-factor by scanning each column of the $X$ matrix. The algorithm of the proposed scheme is given in section 4.2.1.

#### 4.2.1 Algorithm

**Algorithm – BLDAT:**

**Step-1**: Randomly select a binary string of n bits ($P$).

**Step-2**: Encrypt the plaintext with the concerned encryption algorithm to generate the corresponding ciphertext ($C$).

**Step-3**: Encrypt $P_i$'s with the particular encryption algorithm to generate $C_i$'s, where $P_i = P \oplus e_i$and $e_i$ is a string of zeros with the $i^{th}$ bit 1 and $E(P_i) = C_i$.

**Step-4**: $X_i = C_i \oplus C$ is stored in the $i^{th}$ row of the matrix of size $n \times n$ where $n$, the number of bits is in the original plaintext.

**Step-5**: Find the number of $1s$ in each column (j).

As it is evident from the algorithm, there are n bits in the block and at every instance we had changed only one bit, as a result there are n blocks such that $H(P, P_i[i] = 1)$, where H denotes the Hamming Distance. The test finds, the number of times, a particular bit has changed when each of the n newly generated blocks are encrypted using the underlying cipher with respect to the ciphertext of the original block. Ideally each bit should change n/2 times, if a particular bit has changed with very high or very low frequency, it might motivate an attack.

The number of times a particular bit has changed is referred to as the vulnerability factor of the bit. An extremely low or extremely high vulnerability factor associated with a particular bit may act as a motivation for attackers to exploit this idea.

The time taken to construct the $X$ matrix is $O(n^2)$ and the time taken to determine the bit-vulnerability factors is also $O(n^2)$. So, the time complexity of the algorithm is $O(n^2)$.

## 4.3 Experiment

A single block cipher has been used for applying the scheme. The ciphertext generated by the block cipher is subjected to the proposed test analysis.

### 4.3.1 Describing the Test Cipher

The analysis of the scheme is done using a simple Test Cipher. The Test Cipher is a simple substitution – permutation network which takes 8-bit block as input and produces 8-bit cipher block. At first, as found in **[4]**, the 8 bit block is bit wise XOR-ed with an 8-bit key and then passed through two 4-bit S-boxes (for simplicity the two S-boxes are considered to be identical). The outputs of the S-boxes are permuted to generate the ciphertext. The block diagram of the test cipher is depicted by Figure 4.1.

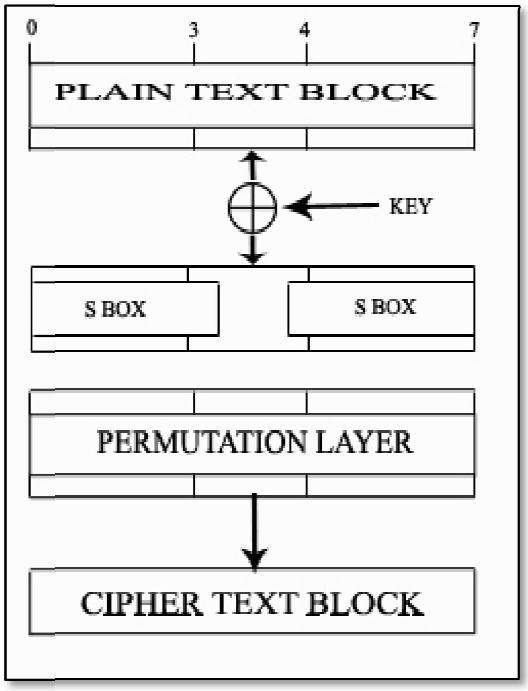


***Figure 4.1: Block diagram of test cipher***

### 4.3.2 Objective

The objective is to determine the secret key as is the case with other cryptanalysis techniques.

### 4.3.3 Assumptions

It is assumed that the plaintext block is known, the corresponding ciphertext block is also known and the results of the bit-level diffusion test are available. Using exhaustive key search method, the correct key can be determined with 28 trials (in the worst case). The sub-goal would be to reduce the number of trials using the results of BLDAT. If it is observed from the results of the BLDAT that a particular bit (say $j^{th}$ bit) of the cipher block seldom changes i.e. if the $i^{th}$ bit of the plaintext is 1/0, and remains 1/0 in the ciphertext at $j^{th}$ position with a very high degree of probability, then it will be easy to identify the particular key bit. Linear cryptanalysis and differential cryptanalysis are well known for mapping an input plaintext bit to an output ciphertext bit. If the observed plaintext and ciphertext are dissimilar, then it is clearly due to the key bit which got XOR-ed with the $i^{th}$ plaintext bit and will be a 1with a very degree of probability. And if both the observed bits are the same, then it implies that the key bit has not affected the plaintext bit which in turn implies that the bit is a 0 with a very high degree of probability.

#### 4.3.4 Experimental Results of BLDAT

Two well-known ciphers, namely Data Encryption Standard and Rijndael (Advanced Encryption Standard) are put to test, and the results obtained are analyzed in **[1]**. Say $\delta$ is deviation where $\delta = 0$ is the ideal case. Table 4.1 and Table 4.2 lists the results of BLDAT on DES and Rijndael Cipher (AES) respectively.

| Key | No. of Deviations | | | |
|---|---|---|---|---|
| | $\boldsymbol{\delta = 0}$ | $\boldsymbol{\delta = 4}$ | $\boldsymbol{\delta = 8}$ | $\boldsymbol{\delta > 8}$ |
| Sparse Key | 5 | 40 | 16 | 3 |
| Moderately Dense Key | 5 | 47 | 9 | 3 |
| Dense Key | 5 | 43 | 13 | 3 |
| Random Key | 7 | 34 | 21 | 2 |

**Table 4.1. Experimental result on DES**

| Key | No. of Deviations | | | |
|---|---|---|---|---|
| | $\boldsymbol{\delta = 0}$ | $\boldsymbol{\delta = 8}$ | $\boldsymbol{\delta = 16}$ | $\boldsymbol{\delta > 16}$ |
| Sparse Key | 8 | 89 | 13 | 1 |
| Moderately Dense Key | 11 | 99 | 18 | 0 |
| Dense Key | 5 | 102 | 20 | 1 |
| Random Key | 6 | 97 | 25 | 0 |

**Table 4.2. Experimental result on AES**

## 4.4 Analysis of BLDAT

The results obtained from both DES and AES are analyzed using the established statistical tools to finally draw the conclusion. The Chi-square ($\chi^2$ ) test has been used to determine the goodness of fit between theoretical and experimental data. The observed values and expected values are to be tested here.

### 4.4.1 Chi- Square ($\chi^2$) test on experimental result of DES

In the experiment of BLDAT for DES, with 64 bit plaintext block and 56 bit key, the observed bit changes in ciphertext block for every bit change in plaintext given in Table 4.3.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| v[0]=29 | v[8]=27 | v[16]=30 | v[24]=38 | v[32]=32 | v[40]=37 | v[48]=37 | v[56]=24 |
| v[1]=38 | v[9]=37 | v[17]=36 | v[25]=33 | v[33]=32 | v[41]=28 | v[49]=30 | v[57]=29 |
| v[2]=31 | v[10]=31 | v[18]=30 | v[26]=38 | v[34]=30 | v[42]=25 | v[50]=33 | v[58]=35 |
| v[3]=35 | v[11]=32 | v[19]=28 | v[27]=34 | v[35]=33 | v[43]=32 | v[51]=24 | v[59]=26 |
| v[4]=26 | v[12]=33 | v[20]=24 | v[28]=28 | v[36]=31 | v[44]=31 | v[52]=24 | v[60]=23 |
| v[5]=28 | v[13]=29 | v[21]=34 | v[29]=31 | v[37]=34 | v[45]=33 | v[53]=40 | v[61]=27 |
| v[6]=28 | v[14]=38 | v[22]=39 | v[30]=30 | v[38]=36 | v[46]=31 | v[54]=35 | v[62]=30 |
| v[7]=26 | v[15]=32 | v[23]=32 | v[31]=31 | v[39]=32 | v[47]=42 | v[55]=31 | v[63]=33 |

**Table 4.3. Observed bit changes in ciphertext using DES**

The change of ciphertext block is ideally *n/2* where n is the block size and is estimated 32. Table 4.4 is constructed to calculate the Chi-square distribution and goodness of fit for Chi-square.

| (T) | (O) | (E) | O-E | $(O-E)^2$ | $Y=(O-E)^2/E$ |
|---|---|---|---|---|---|
| 3 | 29 | 32 | -3 | 9 | 0.28125 |
| 4 | 38 | 32 | 6 | 36 | 1.125 |
| 8 | 31 | 32 | -1 | 1 | 0.03125 |
| 3 | 35 | 32 | 3 | 9 | 0.28125 |
| 3 | 26 | 32 | -6 | 36 | 1.125 |
| 5 | 28 | 32 | -4 | 16 | 0.5 |
| 2 | 27 | 32 | -5 | 25 | 0.78125 |
| 3 | 37 | 32 | 5 | 25 | 0.78125 |
| 6 | 33 | 32 | 1 | 1 | 0.03125 |
| 7 | 32 | 32 | 0 | 0 | 0 |
| 6 | 30 | 32 | -2 | 4 | 0.125 |
| 2 | 36 | 32 | 4 | 16 | 0.5 |
| 4 | 24 | 32 | -8 | 64 | 2 |
| 3 | 34 | 32 | 2 | 4 | 0.125 |
| 1 | 39 | 32 | 7 | 49 | 1.53125 |
| 1 | 25 | 32 | -7 | 49 | 1.53125 |
| 1 | 42 | 32 | 10 | 100 | 3.125 |
| 1 | 40 | 32 | 8 | 64 | 2 |
| 1 | 23 | 32 | -9 | 81 | 2.53125 |

**Table 4.4. DES observed values with corresponds to estimated value with their occurrence**

Where T is occurrences of observed value, O is observed value, E is estimated value. Using these values the chi-square is calculated with the consideration of occurrence of values and the calculated chi-square value of the experimental data is:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

$$\chi^2 = 37.25$$

and Figure. 4.2 is the graphical representation of observed value, estimated value and calculated value:

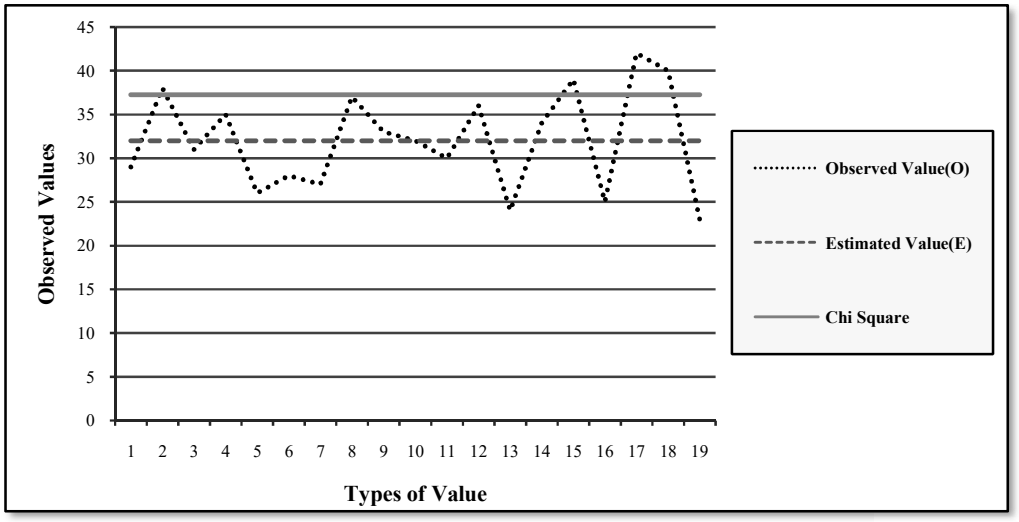


***Figure 4.2: Graph for Chi-square of DES***

In the experimental result (Table 4.4) it is visible that there exist 19 different sets of value $(v)$ i.e. number of time changes of every bit in ciphertext while changing every bit of plaintext for at most once. Therefore the Degree of Freedom (*df*) is:

$$df = v - 1 = 19 - 1 = 18$$

From the Table 4.4 it is now easy to calculate chi-square value for $df = 18\ and\ \alpha = .005$ is:

$$GF_{18,.005} = \chi^2_{18,.005} = 34.71875$$

Moreover, chi-square value for $df = 18\ and\ \alpha = 5\%$ is:

$$GF_{18,5\%} = \chi^2_{18,5\%} = 31.5$$

From the chi-square distribution table **[20]**, is found that $\chi^2_{18,.005}$ is 37.156 and $GF_{18,.005} < 37.156$.

So, according to **[21]** it may be concluded that either (i) this model is valid but that a statistically improbable excursion of $\chi^2$ has occurred, (ii) too conservatively, over-estimated the values of α or (iii) data is 'too good to be true'.

**4.4.2 Chi- Square ($\chi^2$) test on experimental result of AES**

In the experiment of BLDAT for AES, for 128 bit plaintext block with 56 bit key and observed bit changes in ciphertext block for every bit change in plaintext. The Chi-square ($\chi^2$) test has been used. The change of ciphertext block is ideally $n/2$ where n is the block size and is estimated 64. The calculated chi-square value of the experimental data of AES is:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

$$\chi^2 = 73.609375$$

Fig. 4.3 is the graphical representation of observed value, estimated value and calculated value.

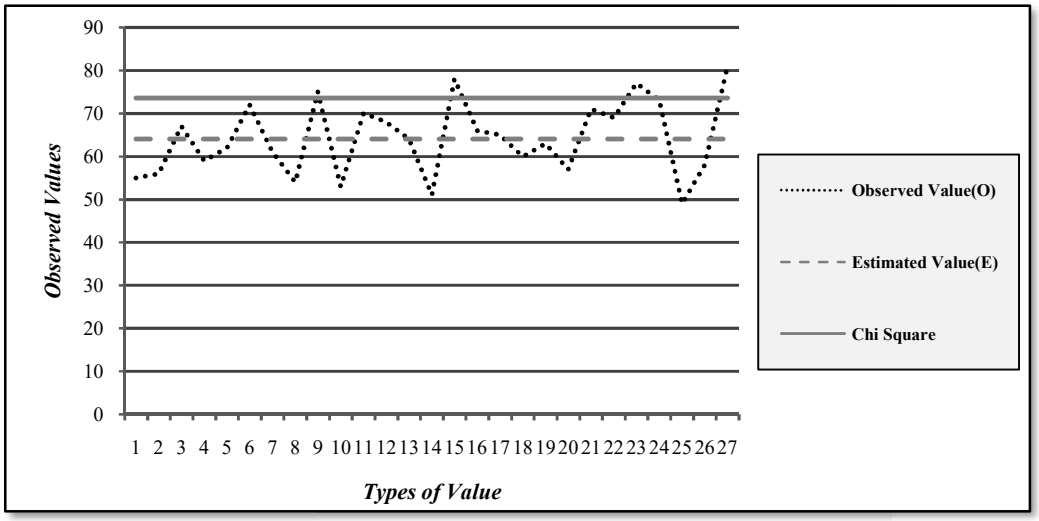


***Figure 4.3: Graph for Chi-square of AES***

In the experimental it is visible that there exist 27 different sets of value ($v$) i.e. number of time changes of every bit in ciphertext while changing every bit of plaintext for at most once. Therefore the Degree of Freedom (*df*) is:

$$df = v - 1 = 27 - 1 = 26$$

Now it is easy to calculate chi-square value for $df = 26 \; and \; \alpha = .005$ is:

$$GF_{26,.005} = \chi^2_{26,.005} = 69.09375$$

Moreover, chi-square value for $df = 26 \; and \; \alpha = 5\%$ is:

$$GF_{26,5\%} = \chi^2_{26,5\%} = 63.8046875$$

From the chi-square distribution table **[20]**, is found that $\chi^2_{26,.005}$ is 48.290 and $GF_{26,.005} > 48.290$.

Therefore, it may be concluded that either (i) this model is valid one but that a statistically improbable excursion of $\chi^2$ has occurred or (ii) that this model is poorly chosen that an unacceptable large value of $\chi^2$ has resulted **[21]**. The theory of chi-square test relies on the assumption that chi-square is the sum of the squares of random normal derivatives, that is, that each $x_i$ is normally distributed over its mean value $\mu_i$.

## 4.5 Conclusion on BLDAT

Even if the Hamming Distance of the plaintext block and the ciphertext block is ideal i.e., $n/2$, where $n$ is the block size, the key space can be reduced using a scheme such as the proposed BLDAT. The scheme may be clubbed with other tests for comprehensive analysis of block ciphers.

# Chapter 5: Analysis of Confusion in S-Boxes through SAC Test: 1 Bit Alteration †

The security of a block cipher, in general, depends on the characteristics of Substitution box (S-box) because it is the only non-linear component of a block cipher. The design and characteristics of S-boxes of a block cipher are central measures of resistance against all cryptanalytical techniques. So, analysis of an S-box before it could be implemented is very much important. This chapter involves one such novel bit-level confusion analysis test for S-boxes. The method has been used to test the strengths of S-boxes of DES and AES using Strict Avalanche Criterion (SAC) matrix.

## 5.1 Introduction

The strength of an encryption algorithm lies in the confusion and diffusion properties in it. The only non-linear component included in many block ciphers, called Substitution box (S-box), provides this functionality to the algorithm. The qualities of the algebraic and statistical properties of an S-box define the strength of an algorithm and thus are a vital source of analysis.

This chapter discusses a novel approach for statistical analysis of the properties of an S-box. The approach includes the analysis of Strict Avalanche Criterion (SAC) matrix using statistical measures like coefficient of variance, correlation, standard deviation and mean of standard deviation.

## 5.2 Strict Avalanche Criterion (SAC)

The Strict Avalanche Criterion (SAC) was introduced by A.F. Webster and S. E. Tavares [7] and according to them "If a function is to satisfy the strict avalanche criterion, then each of its output bits should change with a probability of one half whenever a single input bit $x$ is complemented to $\bar{x}$ **[7]**. SAC can be mathematically represented as:

$$\forall x, y | H(x, y) = 1,$$
$$avg(H(F(x), F(y))) = {}^{n}/_{2}$$

i.e., a very small change in input produces reasonable change in output. Hence, if $F$ has the avalanche effect then the Hamming Distance between the input and the output generated by changing one of its bits should be close to $n/2$.

† This chapter is referenced from the published research paper: "A Novel Technique for Analysing Confusion in S-Boxes, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2016, ISSN(Online): 2320-9801, ISSN (Print) : 2320-9798, DOI: 10.15680/IJIRCCE.2016. 0406253".

*Avalanche Effect*: Feistal **[27]** has proposed a function $\boldsymbol{f}: \mathbb{Z}_2^{\boldsymbol{n}} \rightarrow \mathbb{Z}_2^{\boldsymbol{m}}$ that defines the avalanche effect if and only if

$$\sum_{x \in \mathbb{Z}_2^n} wt\big(f(X) \oplus f(X \oplus C_i^n)\big) = m2^{n-1} \text{ for all } i\ (1 < i < n).$$

*Completeness*: Kam and Davida **[28]** proposed the completeness condition that each output bit depends on all input bits of substitution by a function $\boldsymbol{f}: \mathbb{Z}_2^{\boldsymbol{n}} \rightarrow \mathbb{Z}_2^{\boldsymbol{m}}$ is complete if and only if

$$\sum_{x \in \mathbb{Z}_2^n} \boldsymbol{f}(\boldsymbol{X}) \oplus \boldsymbol{f}(\boldsymbol{X} \oplus \boldsymbol{C_i^n}) > (0,0,\ldots,0) \text{ for all } i\ (1 < i < n).$$

## 5.3 Proposed Method

The steps involved in the proposed technique [47] include:

- Generate SAC matrix by altering each bit of the input to the S-box.
- Analysis of Coefficient of Variance of the generated SAC matrix.
- Analysis of frequency of various avalanche effects from the generated SAC matrix.
- Analysis of frequency of various differential values from the generated SAC matrix.
- Analysis of Hamming Weights according to bit positions from the generated SAC matrix.

## 5.4 Algorithm

**Algorithm – Bit Level Confusion Analysis of S-Box:**

Input: S-box with length *m*, where *m* is the number of bits.

Step 1: Choose a random number $P \in \mathbb{Z}_2^m$. Find the corresponding output value of S-box:

$$C = S(P)$$

Step 2: Change the $P_i's$ to find their corresponding output values $C_i's$. $P_i$ is the new $P$ by altering $i^{th}$ bit to its complement.

Step 3: $X_i = C_i \oplus C$ is stored in the $i^{th}$ row of a matrix of size $x \times y$ where $x$ is the number of bits of $P$ and $y$ is the number of bits of $C$.

Step 4: Find the number of 1s in each column $(j)$.

As it is evident from the algorithm that there are $n$ bits in the block and at every instance only one bit has been changed, as a result there are $n$ blocks such that $H(P, P_i[i] = 1)$, where H denotes the Hamming Distance. The test finds the number of times a particular bit has changed when each of the $n$ newly generated S-box outputs with respect to the S-box output of the original input. Ideally each bit should change

*n/2* times, because if a particular bit has changed with very high or very low frequency, it might become a vulnerable target for an attack.

The number of times a particular bit has changed is referred to as the vulnerability factor of the bit. An extremely low or extremely high vulnerability factor associated with a particular bit may act as an area of exploitation for the attackers.

The time taken to construct the $X$ matrix is $O(n^2)$ and time taken to determine the bit-vulnerability factors is also $O(n^2)$. So, the time complexity of the algorithm is $O(n^2)$.

