

Analysis of Linear cryptanalysis for lightweight AES block cipher

BY

Nur Mohammad Sarker

162-19-1891

Md. Wahiduzzaman

162-19-1892

Muhammad. Abdur Raihan

162-19-1910

This Report Presented in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Electronics and Telecommunication Engineering

Supervised By

Ms. Tasnuva Ali

Assistant Professor

Daffodil International University



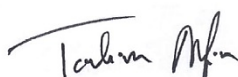
DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA-1207, BANGLADESH

APPROVAL

This thesis, titled "**Analysis of Linear cryptanalysis for lightweight AES block cipher**" submitted by Nur Mohammad Sarkar, Md. Wahiduzzaman, Mohammad Abdur Raihan to the Department of Electronics and Telecommunication Engineering (ETE), Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the bachelor degree in Electronics and Telecommunication Engineering and approved as to its style and contents.

BOARD OF EXAMINERS



Associate Prof. Md. Arefin Taslim

Chairman

Head

Department of ETE Faculty of Engineering
Daffodil International University

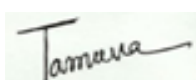


Professor Dr. A K M Fazlul Haque

Internal Examiner

Professor & Associate Dean

Faculty of Engineering
Daffodil International University



Ms. Tasnuva Ali

Internal Examiner

Assistant Professor

Department of ETE
Daffodil International University



Dr. Saeed Mahmud Ullah

External Examiner

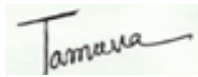
Associate Professor

Department of Electrical and Electronics Engineering
University of Dhaka

DECLARATION

We thus proclaim that this examination is our work and effort under the supervision of **Ms. Tasnuva Ali, Assistant Professor, Department of Electronics, and Telecommunication Engineering (ETE)**, Daffodil International University, Dhaka. It has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Supervised By:



Ms. Tasnuva Ali,
Assistant Professor
Faculty of Engineering
Daffodil International University

Submitted By:

Nur Mohammad Sarkar
162-19-1891

Md. Wahiduzzaman
162-19-1892

Muhammad. Abdur Raihan
162-19-1910
Department of ETE
Daffodil International University

DEDICATION

This thesis is wholeheartedly committed to our dearest guardians, who have been our source of motivation and gave us strength when we thought of surrendering, who consistently provide their ethical, spiritual, feeling and financial support.

ACKNOWLEDGEMENT

We would like first and foremost to express and to extend our appreciation to the Almighty Allah in all facets of our lives for his blessing, acceptance, safety, mental strength, and wisdom. All of the applause to Allah that this thesis is fulfilled.

The real soul of attaining an aim is via proficiency and strict discipline. With no collaboration, motivation, and the help of distinct personalities, we would never have triumphed.

This work wouldn't have been possible without the support and guidance of **Ms. Tasnuva Ali, Assistant Professor, Department of Electronics, and Telecommunication Engineering (ETE)** Daffodil International University, Dhaka, that appears to become our research manager for her endless patience, and educated advice continual encouragement, continuous and playful supervision, constructive criticism, and invaluable advice, studying poor drafts and correcting them at all stages have made it possible to finish this study. Additionally, we picked this subject and developed the study. In-depth Knowledge and keep interested in our manager in cryptography and motivated us to carry out this study.

We thank all of our faculty members, both the Electronics and Telecommunications Engineering Department, for their kind help in completing the thesis. We'd also want to thank the employees of the ETE Department of Daffodil International University.

We must sincerely comprehend with due respect that our family members and our friends are always supportive and individual of our schedule because of their support and assistance.

Nur Mohammod Sarkar

Md. Wahiduzzaman

Muhammad. Abdur Raihan

ABSTRACT

This research work represents the analysis of linear cryptanalysis of lightweight AES block ciphers. The linear cryptanalysis is widely known as one of the primary tools for S-Box security design. The AES is effective for fast, light and hardware reduction systems whereas the lightweight algorithm is used for smooth application. Therefore, the S Box is usually used to obscure the connection between the main text and the cipher text. This S Box design is basically two types: LUT S-Box and composite S-Box. The composite S Box design is based on Affine Transformation (AT) where different affine matrices and constants are used to enhance the security of the design. Hence, this research is focused on using different affine constants to find affine transformation and linear cryptanalysis of different S Boxes to examine the linear attacks on cipher text. The 4X4 affine matrix and 16 different affine constants have been used for this work. The result has been evaluated on strong linear values among 16 different constants. Finally, the S-Box has been proposed according to the best linear values which are more effective and sensible in data protection on lightweight AES block ciphers.

TABLE OF CONTENTS

Contents	page
Title page	i
Approval page	ii
Declaration	iii
Dedication	iv
Acknowledgment	v
Abstract	vi
Table of contents	vii-x
CHAPTER ONE	
1.1 General introduction/Overview	1
1.2 Aims and Objective	2
1.3 Motivation	2
1.4 About Research	3
CHAPTER TWO	
2.1 Related Works and Contributions	4
2.2 Selection of Lightweight Cryptography	4
2.2.1 Hardware Implementation	5
2.2.2 Software Implementation	5
2.3 Selection of AES	5
2.4 Encryption with AES	6
2.5 S-box (Substitution box)	7
2.6 Look-up Table	7
2.7 Affine Transformation	8

2.8	Linear Cryptanalysis	9
-----	----------------------	---

CHAPTER THREE

3.1	Methodology	11
3.2	Kinds of cryptanalysis	11
3.2.1	Differential Cryptanalysis	11
3.2.2	Linear Cryptanalysis	12
3.3	The block diagram of Linear Cryptanalysis calculation and explanation and steps	12
3.3.1	Linear Cryptanalysis Calculation	13
3.3.2	Affine Transformation	13
3.3.3	Selection of 2D Affine Matrix	13
3.4	The input of Linear Cryptographic algorithm	14
3.4.1	Constant for Linear Cryptographic Algorithm	15
3.4.2	Linear Approximation Calculation	15
3.4.3	Matsui's Algorithms	17
3.4.4	Matsui's Algorithm (i)	17
3.4.5	Matsui's Algorithm (ii)	18
3.5	Piling-Up Lemma	19
3.5.1	Piling-Up Lemma (Matsui)	19
3.5.2	Linear Expressions for S-Boxes	20
3.5.3	Linear estimation of a few rounds	22
3.5.4	Extracting final round key bits	23
3.5.5	Finding Linear Expressions	24
3.5.6	Matsui Operation of Linear Expression	25

CHAPTER FOUR

4.1	LAT Calculation	27
4.2	The key difference of particular value for variables 0110 and 1010	30
4.3	Proposed S-Box:	35
4.4	Analysis and Discussion	37

CHAPTER FIVE

5.1	Conclusion	39
5.2	Future work	40

List of figures

Figure 2.1	The general structure of encryption and decoding in Advanced Encryption Standard	6
Figure 2.2	S-box operation	7
Figure 2.3	Lookup table-based S-BOX	7
Figure 3.1	Block diagram of linear cryptanalysis calculation, explanation, and steps	12
Figure 3.2	Structure of Two-dimensional Affine Transformation	13
Figure 3.3	The graphical portrayal of the expression $X_2 = Y_1 \oplus Y_2$ on a 4-round S-box.	20
Figure 3.4	Linear Substitution permutation network	24

List of Tables

Table 3.1	Look-Up Table (LUT) Representation in hexadecimal	14
Table 3.2	Constant value representation in hexadecimal	15
Table 3.3	A few linear expressions of the Simple 4-Bit S-Box	21
Table 4.1	Linear Approximation Table	28
Table 4.2	Approximate linear bias number calculation for two S-boxes	29
Table 4.3	Linear Approximation Counter number for two constants	31
Table 4.4	The quantity of Sub key bias for two constants has been calculated.	32
Table 4.5	Calculation of Linear Attack Key Bias Difference Subject to Linear Best Bias.	34
Table 4.6	Proposed S-Box	35
Table 4.7	Linear key Bias for the all constant of specific input from the LUT.	36

List of Graph

Graph 4.1	Probability bias of constant 0110 and 1010	30
Graph 4.2	The counter number for Two Different Constant.	32
Graph 4.3	Bias Difference for Two Different Constant	33
Graph 4.4	The overall level of the bias varies by the constant	35
Graph 4.5	Showing the key difference between all constant.	37

Analysis of Linear cryptanalysis for lightweight AES block cipher

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In cryptography, linear cryptanalysis is a well-known form of cryptanalysis with discovering affine approximations into cipher's improvement. Linear cryptanalysis is just one of the most broadly used tools on block ciphers. The inherent cryptanalysis version was previously created in 1993 by M. Matsui because of a technological attack on a lightweight algorithm[1]. It's currently used broadly speaking on block ciphers around the area of cryptanalysis with an important starting place for advancing complicated strikes.

Linear cryptanalysis sets a linear link between the security of plain text, the cipher text and also the secret. This analysis intends to reach a linear approximation utilizing different continuous bits[2]. In linear cryptanalysis the method is to determine the expression of the form on which there is a high or low probability of occurrence. Linear Cryptanalysis leads to substantial computational savings over trying all possible values of the key.

We use Algorithms are automated to solve objectives and often a solution to a specific problem. Algorithms perform crucial functions in data security. Here we used Lightweight algorithm because it has a small footprint and less computational complexity. It makes a wide range of applications for cryptography to compile limited devices and its associated international standardization and guidelines are currently under consideration. Lightweight algorithm has functions to enable secure encryption, even for devices with limited resources[3].

The Advanced Encrypting Standard (AES), recognized as a lightweight block cipher, was established at the National Institute of Standards and Technology (NIST) in 2001[4] . In this research, the Lightweight AES is used because of its effective hardware reduction systems, and faster applications

1.2 AIMS AND OBJECTIVE

This research is rendered with the aid of the lightweight AES cryptographic algorithm for linear cryptanalysis.

- Firstly, the thesis will be demonstrated in Linear Cryptanalysis of lightweight AES block cipher.
- The linear analysis will be done using different constant values of S-Box which design is based on Affine Transformation, different affine matrices and constants.
- Finally, the S-Box has been proposed on depending the most efficient and sensitive linear values after linear cryptanalysis calculation.

