HEURISTIC ANALYSIS
OF
SUBSTITUTION BOX

A DISSERTATION

Submitted towards the fulfillment of the

requirement for the award of the degree

of

MASTER OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

By

ANUPAM AGARWAL

Enrollment No. 13535011

Under the supervision of

Dr. Sugata Gangopadhyay

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY - ROORKEE

ROORKEE – 247667 (INDIA)

MAY- 2016

AUTHOR’S DECLARATION

I hereby declare that the work present in this thesis with title “**Heuristic Analysis Of S-Box**” toward the partial fulfillment of the requirements for the award of the degree of Master of technology submitted at Department of Computer Science and Engineering, IIT Roorkee is my own work under the guidance of Dr. Sugata Gangopadhyay, Associate Professor at IIT Roorkee during period May 2014 – May 2016.

The work contained in this thesis has not been previously submitted for a degree or diploma at any higher education institution. To the best of my knowledge and belief, the presented thesis contain no material previously published or written by another person except where due reference is made.

Date: MAY 2016 Anupam Agarwal

CERTIFICATE

It is certified that the work contained in this thesis titled “**Heuristic Analysis Of Substitution Box**” by “**Anupam Agarwal**” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Date: MAY 2016

Dr. Sugata Gangopadhyay
Associate Professor
IIT Roorkee

ACKNOWLEDGEMENT

I would like to thank Dr**. Sugata Gangopadhyay** for introducing me to the world of cryptography and his guidance with this project on “**Heuristic Analysis of Substitution Box**”, especially for providing me with the materials and readings. Provided material helped me understand the concept better and through which I was able to programme my thesis topic efficiently. I am very pleased to thank **Ronak, Angad** and **Bimal Sir** for their heartily friendship and support. Last but not least, for my **parents** for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Anupam Agarwal

ABSTRACT

This report represents an investigation about Substitution Box (S-Box) used in cryptography system. S-Box is one of prime part of cryptographic algorithm. Various properties of Substitution Box makes cryptographic algorithm susceptible to attacks. Will first study attacker strategy with possible attacks on cryptographic system and then about S-Box design principles. The goal of any assaulter is to find the key by an effective technique (finds properties which distinguished cipher text from random text). Then we will study about the methods by which the cryptographic algorithm is supposed to be compromised. S-Box should produce output which is comparable to any random generator. Here in order to measure the deviation of Substitution box from a randomly generated permutation, I have generated large number of random permutations and then measured it with the distribution of used terms in every vectors, analysis them with respect to actual figures fetched from AES Substitution box.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1

INTRODUCTION

1.1 Utility of Cryptography

Information in any form is very crucial, whether it will be financial, military and legal. Therefore, safeguarding above information is very important; it can only be achieved through proper implementation of cryptographic tools. Protection of information is achieved by cryptography through its encryption and decryption techniques. Seeing/Modification of information is not allowed by cryptography.

We have two types of cipher in cryptographic algorithm viz. symmetric cipher and asymmetric cipher Symmetric cipher employs single key while two types of key i.e. Public and Private Key is used in asymmetric cipher. Attacker always wants to see/modify the information, and for this motive they have to break the cipher by cryptanalysis. Hence, it should be ensured that cryptographic algorithm in every form should be strong enough to resist the attack and protect the information by not letting it not known to attacker.

The goal of a sender and a receiver will be to use cryptography and to check that some characteristics of the data are not exposed. For instance, the integrity of data, that whether the data has been changed or not changed, the privacy of information, what information has been transported, and there are certain other objectives which has to be lived up to. So, typically, cryptography gives us some mechanisms to carry these operations forward, so that we can satisfy properties needed for security when we transfer information over an insecure channel.

1.2 Attacker Strategy

The cryptographic algorithmic rule needs to be planned to achieve security goals and this is the central rule in cryptography, it believe on a piece of data, which is also called as the secret key. So, the strategy is as that, everyone knows the algorithmic rule, the algorithmic rule are present in public domain, but the piece of information that the attacker do not know is the key. So, the goal of any attacker is typically, is to find out the key. So, if he succeeds in finding out the key, or deduced the secret key by an effective proficiency, then the cryptographic algorithmic rule is compromised.