## 5.5 Experimental Results

**Coefficient of Variance Analysis of generated SAC of S-boxes of DES:**

S-boxes are the only non-linear elements in DES design. There are 8 S-boxes with the structure of matrix 4 × 16. Each of the unique selection function $S_1, S_2, S_3, \ldots, S_8$, takes a 6-bit block as input and yields a 4-bit block as output. The structure of S-box input and output of DES is given in Figure 5.1 **[29]**

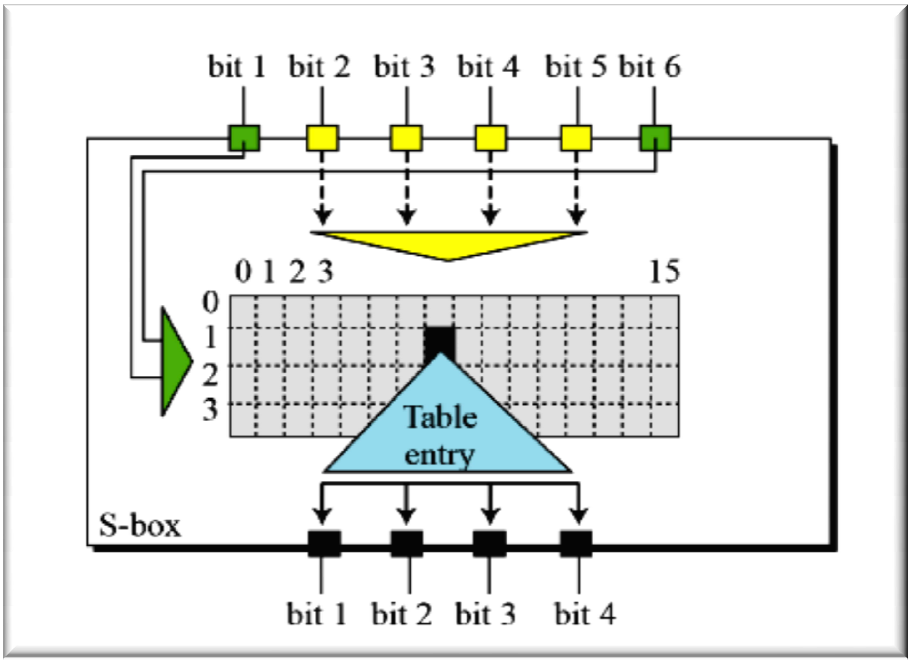


***Figure 5.1: Structure of S-box input and output of DES***

Evaluation of output is:

**Step1:** S= matrix 4 × 16, values from 0 to 15.

**Step2:** B (6-bit input) = $\boldsymbol{b_1}b_2b_3b_4b_5\boldsymbol{b_6}$

**a.** $\boldsymbol{b_1b_6}$ ⟶ $r$ = row of the matrix (2 bits: 0, 1, 2, 3).

**b.** $b_2b_3b_4b_5$ ⟶ $c$ = column of the matrix (4 bits: 0,1,2,….,15).

**Step3:** C (4-bit output) = Binary representation S($r,c$).

A SAC matrix has been generated from every possible input of every S-box and 1s of every column of every output of every S-box have been calculated. Coefficients of Variance of all S-boxes have been calculated on the number of 1$s$ available in each column. Some of the generated SAC matrices have been given as examples in Table 5.2 and Table 5.3. For all possible 64 inputs and outputs of all 8 S-boxes, the sum of 1$s$ of every column is termed as V-vector.

| SAC Matrix of Input – 0 of S-box 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| V-vector of Input – 0 | | | |
| 5 | 3 | 4 | 2 |

Table 5.2. SAC Matrix of input 0 of S-box 0 of DES

| SAC Matrix of Input – 0 S-box 1 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| V-vector of Input – 0 | | | |
| 4 | 5 | 4 | 3 |

Table 5.3. SAC Matrix of input 0 of S-box 1 of DES

### 5.5.1 Experimental Results for DES S-boxes

The results of the proposed test on Data Encryption Standard (DES) **[30]** S-boxes are depicted in Table 5.4.

| S-box# | Observed Mean | Variance | Standard Deviation | Coefficient of Variance |
|---|---|---|---|---|
| 0 | 3.71875 | 1.436523 | 1.198551 | 0.322299 |
| 1 | 3.796875 | 1.036865 | 1.018266 | 0.268185 |
| 2 | 3.9375 | 1.214844 | 1.1022 | 0.279924 |
| 3 | 3.6875 | 1.027344 | 1.01358 | 0.274869 |
| 4 | 3.796875 | 1.224365 | 1.10651 | 0.291427 |
| 5 | 3.90625 | 1.209961 | 1.099982 | 0.281595 |
| 6 | 3.9375 | 1.417969 | 1.190785 | 0.302422 |
| 7 | 3.75 | 0.953125 | 0.976281 | 0.260342 |

Table 5.4. Experimental Results for DES S-box

The column graph of co-variance (CV) for every S-box of DES is shown in Figure 5.3. The column graph of Standard Deviation for every S-box of DES is shown in Figure 5.4.

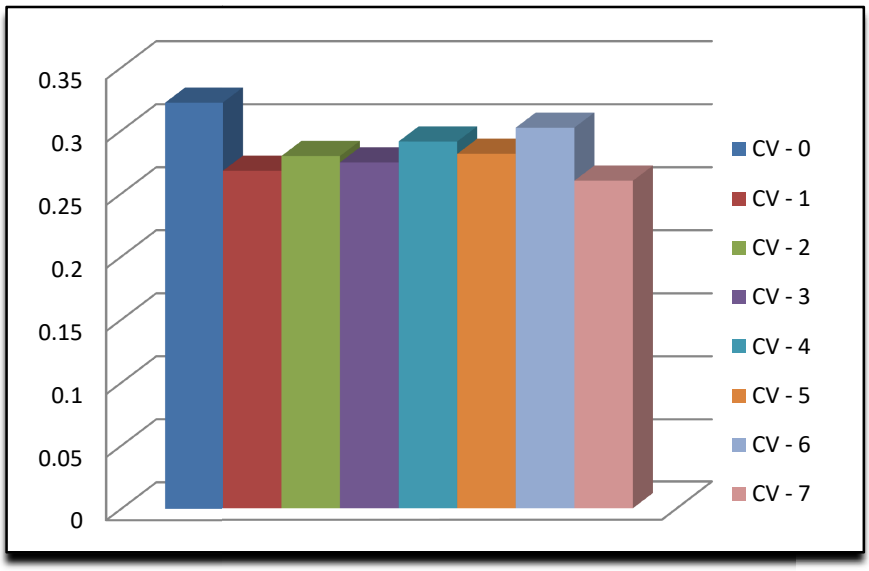


*Figure 5.2: Analysis of Co-variance for S-boxes of DES*

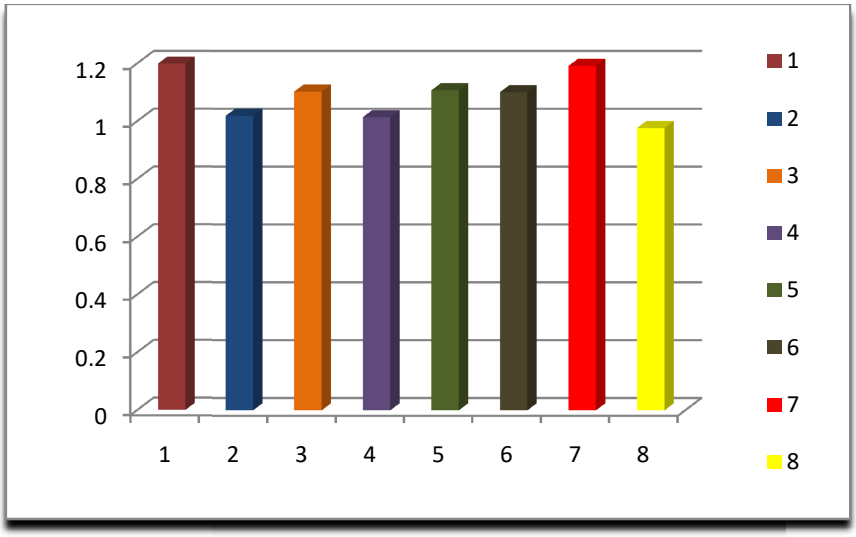


*Figure 5.3: Analysis of Standard Deviation for S-boxes of DES*

The Comparison line graph of Standard Deviation (SD) and Coefficient of Variance (CV) of every S-box of DES is shown in Figure 5.4.

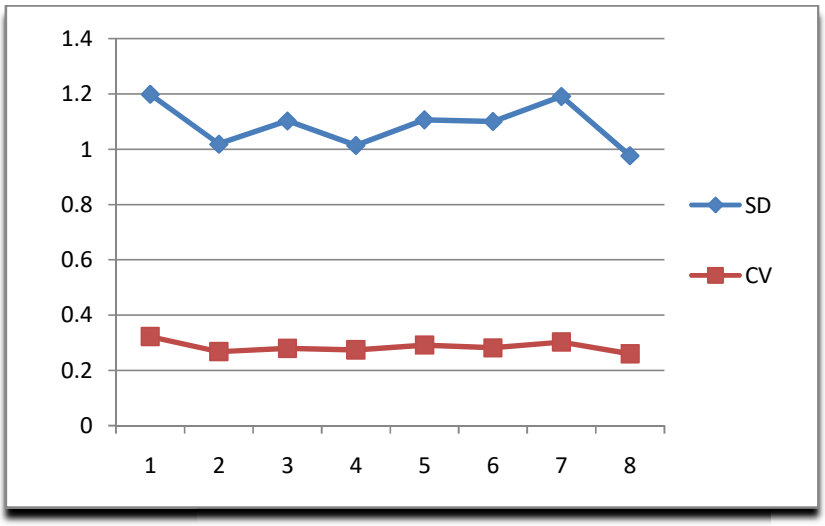


*Figure 5.4: Comparison of SD and CV for S-boxes of DES*

**Coefficient of Variance Analysis of generated SAC of S-box of AES:**

The single S-box is the only non-linear element in AES design. The S-box has the matrix structure of $16 \times 16$ and elements of the s-box are individually an hex value. For each of the unique inputs of 8-bit block it yields an 8-bit blocks as output and evaluation of output is:

**Step1:** S= matrix $16 \times 16$, in hex, values from 0 to F.

**Step2:** B (8-bit input) = $b_1 b_2 b_3 b_4 \mathbf{b_5 b_6 b_7 b_8}$

**a.** $b_1 b_2 b_3 b_4$ ⟶ $r$ = row of the matrix for output.

**b.** $\mathbf{b_5 b_6 b_7 b_8}$ ⟶ $c$ = column of the matrix for output

**Step3:** C (8-bit output) = Binary representation of hex value of S($r$, $c$).

A SAC matrix has been generated from every possible input of every S-box and 1s of every column of every output of every S-box have been calculated. Coefficients of Variance of all S-box have been calculated on the number of 1s available in each column. Some of the generated SAC matrices are given as examples in Table 5.5 and Table 5.6. For all 6 inputs and outputs of S-box, the sum of 1s of every column is termed as V-vector.

| Input : 11000011 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Original Output : 00101110 | | | | | | | |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| V-vector of input 11000011 - $(195)_{10}$ | | | | | | | |
| 6 | 4 | 4 | 4 | 4 | 3 | 4 | 5 |

Table 5.5. SAC Matrix of input 11000011 of AES S-box

| Input : 10101010 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Original Output : 10101100 | | | | | | | |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| V-vector of input 10101010 – $(170)_{10}$ | | | | | | | |
| 3 | 2 | 2 | 5 | 6 | 3 | 3 | 6 |

Table 5.6. SAC Matrix of input 10101010 of AES S-box

### 5.5.2 Experimental Results for AES S-box

The results of the proposed test on the Advanced Encryption Standard (AES) **[31]** S-box are depicted in Table 5.7.

| Input | Observed Mean | Variance | Standard Deviation | Coefficient of Variance |
|---|---|---|---|---|
| $(195)_{10}$ | 4.25 | 0.6875 | 0.829156 | 0.195096 |
| $(170)_{10}$ | 3.75 | 2.4375 | 1.561249 | 0.416333 |
| $(204)_{10}$ | 4.125 | 1.60937 | 1.268611 | 0.307542 |
| $(105)_{10}$ | 3.125 | 1.60937 | 1.268611 | 0.405956 |
| $(45)_{10}$ | 3.625 | 3.98437 | 1.99609 | 0.550645 |
| $(210)_{10}$ | 4.375 | 1.23437 | 1.111024 | 0.253948 |

Table 5.7. Experimental Results for AES S-box

The column graph of co-variance of every input for S-box of AES is shown in Figure 5.5. The column graph of Standard Deviation of every input for S-box of AES is shown in Figure 5.6.

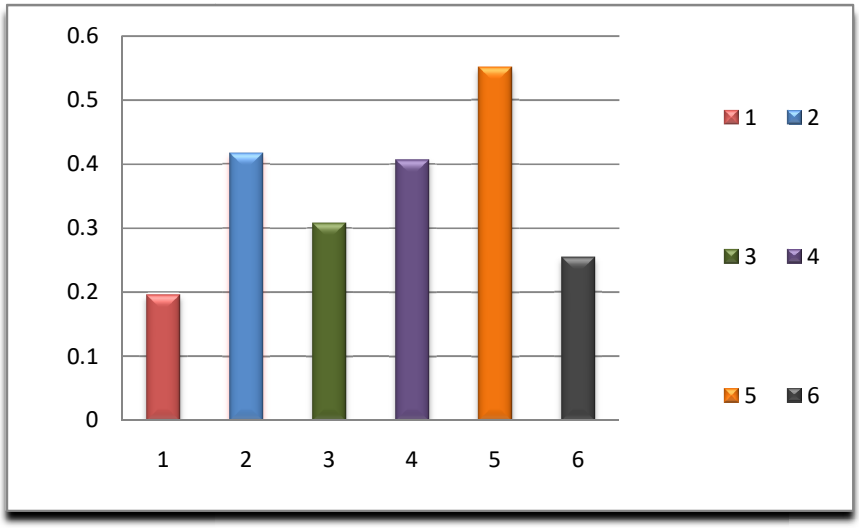

*Figure 5.5: Analysis of Co-variance of Inputs of S-box of AES*

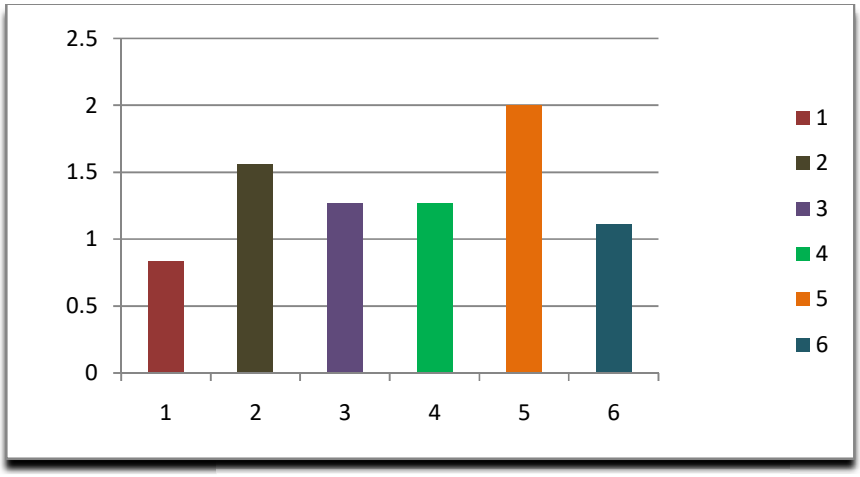


*Figure 5.6: Analysis of Standard Deviation of Inputs of S-box of AES*

The Comparison line graph of Standard Deviation (SD) and Coefficient of Variance (CV) of every input for S-box of AES is shown in Figure 5.7.

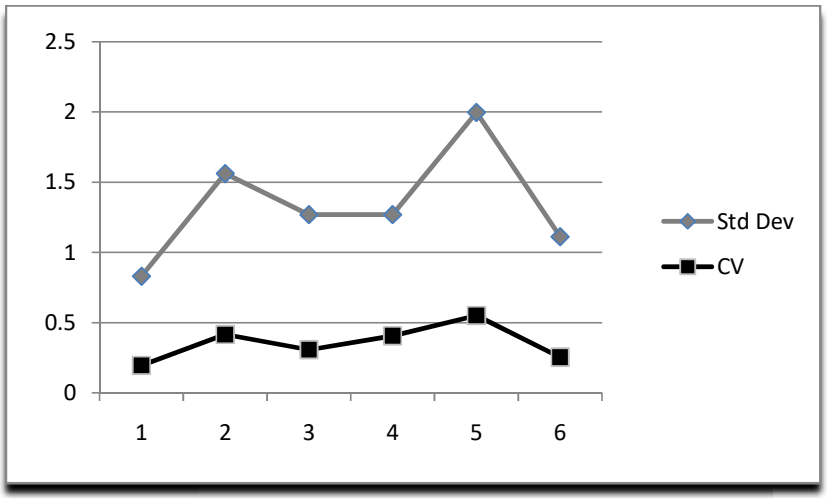


*Figure 5.7: Comparison of SD and CV for S-box of AES*

## 5.6 Discussion

After computing results of the proposed test algorithm on every possible input for every S-box of DES and six numbers of random inputs for single S-box of AES, coefficient of variance has been calculated as a statistical measure of the dispersion of data point in a data series around the mean. The coefficient of variance (CV) is calculated as $CV = Std.Deviation\ /\ Mean$.

By the analysis of coefficient of variance in case of S-boxes of DES, it is closer to 0.3 that is lower end of the spectrum and indicates that S-boxes of DES perform well with respect to the test.

On the other hand the result obtained from the single S-box of AES varies from 0.2 to 0.55 which is also lower end of the spectrum and indicates that S-box of AES perform well with respect to the proposed test algorithm.

## 5.7 Conclusion

As discussed in section 5.1, confusion and diffusion are the two fundamental aspects of a block cipher for analyzing its cryptographic strength. There are many methods to test diffusion but this proposed test could very serve well to analyze the confusion characteristic too, and may be included as a part of a comprehensive test suite for analyzing cryptographic strength of block ciphers.

# Chapter 6: S-Box Confusion Analysis using 2- Bit Alteration [†]

This chapter is concerned with methods of test that are capable of analyzing any practical block cipher, irrespective of the internal structure. The characteristics of the non-linear component of a block cipher, substitution box (S-box), in general, defines the security level of that block cipher. The resistance against all cryptanalytical techniques may be measured by the structure and characteristics of the S-box(es) of the underlying block cipher. This chapter involves one such novel approach for Bit-Level Confusion Analysis of S-boxes. The method has been used to test the strengths of S-boxes of Data Encryption Standard (DES) and Advanced Encryption Standard (AES) using Strict Avalanche Criterion (SAC) matrix.

## 6.1 Introduction

Block ciphers such as DES and AES use the idea of substitution and permutation to consolidate the security aspect. Nevertheless, there are various attacks on cryptographic systems which are very alarming. The understanding of the structure of attacks helps cryptanalysts to dig out the reasons behind the structure of cryptographic algorithms.

This chapter proposes a novel technique for the statistical analysis of an S-box by analyzing the Strict Avalanche Criterion (SAC) using a 2-bit approach over the single-bit approach. The approach includes analysis of SAC matrix by changing all possible pairs of 2-bit data of the output cipher for every input to S-boxes of DES and AES, and computing coefficient of variance, correlation, standard deviation and mean of standard deviation.

## 6.2 Related Work

As discussed earlier, the strength of a cipher can be defined by the confusion and diffusion properties of the algorithm and its performance depends on the strength of the algorithm. The methods used for the measuring the strength using algebraic and statistical analyses are further discussed in section 6.2.1.

### 6.2.1 Proposed Approach

[†] This chapter is referenced from the published research paper: "Implementation of SAC Test for Analyzing Confusion in an S-BOX Using a Novel Technique, International Journal of Scientific Research in Computer Science Application and Management Studies, Volume 7, Issue 3, ISSN 2319-1953, Pg: 182, May, 2018".

The strength and performance analyses of DES and AES using relationship of coefficient of variance, correlation, standard deviation and mean of standard deviation has been discussed in the paper titled *A Novel Technique for Analyzing Confusion in S-boxes* **[74]**. This paper comprises the generation of SAC matrix of S-boxes for both DES and AES considering every individual bit of original cipher that has been changed from $0\ to\ 1\ or\ 1\ to\ 0$. Then the occurrences of $1s$ for each and every output bit for every input element of all S-boxes have been analyzed to draw the conclusion. The vulnerability of the output cipher got measured by measuring the avalanche of bit change in the output for each input element of S-boxes.

The study continues to learn different types and aspects of attacks and reasons of attacks on block cipher. Crypt-analytic attacks and Implementation attacks are the two classes of attack where the first class of attack tries to attack mathematical weaknesses in the algorithm, while the second class of attack tries to attack the specific implementation of the cipher.

The paper *On the statistical testing of Block Cipher* **[2]** discusses two problems of block cipher needed to be solved by the cryptanalyst, and if one of these remains unsolved, the block cipher is said to unbreakable. The problems are:

- To find an algorithm that is distinguishing for given block cipher.
- To find an algorithm that is key-subset distinguishing for a given block cipher and for a given decomposition of the key space.

The security of block cipher against linear cryptanalysis and the comparison between the actual theoretical security and practical security approach have been discussed in Provable Security of *Block Ciphers against Linear Cryptanalysis: A Mission Impossible. An Experimental Review of Practical Security Approach and the Key Equivalence Hypothesis in Linear Cryptanalysis* by G. Piretand and F.-X. Standaert **[32]**.

In an experiment presented by Knudsen et al. **[3]** to evaluate the relevance of use of characteristics for arguing the security of a construction, the practical security approach is that if the best linear approximation of a given cipher is key-dependent, it can be hardly exploited by an actual adversary.

In *New Analysis Methods on Strict Avalanche Criterion of S-box* by Phyu Phyu Mar and Khin Maung Latt **[33]**, three analysis methods of SAC of S-box have been proposed:

- Analysis of the frequency of various hamming weights (Avalanche Effect).
- Analysis of the frequency of various differential values Y (Completeness).
- Analysis of hamming weight is according to the bit position (Strong S-box).

Igil VERGL, Melek D. YCEL **[34]** defined and concluded the relative avalanche error, $\epsilon_{AVAL}$, and relative SAC error, $\epsilon_{SAC}$ , which indicate how close a randomly chosen S-box satisfies the mentioned criteria of AVAL and SAC, respectively. They obtained

the relative errors, $\epsilon_{AVAL}$and $\epsilon_{SAC}$, found in an ensemble of randomly generated $n \times n$ S-box, corresponding to their maximum value of the parameters $\epsilon_{AVAL}$ and $\epsilon_{SAC}$ , respectively.

## 6.3 SP Network and S-box

Claude Shannon proposed the Substitution Permutation (SP) network as the heart of modern cryptography **[35]**. To confirm the confidentiality of bits of data in encryption or decryption, SP network, as shown in figure 6.1, performs three steps **[36]**.

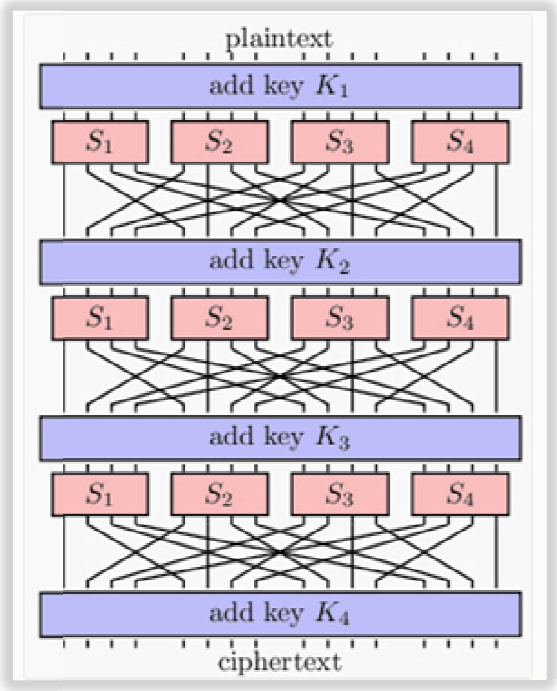


***Figure 6.1: Substitution Permutation Network***

- A function of the transformation key, called subkey, is X-ORed into the input data bit.
- A substitution function $\mathbb{Z}_{2^n} \rightarrow \mathbb{Z}_{2^m}$ $(i = 1,2, \ldots\ldots, 4\ in\ figure\ 6.1)$ replaces $n$ bits of data by another set of $n$ bits to introduce confusion into the data. The replacement is performed using lookup tables called Substitution Boxes or S-boxes.
- A permutation function *P* shuffles the bits to cause diffusion within the data.

These three steps together form a round of the SP network which is repeated several times.

S-boxes of a block cipher are Boolean mapping from $\{0,1\}^m \rightarrow \{0,1\}^n$ in the form $m \times n$, where there are $n$ component functions, each being a map from $m$ bits to $n$ bits **[37]**. Each successful linearization approximation can help to break a few bits of the key. Existing methodologies are suboptimal and they are not capable of finding the strongest S-box **[36]**.