1.3 MOTIVATION

This research is motivated by the fact that, many research articles have been worked on fixed constant value and carried out the Affine Transformation randomly. However, the result of linear approximation was considered with fixed constant and S-Box values. Therefore, the objective of the research is to work with variable constants with a 4 by 4 affine matrix to check the effect of linear cryptanalysis for these different constants and S-Box values.

1.4 ABOUT RESEARCH

In this analysis, the purpose was dealing with the related investigate on Affine Transformation (AT) in which numerous matrices and constants are used to improve design security. This analysis has been carried out using the 4X4 matrix and sixteen distinct affine constants. In this article, the chapter one describes the general introduction, aims, objective and motivation.

In chapter two, it will discuss about the review of the discovery and history of types of Cryptography, Lightweight Cryptography, Linear Cryptanalysis, Advanced Encryption Standard (AES), Substitution Box (S-Box), Look Up Table (LUT) and Affine Transformation (AT). The chapter 3 will give the methodology, whereas the chapter 4 will analysis the data and finally calculate Linear Cryptanalysis and S-Box. The final result and summery will be given in Chapter 5.

Chapter 2

2.1 RELATED WORKS AND CONTRIBUTIONS:

S-box is the only component to perform nonlinear operations in cipher blocks. The number of machines increases as parallel s-boxes are used. But their S box was not repaired in the journal Jitendra, as they did not give priority to lining strikes in leading to S box defense [5]. We test the security capacity of the S box according to a theoretical cryptanalysis system, which is a distinct linear attack. Ensuring the avalanche with a standard 1 box results in the main generation. Due to the linear approximation using the top one, they did not merit that[6].

2.2 SELECTION OF LIGHTWEIGHT CRYPTOGRAPHY:

The Lightweight cryptography could be useful in IoT and RFID gadgets because of their lightweights and faster speed. However, the public (NIST) and worldwide (ISO/IEC) institutions define the gadget array as follows:

1. Conventional cryptography: Servers and Desktops; Tablets and smartphones.

2. Lightweight cryptography: Embedded Systems; RFID and Sensor Networks.

The Embedded systems typically celebrate 8-bit, 16-bit, along with 32bit microcontrollers and that will struggle to accommodate ongoing requests for routine cryptography plans. Additionally, there's even a strong marketplace for 4-bit chips. However, the RFID and sensor system gadgets, have restricted amounts of entry ways accessible for security and also are usually profoundly pushed with the force station on the device[7].

2.2.1 HARDWARE IMPLEMENTATION

In hardware implementation of these lightweight cryptography primitives, the code size, memory consumption, and energy intake will be the vital matrix. It will nicely praise in lightweight cryptography, the precise kind of circuit (like the clock), memory and saving of those inner conditions, and also essential conditions that ought to be taken into account[8].

2.2.2 SOFTWARE IMPLEMENTATION

In the software implementation case, the execution dimensions, power consumption and the throughput (bytes per cycle) are more preferable metrics for the lightweight applications. The Lightweight AES is much smaller, the better for S-Box implementation[9].

2.3 SELECTION OF AES (ADVANCED ENCRYPTION STANDARD)

The AES Rijndael is an asymmetrical lightweight block cipher made by Joan Daemen and Vincent Rijmen that was picked by NIST in the year 2000.[10] The Advanced Encryption Standard is a part of the Rijndael block number created by two Belgian encryption[11]. In terms of properties such as security, productivity, efficiency, performance, flexibility, and implementation efficiency, the Rijndael algorithm gets the biggest scores. As it is faster, lighter, and has minimum use of hardware, this algorithm is also effective in any system implementation. Lightweight block ciphers were developed for smooth implementation in technology. A means of establishing a technique for implementing AES as a lightweight block cipher is an immediate prerequisite. A common block cipher used for many purposes is AES. A standard (Rijndael) algorithm along with the improved version of the AES algorithm using mixed columns, parallel replacement bytes, and shift rows uses the round key into the encryption, and also the opposite of the decryption measures. The plan is to be made in each phase which is created and stored the storage requires a good deal of space to implement [12].

2.4 ENCRYPTION WITH AES

The AES's encryption period can be broken up into three phases: the underlying round, the basic rounds, and the last round. The entirety of these phases use similar sub-tasks in a variety of

combinations as follows [13]

- Opposite Last Round
 - AddRoundKey
 - ShiftRows
 - SubBytes
- Opposite Primary Round
 - AddRoundKey
 - MixColumns
 - ShiftRows
 - SubBytes
- Opposite Starting Round
 - AddRoundKey

The total rounds of AES are rehashed a set number of times for every variation of AES. AES- 128 utilizations nine emphases of the primary round, AES-192 utilizations 11, and AES-256 utilizations 13[14].

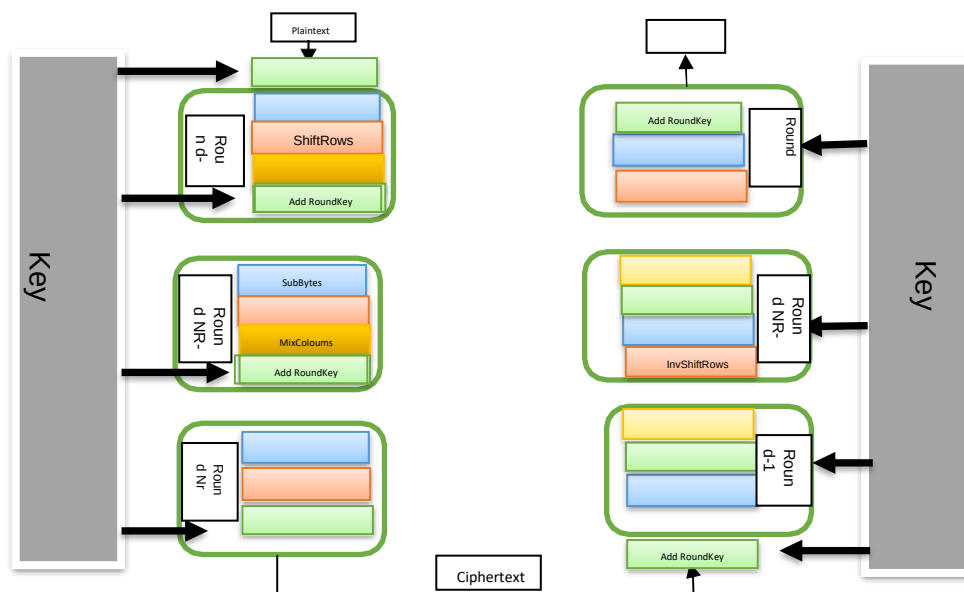


Fig 2.1 General structure of encryption and decoding in Advanced Encryption Standard

The four sub-tasks of AES AddRoundKey, SubBytes, ShiftRows, and MixColumns. These are clarified in more detail in the accompanying subsection[15].

2.5 S-BOX (SUBSTITUTION BOX)

S-box (substitution-box) is a simple part of symmetric important algorithms that performs substitution[16].

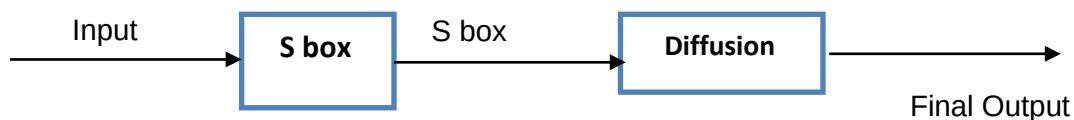


Fig 2.2: S-Box Operation.

In block ciphers, it is generally utilized to obscure the association between the main and the cipher text. Substitution-Box (S-BOX) will be the most crucial block from the Advanced Encryption Standard (AES) algorithm which absorbing 75 percent of the overall power through encryption. The main notion to employ S-BOX would be get an exceptional byte substitution [17].

2.6 LOOK-UP TABLE

The S Box hardware implementations have two mainly common and diverse processes. One of these techniques is indeed made up of this instant with the S Box substitution table, stored in the LUT search table. The type S box is fast but the hardware is expensive. The improvement of the S box works using composite field arithmetic operations is based on another technique. Thus, in simple GP (2^4) and additional multiplications

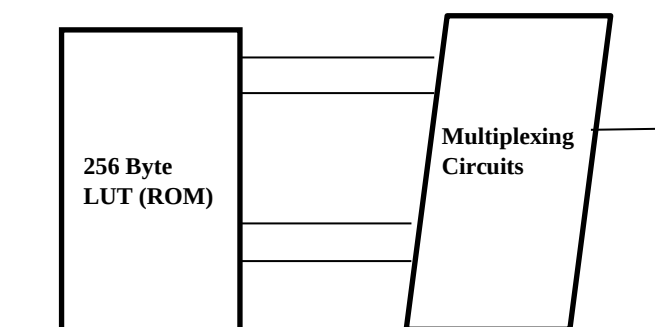


Fig 2.3: Lookup table-based S-BOX [16].

An inverse location in $GF(2^8)$ is reduced to the inverse. It goes back to the suggestion of Rijmen.

This decrease enables relatively simple hardware implementation of this S-box. Although the execution of the S-box in each area is significantly smaller than the LUT-based execution and can be

pipelined more time is needed for energy consumption. Look up table (LUT) includes a large memory size used for ROM or EPROM. LUT based S-BOX implementation is straightforward and requires majority memory. The LUT S-BOX carries a rigid architecture and suffers from inevitable delay because LUT possesses predefined entrance time. The memory is going to be used to maintain the worth and scanning will be finished alongside rows and columns to bring the desired value exhibited in cerebral 8 bits data that is substituted will be utilized to find the appropriate "to be substituted" value in the table. For this, 8 bits of information as a mix of 2 nibbles. This will provide the amount of the row in where information is to be pulled and then will provide the amount of this column in where information is to be hauled.

2.7 AFFINE TRANSFORMATION

Affine can be an integrated system of mapping that maintains objects, straight lines and planes. When the sets of parallel lines have been finely transformed, they continue to be parallel. The elegant transforming technology can typically be used appropriately for geometrical or even none ideally articulated distortions. For instance, panoramic sewing and sensor object collecting use affine transformations with a wide-angle-lens distortion. To prevent distortion, it is utilized to transform to a sufficient, horizontal communication and attribute-absorption program. This provides smoother connections and estimates without tracking chromatic aberration.