Two important principles of cryptography are confusion and diffusion. In diffusion, effect of one bit of input bit is spread among more than half of the output bits. In confusion, the distribution dependence of output bits by input bits is try to made completely lost. Round repetition is a phenomenon that used to employ above principles in cryptographic system.

CHAPTER 2

LITERATURE SURVEY

2.1 Modern Block Ciphers

All block ciphers consists of four transformations as follows:-

1 Key Adding: - Key is added in each round to part of plaint text which ultimately results into cipher text. It is basically a XOR operation between a part of plain text and a part of the key resulting in cipher text for next stage.

2 Permutation: - In permutation box, input bit is permuted to make it difficult for attacker to do cryptanalysis.

3 Mixing: - It consists of shifting the bits by any multiplier.

4 Substitution: - In substitution box, input bit is substituted to make it difficult for attacker to do cryptanalysis.

Also, there are three types of cryptographic objects:-

1 Public Key: - It employs different key for encryption as well as decryption. One is private key and other is public key. Examples are Rivest-Shamir-Adleman (RSA), Elliptic Curve algorithms.

2 Private Key: - It employs same key for encryption as well as decryption. Examples are DES, 3 DES, AES. It is used when encryption of large amount of data is required.

3 Pseudo Random Generators :- Random number generator used in cryptographic algorithms are guided by algorithm. Hence, it can’t be truly random number generators. It is said to be Pseudo Random Generators

Cryptography generally transforms plain text into cipher text of same length viz 128 bits or 256 bits. Large key size of AES makes it more secure than DES. It is good for both eight and thirty two bit processor. It is also noticed that substitution boxes are only non-linear component in AES cryptographic algorithm and thus they are very important part of cryptographic strength. In this thesis we have learnt about substitution box properties which are very much helpful in determination of strong ness of ciphers. We generally prefer to use small amount of good substitution box in block cipher.

It has been found that randomly selected substitution boxes have large probability of above mention desirable properties. This thesis work outlines the effort made to select least pseudo random substitution box and it is least deviated from the mean of large number of random substitution boxes. Blowfish, Khufu are two famous and desirable key dependent substitution box of the past. Each of Blowfish, Khufu utilizes cryptosystems to be able to generate substitution box. The algorithm generated by us should have fair cryptographic strength with a good resistive property against attacks. Various cryptanalysis attacks such as linear or differential cryptanalysis cant able to compromise the secrecy of information.

2.2 Advanced Encryption Standard (AES)

Advanced Encryption Standard consists of 128 bits key data block in general but have various type of tailor made key length of 128,192 or 256 bits. As we all know initial state in every cryptographic algorithm is called as plain text and the final or last stage is known as cipher text. Advanced Encryption

Standard works on four bytes (four times) 2-D array, popularly known as State. Each state have four rows of bytes.

Every round function comprises of four transformations:-

- Sub Bytes: - It is a nonlinear transformation. It operates using substitution Box table on every byte present in the state. Table number or substitution box number is calculated by field inversion (finite) and transformation (affine).

- Shift Rows: - It is a shifting operation (circular). It shifts the rows of a particular state guided by actual byte number also known as offset. This offset is also popularly known as index (row). Shifting ultimately injects confusion in bits of round function.

- Mix Columns: - As the name suggests, it used to mix the bytes present in each column. Multiplying each column by x^4+1 modulo can effectively do the task of mixing each column present in the state.

- Add Round Key: - Each round of round function has different key. Different part of key is fed in each round of round function. Key is added in each round to part of plaint text which ultimately results into cipher text. It is basically a XOR operation between a part of plain text and a part of the key resulting in cipher text for next stage.

Each round function is applied variable number of times depending on key length. It can be 14, 12 or 10 which depends entirely on key length. For example;-