In Feistel network, a part, highly non-linear and difficult to cryptanalyze, consists of the S-boxes in the function $f$ of figure 6.2. The security of S-boxes is highly important and security requirements like strict avalanche criterion (SAC) and others have found their way into the design principles to enhance S-box security.

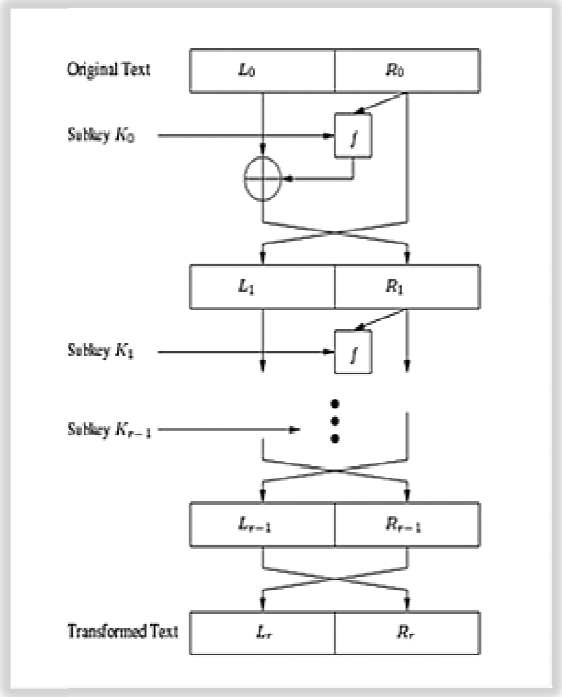


***Figure 6.2: Rounds of Feistel Cipher***

### 6.3.1 Confusion / Diffusion Primitives

Among the various primitives, for example in DES, confusion / diffusion primitives may be termed as bit-twiddling constructions. Typical use of straight-line code of simple Boolean operations creates computational hardness by doing enough strange stuff. Some cryptographic aims are not known to be solvable starting from confusion / diffusion primitive **[38]**.

### 6.3.2 Criteria and Definitions

#### *6.3.2.1 Avalanche Criterion*

The Avalanche (AVAL) Criterion **[39, 40]** is defined by Feistal et. al. which is the property of S-boxes and SPN's. If a small bit variation in the input plaintext/key of a block cipher results in a large difference in ciphertext bits, it is said to cause an *avalanche*. An $n \times n$ S-box is said to satisfy the AVAL criterion if for all $i = 1, 2, \ldots, n$:

$$\frac{1}{2^n}\sum_{j=1}^{n} W\left(a_j^{e_i}\right) = \frac{n}{2} \qquad (1)$$

where

$$W\left(a_j^{e_i}\right) = \sum_{all\ X \in \{0,1\}^n} a_j^{e_i} \tag{2}$$

is the total change in the $j^{th}$ avalanche variable, $a_j^{e_i}$, computed over the whole input of size $2^n$.

#### *6.3.2.1 Strict Avalanche Criterion (SAC)*

Webster and Tavares in "On the Design of S-boxes" **[7]** defined completeness and avalanche properties into the Strict Avalanche Criterion (SAC). An S-box satisfies the SAC if

$$\frac{1}{2^n} W\left(a_j^{e_i}\right) = \frac{1}{2} \tag{3}$$

*Avalanche Effect*: The function $f: \mathbb{Z}_2^n \rightarrow \mathbb{Z}_2^m$, proposed by Feistel **[39]**, defines the avalanche effect if and only if

$$\sum_{x \in \mathbb{Z}_2^n} wt\left(f(X) \oplus f(X \oplus C_i^n)\right) = m2^{n-1} \tag{4}$$

for all $i (1 < i < n)$.

## 6.4 Proposed Method

**[48]** A 2-Bit approach to analyze confusion in S-boxes using SAC has been proposed here. This proposed method will be compared with the conclusion of bit-level confusion analysis of S-box **[1]**. The steps involved include:

- Analysis of frequency of various avalanche effects from generated SAC using 2-bit approach.
- Analysis of Hamming weight according to bit position from generated SAC using 2-bit approach.
- Analysis of Coefficient of Variance of generated SAC using 2-bit approach.
- Analysis of frequency of various differential values from generated SAC using 2-bit approach.

### 6.4.1 Proposed Algorithm

**Algorithm – A 2-Bit Approach Confusion Analysis of S-Box**

**Input**: S-box with length $n$, where $n$ is the number of bits.

***Step1***: Choose a random number $P \in \mathbb{Z}_2^n$. Find the corresponding output value of S-box:

$$C = S(P)$$

***Step2***: Change the$P_i's$ to find their corresponding output values, $C_i's$. $P_i$ is the new $P$ by altering $i^{th}\ and\ j^{th}$ bits to their complements.

***Step3***: $X_i = C_i \otimes C$ is included in the $i^{th}$ row of a matrix of size $x \times y$, where $x$ is the number of bits of $P$ and $y$ is the number of bits of $C$.

***Step4***: Find the count of 1s in each column of generated SAC matrix.

For every $n$ bits in the block cipher and every time consecutive two bits have been changed and the Hamming Distance ($H$) is $H(P, P_{ij}[ij] = 1)$. The proposed test finds the count of change in bits for every $n$ newly generated S-box output by comparing with the original S-box output. Ideally, each bit should change for $^n/_2$ times and if a bit has changed with very high or low frequency then it may be attack prone. The count of bit change is the vulnerability factor of the bit and high or low vulnerability motivates attackers to attack the area. The time complexity for, both, generating the $X$ matrix and determination of the bit-vulnerability is $O(n^2)$ , so the overall time complexity of the algorithm is $O(n^2)$.

## 6.5 Experimental Results

### 6.5.1 Coefficient of Variance Analysis of Generated SAC of S-box of DES

A SAC matrix has been generated for every possible input to every S-box of DES. Using the proposed algorithm, $1s$ of every column of output of S-boxes have been counted. The term V-vector (**V**ulnerability factor) has been used for the sum of 1s in every column for all 64 possible inputs and corresponding outputs of the 8 S-boxes of DES. Some of the generated SAC matrices have been given as examples in Table 6.1.0 and Table 6.2.0, and are compared with the tables which are generated using 1-bit alteration method, viz. Table 6.1.1 and Table 6.2.1.

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| V-vector of Input – 0 | | | |
| 6 | 2 | 6 | 0 |

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| V-vector of Input - 0 | | | |
| 5 | 3 | 4 | 2 |

Table 6.1.0. SAC Matrix of Input 0 of S-box 0 using 2-bit Alteration

Table 6.1.1. SAC Matrix of Input 0 of S-box 0 using 1-bit Alteration

### *6.5.1.1 Experimental Results for DES S-boxes*

The experimental results of proposed test on Data Encryption Standard (DES) **[30]** S-boxes are given in Table 6.3

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| V-vector of Input – 0 | | | |
| 6 | 6 | 5 | 1 |

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| V-vector of Input – 0 | | | |
| 4 | 5 | 4 | 3 |

Table 6.2.0. SAC Matrix of Input 0 of S-box 1 using 2-bit Alteration

Table 6.2.1. SAC Matrix of Input 0 of S-box 1 using 1-bit Alteration

| S-box# | Observed Mean | Variance | Standard Deviation | Coefficient of Variance |
|---|---|---|---|---|
| 0 | 3.007813 | 1.648376 | 1.283891 | 0.426852 |
| 1 | 3.039063 | 1.264099 | 1.124322 | 0.369957 |
| 2 | 3.015625 | 1.140381 | 1.067886 | 0.354118 |
| 3 | 2.992188 | 1.281189 | 1.131896 | 0.378284 |
| 4 | 2.980469 | 1.698837 | 1.303395 | 0.437312 |
| 5 | 3.078125 | 1.384521 | 1.176657 | 0.382264 |
| 6 | 3.03125 | 1.499023 | 1.224346 | 0.403908 |
| 7 | 3 | 1.648438 | 1.283915 | 0.427972 |

Table 6.3. Experimental Results of DES S-boxes

The column graph analysis of co-variance (CV) and standard deviation (SD) for every S-box of DES is shown in Figure 6.3 and Figure 6.4, respectively.

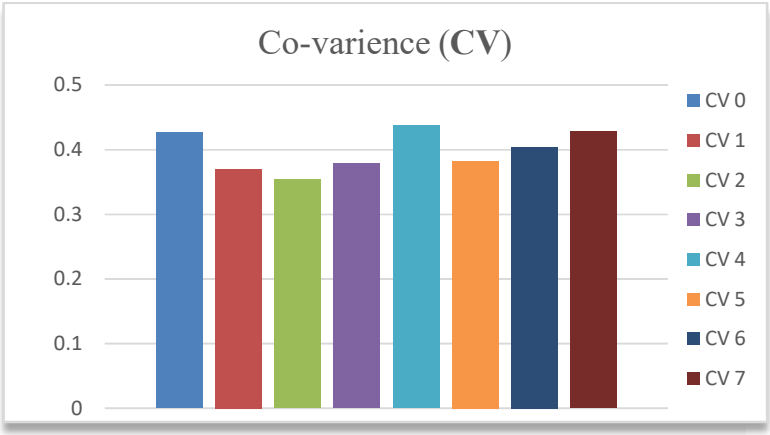


*Figure 6.3: Analysis of Co-variance (CV) for S-boxes of DES*

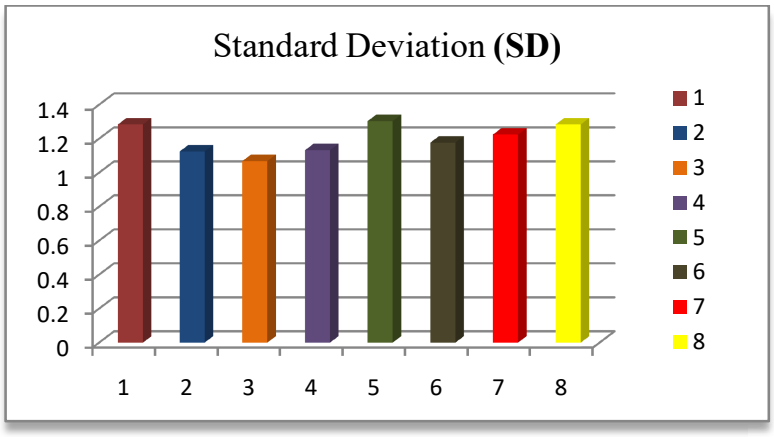


*Figure 6.4: Analysis of Standard Deviation (SD) for S-boxes of DES*

The comparison line graph of Coefficient of Variance (CV) and Standard Deviation (SD) for DES is shown in figure 6.5.

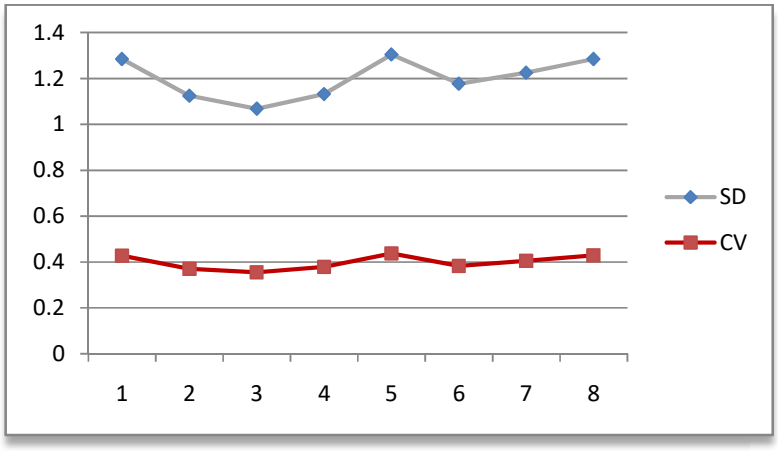


*Figure 6.5: Comparison Analysis of SD and CV for S-boxes of DES*

### 6.5.1 Coefficient of Variance Analysis of Generated SAC of S-box of DES

Using the proposed algorithm, 1*s* of every column of every output of SAC matrix generated from every possible input to the only S-box of AES have been counted. Some of the generated SAC matrices have been given as examples in Table 6.4.0 and Table 6.5.0, and are compared with the tables which are generated using 1-bit alteration method, viz. Table 6.4.1 and Table 6.5.1.

| Input : 11000011<br>Original Output : 00101110 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| V-vector of input 11000011 | | | | | | | |
| 3 | 6 | 3 | 4 | 1 | 5 | 2 | 4 |

**Table 6.4.0. SAC Matrix of input 11000011 to AES S-box using 2-bit Alteration**

| Input : 11000011<br>Original Output : 00101110 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| V-vector of input 11000011 | | | | | | | |
| 6 | 4 | 4 | 4 | 4 | 3 | 4 | 5 |

**Table 6.4.1. SAC Matrix of input 11000011 to AES S-box using 1-bit Alteration**

| Input : 10101010<br>Original Output : 10101100 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| V-vector of input 10101010 | | | | | | | |
| 4 | 5 | 3 | 2 | 6 | 4 | 4 | 4 |

**Table 6.5.0. SAC Matrix of input 10101010 to AES S-box using 2-bit Alteration**

| Input : 10101010 | | | | | | | |
|---|---|---|---|---|---|---|---|
| Original Output : 10101100 | | | | | | | |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| V-vector of input 10101010 | | | | | | | |
| 3 | 2 | 2 | 5 | 6 | 3 | 3 | 6 |

**Table 6.5.1. SAC Matrix of input 10101010 to AES S-box using 1-bit Alteration**

***6.5.2.1 Experimental Results for AES S-box***

The experimental results of the proposed test on the Advanced Encryption Standard (AES) **[31]** S-box are given in the Table 6.6.

| **Input** | **Observed Mean** | **Variance** | **Standard Deviation** | **Coefficient of Variance** |
|---|---|---|---|---|
| $(195)_{10}$ | 3.5 | 2.5 | 1.5 | 0.428571 |
| $(170)_{10}$ | 4 | 1.25 | 1.118034 | 0.279508 |
| $(204)_{10}$ | 3.625 | 1.734375 | 1.316957 | 0.363298 |
| $(105)_{10}$ | 4.125 | 1.859375 | 1.363589 | 0.330567 |
| $(45)_{10}$ | 2.875 | 1.859375 | 1.363589 | 0.474292 |
| $(210)_{10}$ | 3.625 | 1.234375 | 1.111024 | 0.306489 |

**Table 6.6. Experimental Results of AES S-box**

The column graph analysis of co-variance (CV) and standard deviation (SD) for the S-box of AES is shown in figure 6.6 and figure 6.7, respectively.

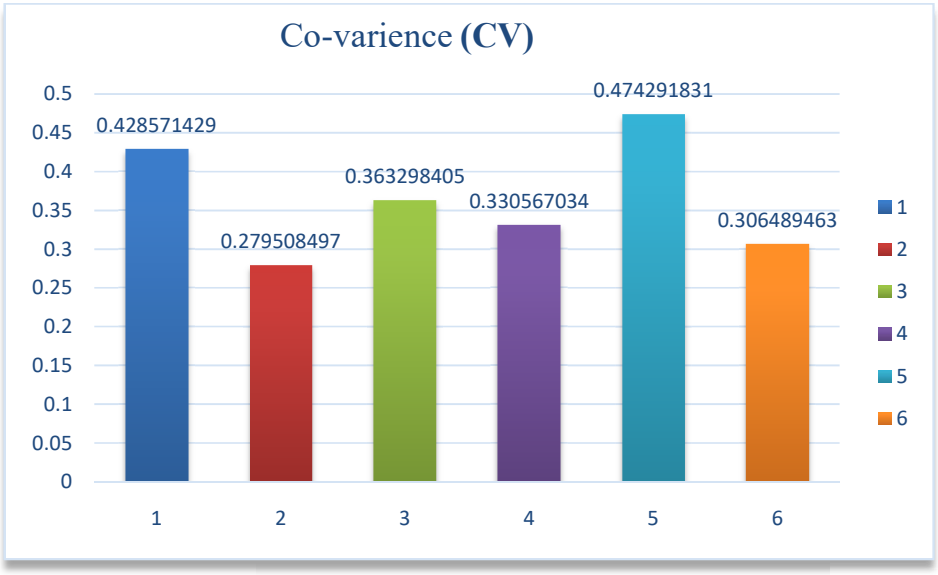


*Figure 6.6: Analysis of Co-variance (CV) for the S-box of AES*

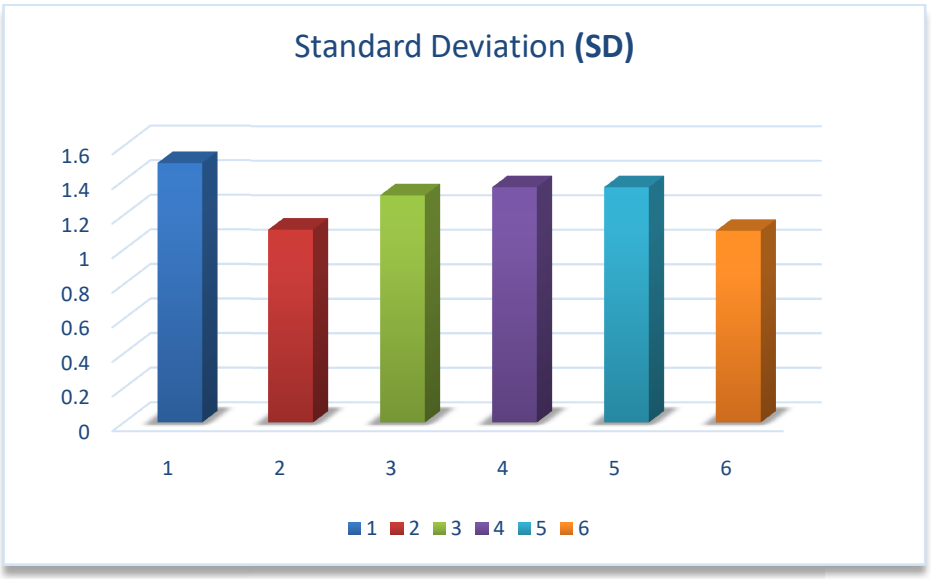


*Figure 6.7: Analysis of Standard Deviation (SD) for the S-box of AES*

The comparison line graph of Coefficient of Variance (CV) and Standard Deviation (SD) for AES is shown in figure. 6.8.

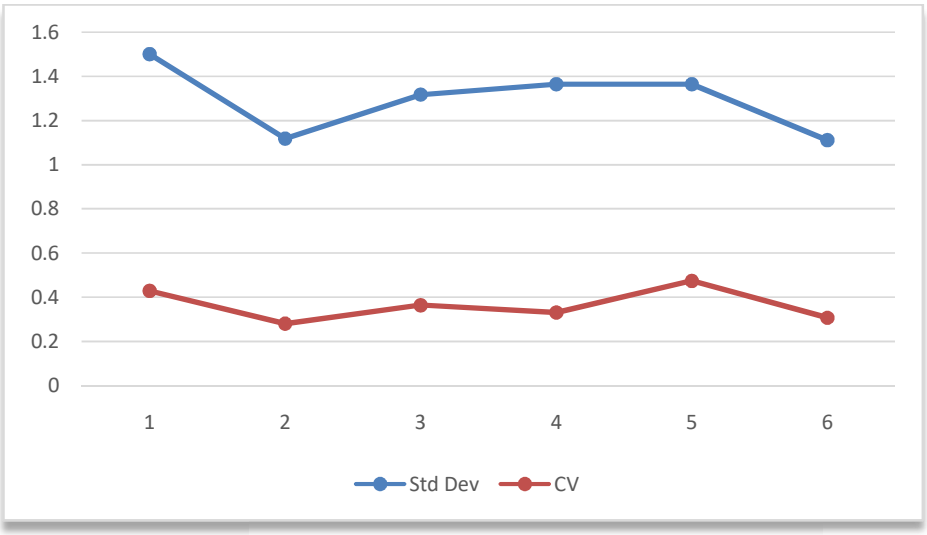


***Figure 6.8: Comparison Analysis of (SD) and (CV) for the S-box of AES***

## 6.6 Discussion

The outputs corresponding to every possible input to every S-box of DES and six numbers of randomly chosen inputs to the single S-box of AES have been computed using proposed algorithm and coefficient of variance has been calculated as a statistical measure of dispersion of data point in a data series around the mean. The coefficient of variance (CV) = Standard Deviation / Mean.

The coefficient of variance in case of S-boxes of DES ranges between 0.3 and 0.4 which is around the lower end of the spectrum and indicates that S-boxes of DES perform mostly well with respect to the proposed test.

For the S-box of AES, on the other hand, the obtained result varies from 0.2 to 0.4 which is also lower end of the spectrum, which indicates that the performance of the S-box of AES is also pretty well with respect to the proposed test.

## 6.7 Conclusions

The cryptographic strength of a block cipher can be measured using two fundamental aspects, confusion and diffusion. There exist many methods to test diffusion and confusion characteristics of cryptographic algorithms. In Chapter 5, the confusion characteristics has been measured by altering every 1-bit of all possible original inputs and SAC matrices have been generated for S-boxes of DES and AES. The occurrence of 1s in every column has been statistically analyzed. In this proposed method, the confusion characteristics are also measured by generating SAC matrices for S-boxes of DES and AES by altering 2-bits of all possible original inputs. The statistical analysis using the proposed method shows good comparative results and will serve very well to analyze the confusion

characteristics. This method may be included as a part of a comprehensive test suite for analyzing the cryptographic strength of a block cipher.

In future, the analysis of the strength of block ciphers against cryptographic attacks may be extended to parity analysis for all possible inputs to S-boxes of DES and AES. Analyses to explore strength of the S-boxes of DES and AES against the Boomerang Attack or Related-Key Boomerang attack are in the pipeline.

# Chapter 7: SAC Analysis with Truncated Differentials †

In this chapter SAC matrices have been implemented for S-boxes of DES and AES to implement analysis of higher order differentials, known as truncated differentials **[49]**. This new approach will help cryptanalysts to determine the vulnerability of ciphers. After getting the original outputs corresponding to the input strings, inputs to S-boxes of DES and AES are then truncated into two parts, strings $(\mathbf{a}, \mathbf{b})$, of equal bit length Then each bit of both $\boldsymbol{a}$ and $\boldsymbol{b}$ is changed one after the other to get the new input and its corresponding output. Using all outputs of every possible input, SAC matrices are generated for statistical and truncated differential analysis to reach the conclusion.

## 7.1 Introduction

The Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are the well known and most widely used cryptosystem developed by IBM in the mid 70's. The method which analyzes the effect of differences in plaintext pairs on the differences of generated ciphertext pairs is called Differential Cryptanalysis. In $n-$ iterated cryptosystem, iterations are called a round and the cryptosystem is called $n-round$ cryptosystem. A function of the output of previous round and a *subkey* which is a key dependent value calculated using a key scheduling algorithm is called the *round function*.

For a given plaintext $P = (P^L, P^R)$ and round keys $(r)$ $K_1,\ K_2, \ldots..,\ K_r$ the ciphertext $C = (C^L, C^R)$ is generated which is computed in $r$ rounds **[49]**. The differential attacks can take place on the calculated differences between these pair of plaintexts and their corresponding ciphertext pair using a non-uniform probability distribution.