Mathematical restrictions, creation, extension, reflection, revolution, shear, transformations of similitude, winding resemblances and perception are all affinity transformations, identical to their blends. In short, an affine transition is a bit of revolution, interpretation, extension, and shear [18].

2.8 LINEAR CRYPTANALYSIS

The Linear Cryptanalysis is an intrusion known from plaintext that implies that have plaintexts and cipher texts which they all want to identify and encrypted with almost the same key. Howard Heys offers a nice creation platform to learn this methodology by a basic cipher (which is how I get to the subject) is another valuable source for linear cryptanalysis (and differential) Give reference number. So far, we are well aware of the significance of cryptanalysis. The "linear" in linear cryptanalysis is that we ought to use linear binary equations to maneuver and resolve them. This means that the coefficients behave like linear binary formulas

$$A \oplus B \oplus C \oplus \cdots \oplus Y = Z$$

This equation is the same because of the XOR operator's flipping capacity

$$A \oplus B \oplus C \oplus \cdots \oplus Y \oplus Z = 0$$

(Obtained on both sides by taking the previous equation and XORing Z).

These binary equations are structured to balance the inputs and outputs of different parts of the cipher. For instance, we might want to have an approximation of double round inputs bits-X0 and X2-with triple rounded output bits-Y1, Y2, and Y4-which would appear like

$$X_0 \oplus X_2 \oplus Y_1 \oplus Y_2 \oplus Y_4$$

This calculation certainly won't always be valid. Several other calculations will be generated and we'll finally use these connections to construct a linear "approximation" of the entire chip in which the final outputs bits consist of main, plaintext, and cipher text. The more detailed the strategy, the less work we have to do to find the key.

A short note: if we have an expression equal to 1 (instead of 0), so the calculations are not quite operating.

$$A \oplus B \oplus C = 1$$

The system of linear cryptanalysis structures the cipher mostly as a linear equation, utilizing plaintext, cipher text, and primary bits as the variables in the equation, identical to the ones above. For all of these ciphers, concerning the plaintext, the encryption process is reasonably random-looking, so that every arbitrary linear expression can be valid around half the time. E.g. if we analyze all of the cipher text bits, we can see that, the bit is 0 half of the time and 1 half of the time. Linear cryptanalysis seeks to find expressions that are not as random: the chance of finding a 0 or 1 is not even 1/2. The success rate of identifying the key from

These expressions increase as we discover expressions with probabilities further away from $1/2$.

The linear encryption algorithm was first defined by Matsui and Atsuhiro in the early 90s in a known plain-text attack against FEAL and AES. This systematic method seeks to connect the algorithm components' inputs and outputs to provide details about the key bits which are used to encrypt these components by resolving a system of linear equations[19].

Also, linear cryptanalysis is a statistical attack in any single situation. It is not guaranteed to function. Most of the time, however, it does function. The methods in this chapter and the next are intended to take advantage of flaws in some of the structures of the cipher. Ideally, a cipher would have almost completed diffusion and confusion; that is, without understanding the key, there would have been no simple way to make assumptions about the output based on the input. No cipher may, however, have genuinely perfect diffusion.

The method of linear cryptanalysis is presented and analyzed in this segment. We also analyze the usefulness of the system against several ciphers. Finally, we demonstrate some extensions and improvements to how it operates, for different values.

CHAPTER 3

MATERIALS AND METHODOLOGY

3.1 METHODOLOGY

In the most well-known case, we have been expecting that, everything about the encryption calculation is separated from the private key. Hence, the linear cryptanalysis of lightweight block cipher is calculated in three different stages which is clearly described below:

- In this Chapter, the process of calculation will be discussed about Linear Cryptanalysis, their related formulas and rounds.
- The best constants will be chosen with the help of Affine Transformation. Therefore, the constant effect is also analyzed on output characteristics.
- This linear estimation is a capacity that relates the plaintext bits, the cipher text bits, and the bits of the private key. Thus, the proposed linear approximation x will give the best data security compared to other constant values.

3.2 KINDS OF CRYPTANALYSIS

The two main types of statistical cryptanalysis are linear and differential attacks. They have numerous versions and enhancements like the multidimensional linear attack as well as the truncated differential attacks.

3.2.1 DIFFERENTIAL CRYPTANALYSIS

One of the first mathematical attacks on a large bunch of block ciphers will be Biham's and Shamir's differential attacks[20]. Many variations, enhancements, and generalizations are proposed since its invention in the early nineties. So-called compressed differential attacks were introduced with Knudsen back in 1994 [21].

3.2.2 LINEAR CRYPTANALYSIS

Some general ways of analyzing block ciphers were presented in the previous chapters. Methods were "common" in that there were not many studies of a single cipher; attacks work equally well on several different groups of ciphers. Any errors exploited were not so intrinsic in the ciphers: the detailed study of the ciphers was not what prompted the attacks. Although several crypt analytical techniques could be focused on an in-depth study of a cipher, we want to concentrate on a few different groups of these ciphers. In this chapter, we are beginning to discuss a very important class of modern techniques of cryptographic algorithms — linear cryptanalysis[22].

3.3 THE BLOCK DIAGRAM OF LINEAR CRYPTANALYSIS CALCULATION AND EXPLANATION

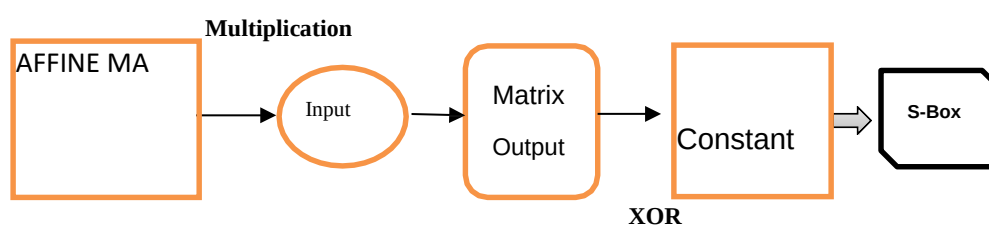


Fig 3.1 Block diagram of linear cryptanalysis calculation.

STEPS:

1. The first step is to multiply the input of the search table with a matrix of 4 by 4, and the values are applied.
2. In the second step, the matrix results are xor with the indicated input sequence consistency, one per relative input resulting from the output of that input being considered.
3. In this way, the same rule is followed in respect of 16 constants in the initial step for different constants with the same matrix output.
4. The next move is to insert with the specified S-boxes linear expressions, attacking the linear relative to each constant, to see whether for a certain constant has any reaction on the S-box.

Some formulas have been used to calculate LAT in the S-box, a linear expression, a linear influencing number bias, the frequency of the S-box, the dependency of the S- box on the

Individual inputs. The piling up lemma of Matsui is one of the algorithms that will be implemented.

3.3.1 LINEAR CRYPTANALYSIS CALCULATION

The next generally applicable attack on block ciphers will be Matsui's linear attack [23]. This debut will many improvements and extensions will make for S-box Linear attack which will ensure that the observe efficiency of lightweight block cipher.

3.3.2 AFFINE TRANSFORMATION

An affine transformation is a transformation in which the jelly is not perfect correlation (all focused on one point lie first on a transformation stage) and the proportions (e.g. the midpoint of the line section remains in the middle of the conversion stage). Affine thus reveals a special class of projective transformations that do not transfer articles from the endless way or the relative plane of space R^3 into the plane. An affine transformation is often referred to as affection which is explanation of the revolution, interpretations, enlargements, and shears.

3.3.3 SELECTION OF 2D AFFINE MATRIX

The 2D affine conversion effect a picture 3X2 matrix directly uses a 2D conversion matrix and the six indicator preposition mode uses a spatial change to do something. The researcher can use this effect to rotate, scale, type, or translate an image or it could be a mix of those functions from the other end. One of the most essential keystones that protect the equilibrium line with any three focused S-boxes in the separation ratio would be a non-linear, reverse conversion from the affine conversion to the AES level security consequences.

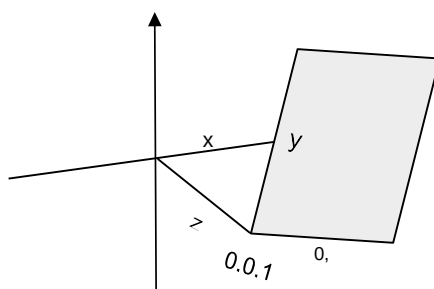


Fig 3.2 Structure of two-dimensional Affine Transformation.

First, the affine matrix was analyzed from the S-box. We need to take a 2D affine matrix. Allow the affine matrix value with 2D 4 with 4 0 0 1 0 1 0 1 1 0 0 0 1 then it multiplies with all our input 4 bits. This could lead to the conversion of Affine into 2D, represented as linear transformations in 3D. First, let us make our 2D distance in 3D by creating one plane in our lift $z = 1$. The old source is currently at a 3D distance (0, 1). In this way, it maps any value other than itself, as long as we preserve the 3D origin (0, 0, and 0) as free. The 2D affine transformation is given:

$$x' = Ax + b$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} a1 \\ a2 \\ a3 \\ a4 \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} b1 \\ b2 \end{pmatrix}$$

Equivalent to a 3D linear transformation

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a1 & a2 & b1 \\ a3 & a4 & b2 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

3.4 THE INPUT OF LINEAR CRYPTOGRAPHIC ALGORITHM

The search tables can be organized as key-value pairs, where the key data objects are checked (shown above) and the values are $\oplus$ indicated the actual data. Occasionally, the search table contains only data elements (values only, not key-value pairs). The table and key positions are equivalent, for example, a 256-bit (BIT 0, bit 1). Lookup tables are widely used to write and confirm values by fitting a set of valid (or invalid) items in a variety of specific programming languages, (tags or even offsets) that can be incorporated for title purposes.