KEY LENGTH	ROUND FUNCTION
128	10
192	12
256	14

Table 1: Key Length and Round Function Relationship

2.3 Substitution Boxes

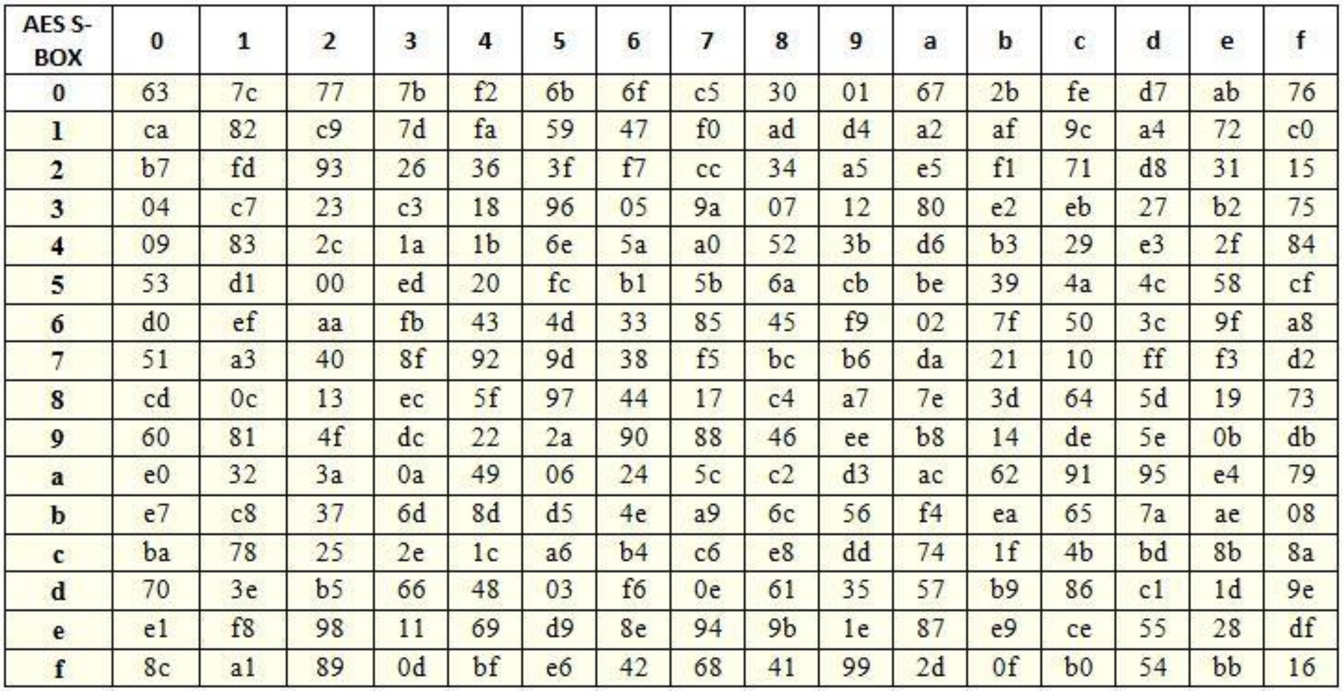
Confusion of bits can be achieved by applying substitution boxes in cryptographic algorithm. It is nonlinear transformation of algorithm which is very crucial phenomena for each modern block cipher algorithm. It is also proved to be very tough against linear and differential cryptanalysis. These transformations are employed by look up tables of substitution boxes.

Substitution boxes employ two transformations in the Advanced Encryption Standard which are listed as follows:-

1 Ist Transformation: - Finding the inverses (multiplicative) in Galois fields or GF (2^8).

2 2nd Transformation: - Second Transformation is affine transformation. Main aim of using this transformation after multiplicative inverses is that all possible algebraic attacks can be mounted.

Substitution Box employs as a lookup table in which every input bit is substituted by other bit. For example in substitution box of Advanced Encryption Standard every bit among eight bit input is substituted by output bit as per below lookup table.



AES S-BOX	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

Table 2 : AES S-Box

Boolean function is very important part which is extensively used inside substitution box of many cryptographic algorithms like AES. Boolean function typically maps multiple input bits to one of the output bit. Boolean function is not only employs mapping phenomena but successfully implements cryptographic methods like confusion and diffusion as described earlier effectively. It is also used in key generation for stream cipher. Hash f (n) also uses the same in round f (n). There are large number of properties of Boolean f(n) like non-linearity, correlation immunity etc. which makes algorithm resists against cryptographic attacks.

CHAPTER 3

CRYPTOGRAPHIC GOALS AND ATTACKS

3.1 Cryptographic Goals

The three main goals of Cryptography:-

Confidentiality: - Hide the information from access which is not authorized, or a person which is not legally authorized cant access the info. unless the access given.

Integrity: - The information must be forbid from modification, by anyone who is not legalized to do it.