Conventional differential is a difference of two bit-strings of the same length. A differential that predicts only parts of an $n$-bit value is called truncated differential **[49]**. Let $(\mathbf{a}, \mathbf{b})$ be the $i-$round differential. If $a'$ is a subsequence of $a$ and $b'$ is subsequence of $b$, then $(\boldsymbol{a}', \boldsymbol{b}')$ is called an $i-$round truncated differential. This type of analysis has been implemented for encryption algorithms and keys, but no existing work has been found that uses this approach to analyze S-boxes.

In this chapter, a new way of implementing truncated differential on generated SAC matrices of every possible input and corresponding output of S-boxes of DES and AES is proposed and the vulnerability to various attacks is statistically analyzed.

---

† This chapter is referenced from the published research paper: "A New Approach towards Confusion Analysis of S-boxes using Truncated Differential Cryptanalysis, International Journal of Computer Sciences and Engineering, Volume-7, Issue-1 E-ISSN: 2347-2693".

## 7.2 Preliminaries

In this section some of the related works on differential cryptanalysis has been included to facilitate futher discussions.

Xuejia Lai discussed the higher order derivatives and differential cryptanalysis in **[50]**. His definition on higher order derivative is:

Let, $(S,+)$ and $(T,+)$ be Abelian groups. For a function $f : S \longrightarrow T$, the derivatives of the $f$ at point $a \in S$ is defined as

$$\triangle_{\boldsymbol{a}} \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}+\boldsymbol{a}) - \boldsymbol{f}(\boldsymbol{x}) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

The derivatives of $f$ is a function from $S$ to $T$ and define the $i-th$ derivative of $f$ at $(a_1, a_2, \ldots., a_i)$ as

$$\triangle^{\boldsymbol{i}}_{\boldsymbol{a_1},\boldsymbol{a_2},\ldots.,\boldsymbol{a_i}} \boldsymbol{f}(\boldsymbol{x}) = \triangle_{\boldsymbol{a}} \left(\triangle^{\boldsymbol{i-1}}_{\boldsymbol{a_1},\boldsymbol{a_2},\ldots.,\boldsymbol{a_{i-1}}} \boldsymbol{f}(\boldsymbol{x})\right) \quad \ldots\ldots\ldots\ldots\ldots \quad (2)$$

where $\triangle^{i-1}_{a_1,a_2,\ldots.,a_{i-1}} f(x)$ being the $(i-1)-th$ derivative of $f$.

The derivatives for binary functions are computed with the group of bitwise XOR operations denoted by $\oplus$.

The differential and probability of a differential is defined as a couple of $(\mathbf{a}, \mathbf{b})$ and $P(\triangle y = b | \triangle x = a)$.

Biham and Shamir in *Differential Cryptanalysis of DES-like Cryptosystems* **[51]** introduced differential cryptanalysis on DES considering the iterative rounds based on S-boxes, bit permutations, arithmetic operations and the exclusive-OR operations.

Kaisa Nyberg in **[52]** discussed differentially uniform mapping for cryptography with other cryptographic properties like large distance from affine functions, high nonlinear order and efficient computability. Special focus has been given on the nonlinearity properties of round functions and it seems that the security of the cryptosystem may be increased by increasing the size of S-boxes or may be by replacing the set of small parallel substitution by one large transformation with desirable properties.

Nyberg and Knudsen in **[53]** have shown that DES-like iterated ciphers are provably resistant against differential attacks.

It is shown in **[54]** that perfect non-linear mappings from $GF(2)^m \longrightarrow GF(2)^n$ only exist for an even $m$ and $m \geq 2n$ and they can be included in DES-like ciphers with expansion mappings that double the block length.

In *Security of E2 against Truncated Differential Cryptanalysis* **[55]**, truncated differential cryptanalysis has been done to ensure security of E2. They evaluated the security against the attacks using truncated differentials with bitwise differentials.

An improved truncated differential cryptanalysis of KLEIN **[56]** has been introduced by Rasoolzadeh, Shahram, et al. KLEIN is a type of light weighted block cipher which has three variants, namely KLEIN-64, -80, -96 with having 12, 16 and 20 rounds, respectively. It has an SPN structure which combines 4-bit S-boxes with AES's MixColumn.

Truncated differential cryptanalysis has also been done for Camellia block cipher, which was cooperatively designed by NTT and Mitsubishi Electric Corporation and submitted to NESSIE by S. Lee, S. Hong et.al **[57]**. They presented truncated differential cryptanalysis for the modified Camellia with 7 rounds and found 8-bit key with $3.2^{81}$ plaintexts and for Camellia with 8 rounds they found 16-bit key with $3.2^{82}$ plaintexts.

Forrié R. **[8]** has introduced a Boolean function $f(\underline{x})$ with $n$ bits input and one bit output. Walsh-transform has shown to be very useful for the statistical analysis of properties of Boolean functions. A Boolean function $f(\underline{x})$ fulfills the SAC if and only if, for all $i \in \{1,2, \ldots \ldots, n\}$, its Walsh-transform

$$\hat{F}(\underline{w}),\ \underline{w} = [w_1, w_2, \ldots\ldots, w_n], \qquad \ldots\ldots\ldots\ldots (3)$$

fulfills

$$\sum_{\underline{w} \in Z_2^n} (-1)^{w_i} . \hat{f}^2(\underline{w}) = 0, \qquad \ldots\ldots\ldots\ldots (4)$$

where $Z_2^n$ denotes the $n$-dimensional vector space over the finite field GF(2). Theorem stated here proves that a Boolean function fulfills the SAC if and only if it is 50% dependent on each of its input bits.

**Definition 1 [8]**: A function $\hat{f} : Z_2^n \longrightarrow \{1, -1\}$ $(resp. Z_2^n \longrightarrow \{1,0\})$ is said to be 50% dependent on its i-th input bit $x_i$ if and only if any two $n - tuples$ $\underline{x}$ and $\underline{x_i}$ that differ only in bit $i$ are mapped onto two different values with probability $1/2$ and onto the same value with the same probability $1/2$.
or formally,

$$\sum_{\underline{x} \in Z_2^n} \hat{f}(\underline{x}) \cdot \hat{f}\left(\underline{x} \oplus \underline{c_i}\right) = \mathbf{0} \qquad \ldots\ldots\ldots\ldots\ldots (5)$$

for {1,-1} – valued functions, and

$$\sum_{\underline{x} \in Z_2^n} \hat{f}(\underline{x}) \oplus \hat{f}\left(\underline{x} \oplus \underline{c_i}\right) = \mathbf{2^{n-1}} \qquad \ldots\ldots\ldots\ldots\ldots (6)$$

for {1,0} – valued functions.

Thus a Boolean function fulfills the SAC if and only if it is 50% dependent on each of its input bits.

The sufficient condition for a function to be 50% dependent on one or more of its input bits can be extracted from the following theorem.

**Theorem 1 [8]**: If for some non-zero $\underline{c} \in Z_2^n$ and for all $\underline{w} \in Z_2^n$

$$\hat{f}^2(\underline{w}) = \hat{f}^2(\underline{w} \oplus \underline{c}) \quad \ldots\ldots\ldots\ldots (7)$$

holds, and if $\underline{c}$ has Hamming weight $m$ ($c_1 = c_1 = \cdots = c_m, 1 \leq m \leq n$), then $\hat{f}(\underline{x})$ is 50% dependent on the input bits $x_1, x_2, \ldots, x_m$.

## 7.3 Differential Attacks and Truncated Differential

If we consider a Feistel cipher with block size $2n$ with $r$ rounds then the round function $g$ is:

$$g: GF(2)^n \times GF(2)^n \times GF(2)^m \rightarrow GF(2)^n \times GF(2)^n$$
$$\boldsymbol{g(X,Y,Z) = (Y, f(Y,Z) + X)} \quad \ldots\ldots\ldots\ldots (8)$$

where $f$ can be any function accepting two parameters, $n$ bits and $m$ bits and producing $n$ bits. For any given plaintext $P = (P_L, P_r)$, the differential attacks exploit the difference of certain plaintexts and the difference of corresponding ciphertexts with non-uniform probability distribution.

The number of times the right key is counted over the number of times a random key is counted is called signal to noise ratio **[49]** and can be represented as:

$$\boldsymbol{S/N = \frac{|K| \times p}{\gamma \times \lambda}} \quad \ldots\ldots\ldots\ldots (9)$$

where $p$ is the probability of differential used in the attack, |K| is the number key, $\gamma$ is the key suggested for each pair of plaintext and $\lambda$ is the ratio of non-discarded pairs to all pairs. If $S/N \leq 1$ then a differential attack will not succeed **[51]**.

For generic differential attack on $2n$ Fiestel cipher, the prediction is done on $n$ bits of ciphertext. The differential that predicts only a part of $n$ bits is called truncated differential. Attackers have more freedom in choosing plaintexts or ciphertexts when using the truncated differential. So, ensuring strength and security against truncated differential analysis can provide a more strict evaluation of the security against differential cryptanalysis **[55]**.

## 7.4 Design of an S-box and SP Network

To get the advantages for attack on cryptosystem, cryptanalyst can collect information about statistical properties if strict avalanche criterion (SAC) or avalanche variable independence requirement is not satisfied **[7]**.

To satisfy both SAC and avalanche variable independence requirement, a cryptographic transformation has been discovered as substitution boxes or S-boxes. An S-box should randomly select potentially invertible and single output bit function that satisfies SAC. Moreover when each input bit is complemented, the resulting avalanche variables are pair-wise independent **[7]**. The S-box that satisfies both of these properties is referred as perfect S-box.

As the heart of modern cryptography, Claude Shannon **[58]** proposed the Substitution Permutation (SP) network, and to confirm the confidentiality of bits of data in encryption or decryption of SP network, three basic steps has been described in **[36]** as:

a. Subkey is X-ORed with input data bits.

b. A substitution function $S_i : \mathbb{Z}_{2^n} \rightarrow \mathbb{Z}_{2^m}$ replaces $n$ bits of data to increase confusion with lookup tables called S-boxes.

c. Permutation function shuffles the bits to cause diffusion within the data.

Fig. 7.1 shows a substitution permutation network with 3 rounds. S-boxes can be considered as Boolean mapping from $\{0,1\}^m \rightarrow \{0,1\}^n$ in the form $n \times n$ . S-boxes are also the only non-linear form of operation in an encryption process.

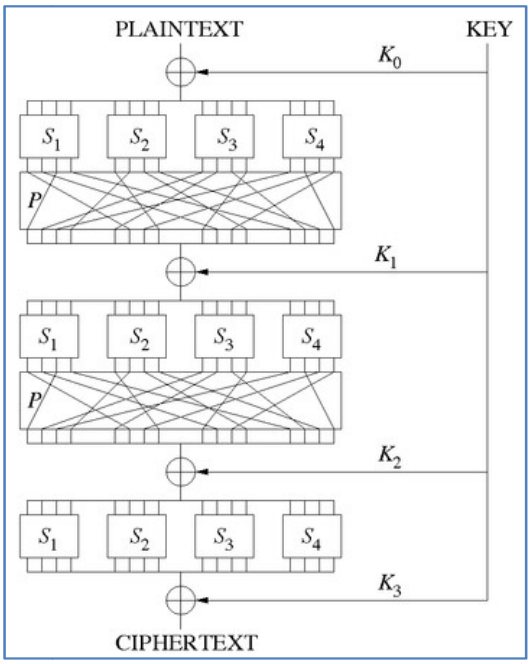


***Figure 7.1: SP Network with 3 rounds***

## 7.5 Proposed Method

Two different approaches have been presented **[59][60]**, where confusion of S-boxes has been analyzed statistically.

In the first approach **[59]**, the SAC (Strict Avalanche Criterion) matrix has been generated in comparison to the original ciphertext. All elements of DES S-boxes and AES S-box take inputs individually and the SAC matrix is generated from the original ciphertext along with ciphertexts generated from the every one alternative bit alteration of the original inputs. By using the generated SAC matrix, the vulnerability of every bit of the ciphertext has been statistically computed and discussed.

In the other approach **[60]**, the SAC matrix was generated with original ciphertext and ciphertexts of every two alternative bit alteration of original inputs. The method has been used for all 8 S-boxes of DES and the S-box of AES.

A totally new approach, the truncated differential cryptanalysis is being proposed here and implemented on S-boxes of DES and S-box of AES with the statistical analysis on

the generated SAC. The proposed method has been compared with the conclusion of 2-bit approach **[60]** of confusion analysis of S-box.

The proposed method involves the following:

1) Analysis of frequency of every bit column-wise and its various avalanche effects from the generated SAC using truncated approach.
2) Coefficient variance analysis of generated SAC.
3) Analysis of frequency of various differential values from the generated SAC.

Using the V-vector (Vulnerability Vector) **[62]**, the proposed algorithm given in section 7.5.1.

### 7.5.1 Proposed Algorithm

**Algorithm – Confusion Analysis of S-boxes using Truncated Differential Cryptanalysis**

**Input**: S-box with length $n$, where $n$ is the number of bits.

**Step1**: Choose a random number $P$, within the range of the S-box, where $P \in \mathbb{Z}_2^n$. Find the corresponding output value of S-box:

$C = S(P)$

**Step 2**: Change the $P_i$s to find their corresponding output values $C_i$s.

$P_i$ may be generated by:

- Truncating the original input $P$ of same size in $P(a, b)$ and getting two parts of $P$ as $P_a$ and $P_b$.
- Change the every individual bit of both $P_a$ and $P_b$, for every iteration.
- Then concatenate $P_a$ and $P_b$, in every iteration, to generate $P_i$.

**Step 3**: SAC matrix has to be created by $S_i = C_i \oplus C$, which to be included in the $i^{th}$ row of a matrix of size $m \times n$, where $m$ is the number bits of $P$ and $n$ is the number of bits of $C$.

**Step 4**: Find the count of $1s$ in each column of generated SAC matrix.

## 7.6 Experimental Results

### 7.6.1 Coefficient Variance (CV) Analysis of Generated SAC of S-boxes of DES using Truncated Differential Method

By following the proposed algorithm, using truncated plaintext $P(a, b)$, a SAC matrix has been generated for every possible input of every S-box of DES. The $1s$ of every column output of S-boxes have been counted and the sum of $1s$ of every column is being identified as V-vector (**V**ulnerability Vector) and computed for all 64 possible inputs and corresponding outputs of the 8 S-boxes. Some of the generated SAC matrices has been given in Table 1.0 and 2.0 and are compared with tables 1.1, 2.1 and 1.2, 2.2 which are generated from 2-bit **[60]** and 1-bit **[59]** alteration methods. The line graph of frequencies of V-vector for all 8 S-boxes of DES is showed in Fig. 7.2.

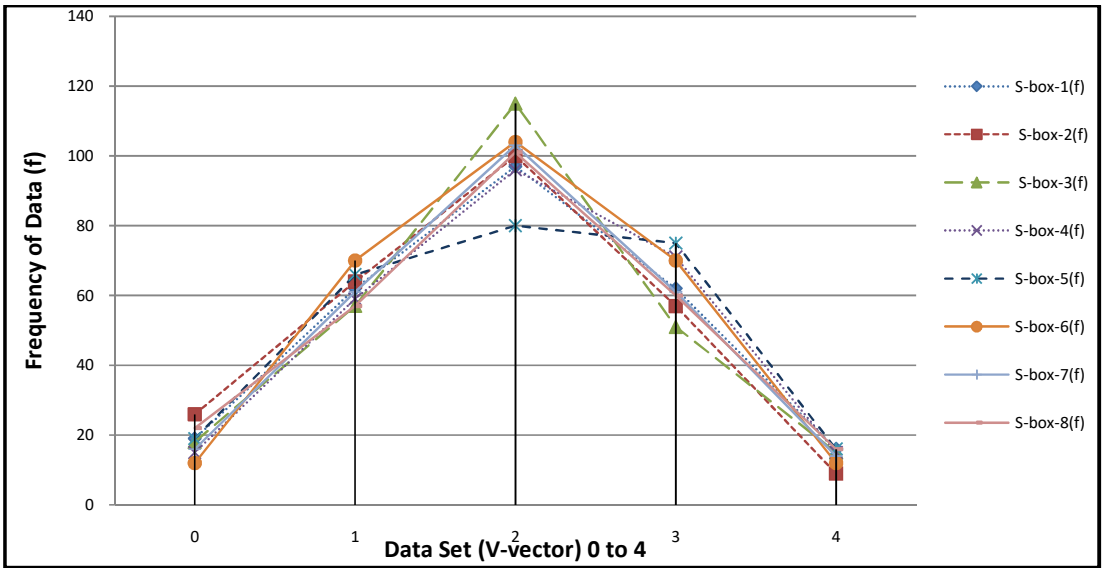


*Figure 7.2: Line Graph of Frequencies of V-vector of S-boxes of DES*

| **SAC Matrix of Input - 0** | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| **V-vector of Input - 0** | | | |
| 3 | 3 | 1 | 3 |

**Table 7.1.0: SAC Matrix of input 0 of S-box 0 using Truncated Differential Approach**

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| V-vector of Input - 0 | | | |
| 6 | 2 | 6 | 0 |

**Table 7.1.1: SAC Matrix of input 0 of S-box 0 using 2-bit Alteration Approach**

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| V-vector of Input - 0 | | | |
| 5 | 3 | 4 | 2 |

**Table 7.1.2: SAC Matrix of input 0 of S-b0x 0 using 1-bit Alteration Approach**

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| V-vector of Input – 0 | | | |
| 3 | 3 | 2 | 1 |

Table 7.2.0: SAC Matrix of input 0 of S-box 1 using Truncated Differential Approach

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| V-vector of Input – 0 | | | |
| 6 | 6 | 5 | 1 |

Table 7.2.1: SAC Matrix of input 0 of S-box 1 using 2-bit Alteration Approach

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| V-vector of Input – 0 | | | |
| 4 | 5 | 4 | 3 |

Table 7.2.2: SAC Matrix of input 0 of S-box 1 using 1-bit Alteration Approach

### 7.6.2 Experimental Results for DES S-boxes

The experimental results of proposed test are in Table 7.3:

| S-box | Observed Mean | Variance | Standard Deviation | Coefficient of Variance |
|---|---|---|---|---|
| 1 | 1.976563 | 1.030701 | 1.405903 | 0.711287 |
| 2 | 1.839844 | 0.993881 | 1.356408 | 0.737241 |
| 3 | 1.953125 | 0.935303 | 1.397542 | 0.715542 |
| 4 | 2.046875 | 0.974365 | 1.43069 | 0.698963 |
| 5 | 2.044719 | 1.097519 | 1.418351 | 0.705044 |
| 6 | 1.953125 | 0.872803 | 1.397542 | 0.715542 |
| 7 | 1.980469 | 0.948837 | 1.407291 | 0.710585 |
| 8 | 1.964844 | 1.049545 | 1.401729 | 0.713405 |

Table 7.3: Experimental Results of Proposed Test on S-boxes of DES

### 7.6.3 Coefficient Variance (CV) Analysis of Generated SAC of S-box of AES using Truncated Differential Method

By following the proposed algorithm, using truncated plaintext $P(a, b)$, a SAC matrix has been generated for S-box of AES. $1s$ of every column output of S-box has been

counted and the sum of $1s$ of every column, is being identified as V-vector (**V**ulnerability Vector), has calculated for some example inputs and corresponding outputs of the S-box. Some of the generated SAC matrices has given below in Table 7.4.0 and 7.5.0 and are compared with tables 7.4.1, 7.5.1 and 7.4.2, 7.5.2 which are generated from 2-bit [60] and 1-bit [59] alteration method. The line graph of frequencies of V-vector for all inputs using S-box of AES is showed in Fig. 7.3.

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **V-vector of input 11000011** | | | | | | | |
| 1 | 3 | 1 | 0 | 3 | 2 | 2 | 1 |

**Table 7.4.0: SAC Matrix of Input 11000011 to AES S-box using Truncated Differential Approach**

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| **V-vector of input 11000011** | | | | | | | |
| 3 | 6 | 3 | 4 | 1 | 5 | 2 | 4 |

**Table 7.4.1: SAC Matrix of Input 11000011 to AES S-box using 2-bit Alteration Approach**

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 |
| **V-vector of input 11000011** | | | | | | | |
| 6 | 4 | 4 | 4 | 4 | 3 | 4 | 5 |

**Table 7.4.2: SAC Matrix of Input 11000011 to AES S-box using 1-bit Alteration Approach**

| Input : 10101010 | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| **V-vector of input 10101010** | | | | | | | |
| 1 | 2 | 2 | 3 | 2 | 3 | 2 | 3 |

**Table 7.5.0: SAC Matrix of Input 10101010 to AES S-box using Truncated Differential Approach**

| Input : 10101010 | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| **V-vector of input 10101010** | | | | | | | |
| 4 | 5 | 3 | 2 | 6 | 4 | 4 | 4 |

**Table 7.5.1: SAC Matrix of Input 10101010 to AES S-box using 2-bit Alteration Approach**

| Input : 10101010 | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| **V-vector of input 10101010** | | | | | | | |
| 3 | 2 | 2 | 5 | 6 | 3 | 3 | 6 |

**Table 7.5.2: SAC Matrix of Input 10101010 to AES S-box using 1-bit Alteration Approach**

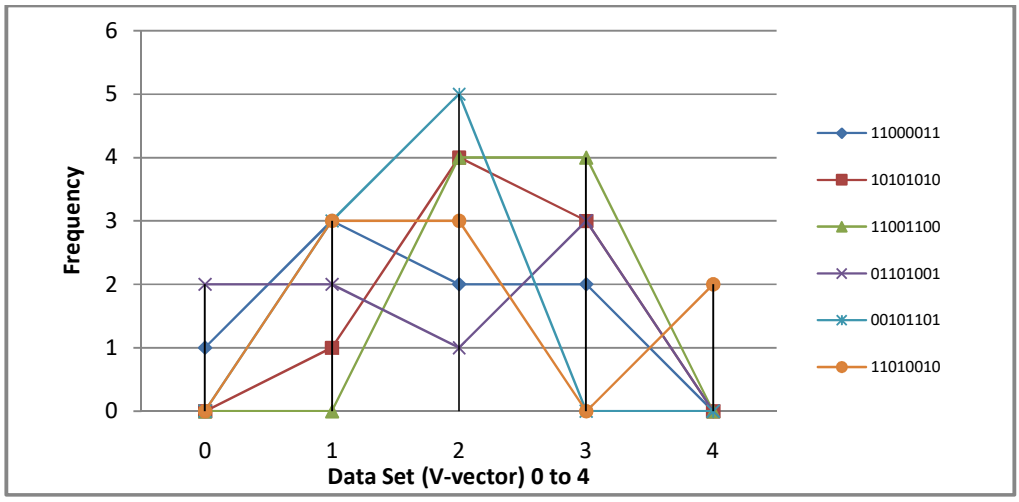


***Figure 7.3: Line Graph of Frequencies of V-vector for Inputs using S-box of AES***

### 7.6.4 Experimental Results for AES S-box

The experimental results of proposed test on the S-box of AES are in Table 7.6:

| Input | Observed Mean | Variance | Std. Deviation | Coefficient of Variance |
|---|---|---|---|---|
| $(195)_{10}$ | 1.625 | 0.984375 | 0.992157 | 0.610558 |
| $(170)_{10}$ | 2.25 | 0.4375 | 0.661438 | 0.293972 |
| $(204)_{10}$ | 2.5 | 0.25 | 0.5 | 0.2 |
| $(105)_{10}$ | 1.625 | 1.484375 | 1.218349 | 0.749753 |
| $(45)_{10}$ | 1.625 | 0.234375 | 0.484123 | 0.297922 |
| $(210)_{10}$ | 2.125 | 1.359375 | 1.165922 | 0.548669 |

**Table 7.6: Experimental Results of Proposed Test using AES S-box**

## 7.7 Discussion

By using the proposed algorithm, coefficient of variance has been calculated as a statistical measure of dispersion for the output corresponding to all possible inputs of every S-box of DES and six numbers of randomly chosen inputs to the single S-box of AES.