Input	0000	0001	0010	0011	0100	0101	0110	0111
	1000	1001	1010	1011	1100	1101	1110	1111

Table 3.1 Look-Up Table (LUT) Representation in hexadecimal

3.4.1 CONSTANT FOR LINEAR CRYPTOGRAPHIC ALGORITHM

A collection and well-defined number or alternative mathematical unit significance is often characterized by the terms mathematical constant or physical constant. This job comes with an unchanged significance and can be an average of a variable that doesn't rely on

the significant factor involvement. This is the case, as an example, for an integration constant that's an arbitrary constant (i.e. does not rely upon an integration variable) and can be applied to a specific anti-derivative as a way to obtain all the anti-derivatives of the job specified.

For our thesis purpose using 16 different constant numbers in binary format that is (0-15) Hexadecimal value.

Constant	0000	0001	0010	0011	0100	0101	0110	0111
	1000	1001	1010	1011	1100	1101	1110	1111

Table 3.2 Constant value representation in hexadecimal

After multiplying the LUT value in the affine matrix, we get the matrix output. That (XOR) is the constant after the value with the individual input that we get from LUT's table. We will get 16 kinds of matrix outputs, with each constant being set per round and then XOR will be the output for 16 different constants.

3.4.2 LINEAR APPROXIMATION CALCULATION

The few outputs and inputs are mentioned here in each round. The input bit has been modified, but with a logical simultaneous and even true table, different variables can play a different part. The output value will be evaluated in the same way like the 16-bit logical aggregation method. As the whole result is stored in binary, thus all input and output count numbers are also binary. Each iteration will have a binary difference and will eliminate binary inputs and outputs.

$$\text{Linear approximation} = \text{Total count number} - (\text{S box size}/2)$$

1st iteration:

Input masked binary= *first input* * *Constant* (AND gate) Output masked
binary= (*constant*) * *first output*

2nd iteration:

Input masked binary= *Second input* * *Constant* (AND gate) Output masked
binary= (*constant*) * *Second output*

The process is repeated up to 16th iteration.

After all calculations of iterations, the similitude number is found which is proved by the conditions of the truth table. The total number of such inputs and outputs is the number of all similar numbers. These Input and output variables are specified with LAT variables. As the numbers are different, the input-output can vary according to their resemblance.

$$\text{Linear approximation} = \text{Total count number} - 8$$

3.4.3 MATSUI'S ALGORITHMS

For linear cryptanalysis, Matsui provides two techniques, but only one is used frequently. Suppose we have an algorithm that includes plaintext bits, cipher text, and a key, such as $P_0 \oplus P_2 \oplus \dots \oplus C_1 \oplus C_2 \oplus \dots = K_1 \oplus K_3 \oplus \dots$

If the bits have strong propagation and uncertainty properties, this equation occurs at about half the time, quite supposedly and unpredictably. If the probability of the events is different from 1/2, may conclude that this is a bias. When p is probable to be valid, $-1/2$ is the bias — that is, the discrepancy between p and 1/2, ignoring the consequence sign (the absolute value). It will know a few values from the bits P and C and search for the bits K if they suspected a known-plaintext attack. The above equation is used by Matsui's Algorithm 1 and its bias to provide us with a single piece of knowledge concerning the key. The following and this methodology both use the highest probability principle which is the most possible cause to be assumed[22].

3.4.4 MATSUI'S ALGORITHM 1

Let the expression of a linear form: Suppose P in this equation.

1. Cipher text for a specific key (mentioned below for calculation probability). Let we have

been selected different pairs of suitable plaintexts.

2. Measure to the left from the linear expression above with the plain-cipher text coordinate.

Let it be 0 below that cycle.

3. If T crosses half and P crosses $1/2$, then we conclude that on the right (Kbit XORed) 0.

I. So T is less than half a pair and P is less than $1/2$ for the bit XOR, so you can still guess 0.

II. If not, guess the K-bits' XOR has to be 1.

These final three steps are present since we believe that our estimates and probabilities mean that if P is less than $1/2$ then we can conclude that the Classification of the key is incorrect and if P is greater than $1/2$ then the equation is generally valid. If we assume it is going to be wrong and the relationship of 0 on the side then it has to be 1 on the right. So it's been false and so declares the overall equation is incorrect. If we have a huge number of plaintexts - some corresponding equations with script pairs and large orthodoxy, then we can create a series of key bit calculations that can be used to bring out the key bits. Most powerful ciphers have a very small bias for the entire S-box (usually about half of the time any linear expression is correct). Fortunately, Matsui has published another algorithm on key bits that theoretically reveals to recover not only linear words relying on a certain numerical but also plain-cipher text pair[22].

Here it estimates to how many times the evaluation follows rather than comparing the true count of the left and trying to correct each K-bit sequence.

3.4.5 MATSUI'S ALGORITHM 2

Suppose a linear form expression: we need to know the probability of this equation, $p[23]$.

- Just a unique connection that we want to get is the code pair - collect the number of planes (according to the constant).
- Calculate the main bit of each candidate set, that instance number is a T-value for the truth of the linear equation plain-cipher text pair.
- Select the most distant candidate key out half a pair. So, if the number of N pairs, measure T ($N / 2$) (absolute difference value) - for each candidate ($N / 2$) and several major candidates give the best optimal form of the result.

The idea here is that, we all aim at the potential range of key bit values. If it has a 4-bit key for analysis and can detect a 32-bit key-specific expression, which will take 220 actions per second ($220+32$) = (252) 32-bit operating Process. The following sections are our support numbers, timely estimates as well as linear strategy detection, and plain-cipher text couples location.

3.5 PILING-UP LEMMA

In the attack, there is join a few linear relationships on individual S-boxes for a linear connection over the initial three cycles. Here the potential bias for the next linear connection can go through the pile theme:

Let X_1, X_2 be two independent binary random variables and, $P_1 = Pr [X_1 = 0]$, $P_2 = Pr [X_2 = 0]$, then

$$\begin{aligned} Pr [X_1 \oplus X_2 = 0] &= Pr [X_1 = X_2] = Pr [X_1 = 0, X_2 = 0] + Pr [X_1 = 1, X_2 = 1] \\ &= P_1 P_2 + (1 - P_1)(1 - P_2) \end{aligned}$$

Let $\varepsilon_1, \varepsilon_2$ be the probability biases $X_1 = 0$ and $X_2 = 0$: $P_1 = \frac{1}{2} + \varepsilon_1$, $P_2 = \frac{1}{2} + \varepsilon_2$, then

$$Pr [X_1 \oplus X_2 = 0] = \frac{1}{2} + 2\varepsilon_1 \varepsilon_2$$

Hence the probability bias $X_1 \oplus X_2 = 0$ is $\varepsilon_{12} = 2\varepsilon_1 \varepsilon_2$

This can be generalized to n independent binary random variables.

3.5.1 PILING –UP LEMMA (MATUSI)

For n independent binary random variables $X_1, \dots, X_n$ with probability biases

$$\begin{aligned} \varepsilon_i &= Pr [X_i = 0] - \frac{1}{2} \\ Pr [X_1 \oplus \dots \oplus X_n = 0] &= \frac{1}{2} + 2^{n-1} \prod_{i=1}^n \varepsilon_i \end{aligned}$$

Or equivalently,

$$\varepsilon_{1, \dots, n} = 2^{n-1} \prod_{i=1}^n \varepsilon_i$$

Note that if $P_i = 0$ for all i, then $Pr [X_1 \oplus \dots \oplus X_n = 0]$ is either 0 or 1.

Also, if only one if $P_i = \frac{1}{2}$ then $Pr [X_1 \oplus \dots \oplus X_n = 0] = \frac{1}{2}$.

3.5.2 LINEAR EXPRESSION FOR S-BOXES

The first step towards a complete linear equation for Matsui's Algorithms is to understand basic linear terms are measured and their biases decided.

we have four potential input Bits and Four possible output bits that could either maintain or omit in each linear expression, then it has to appear through $2^4 \times 2^4 = 2^{4+4} = 2^8 = 256$ unique expressions. Additionally, we must try out all probable values of the input-output value pairs which are 23 (because we now have 3 input bits and the output is completely determined by them). This provides us $2^8 \times 2^8 = 2^{16} = 65536$ surgeries in total about the S-box.

Generalizing this, using an n-piece S-box (n-bit input, n-bit output), we have to Perform several operations equivalent to 2^{3n} [24]

The basic features of this linear approximation table can be mentioned. As an example, the probability that the output bit is not empty somewhat equal to the sum of any quantity in any input is precisely equal to $1/2$. Because the output bit must have an equivalent number of zeros in any linear combination and get a bijective S-box. Also, the linear combination between any output piece will be equal to $+1/2$ bias and the linear combination of any input piece leading to a table value of $+8$ from the top left corner. Thus, the best rows in this table are all zeros, except the value on the left. Similarly, the first column is all zeros except its predecessor. Additionally, it may be mentioned that the amount of any row or any column should be $+8$ or even -8 .