Availability: - The network and communication should not be busy enough, that the information is not approachable to the user, who is authorized.

Cryptographic algorithmic rule needs to be formed to achieve above objectives. Everyone knows algorithms, the algorithmic rules are present in public arena, but attacks do not know the piece of information, which we call as secret key. The objective of any attacker is basically, to deduce the key. So, if he is able to deduce the secret key, by an effective technique, then the cryptographic algorithmic rule is said to be broken.

Also, the objectives which cryptographic algorithms guarantees like confidentiality, integrity and availability will not exist any longer, because the basic algorithmic rule on which above objectives are accomplished, are endangered by the attacks.

3.2 Exploiting Cryptography Algorithm

As per [5], cryptographic algorithm has a 128 bit key. Meaning of 128 bit key is that you have 2 ^ 128 possible key values. Therefore, one thing that attacker can supposed to do is, for example, try all the combinations of 2 ^ 128 possibilities which is also known as brute force search.

Here the attacker does not exploit the characteristic feature of a cryptographic algorithmic rule, but instead just searches all the keys which are possible. 2 ^ 128 is a large number, which is more than particles present in this universe. Hence, it is extremely difficult as a practical attacker to search for such large number of keys. The objective of cryptographic algorithmic rule is to assure that an attack does not exist in universe, which is good or better than brute force search. Now, for example, for a 128 bit key, if an attack has developed against a cryptographic algorithmic rule, which takes 2 ^ 127 searches. So, in technically language, it is still an attack which is not practical, but still we will classify it as an attack, and look as cryptographic algorithmic rule which is technically exposed.

3.3 Cryptographic Attack

All attacks are typically distinguishers. Meaning of distinguishers is that all ciphers, which are good, translate the plaintext to something which appears as random. For example, we consider a normal cryptographic algorithmic rule and put it over an alphabetic text, for instance, English text as the plaintext and apply cryptographic algorithmic rule to get cipher text.Distribution of English language is a peculiar distribution. For example, “e” is one of the most common letters in English literature. These properties exist in languages that attacker exploit.

Therefore, for designing a good cryptographic algorithm, we have to take attacks into consideration. The goal of a strong cipher is to make distribution look as random as possible to an attacker or any other who is very keen in observing output. For example, in a plaintext, there is a distinct distribution. Then, the goal of ciphering algorithmic rule is to make this distribution partially or completely lost or we can say that the distribution of cipher text should appear random, but ciphering algorithmic rule consists of sequence of mathematical steps, so it cannot be random, it can at best be made as pseudo random. Hence, definitely it is difficult to differentiate from random, but it is never be random.

CHAPTER 4

CRYPTANALYSIS

4.1 Definition

Cryptanalysis is a field of cryptology which tries to find the value of key but only finding the value of key is not sufficient, it should be typically better than the brute force search. It tries to evolve methods to obtain the secret key which is better than a brute force search.

4.2 Cryptanalysis Models

Cipher text only attack: -The opponent have a string of cipher text which means that opponent (the attacker) has access to cipher text only to deduce the key.

Known plaintext attack: -The opponent has the plaintext and also has the corresponding cipher text. This strategy is more relevant in respect to asymmetric ciphers.

Chosen plaintext attack: -Opponent has the facility of choosing plaintext and can obtain the respective cipher text. Here the attacker not only knows plaintext but he/she can also choose plaintext.

Chosen cipher text attack: -Here attacker has got temporary access to decryption function. So he/she can choose cipher text and also decrypt it to get the corresponding plaintexts, but one thing should be kept in mind that in chosen cipher text an opponent is given a huge number of cipher text who using its temporary decryption function will obtain respective plaintext.

Boolean function is very important part which is extensively used inside substitution box of many cryptographic algorithms like AES. Boolean function typically maps multiple input bits to one of the output bit. Boolean function is not only employs mapping phenomena but successfully implements cryptographic methods like confusion and diffusion as described earlier effectively. It is also used in key generation for stream cipher. Hash f (n) also uses the same in round f (n). There are large number of properties of Boolean f(n) like non-linearity, correlation immunity etc. which makes algorithm resists against cryptographic attacks.