From the generated SAC matrix from each S-box of DES, the vulnerability vector (V-vector) has been calculated by summation of $1's$ of every column of SAC matrices and by using the data set ranging from 0 to 4 and it's frequency of appearance, the line graph has been plotted in Fig. 7.2. To calculate the coefficient of variance of every S-box, statistical mean, variance and standard deviation have also been calculated. It is found that the coefficient of variance (CV) ranges from 0.69 to 0.73, where CV < 1 and average coefficient of variance of S-boxes of DES is 71%.

Using the single S-box of AES, the V-vector has been calculated in same way and using data set ranging from 0 to 4 and it's frequency of appearance, the line graph has been plotted in Fig. 7.3. To calculate the coefficient of variance of the inputs using single S-box, statistical mean, variance and standard deviation have also been calculated. The coefficient of variance (CV) is found to range from 0.20 to 0.74 where CV < 1 and average coefficient of variance of S-box of AES is 45%.

## 7.8 Conclusion

Confusion and diffusion are the two major aspects to measure of the strength of a block cipher and there exist different methods to test diffusion and confusion characteristics of cryptographic algorithms. In this proposed method, to test the confusion characteristic, the truncated differential approach has been used to analyze statistically the occurrence of $1's$ of every column of SAC matrices of DES S-boxes and AES S-box.

For the both cases of AES and DES, the coefficient of variance (CV) is ranging from 0.69-0.73 and 0.2-0.74, respectively, which are in the lower end of the spectrum which indicating that performances of S-boxes of DES and AES are pretty good with respect to the proposed test using truncated differential approach. The average CV of DES and AES is 71% and 45%, respectively, which helps us to draw the conclusion that the performance of S-box of AES is better than the performance of S-boxes of DES.

The proposed truncated differential approach of testing of confusion characteristics of S-boxes will lead us to draw an algorithm of testing Boomerang Attack.

# Chapter 8: Boomerang-style Cryptanalysis on S-boxes [†]

In recent times, there exist several approaches of differential-style attacks like truncated differential attack, high-level differential attack, boomerang attack etc. This chapter includes the study of boomerang-style attack on S-boxes and a new SAC analysis approach to test the strength of S-boxes over boomerang-style attack. The proposed analysis is tested on each input elements of 8 S-boxes of DES and 8 input element of the lone S-box of AES. The vulnerability factor *n/2* has been measured by calculating all $\mathbf{1's}$ of every column from the generated SAC matrix. Finally a comparison of standard deviation, coefficient variance and other factors show the way towards the conclusion.

## 8.1 Introduction

Differential cryptanalysis is one of the most important cryptanalytic techniques in cryptology. All published block ciphers may probably be broken by using differential cryptanalysis. So, one of the most important responsibilities of the block cipher designer is to ensure protection and security against the differential cryptanalysis.

If the upper bound probability of any differential characteristic is tentatively $p$, the designer of the algorithm presumes, following the "folk theorem", that differential attack requires at least $1/p$ text to break the cipher **[63]**. Unfortunately the "folk theorem" is not always right. There is a type of differential attack, called Boomerang attack, which can allow an adversary to beat the $1/p$ bound in some cases.

In this chapter, the possibility of boomerang type attack on S-boxes of cryptographic algorithms has been explored. To represent the approach, the boomerang approach has been implemented to generate a SAC matrix for all possible inputs of all S-boxes of DES and all possible input of the S-box of AES. The generated SAC matrices are analyzed statistically to determine the vulnerability of the S-boxes.

## 8.2 Review of Existing Work

D. Wagner **[63]** introduced an attack called Boomerang attack. This attack led to prove that the so called folk theorem is not always right. According to the folk theorem, differential attack requires at least $1/p$ text to break a cipher where, $p$ is the upper bound probability of any differential characteristic. According to his research, if the

[†] This chapter is referenced from the published research paper: "A Novel Technique for SAC Analysis of S-Box for Boomerang-Style Attacks , International Journal of Computer Sciences and Engineering, Volume-7, Issue-5 E-ISSN: 2347-2693".

best characteristic for half of the rounds of the cipher has probability $q$, then the boomerang attack can be used successfully on $O(q^{-4})$ chosen text. In some cases, it may have $q^{-4} \ll p^{-1}$, in which boomerang attack allows one to beat the folk theorem bound. Boomerang attack also sometimes uses some extensive structures that are available in conventional differential attack.

Boomerang connectivity table is a good cryptanalysis tool **[64]**. The issue of dependency of two characteristics in a block cipher $E_m$ has been revisited and a new tool called Boomerang Connectivity Tool (BCT) has been proposed, which evaluates $r$, where $r$ is the differential propagation probability among the texts, in a systematic and easy-to-understand way when $E_m$ is composed of a single S-box layer. BCT shows that the probability around the boundary may be even higher than $p$ or $q$.

A variant of differential cryptanalysis against the block ciphers, Impossible Boomerang Attack (IBA), was introduced by Choi et. al. **[65]**. This research approach is the combination of differential cryptanalysis and boomerang attack.

Amplified Boomerang attack was introduced by Kim et. al. **[66]**. This research is dedicated on SHACAL, which is a 4-round block cipher **[67]**. SHACAL was designed by using hash standard SHA-1in encryption mode for the first time. Kim et. al. proposed a 10-step differential characteristic with probability $2^{-12}$in rounds 2 and 4. Using this characteristic, they described a 36-step boomerang distinguisher. With this distinguisher they devised amplified boomerang attacks on reduced round SHACAL with different size of key.

A key recovery attack on the full round of SQUARE using a related key boomerang distinguisher was proposed Koo et. Al. **[68]**. They constructed a 7-round related key boomerang distinguisher with probability $2^{-119}$by finding local collision **[69]**, and calculated its probability using ladder switch and local amplification techniques.

Analysis of vulnerability factor for truncated differential using SAC is being proposed here **[70]**. This research approach describes and analyzes the truncated differential on S-boxes of DES and AES.

## 8.3 Boomerang Attack

The attack considers four plaintexts $A, A', B, B'$ along with their ciphertexts $C, C', D, D'$. If $E(\cdot)$ is the encryption operation and divide the cipher into $E = E_1 \circ E_0$ where $E_0$ is the first half of the cipher and $E_1$ is the last half then differential characteristics are $\Delta \rightarrow \Delta^*$ for $E_0$ and $\nabla \rightarrow \nabla^*$ for $E_1^{-1}$. The boomerang attack is depicted in fig. 8.1.

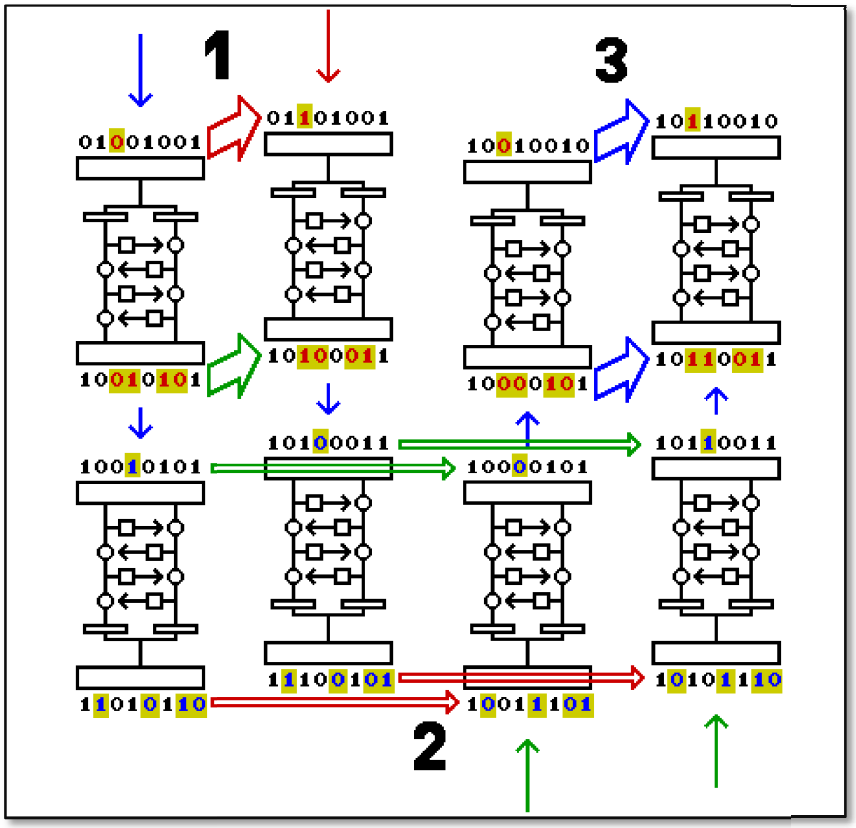


*Figure 8.1: Boomerang Attack*

## 8.4 Proposed Method

The work of this chapter has started with the confusion analysis of S-boxes. There are several previous research works in **[70]**, **[59]**, **[60]** where confusion of S-boxes has been analyzed as differential cryptanalysis. In this chapter, most importantly, confusion of S-boxes has been analyzed against the boomerang type attack.

In **[70]**, the truncated differential cryptanalysis approach has been implemented on S-boxes of DES and S-box of AES with a new proposal of the statistical analysis on the generated SAC. The proposed method has been compared with the conclusion of 2-bit approach **[60]** of confusion analysis of S-box.

In **[59]**, all elements of DES S-boxes and AES S-box take inputs individually and the SAC matrix is generated from the original ciphertext along with ciphertexts generated from the every one alternative bit alteration of the original inputs. In the generated SAC matrix, the vulnerability of every bit of the ciphertext has been statistically computed and discussed.

In the other approach **[60]**, the SAC matrix was generated with original ciphertext and ciphertexts of every two alternative bit alteration of original inputs. The method has been used for all 8 S-boxes of DES and the S-box of AES.

In this chapter, the boomerang-style attack has been implemented using S-boxes of DES and S-box of AES and a SAC has been generated. The SAC is then analyzed with a new proposed statistical analysis.

The proposed analysis in this chapter involves the following:

1) Analysis of frequency of every bit column-wise and its various avalanche effects from the generated SAC.
2) Coefficient variance analysis of generated SAC.
3) Analysis of frequency of various differential values from the generated SAC.

Using the V-vector (Vulnerability Vector) **[62]**, the proposed algorithm is as below:

### 8.4.1 Proposed Algorithm

**Algorithm – Confusion Analysis of S-boxes using Boomerang-style Attack**

**Input**: Elements of S-box with length $n$, where $n$ is the number of bits.

**Step 1**: Choose any random number $P$ within the range of the S-box, where $P \in \mathbb{Z}_2^n$. Find the corresponding output value of S-box:

$$C = S(P)$$

**Step 2**: Change the $P_i$s to find their corresponding output values $C_i$s.

$P_i$ may be generated by:

- Generating all possible pairs of input bits by using:
$$\frac{n(n-1)}{2}$$
- Making the number of pairs multiple of the input bit size by padding $0's$ at left side.
- Changing every even bit of every pair.
- Combining pairs to generate $P_i$.

**Step 3**: SAC matrix has to be created by $S_i = C_i \oplus C$, which to be included in the $i^{th}$ row of a matrix of size $m \times n$, where $m$ is the number bits of $P$ and $n$ is the number of bits of $C$.

**Step 4**: Find the count of $1s$ in each column of generated SAC matrix.

## 8.5 Experimental Results

### 8.5.1 Coefficient Variance (CV) Analysis of Generated SAC of S-boxes of DES using Boomerang-style Attack Method

By following the proposed algorithm, using boomerang-style plaintext, a SAC matrix has been generated for every possible input of every S-box of DES. The $1's$ of every

column output of S-boxes have been counted for all 64 possible inputs and corresponding outputs of the 8 S-boxes and identified as V-vector (**V**ulnerability Vector). Some of the generated SAC matrices has been given in Table 8.1.0 and 8.2.0 and are compared with tables 8.1.1, 8.2.1 and 8.1.2, 8.2.2 which are generated from truncated **[70]** and 2-bit **[60]** alteration methods. The line graph of frequencies of V-vector for all 8 S-boxes of DES is showed in Fig. 8.2

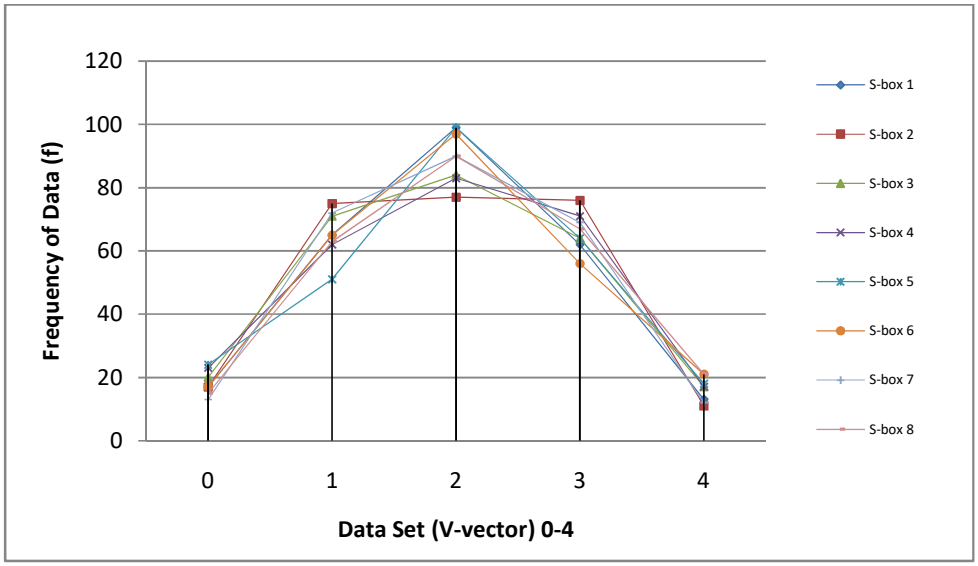


***Figure 8.2: Line Graph of Frequencies of V-vector of S-boxes of DES***

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| **V-vector of Input - 0** | | | |
| 2 | 0 | 3 | 2 |

**Table 8.1.0: SAC Matrix of Input '0' to 'S-box 0' Using Boomerang-style Approach**

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| **V-vector of Input - 0** | | | |
| 3 | 3 | 1 | 3 |

**Table 8.1.1: SAC Matrix of Input '0' to 'S-box 0' Using Truncated Differential Approach**

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| **V-vector of Input - 0** | | | |
| 6 | 2 | 6 | 0 |

**Table 8.1.2: SAC Matrix of Input '0' to 'S-box 0' using 2-Bit Alteration Approach**

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| **V-vector of Input - 0** | | | |
| 3 | 3 | 3 | 1 |

**Table 8.2.0: SAC Matrix of Input '0' to 'S-box 1' using Boomerang-style Approach**

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| **V-vector of Input – 0** | | | |
| 3 | 3 | 2 | 1 |

**Table 8.2.1: SAC Matrix of Input '0' to 'S-box 1' using Truncated Differential Approach**

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| **V-vector of Input – 0** | | | |
| 6 | 6 | 5 | 1 |

**Table 8.2.2: SAC Matrix of Input '0' to 'S-box 1' using 2-Bit Alteration Method**

### 8.5.2 Experimental Results for DES S-boxes

The experimental results of proposed test are in Table 8.3:

| S-box | Observed Mean | Variance | Standard Deviation | Coefficient of Variance |
|---|---|---|---|---|
| 1 | 1.957031 | 0.962997 | 0.981324 | 0.501435 |
| 2 | 1.957031 | 1.025497 | 1.012668 | 0.517451 |
| 3 | 1.949219 | 1.10289 | 1.050186 | 0.538773 |
| 4 | 1.988281 | 1.144394 | 1.069736 | 0.538034 |
| 5 | 2.003906 | 1.105453 | 1.051405 | 0.524678 |
| 6 | 1.996094 | 1.066391 | 1.032662 | 0.517341 |
| 7 | 1.980469 | 0.941025 | 0.970064 | 0.489816 |
| 8 | 2.0625 | 1.066406 | 1.032669 | 0.500688 |

**Table 8.3: Experimental Results of Proposed Test on S-boxes of DES**

### 8.5.3 Coefficient Variance (CV) Analysis of Generated SAC of S-box of AES using Boomerang-style Attack Method

By following the proposed algorithm, using boomerang-style plaintext, a SAC matrix has been generated for S-box of AES. $1s$ of every column output of S-box has been counted and the sum of $1s$ of every column, is being identified as V-vector (**V**ulnerability Vector), has calculated for some example inputs and corresponding outputs of the S-box. Some of the generated SAC matrices has given below in Table 8.4.0 and 8.5.0 and are compared with tables 8.4.1, 8.5.1 and 8.4.2, 8.5.2 which are

generated from truncated **[70]** and 2-bit [60] alteration method. The line graph of frequencies of V-vector for all inputs using S-box of AES is showed in Fig. 8.3.

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **V-vector of input 11000011** | | | | | | | |
| 3 | 3 | 4 | 3 | 4 | 5 | 1 | 3 |

**Table 8.4.0: SAC Matrix of Input 11000011 to AES S-box using Boomerang-style Approach**

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **V-vector of input 11000011** | | | | | | | |
| 1 | 3 | 1 | 0 | 3 | 2 | 2 | 1 |

**Table 8.4.1: SAC Matrix of Input 11000011 to AES S-box using Truncated Differential Approach**

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| **V-vector of input 11000011** | | | | | | | |
| 3 | 6 | 3 | 4 | 1 | 5 | 2 | 4 |

**Table 8.4.2: SAC Matrix of Input 11000011 to AES S-box using 2-bit Alteration Approach**

| **Input : 10101010** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| **V-vector of input 10101010** | | | | | | | |
| 4 | 3 | 3 | 4 | 4 | 5 | 2 | 3 |

**Table 8.5.0: SAC Matrix of Input 10101010 to AES S-box using Boomerang-style Approach**

| **Input : 10101010** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| **V-vector of input 10101010** | | | | | | | |
| 1 | 2 | 2 | 3 | 2 | 3 | 2 | 3 |

**Table 8.5.1: SAC Matrix of Input 10101010 to AES S-box using Truncated Differential Approach**

| **Input : 10101010** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| **V-vector of input 10101010** | | | | | | | |
| 4 | 5 | 3 | 2 | 6 | 4 | 4 | 4 |

**Table 8.5.2: SAC Matrix of Input 10101010 to AES S-box using 2-bit Alteration Approach**

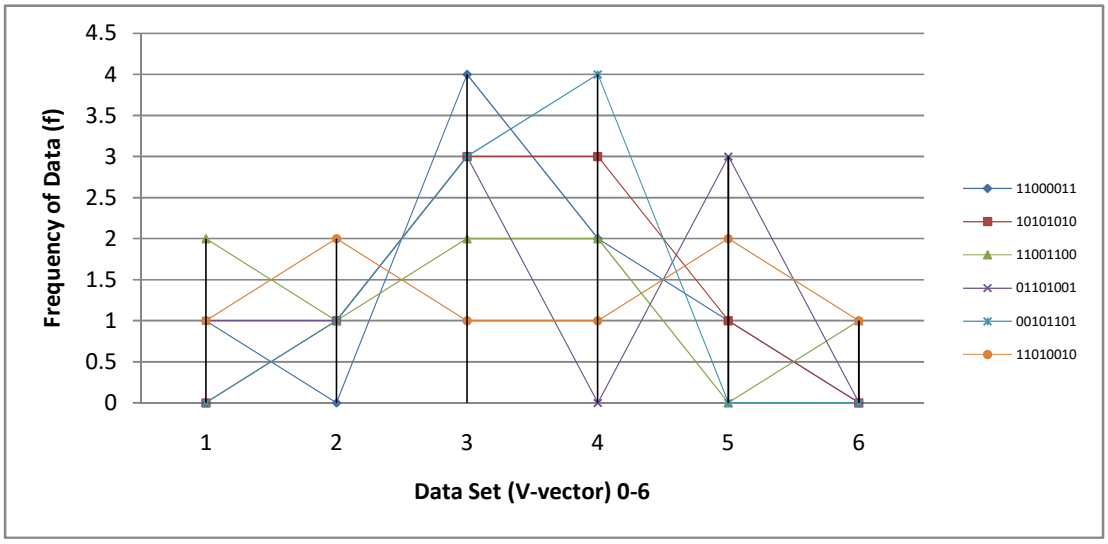


*Figure 8.3: Line Graph of Frequencies of V-vector for Inputs using S-box of AES*

### 8.5.4 Experimental Results for AES S-box

The experimental results of proposed test on the S-box of AES are in Table 8.6:

| Input | Observed Mean | Variance | Std. Deviation | Coefficient of Variance |
|---|---|---|---|---|
| $(195)_{10}$ | 3.25 | 1.1875 | 1.089725 | 0.3353 |
| $(170)_{10}$ | 3.5 | 0.75 | 0.866025 | 0.247436 |
| $(204)_{10}$ | 3 | 0.25 | 1.581139 | 0.527046 |
| $(105)_{10}$ | 3.375 | 1.984375 | 1.408678 | 0.417386 |
| $(45)_{10}$ | 3.375 | 0.484375 | 0.695971 | 0.206213 |
| $(210)_{10}$ | 3.5 | 2.75 | 1.658312 | 0.473804 |

Table 8.6: Experimental Results of Proposed Test using AES S-box

## 8.6 Discussion

By using the proposed algorithm, coefficient of variance has been calculated as a statistical measure of dispersion for the output corresponding to all possible inputs of every S-box of DES and six numbers of randomly chosen inputs to the single S-box of AES.

From the generated SAC matrix from each S-box of DES, the vulnerability vector (V-vector) has been calculated by summation of $1's$ of every column of SAC matrices and by using the data set ranging from 0 to 4 and it's frequency of appearance, the line graph has been plotted in Fig. 8.2. To calculate the coefficient of variance of every S-

box, statistical mean, variance and standard deviation have also been calculated. It is found that the coefficient of variance (CV) ranges from 0.48 to 0.53, where CV < 1 and average coefficient of variance of S-boxes of DES is 51%.

Using the single S-box of AES, the V-vector has been calculated in same way and using data set ranging from 0 to 4 and it's frequency of appearance, the line graph has been plotted in Fig. 8.3. To calculate the coefficient of variance of the inputs using single S-box, statistical mean, variance and standard deviation have also been calculated. The coefficient of variance (CV) is found to range from 0.20 to 0.52 where CV < 1 and average coefficient of variance of S-box of AES is 36%.

## 8.7 Conclusion

Confusion and diffusion are the two major aspects to measure of the strength of a block cipher and there exist different methods to test diffusion and confusion characteristics of cryptographic algorithms. In this proposed method, to test the confusion characteristic, the boomerang-style attack approach has been used to analyze statistically the occurrence of $1's$ of every column of SAC matrices of DES S-boxes and AES S-box.