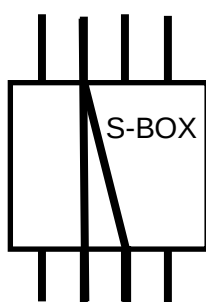


Figure 3.3 the graphical portrayal of the expression $X_2 = Y_1 \oplus Y_2$ on a 4-round S-box. The best

expression of the S-box is given below

S11: $Y_0 \oplus Y_2 \oplus Y_3 = X_1$ with probability $5/8$ and bias $1/8$. . (i) S22:

$Y_2 = X_0 \oplus X_3$ with probability $3/8$ and bias $-1/8$ (ii)

S32: $Y_2 = X_0 \oplus X_3$ with probability $5/8$ and bias $-1/8$ (iii)

S34: $Y_2 = X_0 \oplus X_3$ with probability $3/8$ and bias $-1/8$ (iv)

From equation (i) and (ii) following from Pilling Up-lemma

$$\frac{1}{2} + 2 \left(\varepsilon_1 - \frac{1}{2} \right) \left(\varepsilon_2 - \frac{1}{2} \right), \text{ where } \varepsilon_1 = \frac{5}{8} \text{ and } \varepsilon_2 = \frac{3}{8} \dots\dots (v)$$

From equation (iii) and (iv) following up from Pilling Up-lemma

$$\frac{1}{2} + 2 \left(\varepsilon_1 - \frac{1}{2} \right)^2 \text{ where } \varepsilon_1 = \frac{3}{8} \dots\dots (vi)$$

Here, the All-around key is fixed either 0 or 1 which is depending on the key of the cipher. So using all the equations (i), (ii), (iii), (iv) following up from Pilling Up-Lemma.

$$\frac{1}{2} + 2^3 \left(\varepsilon_2 - \frac{1}{2} \right) \left(\varepsilon_1 - \frac{1}{2} \right)^3$$

EQUATION	BIAS (ε)
$0 = 0$	$1/2$
$X_0 = Y_0$	0
$X_0 = Y_1$	0
$X_0 = Y_0 \oplus Y_1$	0
$X_0 = Y_2$	$1/2$
$X_1 = Y_0$	$-1/2$
$X_2 = Y_1 \oplus Y_2$	$-1/4$

Table 3.3 A few linear expressions of the Simple 4-Bit S-Box

We have generally focused on the value which has the lowest possible number of bits. For low bits of input and output, we can arrange all of the linear cryptanalysis for consisting of building different linear expressions. The syntax of Table 6-1 is somewhat conclusive. It's great for calculating exactly what bits and associated bias effects. But it's not easy to analyze. The use of a notation everywhere in the context of Matsui is quite useful. The input and output bits are represented by a pair of numbers (usually in hexadecimal). In the linear expression would be converted into 1 (in the binary picture) of the number and a certain sum, not present would be 0.

3.5.3 LINEAR ESTIMATION OF A FEW ROUNDS

Using the S-Box LAT as Well as the Pilling Up lemma, we will inform a model where the best way to utilize them to get a linear attack for the initial 3 rounds of this figure. The initial key-mixings, replacements, and stages, last around key-blending, substitute, and also final key- blending. We will use P_i to denote the i-th bit of the plaintext, K_r to denote the i- th bit of the round key, X_i and Y_i denote the i-th input and output bit of S-box $\in$

{11, 14, 21, 44}

➤ Start with the linear approximation, $X_{12,1} \oplus X_{12,3} \oplus X_{12,4} \oplus Y_{12,2} = 0$ over S-box S_{12} which has probability bias -1/8.

Note that, $X_{12,1} = P_5 \oplus K_{1,5}$, $X_{12,3} = P_7 \oplus K_{1,7}$, and $X_{12,4} = P_8 \oplus K_{1,8}$,

Although key-bits are unknown, for a given problem instance, they are fixed. Therefore, we gather sums of key-bits on the right-hand side:

$$P_5 \oplus P_7 \oplus P_8 \oplus Y_{12,2} = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \quad (1)$$

Holds with bias -1/8.

➤ Output bits $Y_{12,2}$ is the 10th bit and remains the 10th bit through the permutation, key mixing is applied and $Y_{12,2} \oplus K_{2,6} = X_{22,2}$ is the input of S-box S_{22} .

➤ Use the linear approximation $X_{22,2} \oplus Y_{22,2} \oplus Y_{22,4} = 0$ over S-box S_{22} with bias -1/4 and obtain $Y_{12,2} \oplus Y_{22,2} \oplus Y_{22,4} = K_{2,6}$ Holds with bias 1/4. (2)

➤ With the Piling-Up, Lemma can combine (1) and (2) to obtain

$$P_5 \oplus P_7 \oplus P_8 \oplus Y_{22,2} \oplus Y_{22,4} = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \quad (3)$$

➤ Output bits $Y_{22,2}$ and $Y_{22,4}$ pass through another permutation and key-mixing that is

$$Y_{22,2} \oplus K_{3,6} = X_{32,2} \text{ And } Y_{22,4} \oplus K_{3,14} = X_{34,2}$$

➤ For both S-boxes, S_{32} and S_{34} use the same linear relation as for S_{22} with the bias of 1/4.

$$X_{32,2} \oplus Y_{32,2} \oplus Y_{32,4} = 0 \text{ With bias } 1/4$$

$$X_{34,2} \oplus Y_{34,2} \oplus Y_{34,4} = 0 \text{ With bias } 1/4$$

Substituting the input bits with the previous output bit,

$$Y_{22,4} \oplus Y_{32,2} \oplus Y_{32,4} = K_{3,6} \text{ With bias } 1/4 \quad (4)$$

$$Y_{22,4} \oplus Y_{34,2} \oplus Y_{34,4} = K_{3,14} \text{ With bias } 1/4 \quad (5)$$

➤ applying the piling up lemma and combining the three linear relations (3), (4), and (5) to obtain

$$P_5 \oplus P_7 \oplus P_8 \oplus Y_{32,2} \oplus Y_{32,4} \oplus Y_{34,2} \oplus Y_{34,4}$$

$$= K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14}$$

The final bias of -1/64 depends on the individual relations' bias and the number of S-boxes actively involved in the relation. To strengthen a cipher, one can thus try to use S-boxes with very small biases and structures that try to maximize the minimum number of active s-boxes.

3.5.4 EXTRACTING FINAL ROUND KEY BITS

When we've got a linear approximation, the previous round for any particular cipher using a suitably big enough chance bias, it could attempt to harness it to regain some pieces of the last round key pieces from the finished key-mixing.[25]

The linear approximation we obtained above

$$Y_{32,2} \oplus K_{4,6} = X_{42,2}; Y_{32,4} \oplus K_{4,8} = X_{42,4}; Y_{34,2} \oplus K_{4,14} = X_{44,2}; Y_{34,4} \oplus K_{4,16} = X_{44,4}$$

$$P_5 \oplus P_7 \oplus P_8 \oplus X_{42,2} \oplus X_{42,4} \oplus X_{44,2} \oplus X_{44,4}$$

$$= K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14}$$

$$\oplus K_{4,16}$$

Let's replace the sum of key bits by zero:

$$P_5 \oplus P_7 \oplus P_8 \oplus X_{42,2} \oplus X_{42,4} \oplus X_{44,2} \oplus X_{44,4} = 0$$

This linear approximation holds with bias -1/64 if the sum of key bits equal to zero and with bias 1/32 otherwise. The magnitude remains the same only sign of the bias changed.

In our attack, the sign has no importance; only the magnitude is substantial to statistically distinguish a correct guess[26].

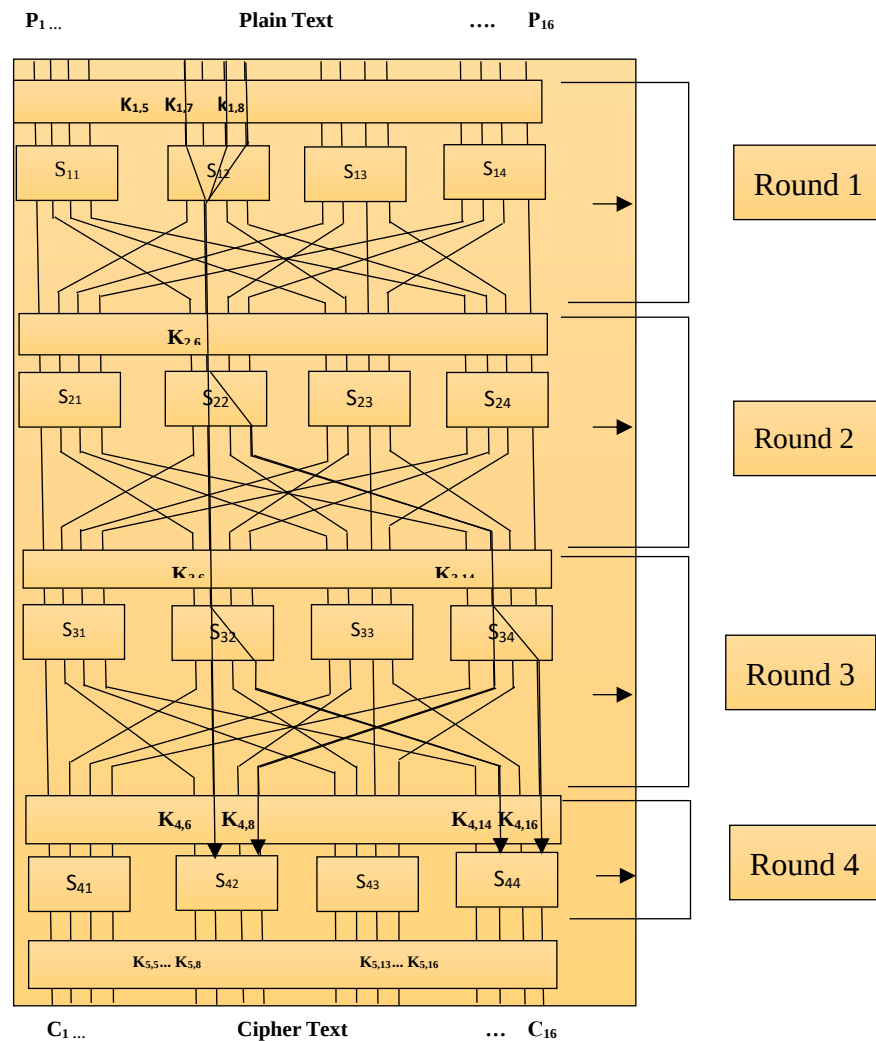


Figure 3.4 Linear Substitution permutation network

3.5.5 FINDING LINEAR EXPRESSIONS

During this period, finding good linear expressions. It should come as no real surprise that finding good linear expressions using high biases is just really a somewhat tough job, more so to locate the “top” for example saying. Matsui created an algorithm to perform these hunts.

3.5.6 MATSUI OPERATION OF LINEAR EXPRESSION

Here, we're choosing the most effective linear term, B , with induction, constructed out of a great suspect, and taking care of the ideal term for any particular chip. We have to assume that we have a very powerful word that's similar to

ROUND 1:

1. In the first circular, which Matsui calls a mask, sets $O1$ for each potential range of output bits, and do the following:
 - ❖ Let $p'1$ as the highest probability for every linear term outputting $O1$.
 - ❖ if $p'1$ is less than the probability of being the best expression than
2. Go through round 2.
3. If no key has been identified, back

ROUND 2:

1. Input bitmask ($I2$) and output bitmask ($O2$) for any possible value:
 - ❖ The input bits leading to the output bits should be possibly $p2$ (i.e., the linear expression $I2$ probability = $O2 = 0$).
 - ❖ If the $p'1$ and $p'2$ expression do not better than that
 - ❖ Go to Round 3 otherwise.
2. If no masks input or output function, go back to round 1.