CHAPTER 5

ANALYSIS OF S – BOX

5.1 Heuristic S-box Design

5.1.1 Research in S-Box Design

Cryptanalysis is a strong way against encryption algorithmic rule. It takes benefit of loop-hole present in Heuristic Substitution box design f(n) of the algorithmic rule, which is generally present in the s-box characteristics. There are good techniques for s-boxes generation and which have characteristics with a high degree of resistance to cryptanalysis and also with a low level of complexity with a very short generation process with such techniques we can design new encryption algorithms that can use readymade s-boxes with desirable characteristics.

5.1.2 Desirable Substitution box Characteristics

1. Strict Avalanche Criteria

Degree of dependence of output bits on input bits should be high through encryption algorithmic rule. This is basic action and should be present in every S-Box. As per SAC, change in one input bit will result in the change in approx. half of the output bits. [4]

2. Non-linear

Substitution box cannot be expressed as the linear input function. It will make the algorithmic rule to be divided, solving a set of unknowns for set of equations, which is easy. [3]

3. Differential Cryptanalysis (DC)

DC is a type of attack which employs the feature of a substitution box i.e. DDT (is a table which indicates how the s-box output varies when the input is varied) which reduce the hardness and complexity of a brute force attack to great extent. [2]

4. Dynamic Nature

AES S-Box is not changeable. If substitution box is static, it will use identical substitution box in every round while dynamic substitution box changes which is key-dependent and also depends on no. of rounds. Algorithm based on dynamic substitution box be given preference to enhance the strength of any cryptographic algorithm. Key dependent S-Boxes are comparatively slower but are more secure against cryptanalysis. AES has not completely broken till now but cryptanalysis on that are going on continuously. [1]

5. Differential Uniformity

Relation between differential uniformity and cryptanalysis is that larger the Differential Uniformity, poorer is the resistance of substitution box against it.

6. Linear Approximation

Relation between differential uniformity and cryptanalysis is that higher the Linear Approximation, poorer is the resistance of substitution box against it.

7. Algebraic Complexity

It is important against attacks such as interpolation other related algebraic attacks. Relation between Algebraic Complexity and cryptanalysis is that higher the Algebraic Complexity, better is the resistance of substitution box against it.

8. Fixed and Opposite Fixed Points

Statistic leakage should be minimized by keeping above points i.e. fixed and opposite points as slow as possible in cryptanalysis. [4]

9. Bit independence criterion

It should be minimized as far as possible because on increasing bits independence, complexity of system design and hence understanding difficulty level of it also increases. [4]

CHAPTER 6

RELATED WORK

6.1 AES Substitution box as 8 dimensional vector valued Boolean f (n)

The Explicit algebraic representation will say about correlation of bits in S - Box when we used it in the block cipher. The algebraic structure of encryption algorithmic rule can help in analysis the difference of the output of encryption rule and random output (true). Above deflections can be seen as a good research area when we have to build the algebraic distinguisher.

ALGORITHM DERIVED:-

1. Input Truth Table Of AES (RIJNDAEL) S-Box (Decimal Form or converted into decimal Form)
2. Convert Truth Table into Bits (Binary Form).
3. Obtained result of Step 2 should be converted into its ANF form to get eight dimensional vector valued Boolean function of AES S-Box.

6.2 Deviations of AES S-box from random permutations

To measure the deflection and difference of the output of encryption rule and random output (true), we have to generate ten thousand permutations (random) of 2^8 elements. Permutations (random) can be generated through any software package. Side by Side, numbers of terms (used) in eight vectors are obtained.

In [7], a method is devised to measure the deflection in output of encryption rule and random output (true). In the method, from the AES S-Box truth table, eight dimensional vector valued Boolean function of AES S-Box is obtained. Through it we can calculate numbers of used terms in all 8 coordinates of AES S-Box.Truth Table of AES S-Box (Decimal Form or converted into decimal Form) are converted into Bits (Binary Form).After that, it will be converted into its ANF form to get eight dimensional vector valued Boolean function of AES S-Box.