For the both cases of DES and AES, the coefficient of variance (CV) is ranging from 0.48-0.53 and 0.2-0.52, respectively, which are in the lower end of the spectrum which indicating that performances of S-boxes of DES and AES are pretty good with respect to the proposed test using boomerang-style approach. The average CV of DES and AES is 51% and 36%, respectively, which helps us to draw the conclusion that the performance of S-box of AES is better than the performance of S-boxes of DES.

The proposed boomerang-style attack approach of testing of confusion characteristics of S-boxes will lead us to draw more testing algorithm on different cryptographic algorithms.

# Chapter 9: Comparative Study of the Proposed Algorithms

This research work carried out with five novel techniques of algorithms and detailed discussion has been done in chapter 4, 5, 6, 7 and 8 respectively. Some computed data and its statistical graphs are also being discussed in chapters respectively. Following are the algorithms that are used in this research work:

## 9.1 BLDAT – Bit Level Diffusion Analysis Test

This algorithm is a bit-wise diffusion analysis test approach of a plaintext input. The feature of this algorithm is that:

- A chosen plaintext has been taken as an input and enciphered with a standard encryption algorithm and the ciphertext has been saved.
- Every individual bits of $n$ number of bit has been altered with $0\ or\ 1$ if it is $1\ or\ 0$ in the original plaintext and then enciphered and $n$ number of output has been saved.
- A SAC matrix has been generated with the original ciphertext and $n$ number of altered ciphertext.
- By a novel approach $1's$ of every column, that is vertically, has been calculated and $\chi^2$ test led to the conclusion of the approach.

## 9.2 Bit Level Confusion Analysis of S-Box

After BLDAT, the research shifted to the S-boxes which led the work towards the confusion analysis of established encryption algorithms. Using this algorithm, S-boxes of DES and AES has been analyzed. The algorithm includes the following steps:

- Elements of S-boxes have been considered as input text and the generated output has been saved as an output element.
- For $n$ number of input bits, every bit has been altered $0\ or\ 1$ for $1\ or\ 0$ in the original input to get $n$ number of output element.
- A SAC matrix has been generated with original output text and $n$ number of altered output text.
- An approach used to calculate $1's$ of every column. The statistical approaches of coefficient of variance (CV) and standard deviation (SD) established a comparative conclusion about the strength of the S-box input.

## 9.3 A 2-Bit Approach Confusion Analysis of S-Box

The 1-Bit alteration approach of confusion analysis of S-Box, the research led to 2-bit alteration method. The followings are the steps included in the algorithms:

- Elements of S-boxes have been considered as input text and the generated output has been saved as an output element.
- For $n$ number of input bits, every consecutive two bit has been altered $0\ or\ 1$ for $1\ or\ 0$ in the original two bit input to get output element.
- A SAC matrix has been generated with original output text and altered output text.
- An approach used to calculate $1's$ of every column. The statistical approaches of coefficient of variance (CV) and standard deviation (SD) established a comparative conclusion about the strength of the S-box input.

## 9.4 Confusion Analysis of S-boxes using Truncated Differential Cryptanalysis

A new approach with Truncated Differential Cryptanalysis has been used in the next phase of the research. The approach includes:

- Elements of S-boxes have been considered as input text and the generated output has been saved as an output element.
- The input text has been divided in two part of four bit each as $I(x, y)$. Every bit of $x$ and $y$ has been altered and generated the output.
- A SAC matrix has been generated with original output text and altered output text.
- An approach used to calculate $1's$ of every column. The statistical approaches of coefficient of variance (CV) and standard deviation (SD) established a comparative conclusion about the strength of the S-box input.

## 9.5 Confusion Analysis of S-boxes using Boomerang-style Attack

The final approach for this research is the establishment of testing against the boomerang-style attack. Boomerang attack on S-boxes is one of the untouched areas in the field of cryptanalysis. It is an approach to face the boomerang-style attack on S-boxes:

- Elements of S-boxes have been considered as input text and the generated output has been saved as an output element.
- From the input text, the algorithm finds all possible pair combination by using the formula $\frac{n(n-1)}{2}$.
- The pairs has been combined to generate the next all possible input text set. Using the new sets of input the output text generated.

- A SAC matrix has been generated with original output text and output text of newly combined input texts.
- An approach used to calculate $1's$ of every column. The statistical approaches of coefficient of variance (CV) and standard deviation (SD) established a comparative conclusion about the strength of the S-box input.

## 9.6 Comparative Study of Experimental Data for DES and AES

A comparative study of all experimental data for DES and AES has been included in this section. All included data are from following proposed algorithms:

- Bit Level Confusion Analysis of S-Box (**1B**).
- A 2-Bit Approach Confusion Analysis of S-box (**2B**).
- Confusion Analysis of S-Boxes using Truncated Differential Cryptanalysis (**TD**).
- Confusion Analysis of S-Boxes using Boomerang-style Attack (**BA**).

### 9.6.1 Comparative Study of S-Boxes of DES

Comparisons of mean, variance, standard deviation and coefficient of variance for 1B, 2B, TD and BA are given in Table 9.1:

| S-box | Mean | | | | Variance | | | | Std. Deviation | | | | Covariance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1B | 2B | TD | BA | 1B | 2B | TD | BA | 1B | 2B | TD | BA | 1B | 2B | TD | BA |
| 0 | 3.71 | 3 | 1.97 | 1.95 | 1.43 | 1.64 | 1.03 | 0.96 | 1.19 | 1.28 | 1.4 | 0.98 | 0.32 | 0.42 | 0.71 | 0.5 |
| 1 | 3.79 | 3.03 | 1.83 | 1.95 | 1.03 | 1.26 | 0.99 | 1.02 | 1.01 | 1.12 | 1.35 | 1.01 | 0.26 | 0.36 | 0.73 | 0.51 |
| 2 | 3.93 | 3.01 | 1.95 | 1.94 | 1.21 | 1.14 | 0.93 | 1.1 | 1.1 | 1.06 | 1.39 | 1.05 | 0.27 | 0.35 | 0.71 | 0.53 |
| 3 | 3.68 | 2.99 | 2.04 | 1.98 | 1.02 | 1.28 | 0.97 | 1.14 | 1.01 | 1.13 | 1.43 | 1.06 | 0.27 | 0.37 | 0.69 | 0.53 |
| 4 | 3.79 | 2.98 | 2.04 | 2 | 1.22 | 1.69 | 1.09 | 1.1 | 1.1 | 1.3 | 1.41 | 1.05 | 0.29 | 0.43 | 0.7 | 0.52 |
| 5 | 3.9 | 3.07 | 1.95 | 1.99 | 1.2 | 1.38 | 0.87 | 1.06 | 1.09 | 1.17 | 1.39 | 1.03 | 0.28 | 0.38 | 0.71 | 0.51 |
| 6 | 3.93 | 3.03 | 1.98 | 1.98 | 1.41 | 1.49 | 0.94 | 0.94 | 1.19 | 1.22 | 1.4 | 0.97 | 0.3 | 0.4 | 0.71 | 0.48 |
| 7 | 3.75 | 3 | 1.96 | 2.06 | 0.95 | 1.64 | 1.04 | 1.06 | 0.97 | 1.28 | 1.4 | 1.03 | 0.26 | 0.42 | 0.71 | 0.5 |

**Table 9.1: Comparative Experimental Results for DES using Different Proposed Algorithms**

The line graph comparisons of standard deviation, variance and coefficient of variance are depicted in Figure 9.1, 9.2 and 9.3 respectively.

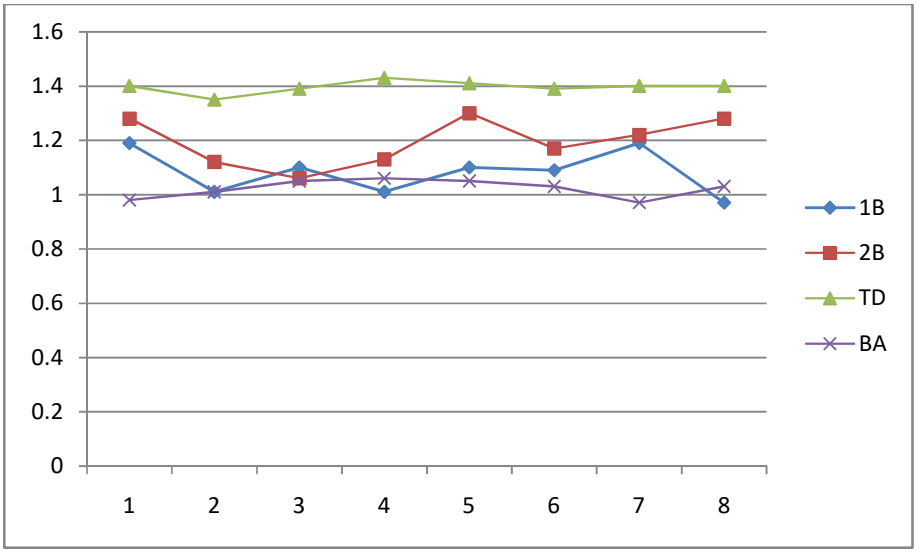


***Figure 9.1: Line Graph of Standard Deviation for Algorithms of S-Boxes of DES***

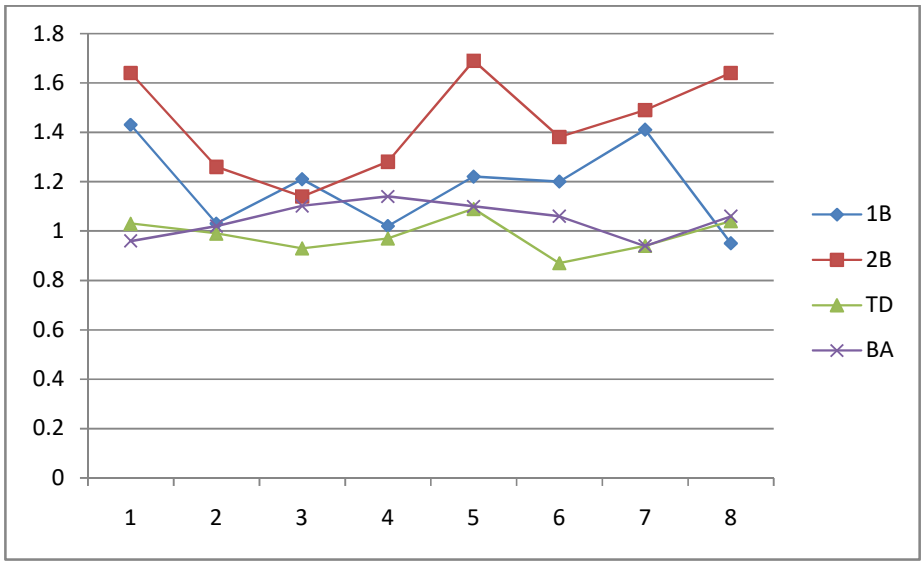


***Figure 9.2: Line Graph of Variance for Algorithms of S-Boxes of DES***

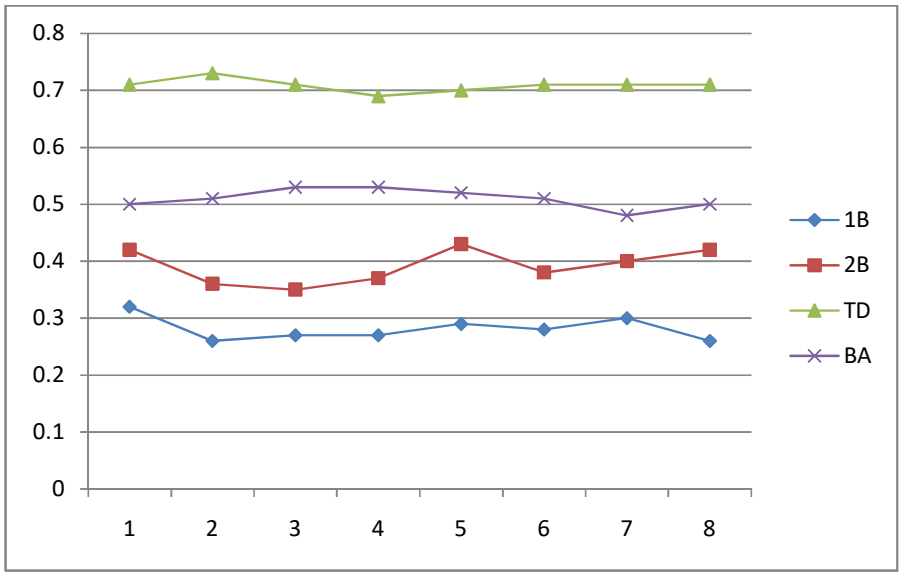


*Figure 9.3: Line Graph of Coefficient of Variance for Algorithms of S-Boxes of DES*

### 9.6.2 Comparative Study of Confusion and Diffusion in S-Box of AES

Comparisons of mean, variance, standard deviation and coefficient of variance for 1B, 2B, TD and BA are given in Table 9.2:

| Inputs | Mean | | | | Variance | | | | Std. Deviation | | | | Covariance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1B | 2B | TD | BA | 1B | 2B | TD | BA | 1B | 2B | TD | BA | 1B | 2B | TD | BA |
| $(195)_{10}$ | 4.25 | 3.5 | 1.62 | 3.25 | 0.68 | 2.5 | 0.98 | 1.18 | 0.82 | 1.5 | 0.99 | 1.08 | 0.19 | 0.42 | 0.61 | 0.33 |
| $(170)_{10}$ | 3.75 | 4 | 2.25 | 3.5 | 2.43 | 1.25 | 0.43 | 0.75 | 1.56 | 1.11 | 0.66 | 0.86 | 0.41 | 0.27 | 0.29 | 0.24 |
| $(204)_{10}$ | 4.13 | 3.62 | 2.5 | 3 | 1.6 | 1.73 | 0.25 | 0.25 | 1.26 | 1.31 | 0.5 | 1.58 | 0.3 | 0.36 | 0.2 | 0.52 |
| $(105)_{10}$ | 3.12 | 4.12 | 1.62 | 3.38 | 1.6 | 1.85 | 1.48 | 1.98 | 1.26 | 1.37 | 1.21 | 1.4 | 0.4 | 0.33 | 0.74 | 0.41 |
| $(45)_{10}$ | 3.62 | 2.87 | 1.62 | 3.37 | 3.98 | 1.85 | 0.23 | 0.48 | 1.99 | 1.36 | 0.48 | 0.69 | 0.55 | 0.47 | 0.29 | 0.2 |
| $(210)_{10}$ | 4.37 | 3.62 | 2.12 | 3.5 | 1.23 | 1.23 | 1.35 | 2.75 | 1.11 | 1.11 | 1.16 | 1.65 | 0.25 | 0.3 | 0.54 | 0.47 |

Table 9.2: Comparative Experimental Results for AES using Different Proposed Algorithms

The line graph comparisons of standard deviation, variance and coefficient of variance are depicted in Figure 9.4, 9.5 and 9.6 respectively.

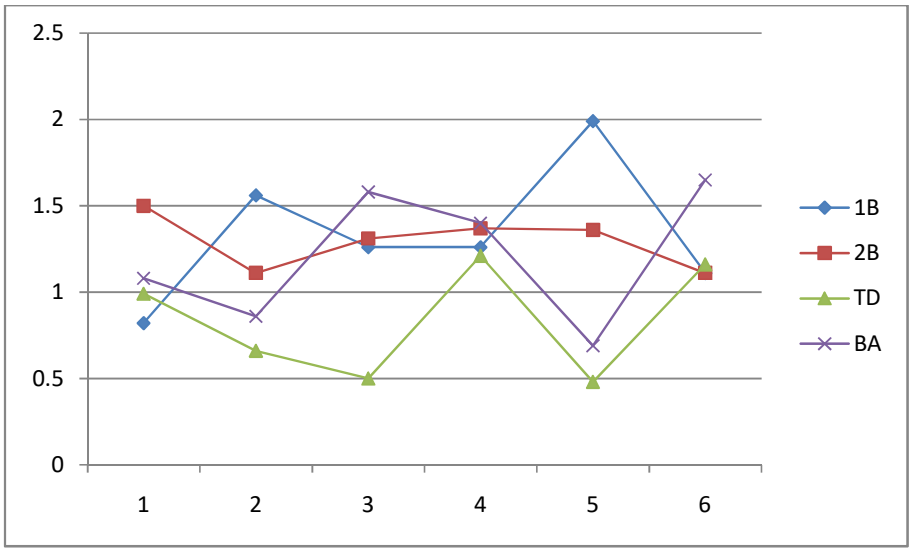


*Figure 9.4: Line Graph of Standard Deviation for Algorithms of S-Box of AES*

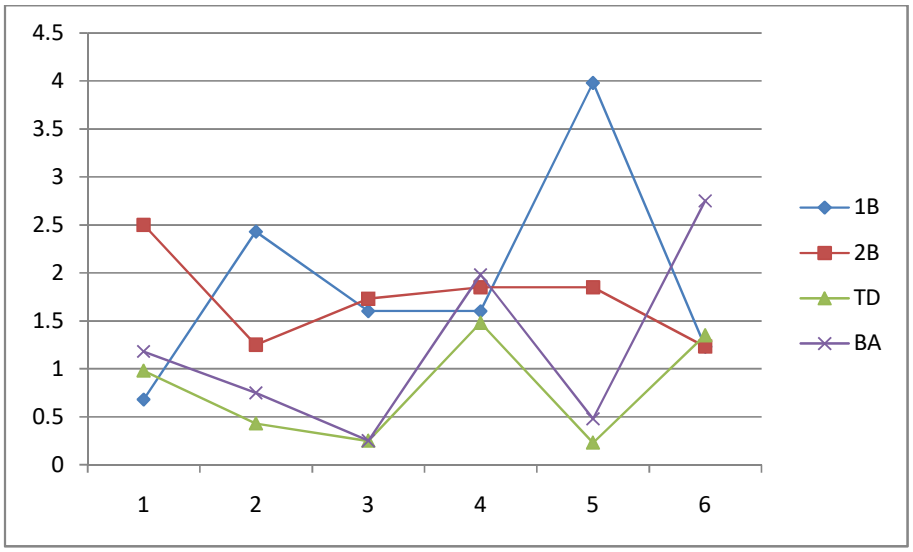


*Figure 9.4: Line Graph of Variance for Algorithms of S-Box of AES*

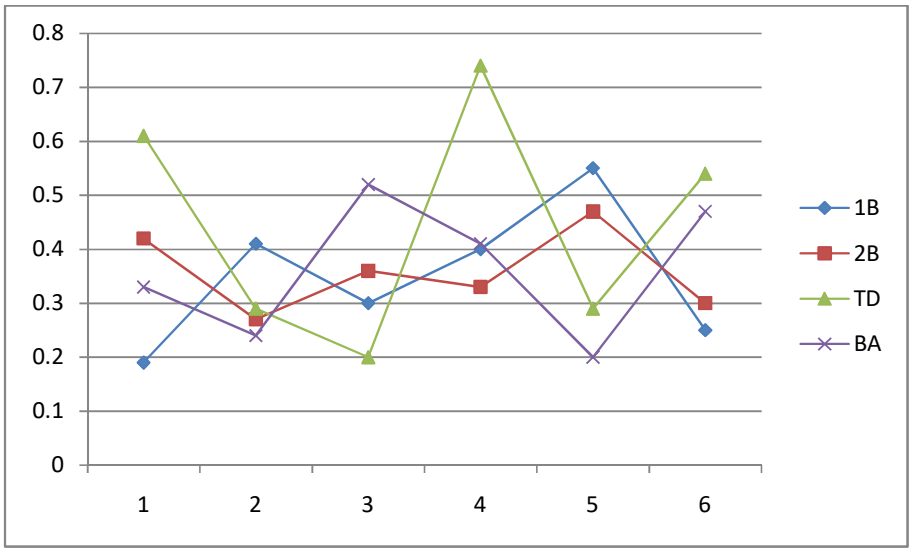


***Figure 9.6: Line Graph of Coefficient of Variance for Algorithms of S-Box of AES***

From the above comparative experimental data of DES and AES, the coefficient of variance (CV) is ranging from 0.26-0.71 and 0.2-0.55, respectively, which are in the lower end of the spectrum which indicating that performances of S-boxes of DES and AES are pretty good with respect to the proposed test. The average CV of DES and AES is 47% and 37%, respectively, which helps us to draw the conclusion that the performance of S-box of AES is better than the performance of S-boxes of DES.

# References


[1] Cristof Paar et. al., "*Understanding Cryptography A Textbook for Students and Practitioners*", Springer International Edition, Indian reprint-2012.

[2] Richard J. De Moliner, "*On the statistical Testing of Block Cipher*", A Dissertation Submitted to the Swiss Federal Institute of Technology, Zurich, 1999.

[3] Lars R. Knudsen and John E. Mathiassen, "*On the Role of Key Schedule in Attacks on Iterated Ciphers*", P. Samaratiet. al.: ESORICS 2004, LNCS 3193, 322-334, Springer-Verlag, 2004.

[4] S. Forsyth and R. Safavi-Naini, "*Automated Cryptanalysis of Substitution Ciphers*", CRYPTOLOGIA, Vol. XVII, No. 4, Oct. 1993.

[5] Gilles Piret and Francois-Xavier Standaert, "*Provable security of block ciphers against linear cryptanalysis: a mission impossible? An experimental review of practical security approach and the key equivalence hypothesis in linear cryptanalysis*", Springer, 2008, 325-338.

[6] Julio Cesar Harnandez Castro, José María Sierra et. al., "*The strict avalanche criterion randomness test*", Mathematics and Computers in Simulation 68(2005), 1-7, Elsevier.

[7] Webster, A.F. & Tavares, "*On the design of s-boxes*", S.F. 1986, Advances in Cryptology – CRYPTO '85, Springer-Verlag, 523-534.

[8] Forre, R.,The strict avalanche criterion: spectral properties of booleans functions and an extended definition. Advances in cryptology, in: S. Goldwasser (Ed.), Crypto'88, Lecture Notes in Computer Science, vol. 403, Springer-Verlag, 1990, pp. 450–468.

[9] H.M. Gustafson, E.P. Dawson and J. Dj. Golic, "*Automated Statistical Methods for Measuring the Strength of Block Ciphers*", Statistics and Computing (1997) 7, 125-135.

[10] DenizToz, Ali Doğanaksoy, MeltemSönmezTuran, "*Statistical Analysis of Block Ciphers*", In: UlusalKriptologiSempozyumu, Ankara, Turkey, pp. 56–66 (2005)

[11] SreenivasuluNagireddy, "*A Pattern Recognition Approach to Block Cipher Identification*", A Thesis for the award of the degree of Master of Science, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, Oct. 2008.

[12] Stefan Lucks, "*Attacking triple encryption*", FSE '98: Proceedings of the 5th International Workshop on Fast Software Encryption, London, UK, 1998, pp. 239-253, Springer-Verlag.

[13] Xuejia Lai, James L. Massey, "*Markov Ciphers and Differential Cryptanalysis*", Advances of Cryptology- EUROCRYPT '91, LNCS 547, Springer-Verlag, 1991.

[14] Doganaksoy A., Ege B., Kocak O., "*Cryptographic Randomness Testing of Block Cipher and Hash Function*", 2010., IACR Cryptology ePrint Archive Report 2010/564.