ROUND 3, 4... ($n - 1$):

1. Let the round number be i .
2. For each possible value of the input bits, I_i :
 - ❖ Set $O_i = O_{i-2} \oplus I_{i-1}$ (because of the Feistel structure).
 - ❖ Set p'_i to be the probability for the linear expression of I_i yielding O_i .
 - ❖ If the linear expression involving $p'_1, p'_2, \dots, p'_i$ is not better than
 - ❖ Otherwise, call the next round ($i + 1$).
3. Return to the previous round.

ROUND N:

1. Set $O_n = O_{n-2} \oplus I_{n-1}$.
2. Set p'_n to be the maximum probability possible for any input giving O_n .
3. If the expression using p'_1, p'_2, p'_n is better than B.
4. Return to the previous round.

Even though Matsui's algorithm functions pretty much for finding grim expressions of AES, it's 2 deficiencies:

- 4 Several expressions researched will probably soon be equal to some other sayings searched, due to rarity from the cipher.
- 5 Several sayings researched aren't truly candidates.

RESULT AND CALCULATION

I N P U T	OUTPUT																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	-4	0	+4	-2	-2	-2	-2	-2	+2	-2	+2
	2	0	+4	-2	+2	0	0	-2	-2	0	0	-2	-2	0	-4	-2	+2
	3	0	0	+2	+2	0	0	+2	+2	+2	-2	-4	0	+2	-2	+4	0
	4	0	0	0	-4	-2	+2	+2	+2	0	0	0	-4	+2	-2	-2	-2
	5	0	+4	0	0	-2	+2	+2	+2	+2	-2	+2	+2	-4	0	0	0
	6	0	0	-2	+2	+2	+2	+4	0	0	+4	-2	-2	-2	+2	0	0
	7	0	0	+2	-2	+2	-2	0	0	-2	+2	0	0	-4	-4	+2	-2
	8	0	+2	0	-2	0	-2	+4	-2	-2	0	+2	0	+2	0	+2	+4
	9	0	+2	0	-2	-4	-2	0	-2	0	+2	-4	+2	0	+2	0	-2
	A	0	-2	-2	0	-4	+2	-2	0	-2	0	0	-2	-2	0	+4	+2
	B	0	+2	+2	0	0	-2	-2	0	+4	+2	+2	-4	0	+2	+2	0
	C	0	-2	-4	+2	-2	-4	+2	0	+2	0	+2	0	0	-2	0	-2
	D	0	+2	-4	-2	+2	0	-2	+4	0	+2	0	+2	+2	0	+2	0
	E	0	-2	+2	0	-2	0	0	+2	+2	+4	0	+2	0	-2	-2	+4
	F	0	-2	-2	-4	+2	0	0	-2	+4	-2	-2	0	-2	0	0	+2

Table 4.1 Linear Approximation Table

After all the calculation of iteration results, only 6 input numbers of input and outputs are logically true. So, the total count number is 6.

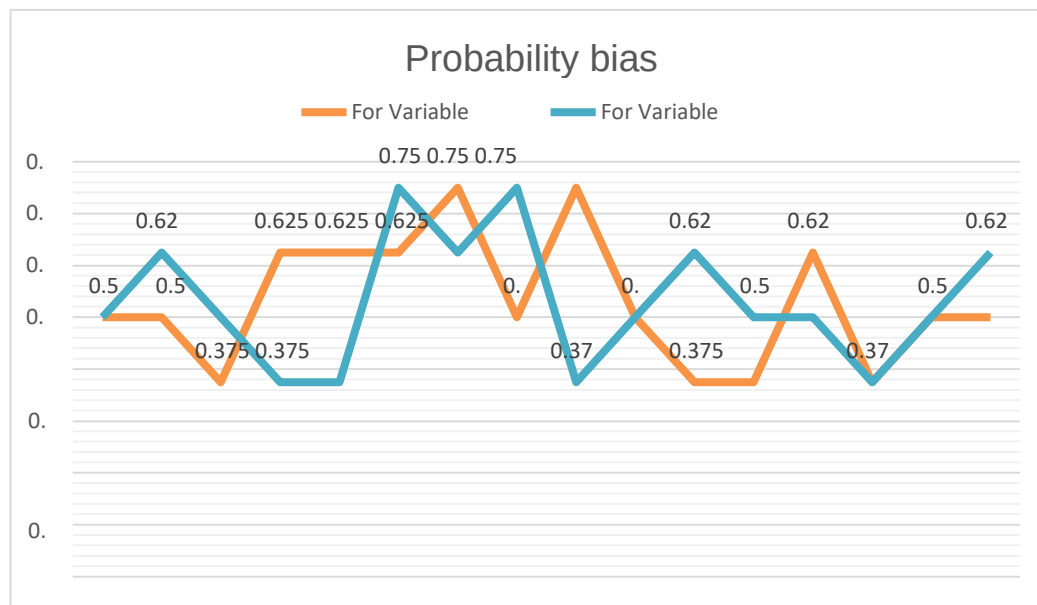
Linear approximation=Total count number-8

So that Linear approximation for input 0110(6) and output 1011(B) is proven which is -2. Suppose the input value will be taken as 0110(6) and the output value will be 1011(B). From the LAT table, we find that value is -2 for the probability of that particular value.

The table below shows the difference in bias between the constants, based on the same input, which shows that the difference in constant will impact the linear attack, even though the input is the same.

I/N	Constant 0110	Constant 1010	Bias 0110	Bias 1010
0000	0110	1010	0.5	0.5
0001	1111	0011	0.5	0.625
0010	1110	0010	0.375	0.5
0011	0001	1101	0.625	0.375
0100	1100	0000	0.625	0.375
0101	1010	0110	0.625	0.75
0110	1001	0101	0.75	0.625
0111	0000	1100	0.5	0.75
1000	1000	0100	0.75	0.375
1001	0010	1110	0.5	0.5
1010	0101	1001	0.375	0.625
1011	1101	0001	0.375	0.5
1100	0011	1111	0.625	0.5
1101	0100	1000	0.375	0.375
1110	1011	0111	0.5	0.5
1111	0111	1011	0.5	0.625
			8.5	8.5

Table 4.2 Approximate linear bias number calculation for two S-boxes



Graph 4.1 Probability biases of constant 0110 and 1010

This is the line graph comprising details regarding the distortions impacting the opponent. We take 2 constant data out of 16 constants. The bias discrepancy between the input parameter and the probability can be seen in the table above.

4.2 THE KEY DIFFERENCE OF PARTICULAR VALUE FOR VARIABLES 0110 AND 1010

The XOR number is used for the input mask. The values of plain text and ciphertext, where the position of the given value has the precursor of the straight line, which is given by the bits running in the spherical function. These 16 bits have a random number. This is a 16-bit size which size is not beyond the S-box size. In the final round, it has to remember that the total S-Box count factor is 65536, where the random number of the output acts as a counter number according to the input of each rounds table. Thus we know that the total counter factor is the s-box factor, where the known number is multiplied by the valid S-box dimension. The known S- box number given in this section is 15 bits and the S- box size is 32 bits. Therefore, the inverse of this value for this round table is $15 * 32 = 480$. Here the first counter number for the given S-box is to be taken randomly for the two constants.

1. In 8 rounds, the linear expression for 0110 operated in pairs, so this will be the first constant counter, multiplied by the specified S-box bit number, $32 * 8 = 256 = 256$. So the number of the first counter is 256, some as the number of the 2nd counter are $256 + 480$ (key difference) = 736, the number of the 3rd counter is $736 + 480 = 1216$, and the number of the counter is 16 inputs.
2. The first counter number would be $32 * 4 = 128$ in almost the same manner as working for 4 rounds in a linear expression with 1010 constant.

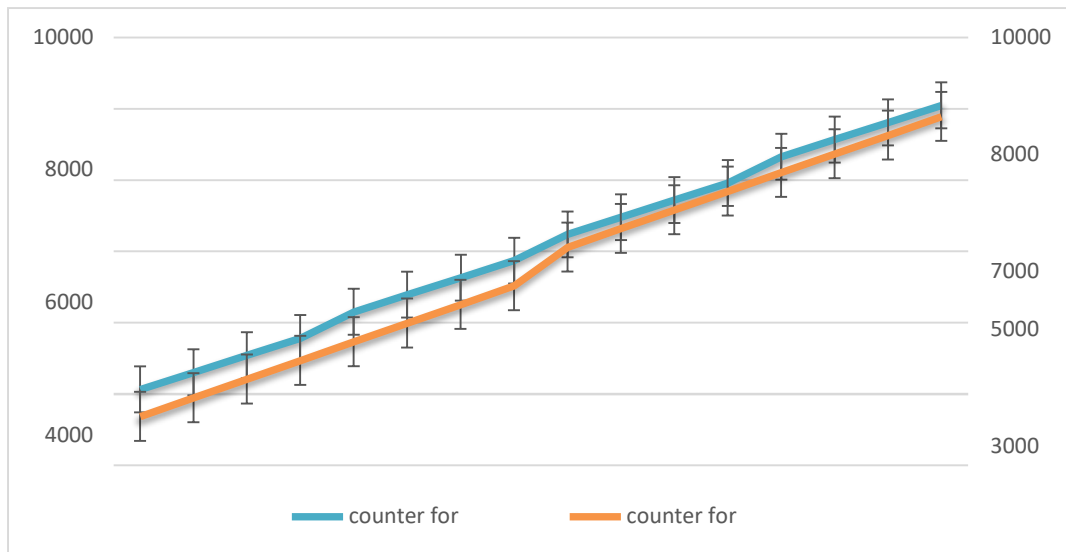
So 128 will be the first counter number for this constant, $128 + 480$ (key difference) = 608 will be the 2nd counter number, $608 + 480 = 1088$ will be the 3rd counter number for 16 multiple output counter values in the same of 16 variables.