[15] Marsaglia, G., "*Diehard battery of test of randomness*", 1996.

[16] Marsaglia, G., Wai Wan, T., "*Some difficult-to-pass test of randomness*", Research supported by Innovation and Technology Support Programme, Government of Hong Kong, Grant ITS/277/0, 2002.

[17] Rukhin, A., Soto, J. et. al., "*A Statistical Test Suite for Random and Pseudo-random Number Generators for Cryptographic Applications*", NIST Special Publication. 800-22.

[18] Soto, J., Bassham, L.: Randomness Testing of the Advanced Encryption Standard Finalist Candidates. In: Computer Security Division. National Institute of Standards and Technology (2000)

[19] Katos,V, 2005. "A Randomness Test for Block Ciphers", Applied Mathematics and Computation, Elsevier Publication, 162(2005), 29-35.

[20] "Chi Square Tests", Chapter 10, http://uregina.ca/~gingrich/ch10.pdf.

[21] "Chi-Square: Testing for Goodness of Fit", www.physics.ucsc.edu/~drip/133/ch4.pdf.

[22] W. Stallings, "*Cryptography and Network Security-Principles and Practices*", 4th ed., Prentice-Hall of India Pvt. Ltd. 2008.

[23] Shannon, C. E. 1949. Communication theory of secrecy systems. Bell System Technical Journal 28-4, pp. 656–715.

[24] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, IEEE Transactions on Information Theory IT-30 (1984), no. 5, 776-780.

[25] Bart Preneel, Cryptographic Properties of Boolean Functions and S-Boxes, KatholiekeUniversiteit Leuven – Faculteit Toegepaste Wetenschappen Arenbergkasteel, B-3001 Heverlee (Belgium), ISBN 90-5682-649-2

[26] Heys, H.M., 2002. A tutorial on linear and differential cryptanalysis. *Cryptologia*, *26*(3), pp.189-221.

[27] H.Feistel, "Cryptography and Computer Privacy", Scientific American, Vol.228, No.5, pp 15-23, 1973.

[28] B.Kam, G.I. Davida, "Structured Design of Substitution-Permutation Encryption Network", IEEE Trans. on Compute. Vol.C-28, No.10, pp.747-753, Oct., 1979.
[29] http://www.tuicool.com/articles/Ub6rui

[30] Coppersmith, D, 1994, “The Data Encryption Standard and its Strength Against Attacks”, IBM Journal of Research and Development, 38(3) 243

[31] Daemen, J, and Rijmen, V (March 9, 2003). "AES Proposal: Rijndael" (PDF). National Institute of Standards and Technology. p. 1.

[32] G. Piret and F.X. Standaert, ”Provable security of block ciphers against linear cryptanalysis: a mission impossible? An experimental review of practical security approach and the key equivalence hypothesis in linear cryptanalysis”, Springer, 2008, pp. 325-338.

[33] PhyuPhyu Mar, KhinMaungLatt, ”New Analysis Methods on Strict Avalanche Criterion of S-boxes”, World Academy of Science, Engineering and Technology 24, 2008, pp. 150-154.

[34] IgilVERGL, Melek D. YCEL, ”Avalanche and Bit Independence Properties for the Ensembles of Randomly Chosen nn S-boxes”, Truk J ElecEngin, VOL. 9, NO. 2, 2001, pp. 137-145.

[35] Shannon, C.E., ”A mathematical theory of communication”, Bell System Technical Journal 27, 1948. pp. 379423.

[36] Ramamoorthy, V., et al., ”The Design of Cryptographic S-boxes Using CPSs. J. Lee (Ed.): CP 2011”, LNCS 6876, Springer-Verlag Berlin Heidelberg, 2013, pp. 54-68.

[37] Mukhopadhyay, D, ”Overview on S-box Design Principles. Crypto and Network Security”. Lecture Note. IIT Kharagpur.

[38] Phillip Rogaway, ”Confusion/Diffusion Primitives”, ECS 227: Modern Cryptography Lecture 2, 1996.

[39] Feistel, H., ”Cryptography and Computer Privacy”, Scientific American, Vol. 228, No. 5, 1973, pp. 15-23.

[40] Feistel, H., Notz, W.A. and Smith, J.L, ”Some Cryptographic Techniques for Machine to Machine Data Communications”, Proc. IEEE, Vol. 63, No. 11, 1975,pp. 1545-1554.

[41] Lecture Note: Attacks on Cryptosystems, Chapter 13, http://www.facweb.iitkgp.ernet.in/~sourav/Attacks_on_cryptosystems.pdf

[42] Wagner D. (1999) The Boomerang Attack. In: Knudsen L. (eds) Fast Software Encryption. FSE 1999. Lecture Notes in Computer Science, vol 1636. Springer, Berlin, Heidelberg

[43] Lecture Note: The Data Encryption Standard, Chapter 2, http://www.facweb.iitkgp.ernet.in/~sourav/DES.pdf

[44] Biham, Eli, and Adi Shamir. *Differential cryptanalysis of the data encryption standard*. Springer Science & Business Media, 2012.

[45] Lecture Note: The Advances Encryption Standard, Chapter 2, http://www.facweb.iitkgp.ernet.in/~sourav/AES.pdf

[46] Mister, Serge, and Carlisle Adams. "Practical S-box design." Workshop on Selected Areas in Cryptography, SAC. Vol. 96. 1996.

[47] Hosseinkhani and Javadi. "Using Cipher Key to Generate Dynamic S-Box in AES Cipher Syst.", International Journal of Computer Science and Security (IJCSS), Volume (6): Issue (1): 2012.

[48] Kamsiah Mohamed, Mohd Nazran Mohammed Pauzi, Fakariah Hani Hj Mohd Ali, Suriyani Ariffin, Nurul Huda Nik Zulkipli. "Study of S-box Properties in Block Cipher", IEEE 2014 International Conference on Computer, Communication, and Control Technology (I4CT 2014), September 2 -4, 2014 - Langkawi, Kedah, Malaysia.

[49] Knudsen, Lars R. "Truncated and higher order differentials." International Workshop on Fast Software Encryption. Springer, Berlin, Heidelberg, 1994.

[50] X. Lal. Higher order derivatives and differential cryptanalysis. In Proc. "Symposium on Communication, Coding and Cryptography", in honor o] James L. Massey on the occasion of his 60'th birthday, Feb. 10-13, 1994, Monte-Verita, Ascona, Switzerland, 1994.

[51] E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1):3-72, 1991.

[52] K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, Advances in Cryptology- Proc. Eurocrypt'93, LNCS 765, pages 55-64. Springer Verlag, 1993.

[53] K. Nyberg and L.R. Knudsen. Provable security against differential cryptanalysis. In E.F. Brickell, editor, Advances in Cryptology - Proc. Crypto'92, LNCS 740, pages 566-574. Springer Verlag, 1993.

[54] Nyberg, Kaisa. "Perfect nonlinear S-boxes." Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1991.
[55] Moriai S., Sugita M., Aoki K., Kanda M. (2000) Security of E2 against Truncated Differential Cryptanalysis. In: Heys H., Adams C. (eds) Selected Areas in

Cryptography. SAC 1999. Lecture Notes in Computer Science, vol 1758. Springer, Berlin, Heidelberg.

[56] Rasoolzadeh, Shahram, et al. "An improved truncated differential cryptanalysis of KLEIN." Tatra Mountains Mathematical Publications 67.1 (2016): 135-147.

[57] Lee, Seonhee, et al. "Truncated differential cryptanalysis of Camellia." International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2001.

[58] Shannon, C.E. A mathematical theory of communication. Bell System Technical Journal 27, 1948. p. 379–423, 623–656.

[59] A.Datta, D.Bhowmick, S. Sinha, A Novel Technique for Analysing Confusion in S-boxes. International Journal of Innovative Research in Computer and Communication Engineering, 2016. 4(6): p. 11608-11615.

[60] A.Datta, D.Bhowmick, S. Sinha, Implementation of SAC Test for Analyzing Confusion in an S-box Using a Novel Technique. International Journal of Scientific Research in Computer Science Applications and Management Studies, Vol. 7, Issue 3, No. 182.

[61] Webster, A.F., Tavares, S.E. On the Design of S-boxes. Advance in Cryptology. Proc. CRYPTO ’85, Springer-Verlag, Berlin, 1986. pp. 523-534.

[62] D.Bhowmick, A.Datta, S. Sinha. A Bit-Level Block Cipher Diffusion Analysis Test. Springer International Publishing Switzerland 2015: S.C.Satpathy et. al. (eds), Proc of 3rd Int. Conf. on Front. of Intell. Comput. (FICTA) 2014-Col. I, Advances in Intelligent Systems and Computing 327. pp: 667-674.

[63] Wagner D. (1999) The Boomerang Attack. In: Knudsen L. (eds) Fast Software Encryption. FSE 1999. Lecture Notes in Computer Science, vol 1636. Springer, Berlin, Heidelberg

[64] Cid C., Huang T., Peyrin T., Sasaki Y., Song L. (2018) Boomerang Connectivity Table: A New Cryptanalysis Tool. In: Nielsen J., Rijmen V. (eds) Advances in Cryptology – EUROCRYPT 2018. EUROCRYPT 2018. Lecture Notes in Computer Science, vol 10821. Springer, Cham.

[65] Choy J., Yap H. (2009) Impossible Boomerang Attack for Block Cipher Structures. In: Takagi T., Mambo M. (eds) Advances in Information and Computer Security. IWSEC 2009. Lecture Notes in Computer Science, vol 5824. Springer, Berlin, Heidelberg.

[66] Kim, Jongsung & Moon, Dukjae & Lee, Wonil & Hong, Seokhie & Lee, Sangjin & Jung, Seokwon. (2002). Amplified boomerang attack against reduced-round

SHACAL. SIACRYPT 2002, LNCS 2501, pp. 243–253, 2002.c©Springer-Verlag Berlin Heidelberg 2002.doi: 10.1007/3-540-36178-2_15.

[67] H. Handschuh, D. Naccache, SHACAL, NESSIE project, October 2001.

[68] Koo, Bonwook & Yeom, Yongjin & Song, Junghwan. (2010). Related-Key Boomerang Attack on Block Cipher SQUARE. IACR Cryptology ePrint Archive. 2010. 73. 10.1587/transfun.E94.A.3.

[69] Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1-18.Springer, Heidelberg (2009)

[70] Avijit Datta, Dipanjan Bhowmik, Sharad Sinha, "A New Approach towards Confusion Analysis of S-boxes using Truncated Differential Cryptanalysis", International Journal of Computer Sciences and Engineering, Vol.7, Issue.1, pp.249-256, 2019.

[71] Cheung, Jennifer Miuling. "The design of S-boxes." PhD diss., Sciences, 2010.

[72] C. Adams and S. Tavares. The structured design of cryptographically good s-boxes. Journal of Cryptology, 3(1):27–41, 1990.

[73] J. Cobas and J. Brugos. Complexity-theoretical approaches to the design and analysis of cryptographical boolean functions. In Computer Aided Systems Theory–EUROCAST2005, Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 2005.

[74] A Novel Technique for Analysing Confusion in S-Boxes, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 6, June 2016, ISSN(Online): 2320-9801, ISSN (Print) : 2320-9798, DOI: 10.15680/IJIRCCE.2016. 0406253

# Appendix 1: List of Publications

## Communicated Papers

| Sl. | Title of Research Paper | Journal Detail | Remarks |
|---|---|---|---|
| 1. | **A New Approach for Measuring SKAC using Bit Relationship Test** | Design, Codes and Cryptography, Springer US, ISSN: 15737586 | UGC Journal No.: 12931 |

## Published Papers

| Sl. | Title of Research Paper | Journal Detail | Remarks |
|---|---|---|---|
| 9. | **A Novel Technique for SAC Analysis of S-Box for Boomerang-Style Attacks** | International Journal of Computer Sciences and Engineering, Volume-7, Issue-5 E-ISSN: 2347-2693 | UGC Journal No.: 63193 |
| 8. | **A New Approach towards Confusion Analysis of S-boxes using Truncated Differential Cryptanalysis** | International Journal of Computer Sciences and Engineering, Volume-7, Issue-1 E-ISSN: 2347-2693 | UGC Journal No.: 63193 |
| 7. | **Implementation of SAC Test for Analyzing Confusion in an S-BOX Using a Novel Technique** | International Journal of Scientific Research in Computer Science Application and Management Studies, Volume 7, Issue 3, ISSN 2319-1953, Pg: 182, May, 2018 | UGC Journal No.: 63611 |
| 6. | **A Novel Scheme for** | Springer Science+Business | |

| | | | |
|---|---|---|---|
| | **Analyzing Confusion Characteristics of Block Ciphers** | Media Singapore 2017, Advances in Intelligent Systems and Computing 458, DOI 10.1007/978-981-10-2035-3_64 | |
| 5. | **A New Perspective of Inferring from the output of Linear Cryptanalysis Attack** | International Journal of Computer Sciences and Engineering Volume-5, Issue-2 E-ISSN: 2347-2693 | UGC Journal No.: 63193 |
| 4. | **An Approach towards Analyzing Strict Key Avalanche Criterion of Block Ciphers** | International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2016 ISSN(Online): 2320-9801 ISSN (Print) : 2320-9798 DOI: 10.15680/IJIRCCE.2016. 0406258 | |
| 3. | **A Novel Technique for Analysing Confusion in S-Boxes** | International Journal of Innovative Research in Computer and Communication Engineering Vol. 4, Issue 6, June 2016 ISSN(Online): 2320-9801 ISSN (Print) : 2320-9798 DOI: 10.15680/IJIRCCE.2016. 0406253 | |
| 2. | **Measuring the Diffusion Characteristic of Block Ciphers: The Bit Relationship Test (BRT)** | International Journal of Computer Sciences and Engineering Volume-3, Special Issue-1 E-ISSN: 2347-2693 | UGC Journal No.: 63193 |
| 1. | **A Bit-Level Block Cipher Diffusion Analysis Test – BLDAT** | Springer International Publishing Switzerland 2015, *Vol. 1*, Advances in Intelligent Systems and Computing 327, DOI: 10.1007/978-3-319-11933- | |

| | | | |
|---|---|---|---|
| | | 5_75 | |

# Appendix 2: One of the Published Paper

IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

## A Novel Technique for SAC Analysis of S-Boxes for Boomerang-Style Attacks

**Avijit Datta[1*], Dipanjan Bhowmik[2], Sharad Sinha[3]**

[1,2,3] University of North Bengal, West Bengal, India

*Corresponding Author: avijit.go2avi@gmail.com, Tel.: +91-9775802114

DOI: https://doi.org/10.26438/ijcse/v7i5.713 | Available online at: www.ijcseonline.org

Accepted: 10/May/2019, Published: 31/May/2019

***Abstract*—** In recent times, there exist several approaches for differential-style attacks like truncated differential attack, high-level differential attack, boomerang attack etc. This paper involves the study of boomerang-style attack on S-boxes and a new SAC analysis approach to test the strength of S-boxes against such attacks. The proposed analysis is tested on each input elements of 8 S-boxes of DES and 8 input elements on the S-box of AES. The vulnerability factor $^{n}/_{2}$ has been measured by calculating all $\mathbf{1's}$ of every column from the generated SAC matrix. Finally a comparison of standard deviation, coefficient of variance and other factors show the way towards the conclusion.

***Keywords*—**Block Cipher, S-box, Differential Cryptanalysis, Boomerang attack, Truncated Differential

### I. INTRODUCTION

Differential cryptanalysis is one of the most important cryptanalytic techniques in cryptology. Published cipher may be broken by using differential cryptanalysis. So, one of the most important responsibilities of the block cipher designer is to ensure protection and security against such cryptanalysis.

The upper bound probability of any differential characteristics is tentatively $p$ and the designer of the algorithm presumes, following the "folk theorem", that differential attack requires at least $^{1}/_{p}$ texts to break the cipher. [1] But unfortunately the "folk theorem" is not always right, as there is a type of differential attack, called Boomerang attack that can allow an adversary to beat the $^{1}/_{p}$ bound in some cases.

In this paper, the possibility of boomerang type attack on S-boxes of cryptographic algorithms has been measured. To represent the approach, the boomerang approach has been implemented to generate a SAC matrix for all possible inputs of all S-boxes of DES and all possible input of the S-box of AES. The generated SAC matrix is leads towards the statistical analysis of vulnerability of attack.

The Section I contains the introduction of the work, section II contains related works of the proposed technique, section III contains discussion on differential cryptanalysis and boomerang attack, section IV defines the design of S-boxes, section V is about proposed method, section VI contains brief experimental results, and finally, sections VII and VIII include the discussion and conclusion respectively.

### II. RELATED WORK

D. Wagner, in his research [1] introduced an attack called Boomerang attack. This attack leads to prove that the so called folk theorem is not always right according to which, differential attack requires at least $^{1}/_{p}$ text to break a cipher where, $p$ is the upper bound probability of any differential characteristics. According to his research, if the best characteristic for half of the rounds of the cipher has probability $q$, then the boomerang attack can be used successfully on $O(q^{-4})$ chosen text. In some cases, it may be possible that $q^{-4} \ll p^{-1}$, wherein boomerang attack allows one to beat the folk theorem bound. Sometimes, Boomerang attack also uses some extensive structures that are available in conventional differential attack.

Boomerang connectivity table is a good cryptanalysis tool [2]. In this research paper, it is revisited with the issue of dependency of two characteristics in a block cipher $E_m$ and proposed a new tool called Boomerang Connectivity Tool (BCT), which evaluates $r$ in a systematic and easy-to-understand way when $E_m$ is composed of a single S-box layer. BCT shows that the probability around the boundary may be even higher than $p$ or $q$.

 

A variant of differential cryptanalysis against the block cipher, Impossible Boomerang Attack (IBA) was introduced by Choi et. al. [3]. This research approach is the combination of differential cryptanalysis and boomerang attack.

Amplified Boomerang attack has been introduced in a research [4] dedicated on SHACAL, which is a 4-round block cipher [6]. SHACAL was designed by using hash standard SHA-1in encryption mode for the first time. Kim et. al. proposed a 10-step differential characteristic with probability $2^{-12}$ in rounds 2 and 4. Using this characteristic, they described a 36-step boomerang distinguisher. With this distinguisher they devised amplified boomerang attacks on reduced round SHACAL with different size keys.

A key recovery attack on the full round of SQUARE using a related key boomerang distinguisher was proposed in [5]. They constructed a 7-round related key boomerang distinguisher with probability $2^{-119}$ by finding local collision[7], and calculated its probability using ladder switch and local amplification techniques.

Analysis of vulnerability factor for truncated differential using SAC has been proposed in [8]. This research approach describes and analyzes the truncated differential on S-boxes of DES and AES.

## III. Differential Cryptanalysis and Boomerang Attack

### A. Differential Cryptanalysis

The method which analyzes the effect of particular differences in plaintext pairs on the differences of the ciphertext pairs is call differential cryptanalysis [9]. Differential cryptanalysis is a chosen-plaintext attack on secret-key block ciphers that are based on iterating a cryptographically weak function $r$ times. The success of attacks on $r$ round cipher depends on the existence of $(r-1)$ -round differentials with higher probability [10]. The "difference" $\Delta X$ between two plaintexts (or ciphertexts) $X$ and $X^*$ can be defined as $\Delta X = X \oplus X^{*-1}$, where $\oplus$ denotes a specified group of operation on the set of plaintexts.

Differential cryptanalysis exploits the fact that the round function $f$ in an iterated cipher is usually cryptographically weak [10]. In a differential cryptanalysis, all the sub-keys are fixed and only the plaintext can be chosen randomly. In differential cryptanalysis attack, differential probability helps us to determine which differential to use in the attack.

### B. Boomerang Attack

Boomerang attack was introduced by David Wagner in [1] where he proved the 'folk theorem' of differential cryptanalysis as wrong. Boomerang attack uses the technique of truncated differential. So, if a cipher is safe from boomerang attack, then it also proves good against truncated differentials. The boomerang attack is a differential attack that attempts to generate a quartet structure at an intermediate value halfway through the cipher.

The attack considers four plaintexts $A, A', B, B'$ along with their cipher-texts $C, C', D, D'$. If $E(\cdot)$ is the encryption operation and divide the cipher into $E = E_1 \circ E_0$ where $E_0$ is the first half of the cipher and $E_1$ is the last half then differential characteristics are $\Delta \rightarrow \Delta^*$ for $E_0$ and $\nabla \rightarrow \nabla^*$ for $E_1^{-1}$. The boomerang attack is given in *fig. 1*.

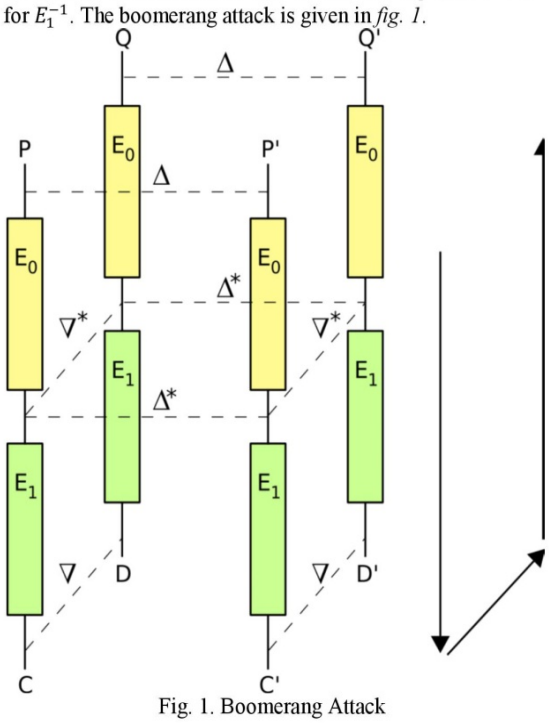


Fig. 1. Boomerang Attack

## IV. Design Aspect of S-box

Substitution boxes (S-boxes) are the only non-linear part of a SP network in a cryptosystem. S-boxes are composed with highly non-linear Boolean functions. SP network is a private key cryptosystem. There are two components of SP network as $\pi_s$ and $\pi_p$. Each permutation $\pi_s$ is called S-box [11]. It replaces a set of input bits with a different set of bits as its output. The following criteria must be there in Boolean function that are responsible for a cryptographically good S-box [12] [13]:

- Bijection requires a 1-to-1 and onto mapping from input vectors to output vectors in the S-box of size $n \times n$ bits.

 

- Strict Avalanche Criterion (SAC) occurs if a change in one input bit $i$ causes each output bit to change with probability of one half.
- Bit independence criterion or correlation-immunity.
- Non-linearity of S-box from input to output.
- Balance of Boolean vectors that are responsible for S-boxes that have the same number of $0's$ and $1's$.

Table 1. Partial Truth Table of S-Box 1in DES system

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_5$ | $x_6$ | $y_1$ | $y_2$ | $y_3$ | $y_4$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |

For the construction of S-box, the function $f: \{0,1\}^n \rightarrow \{0,1\}^m$, with a high non-linearity, there are $2^n$ rows with $m$ columns

## V. PROPOSED METHOD

This research work was started with the confusion analysis of S-boxes. There are several previous research works in [8], [14], [15] where confusion of S-boxes has been analyzed with differential cryptanalysis. In this research, most importantly, confusion of S-boxes has been analyzed against the boomerang type attack.

In [8], the truncated differential cryptanalysis approach has been implemented on S-boxes of DES and S-box of AES with a new proposal of the statistical analysis on the generated SAC. The proposed method has been compared with the conclusion of 2-bit approach [15] of confusion analysis of S-box.

In [14], all elements of DES S-boxes and AES S-box take inputs individually and the SAC matrix is generated from the original ciphertext along with ciphertexts generated from the every one alternative bit alteration of the original input. In the generated SAC matrix, the vulnerability of every bit of the ciphertext has been statistically computed and discussed.
In the other approach [15], the SAC matrix was generated with the original ciphertext and ciphertexts obtained from every two alternative bit alteration of original input. The method has been used for all 8 S-boxes of DES and the S-box of AES.

In this work, the boomerang-style attack has been implemented on S-boxes of DES and AES and SAC matrices have been generated, which are then subjected to the proposed statistical analysis.

The proposed analysis in this paper involves the following:

1) Analysis of frequency of every bit column-wise and its various avalanche effects from the generated SAC.
2) Coefficient variance analysis of generated SAC.
3) Analysis of frequency of various differential values from the generated SAC.

Using the V-vector (Vulnerability Vector) [16,17], the proposed algorithm is as below:

### *A. Proposed Algorithm: Confusion Analysis of S-boxes using Boomerang-style Attack*

**Input**: Elements of S-box with length $n$, where $n$ is the number of bits.

**Step 1**: Choose any random number $P$ within the range of the S-box, where $P \in \mathbb{Z}_2^n$. Find the corresponding output value of S-box: $C = S(P)$

**Step 2**: Change the $P_i$s to find their corresponding output values $C_i$s.

$P_i s$ may be generated as follows:

- Generate all possible pairs of input bits by using:
$$\frac{n(n-1)}{2}$$
- Make the number of pairs multiple of the input bit size by padding $0's$ on the MSB side.
- Change every even bit of every pair.
- Combine the pairs to generate $P_i$.

**Step 3**: Construct the SAC matrix by including $S_i = C_i \oplus C$ in the $i^{th}$ row of a matrix of size $m \times n$, where $m$ is the number bits of $P$ and $n$ is the number of bits of $C$.

**Step 4**: Find the count of $1s$ in each column of generated SAC matrix.

### *B. S-box structure of DES*

The structure of S-box of DES is given in Fig. 2. The number of S-boxes in DES is 8 with the structure of $4 \times 16$ and values ranging from $(0)_{10}$ to $(15)_{10}$. For any 6-bit input, within the ranging value, each S-box of DES generates 4 bits of output.

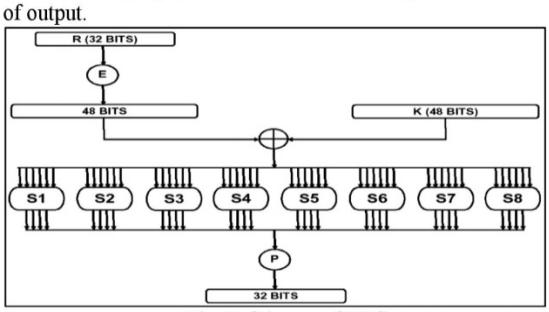


Fig. 2: S-boxes of DES

 

$S = matrix\ 4 \times 16, values\ from\ 0\ to\ 15$
$B\ (6\ bit\ input) = \boldsymbol{b_1}b_2b_3b_4b_5\boldsymbol{b_6}$
$\boldsymbol{b_1b_6} \rightarrow r = row\ of\ the\ matrix\ (2\ bits: 00,01,10,11)$
$b_2b_3b_4b_5 \rightarrow c$
$= column\ of\ matrix\ (0,1,\ldots,15)$
$C\ (output) = Binary\ representation\ S(r,c)$

*C. S-box structure of AES*

The structure of S-box of AES is shown in Fig. 3. There is a single non-linear S-box in AES with matrix structure 16 × 16.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 63 | 7c | 77 | 7b | f2 | 6b | 6f | c5 | 30 | 01 | 67 | 2b | fe | d7 | ab | 76 |
| 10 | ca | 82 | c9 | 7d | fa | 59 | 47 | f0 | ad | d4 | a2 | af | 9c | a4 | 72 | c0 |
| 20 | b7 | fd | 93 | 26 | 36 | 3f | f7 | cc | 34 | a5 | e5 | f1 | 71 | d8 | 31 | 15 |
| 30 | 04 | c7 | 23 | c3 | 18 | 96 | 05 | 9a | 07 | 12 | 80 | e2 | eb | 27 | b2 | 75 |
| 40 | 09 | 83 | 2c | 1a | 1b | 6e | 5a | a0 | 52 | 3b | d6 | b3 | 29 | e3 | 2f | 84 |
| 50 | 53 | d1 | 00 | ed | 20 | fc | b1 | 5b | 6a | cb | be | 39 | 4a | 4c | 58 | cf |
| 60 | d0 | ef | aa | fb | 43 | 4d | 33 | 85 | 45 | f9 | 02 | 7f | 50 | 3c | 9f | a8 |
| 70 | 51 | a3 | 40 | 8f | 92 | 9d | 38 | f5 | bc | b6 | da | 21 | 10 | ff | f3 | d2 |
| 80 | cd | 0c | 13 | ec | 5f | 97 | 44 | 17 | c4 | a7 | 7e | 3d | 64 | 5d | 19 | 73 |
| 90 | 60 | 81 | 4f | dc | 22 | 2a | 90 | 88 | 46 | ee | b8 | 14 | de | 5e | 0b | db |
| a0 | e0 | 32 | 3a | 0a | 49 | 06 | 24 | 5c | c2 | d3 | ac | 62 | 91 | 95 | e4 | 79 |
| b0 | e7 | c8 | 37 | 6d | 8d | d5 | 4e | a9 | 6c | 56 | f4 | ea | 65 | 7a | ae | 08 |
| c0 | ba | 78 | 25 | 2e | 1c | a6 | b4 | c6 | e8 | dd | 74 | 1f | 4b | bd | 8b | 8a |
| d0 | 70 | 3e | b5 | 66 | 48 | 03 | f6 | 0e | 61 | 35 | 57 | b9 | 86 | c1 | 1d | 9e |
| e0 | e1 | f8 | 98 | 11 | 69 | d9 | 8e | 94 | 9b | 1e | 87 | e9 | ce | 55 | 28 | df |
| f0 | 8c | a1 | 89 | 0d | bf | e6 | 42 | 68 | 41 | 99 | 2d | 0f | b0 | 54 | bb | 16 |

Fig. 3: S-boxes of AES

$S = matrix\ 16 \times 16, in\ octal\ value\ 0\ to\ F$
$B\ (8\ bit\ input) = b_1b_2b_3b_4\boldsymbol{b_5b_6b_7b_8}$
$b_1b_2b_3b_4 \rightarrow r = row\ of\ the\ matrix\ for\ output$
$\boldsymbol{b_5b_6b_7b_8} \rightarrow c = column\ of\ the\ matrix\ for\ output$
$C\ (8\ bit\ output) = octal\ value\ S(r,c)$

## VI. EXPERIMENTAL RESULTS

*A. Coefficient Variance (CV) Analysis of Generated SAC of S-boxes of DES using Boomerang-style Attack Method*

By following the proposed algorithm, using boomerang-style plaintext, a SAC matrix has been generated for every possible input for every S-box of DES. The $1's$ of every column output of S-boxes have been counted for all 64 possible inputs and corresponding outputs of the 8 S-boxes and identified as V-vector (**V**ulnerability Vector). Some of the generated SAC matrices has been given in Table 1.0 and 2.0 and are compared with tables 1.1, 2.1 and 1.2, 2.2 which are generated from truncated [8] and 2-bit [15] alteration methods. The line graph of frequencies of V-vector for all 8 S-boxes of DES is showed in Fig. 4.

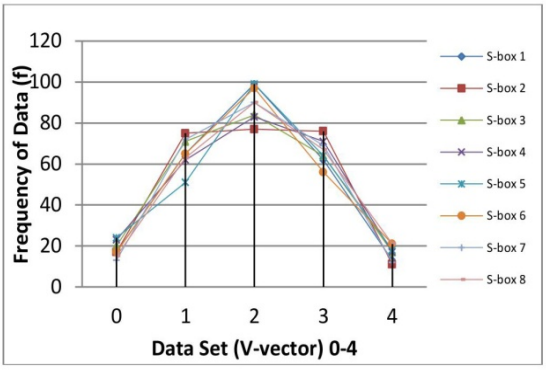


Fig. 4: Line Graph of Frequencies of V-vector for S-boxes of DES

TABLE 1.0 – SAC MATRIX OF INPUT '0' TO 'S-BOX 0' USING BOOMERANG-STYLE APPROACH

| **SAC Matrix of Input - 0** | | | |
|---|---|---|---|
| 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| **V-vector of Input - 0** | | | |
| 2 | 0 | 3 | 2 |

TABLE 1.1 – SAC MATRIX OF INPUT '0' TO 'S-BOX 0' USING TRUNCATED DIFFERENTIAL APPROACH

| **SAC Matrix of Input - 0** | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 |
| **V-vector of Input - 0** | | | |
| 3 | 3 | 1 | 3 |

TABLE 1.2 – SAC MATRIX OF INPUT '0' TO 'S-BOX 0' USING 2-BIT ALTERATION APPROACH

| SAC Matrix of Input - 0 | | | |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| V-vector of Input - 0 | | | |
| 6 | 2 | 6 | 0 |

TABLE 2.0 – SAC MATRIX OF INPUT '0' TO 'S-BOX 1' USING BOOMERANG-STYLE APPROACH

| **SAC Matrix of Input - 0** | | | |
|---|---|---|---|
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 |

 

| V-vector of Input - 0 | | | |
|---|---|---|---|
| 3 | 3 | 3 | 1 |

TABLE 2.1 – SAC MATRIX OF INPUT '0' TO 'S-BOX 1' USING TRUNCATED DIFFERENTIAL APPROACH

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |
| **V-vector of Input – 0** | | | |
| 3 | 3 | 2 | 1 |

TABLE 2.2 – SAC MATRIX OF INPUT '0' TO 'S-BOX 1' USING 2-BIT ALTERATION APPROACH

| SAC Matrix of Input – 0 | | | |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 |
| V-vector of Input – 0 | | | |
| 6 | 6 | 5 | 1 |

### B. *Experimental Results for DES S-Boxes*

The experimental results of proposed test are in Table 3:

TABLE 3 – EXPERIMENTAL RESULTS OF PROPOSED TEST ON S-BOXES OF DES

| S-box | Observed Mean | Variance | Standard Deviation | Coefficient of Variance |
|---|---|---|---|---|
| 1 | 1.957031 | 0.962997 | 0.981324 | 0.501435 |
| 2 | 1.957031 | 1.025497 | 1.012668 | 0.517451 |
| 3 | 1.949219 | 1.10289 | 1.050186 | 0.538773 |
| 4 | 1.988281 | 1.144394 | 1.069736 | 0.538034 |
| 5 | 2.003906 | 1.105453 | 1.051405 | 0.524678 |
| 6 | 1.996094 | 1.066391 | 1.032662 | 0.517341 |
| 7 | 1.980469 | 0.941025 | 0.970064 | 0.489816 |
| 8 | 2.0625 | 1.066406 | 1.032669 | 0.500688 |

### C. *Coefficient Variance (CV) Analysis of Generated SAC of S-box of AES using Boomerang-Style Attack Method*

By following the proposed algorithm, using boomerang-style plaintext, a SAC matrix has been generated for S-box of AES. 1*s* of every column output of S-box has been counted and the sum of 1*s* of every column is being identified as the V-vector (**V**ulnerability Vector). The V-vectors have been calculated for some example inputs and corresponding outputs of the S-box. Some of the generated SAC matrices are given in Table 4.0 and 5.0 and are compared with tables 4.1, 5.1 and 4.2, 5.2 which are generated from truncated [8] and 2-bit [15] alteration method. The line graph of frequencies of V-vector for all inputs using S-box of AES is showed in Fig. 5.

TABLE 4.0 – SAC MATRIX OF INPUT 11000011TO AES S-BOX USING BOOMERANG-STYLE APPROACH

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| **V-vector of input 11000011** | | | | | | | |
| 3 | 3 | 4 | 3 | 4 | 5 | 1 | 3 |

TABLE 4.1– SAC MATRIX OF INPUT 11000011TO AES S-BOX USING TRUNCATED DIFFERENTIAL APPROACH

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **V-vector of input 11000011** | | | | | | | |
| 1 | 3 | 1 | 0 | 3 | 2 | 2 | 1 |

TABLE 4.2 – SAC MATRIX OF INPUT 11000011TO AES S-BOX USING 2-BIT ALTERATION APPROACH

| **Input : 11000011** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 00101110** | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| **V-vector of input 11000011** | | | | | | | |
| 3 | 6 | 3 | 4 | 1 | 5 | 2 | 4 |

TABLE 5.0 – SAC MATRIX OF INPUT 10101010 TO AES S-BOX USING BOOMERANG-STYLE APPROACH

| **Input : 10101010** | | | | | | | |
|---|---|---|---|---|---|---|---|
| **Original Output : 10101100** | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |

 

| V-vector of input 10101010 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 4 | 3 | 3 | 4 | 4 | 5 | 2 | 3 |

TABLE 5.1 – SAC MATRIX OF INPUT 10101010 TO AES S-BOX USING TRUNCATED DIFFERENTIAL APPROACH

| Input : 10101010<br>Original Output : 10101100 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| **V-vector of input 10101010** | | | | | | | |
| 1 | 2 | 2 | 3 | 2 | 3 | 2 | 3 |

TABLE 5.2 – SAC MATRIX OF INPUT 10101010 TO AES S-BOX USING 2-BIT ALTERATION APPROACH

| Input : 10101010<br>Original Output : 10101100 | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| **V-vector of input 10101010** | | | | | | | |
| 4 | 5 | 3 | 2 | 6 | 4 | 4 | 4 |

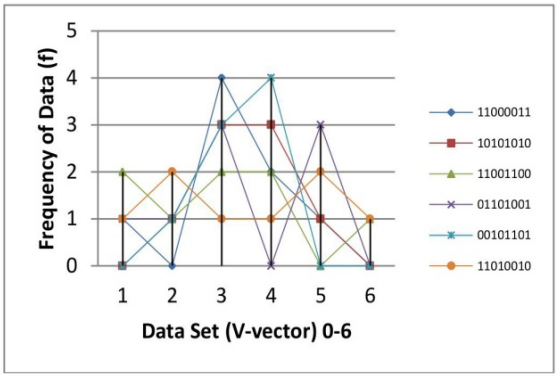


Fig. 5: Line Graph of Frequencies of V-vector for inputs using S-box of AES

*D. Experimental Results for AES S-Box*

The experimental results of the proposed test on the S-box of AES are in Table 6:

TABLE 6 – EXPERIMENTAL RESULTS OF PROPOSED TEST USING AES S-BOX

| Input | Observed Mean | Variance | Std. Deviation | Coefficient of Variance |
|---|---|---|---|---|
| $(195)_{10}$ | 3.25 | 1.1875 | 1.089725 | 0.3353 |
| $(170)_{10}$ | 3.5 | 0.75 | 0.866025 | 0.247436 |
| $(204)_{10}$ | 3 | 0.25 | 1.581139 | 0.527046 |
| $(105)_{10}$ | 3.375 | 1.984375 | 1.408678 | 0.417386 |
| $(45)_{10}$ | 3.375 | 0.484375 | 0.695971 | 0.206213 |
| $(210)_{10}$ | 3.5 | 2.75 | 1.658312 | 0.473804 |

## VII. DISCUSSION

By using the proposed algorithm, coefficient of variance has been calculated as a statistical measure of dispersion for the output corresponding to all possible inputs of every S-box of DES and six numbers of randomly chosen inputs to the single S-box of AES.

From the generated SAC matrix from each S-box of DES, the vulnerability vector (V-vector) has been calculated by summation of $1's$ in every column of each SAC matrix and by using the data set ranging from 0 to 4 and it's frequency of appearance, the line graph has been plotted as shown in Fig. 5. To calculate the coefficient of variance of every S-box, statistical mean, variance and standard deviation have also been calculated. It is found that the coefficient of variance (CV) ranges from 0.48 to 0.53, where CV < 1 and average coefficient of variance of S-boxes of DES is 51%.
The V-vector for the single S-box of AES has been calculated in same way and using data set ranging from 0 to 4 and it's frequency of appearance, the line graph has been plotted as shown in Fig. 6. To calculate the coefficient of variance of the inputs using single S-box, statistical mean, variance and standard deviation have also been calculated. The coefficient of variance (CV) is found to range from 0.20 to 0.52 where CV < 1 and average coefficient of variance of S-box of AES is 36%.

## VIII.CONLUSION

Confusion and diffusion are the two major aspects to measure of the strength of a block cipher and there exist different methods to test diffusion and confusion characteristics of cryptographic algorithms. In this work, a boomerang-style attack approach to test the confusion characteristic has been proposed and was used to analyze statistically the occurrence of $1's$ of every column of SAC matrices of DES and AES S-boxes.

For the both cases of AES and DES, the coefficient of variance (CV) was found to range from 0.48-0.53 and 0.2-0.52, respectively, which are in the lower end of the spectrum indicating that performances of S-boxes of DES and AES are pretty good with respect to the proposed test using boomerang-style approach. The average CV of DES and AES is 51% and 36%, respectively, which helps us to

 

draw the conclusion that the performance of S-box of AES is better than that of S-boxes of DES.

The proposed boomerang-style attack approach for testing confusion characteristics of S-boxes will lead us to formulate more testing algorithm on different cryptographic algorithms.

## REFERENCES


[1] Wagner D. (1999) The Boomerang Attack. In: Knudsen L. (eds) Fast Software Encryption. FSE 1999. Lecture Notes in Computer Science, vol 1636. Springer, Berlin, Heidelberg

[2] Cid C., Huang T., Peyrin T., Sasaki Y., Song L. (2018) Boomerang Connectivity Table: A New Cryptanalysis Tool. In: Nielsen J., Rijmen V. (eds) Advances in Cryptology – EUROCRYPT 2018. EUROCRYPT 2018. Lecture Notes in Computer Science, vol 10821. Springer, Cham

[3] Choy J., Yap H. (2009) Impossible Boomerang Attack for Block Cipher Structures. In: Takagi T., Mambo M. (eds) Advances in Information and Computer Security. IWSEC 2009. Lecture Notes in Computer Science, vol 5824. Springer, Berlin, Heidelberg

[4] Kim, Jongsung & Moon, Dukjae & Lee, Wonil & Hong, Seokhie & Lee, Sangjin & Jung, Seokwon. (2002). Amplified boomerang attack against reduced-round SHACAL. SIACRYPT 2002, LNCS 2501, pp. 243–253, 2002.c©Springer-Verlag Berlin Heidelberg 2002.doi: 10.1007/3-540-36178-2_15.

[5] Koo, Bonwook & Yeom, Yongjin & Song, Junghwan. (2010). Related-Key Boomerang Attack on Block Cipher SQUARE. IACR Cryptology ePrint Archive. 2010. 73. 10.1587/transfun.E94.A.3.

[6] H. Handschuh, D. Naccache, SHACAL, NESSIE project, October 2001.

[7] Biryukov, A., Khovratovich, D.: Related-key cryptanalysis of the full AES-192 and AES-256. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 1-18.Springer, Heidelberg (2009)

[8] Avijit Datta, Dipanjan Bhowmik, Sharad Sinha, "A New Approach towards Confusion Analysis of S-boxes using Truncated Differential Cryptanalysis", International Journal of Computer Sciences and Engineering, Vol.7, Issue.1, pp.249-256, 2019.

[9] Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology, 4(1), 3–72. doi:10.1007/bf00630563

[10] Lai X., Massey J.L., Murphy S. (1991) Markov Ciphers and Differential Cryptanalysis. In: Davies D.W. (eds) Advances in Cryptology — EUROCRYPT '91. EUROCRYPT 1991. Lecture Notes in Computer Science, vol 547. Springer, Berlin, Heidelberg

[11] Cheung, Jennifer Miuling. "The design of S-boxes." PhD diss., Sciences, 2010.

[12] C. Adams and S. Tavares. The structured design of cryptographically good s-boxes.Journal of Cryptology, 3(1):27–41, 1990.

[13] J. Cobas and J. Brugos. Complexity-theoretical approaches to the design and analysis of cryptographical boolean functions. InComputer Aided Systems Theory–EUROCAST2005, Lecture Notes in Computer Science. Springer-Verlag, Berlin, Germany, 2005.

[14] A.Datta, D.Bhowmick, S. Sinha, "A Novel Technique for Analysing Confusion in S-boxes." International Journal of Innovative Research in Computer and Communication Engineering, 2016. 4(6): p. 11608-11615.

[15] A.Datta, D.Bhowmick, S. Sinha, "Implementation of SAC Test for Analyzing Confusion in an S-box Using a Novel Technique." International Journal of Scientific Research in Computer Science Applications and Management Studies, Vol. 7, Issue 3, No. 182

[16] D.Bhowmick, A.Datta, S. Sinha. "A Bit-Level Block Cipher Diffusion Analysis Test." Springer International Publishing Switzerland 2015: S.C.Satpathy et. al. (eds), Proc of 3rd Int. Conf. on Front. of Intell. Comput. (FICTA) 2014-Col. I, Advances in Intelligent Systems and Computing 327. pp: 667-674.

[17] P. Sharma, D. Mishra, V.K. Sarthi, P. Bhatpahri, R. Shrivastava, "Visual Encryption Using Bit Shift Technique", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.3, pp.57-61, 2017


## AUTHORS PROFILE


**Avijit Datta** is a Research Scholar in the Department of Computer Science and Application, University of North Bengal and Assistant Professor of Siliguri Institute of Technology, Siliguri. He received Master of Computer Application (MCA) degree in 2005 from UPTU, UP, India. His research interest is Cryptology.

**Dipanjan Bhowmik** is an UGC-SRF in the Department of Computer Science and Application, University of North Bengal. He received Master of Computer Application (MCA) degree in 2011 from University of North Bengal, WB, India. His research interest is Cryptology.

**Sharad Sinha** is an Assistant Professor of University of North Bengal. He received Ph.D. degree in 2008 and Master of Computer Application (MCA) degree in 1992 from University of North Bengal, WB, India. His research interest is Cryptology, NLP.


 

# Appendix 3: Plagiarism Report

URKUND

## Urkund Analysis Result

| | |
|---|---|
| Analysed Document: | Avijit Datta_Computer Science and Application.pdf (D54569474) |
| Submitted: | 7/25/2019 1:23:00 PM |
| Submitted By: | nbuplg@gmail.com |
| Significance: | 3 % |

Sources included in the report:

https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf
https://www.linkedin.com/pulse/symmetric-block-cipher-versus-stream-darril-gibson
https://thisismyclassnotes.blogspot.com/2017/03/cryptography-stream-cipher-vs-block.html
http://euler.ecs.umass.edu/ece597/pdf/Crypto-Part2-Stream.pdf
http://www1.spms.ntu.edu.sg/~kkhoongm/GUFN_conf.pdf
http://ijarcsse.com/Before_August_2017/docs/papers/Volume_4/8_August2014/V4I8-0108.pdf
3d7da71d-e72a-4a4c-828c-f397b35e43d0
33562e48-847d-4820-8ff7-1802b1e4e003

Instances where selected sources appear:

23