I/N	Constant 0110	Constant 1010	Count 0110	Count 1010
0000	0110	1010	256	128
0001	1111	0011	736	608
0010	1110	0010	1216	1088
0011	0001	1101	1696	1568
0100	1100	0000	2176	2304
0101	1010	0110	2656	2784
0110	1001	0101	3136	3264
0111	0000	1100	3616	3744
1000	1000	0100	4608	4480
1001	0010	1110	5088	4960
1010	0101	1001	5568	5440
1011	1101	0001	6048	5920
1100	0011	1111	6528	6656
1101	0100	1000	7008	7136
1110	1011	0111	7488	7616
1111	0111	1011	7968	8096
Total			65792	65792

Table 4.3 Linear Approximation counter number for two constant.

The counter numbers arrive randomly with the S-box in the case of linear attacks. There are the counter numbers for the two constants in the table below. Much of the time, this counter number depends on the bit size of the S-box. Consider 32-bit S-Box size and 10000 count number of this research to calculate the values of table 4.3.

Considering the values derived from the table in the query, it can be shown that the S-box counter number for the same input differs. Whereas, the 16-bit S-box for the obtained constant has 0 (Total counter number)-65536=65792-65536=256 keys per round for each input.



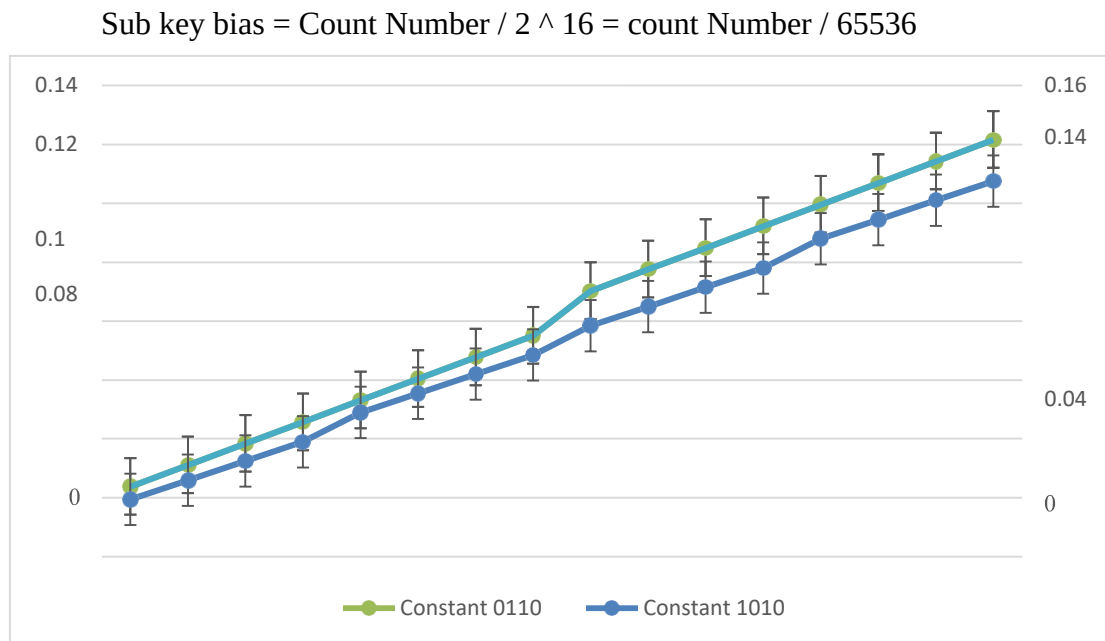
Graph 4.2 Counter numbers for Two Different Constant.

The variance of count between the two constants can be noted, as it can be seen from the given chart. The linear numbers are very near the top two constants.

I/N	Constant 0110	Constant 1010	Sub Key Bias 0110(K1)	Sub Key Bias 1010(k2)
0000	0110	1010	0.0039	0.00195
0001	1111	0011	0.0112	0.00927
0010	1110	0010	0.0185	0.0166
0011	0001	1101	0.0258	0.0239
0100	1100	0000	0.03320	0.03515
0101	1010	0110	0.0405	0.04248
0110	1001	0101	0.0478	0.0498
0111	0000	1100	0.05517	0.0571
1000	1000	0100	0.07031	0.0683
1001	0010	1110	0.07763	0.0756
1010	0101	1001	0.0849	0.0830
1011	1101	0001	0.09228	0.09033
1100	0011	1111	0.0996	0.1015
1101	0100	1000	0.1069	0.1088
1110	1011	0111	0.11425	0.1162
1111	0111	1011	0.12158	0.1235
Total			0.09683	0.18032

Table 4.4 The quantity of Sub key bias for two constants have been calculated.

The bias of the linear attack for each input has been found in contrast to the count number given in the table above, where we have found a relative plaintext/cipher text for each input continuously in each round. So get the value from



Graph 4.2 Bias Differences for Two Different Constant

As conspicuous, the default counter-number of a 32 bit S-Box is $2^{16} = 65536$, so the two S-Boxes' linearly attack is different,

For constant 0110, Bias approx./65536=6346/65536= **0.09683**

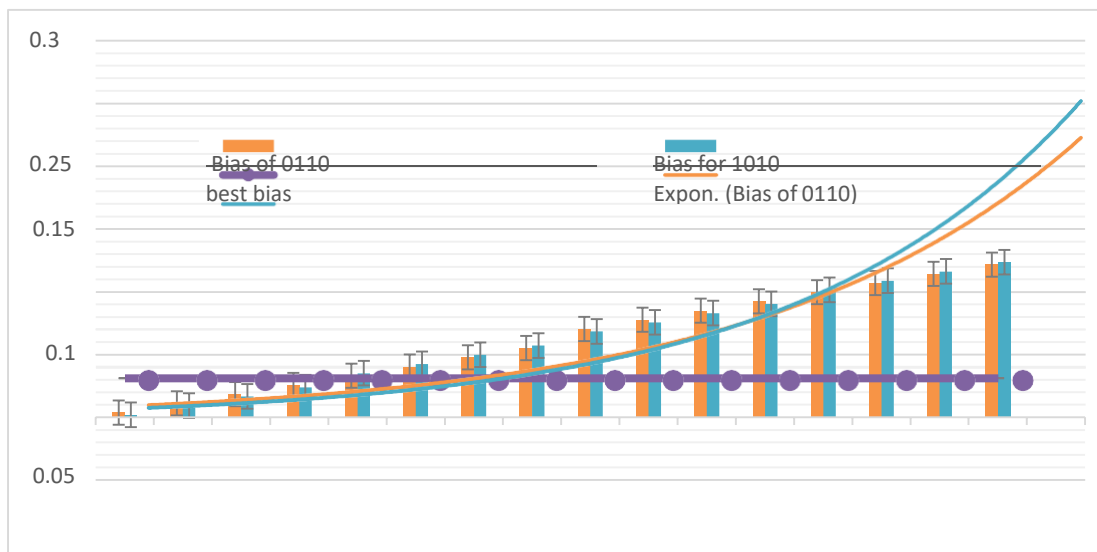
For constant 1010, Bias approx./65536=11817.6/65536= **0.18032**

I/N	Constant 0110	Constant 1010	Bias Differences 0110	Bias Differences 1010
0000	0110	1010	0.02723	0.0293
0001	1111	0011	0.02005	0.02198
0010	1110	0010	0.01275	0.01465
0011	0001	1101	0.00545	0.00735
0100	1100	0000	0.00195	0.0039
0101	1010	0110	0.00925	0.01123
0110	1001	0101	0.01655	0.01855
0111	0000	1100	0.02392	0.02585
1000	1000	0100	0.03906	0.03705
1001	0010	1110	0.04638	0.04435
1010	0101	1001	0.05365	0.05175
1011	1101	0001	0.06103	0.5908
1100	0011	1111	0.06835	0.07025
1101	0100	1000	0.07565	0.07755
1110	1011	0111	0.083	0.08495
1111	0111	1011	0.09033	0.09225
Total			0.6346	1.18176

Table 4.5 Calculation of Linear Attack Key Bias Difference Subject to Linear Best Bias

We are simulating targeting fundamental cipher with the output of 10000 known plaintext/cipher text values and the encryption method for sub key values.

Thus the total bias approximation= 0.06346×10000 (for constant 0110) =6348
= 1.18176×10000 (for constant 1010) = 11817.6



Graph 4.4 The overall level of the bias varies by the constant

The variation in the calculation between the 16 constants can be noticed, as can be seen from the given chart. The linear number is very close to the top two steps. Where the biases of the other constants have been observed to differ considerably.

4.3 PROPOSED S-BOX

The analysis will be applied for all 16 possible constants using programming. The best values of our analysis are 0110 and 1010. Although these two constants, the linear final round keys 0110 reach linear reading for $1/32$ or 0.03125 which is very different. On the other hand, 256 key variations in terms of the same constant are included in each round table. Therefore, the proposed S- Box has been designed with constant 0110 to get the appropriate linear approximation.

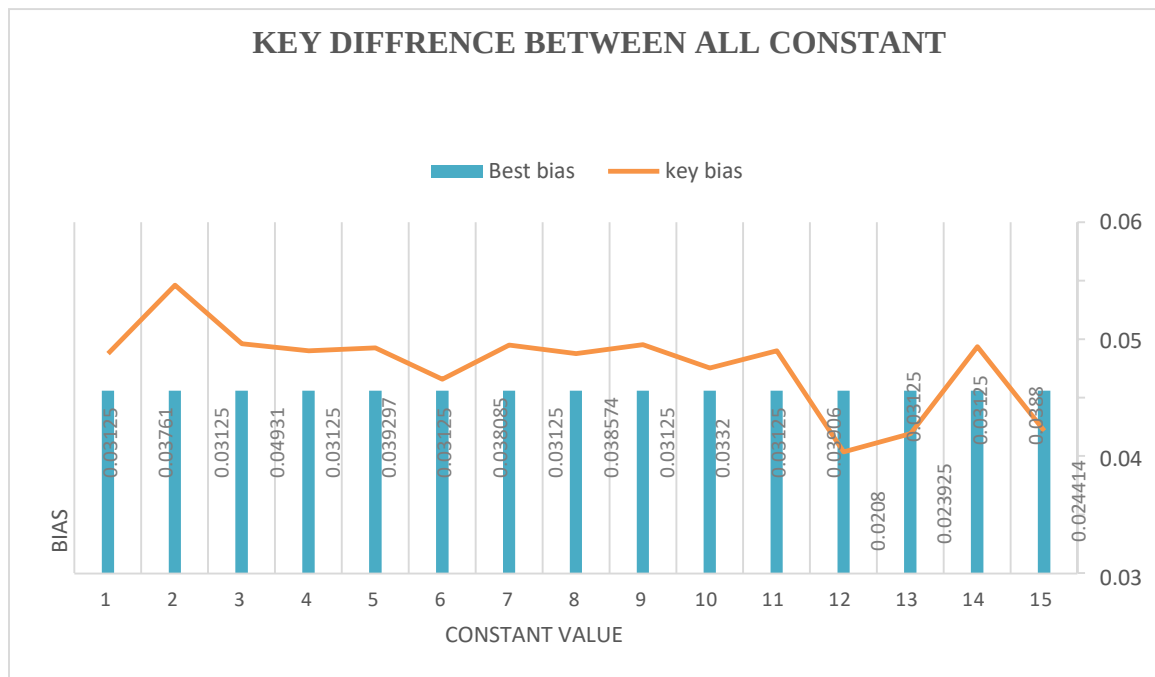
Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
output	6	F	E	1	C	A	9	0	8	2	5	D	3	4	B	7

Table 4.6 Proposed S-Box

Input	Constant	Key Bis
0100	0001	0.03761
0100	0010	0.04931
0100	0011	0.029297
0100	0100	0.038085
0100	0101	0.038574
0100	0110	0.03320
0100	0111	0.03906
0100	1000	0.03759
0100	1001	0.039125
0100	1010	0.03515
0100	1011	0.03808
0100	1100	0.0208
0100	1101	0.023925
0100	1110	0.03880
0100	1111	0.024414

Table 4.7 Linear key Bias for the all constant of specific input from the LUT.

In linear distance calculation, the difference between the values given by the graph is suggested that all the constants for the subject under a linear attack have been separated.



Graph 4.5 The key difference between all constant.

The graph shows the linear crypt of all possible constants linear analysis. This analysis shows the key difference balance between them and also gives the best bias value for linear cryptanalysis.

4.4 ANALYSIS AND DISCUSSION

- The cipher text is primarily a complex S-box feed for the extra volume of cipher text and plain text collection.
- Since there is no clear understanding of the given objective sub key, we have re-verified this effective algorithm with a possible 256 keys.
- The extent of each sub key relies on its S-box to maintain a regular linear position which has been explicitly reviewed.
 - The condition holds $1/2$ probability (like the use of random qualities)
 - If the correct sub key is used more than $1/2$ of the state, it will be established on a comparatively constant basis (sub key to bias) on the S-box.

After verifying both the general text and the cipher text collection, we use the sub-key for the most individual check-in 1/2.

The test results of the linear attacks are 1/32 which represents the fraction 0.03125 which is the same for the two compatible properties 1010 and 0110. But their closest choice is

For constant 1010,

$$0.03515-0.03125=0.0039$$

For constant 0110,

$$0.03320-0.03125=0.0019$$

The predisposition approximation of 0.03195 is exceptionally near the approximations of 1/32. Another huge bias occurs when the erroneous objective half-sub key's assessment is not accurately proportional to the opposite of the key with a linear expression.

This research paper mainly focus on the effect of the different constant values of linear cryptanalysis of S-box. We have taken the compound S-box design which is based on the affine transformation (AT), where different affine matrices and constant are used. Seeing the different constants we take affects the S box. We saw various security enhancement designs when the constant was changing. After taking 16 different constants, two different constants show the best linear values which are more effective and reasonable in data security on lightweight AES block ciphers for S-box.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

Linear cryptanalysis helps to make linear connections between planes, cipher text, and key bits for high probability S-Boxes. In this paper, a 4X4 fixed affine transformation with 16 constant contacts is used to find linear attacks. After finding Linear Attacks, the best value has been proposed to make the strongest S Boxes for better security. After data analysis, it was found that the constant 0110 gives the best result which is near about $\frac{1}{32}$ and **0.03125** among 16 constants. Therefore, our proposed S-Box is based on 0110 constant value which will be more secure compared to other S Boxes.

5.2 Future Work

This thesis work is a preliminary investigation of linear cryptanalysis of lightweight block ciphers like AES. With these vulnerabilities, the researcher will perform differential cryptanalysis and integral cryptanalysis of these constant values. Researcher will perform more analysis on notable block ciphers like DES, IDEA, SERTENT and Blowfish.

REFERENCE:

- [1] M. Matsui, “Linear cryptanalysis method for DES cipher,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 765 LNCS, pp. 386–397, 1994, doi: 10.1007/3-540-48285-7_33.
- [2] J. Daemen and V. Rijmen, “Cryptanalysis,” in *Information Security and Cryptography*, 2020.
- [3] S. Rana, S. Hossain, H. I. Shoun, and M. A. Kashem, “An effective lightweight cryptographic algorithm to secure resource-constrained devices,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 11, pp. 267–275, 2018, doi: 10.14569/ijacsa.2018.091137.
- [4] S. W. An and S. C. Seo, “Highly efficient implementation of block ciphers on graphic processing units for massively large data,” *Appl. Sci.*, vol. 10, no. 11, Jun. 2020, doi: 10.3390/app10113711.
- [5] K. B. Jithendra and T. K. Shahana, “A New Efficient Sbox for Strengthening PRESENT Like Block Ciphers Against Linear Cryptanalysis,” *2019 2nd Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2019*, pp. 507–511, 2019, doi: 10.1109/ICICICT46008.2019.8993397.
- [6] D. Cryptanalysis and H. Heys, “Cryptanalysis of Modern Symmetric-Key Block Ciphers Substitution-Permutation Networks (SPN).”
- [7] W. J. Buchanan, S. Li, and R. Asif, “Lightweight cryptography methods,” *J. Cyber Secur. Technol.*, vol. 1, no. 3–4, pp. 187–201, 2017, doi: 10.1080/23742917.2017.1384917.
- [8] S. Campbell, M. Grichenko, and W. Smith, “Linear cryptanalysis of simplified AES under change of S-box,” *Cryptologia*, vol. 37, no. 2, pp. 120–138, 2013, doi: 10.1080/01611194.2012.660236.
- [9] J. Artiles, D. Chaves, and C. Pimentel, “Linear Cryptoanalysis of the Simplified AES Cipher Modified by Chaotic Sequences,” *J. Commun. Inf. Syst.*, vol. 34, no. 1, pp. 92–99, 2019, doi: 10.14209/jcis.2019.10.

- [10] M. James and D. S. Kumar, "An Implementation of Modified Lightweight Advanced Encryption Standard in FPGA," *Procedia Technol.*, vol. 25, pp. 582–589, 2016, doi: 10.1016/j.protcy.2016.08.148.
- [11] B. Rothke, "A look at the Advanced Encryption Standard (AES)," *Inf. Secur. Manag. Handbook, Sixth Ed.*, pp. 1151–1158, 2007, doi: 10.1201/9781439833032.ch89.
- [12] S. W. An and S. C. Seo, "Highly efficient implementation of block ciphers on graphic processing units for massively large data," *Appl. Sci.*, 2020, doi: 10.3390/app10113711.
- [13] C. Arul Murugan and P. KarthigaiKumar, "Survey on Image Encryption Schemes, Bio cryptography and Efficient Encryption Algorithms," *Mob. Networks Appl.*, pp. 1–6, 2018, doi: 10.1007/s11036-018-1058-3.
- [14] K. S. Ooi and B. C. Vito, "Cryptanalysis of S-DES," *Computer (Long. Beach. Calif.)*, 2002.
- [15] N. Fips, "197: Announcing the advanced encryption standard (AES)," ... *Technol. Lab. Natl. Inst. Stand.* ..., 2001, doi: 10.1016/S1353-4858(10)70006-4.
- [16] A. Kumar and S. Tejani, "S-BOX Architecture S-BOX Architecture," no. January, 2019, doi: 10.1007/978-981-13-3804-5.
- [17] T. G. S. Chandrasekharappa, K. V Prema, and K. Shama, "S-boxes generated using Affine Transformation giving Maximum Avalanche Effect," *Int. J. Comput. Sci. Eng.*, vol. 3, no. 9, pp. 3185–3193, 2011.
- [18] "3D Affine Transformation Matrices."
- [19] S. Dey and R. Ghosh, "A Review of Existing 4-Bit Crypto S-Box Cryptanalysis Techniques and Two New Techniques with 4-Bit Boolean Functions for Cryptanalysis of 4-Bit Crypto S-Boxes*," *Adv. Pure Math.*, vol. 08, no. 03, pp. 272–306, 2018, doi: 10.4236/apm.2018.83015.
- [20] C. Blondeau, G. Leander, and K. Nyberg, "Differential-Linear Cryptanalysis Revisited," *J. Cryptol.*, vol. 30, no. 3, pp. 859–888, 2017, doi: 10.1007/s00145-016-9237-5.
- [21] L. R. Knudsen, "Block Ciphers - The Basics Introduction," 2011.
- [22] M. Matsui and A. Yamagishi, "Chapter 6 Linear Capacity."
- [23] H. Kruppa and A. Shah, "Differential and Linear Cryptanalysis in Evaluating AES candidate algorithms," vol. 8, p. 302, 1385.

- [24] L. Lacko-Bartošová, “Linear and differential cryptanalysis of reduced-round AES,” *Tatra Mt. Math. Publ.*, vol. 50, no. 1, pp. 51–61, 2011, doi: 10.2478/v10127-011-0036-y.
- [25] L. Cryptanalysis, “MasterMath Cryptology 2015 - 2/2 - Cryptanalysis,” vol. 7, pp. 1–9, 2015.
- [26] P. Lemma, “Linear Cryptanalysis,” pp. 1–6, 1993.