ALGORITHM DERIVED :-

1. Random Permutation of 256 elements are generated 10000 times.
2. Each Random Permutation are converted into bits and then into its ANF form.
3. Each random generated ANF permutation are compared with AES S-Box (ANF Form) bitwise.
4. Number of bit wise matching of all S-Box are stored in arrays e.g. B0, B1 etc.
5. Similarly, bit wise matching of their intersection e.g. B0 n B1 are also stored for analysis purpose.
6. Mean of bit wise matching of 10000 random generated ANF permutation are compared with that of AES S- Box

We can also generate large number of random numbers through any random number generator. In [7], author has supposed to generate random permutation of 256 elements 10000 times. Each Random Permutation are converted into bits and then into its ANF form same as we have done above with AES S-Box previously.

At the end we have to compare above two values i.e. Each random generated ANF permutation are compared with AES S-Box (ANF Form) bitwise. Number of bit wise matching of all 10000 Random

Permutation with that AES S-Box are stored in arrays e.g. B0, B1 etc. Similarly, bit wise matching of their intersection e.g. B0 and B1 , B0 and B1 and B3 are also stored. Now for analysis, deviation of mean of bit wise matching of 10000 random generated ANF permutation are compared with that of AES Substitution Box.

These deflections can be used by attacker for cryptanalysis by building the algebraic distinguisher. Thus, we can use above measurement to devise a substitution box with a least deviation from randomly generated permutation. Hence, above mechanism can be used to have a Substitution Box which is least prone to cryptanalysis.

CHAPTER 7

PROBLEM STATEMENT AND PROPOSED METHOD

The goal of a strong cipher is to make distribution look as random as possible to an attacker or any other who is very keen in observing output. For example, if in a plaintext, there is a distinct distribution. Then, the goal of ciphering algorithmic rule is to make this distribution partially or completely lost or we can say that the distribution of cipher text should appear random, but ciphering algorithmic rule consists of sequence of mathematical steps, so it cannot be random. Hence, it is susceptible to cryptanalysis.

Thus, our aim is to search all the 4 by 4 S-Box and get the best S-Box whose number of terms in each vector is as good as mean of large number of randomized S-Box. For the above aim, we have to generate all the possible permutation of 4 by 4 S-Boxes and then compare the number of terms in each permuted S-Box with the mean of 10000 random permutations (For example). S-Box with the least deviation from mean will be our desired S-Box. In other words, it will be the S-Box which is closest to the group of random S-Box. This S-Box will be least susceptible to cryptanalysis.

<u>PROPOSED ALGORITHM :-</u>

1. One by One, generate all the possible permutation of 4 by 4 S-Boxes.
2. Each generated permutation will be converted in Binary Form.
3. It is then changed into ANF form to get eight dimensional vector valued Boolean function of permuted 4 by 4 S-Box.

4. Side by Side Random Permutation of 16 elements are generated 10000 times and stored in array for greater efficiency.
5. Each Random Permutation of Step – 4 are converted into bits and then into its ANF form.
6. Each random generated ANF permutation are compared with that of original 4 by 4 S-Box (ANF Form) bitwise.
7. Number of bit wise matching of all S-Box are stored in arrays e.g. B0, B1 etc.
8. Similarly, bit wise matching of their intersection e.g. B0 and B1 are also stored for analysis purpose.
9. Mean of bit wise matching of 10000 random generated ANF permutations are compared with that of permuted 4 by 4 S-Boxes.
10. Each comparison (difference in mean) is stored in an array and heuristically searches for the best S-Box with least mean difference.

CHAPTER 8

WORK DONE SO FAR

8.1 Implementation

Following is the algorithm for Implementation of AES S-Box deviation from truly random source

8.1.1 Implementation of AES S-Box deviation from truly random source :

1 Start with Truth Table Of AES S-Box.

2 Convert Truth Table first into decimal form and then into Bits (Binary Form).

3 Result of Step 2 should be converted into its ANF by TT_ANF () function to get eight dimensional vector valued Boolean function of AES S-Box.

4 Random Permutation of 256 elements are generated 10000 times by randperm() function.

5 Each Random Permutation are converted into bits and then into its ANF form.

6 Each random generated ANF permutation is compared with AES S-Box (ANF Form) bitwise.

7 Number of bit wise matching of all S-Box are stored in arrays e.g. B0, B1 etc.

8 Similarly, bit wise matching of their intersection e.g. B0 n B1 are also stored for analysis purpose.

9 Mean of bit wise matching of 10000 random generated ANF permutations are compared with that of AES S- Box.

10 Various Graphs are drawn using Matlab inbuilt function “plot” for representation and analysis purposes.

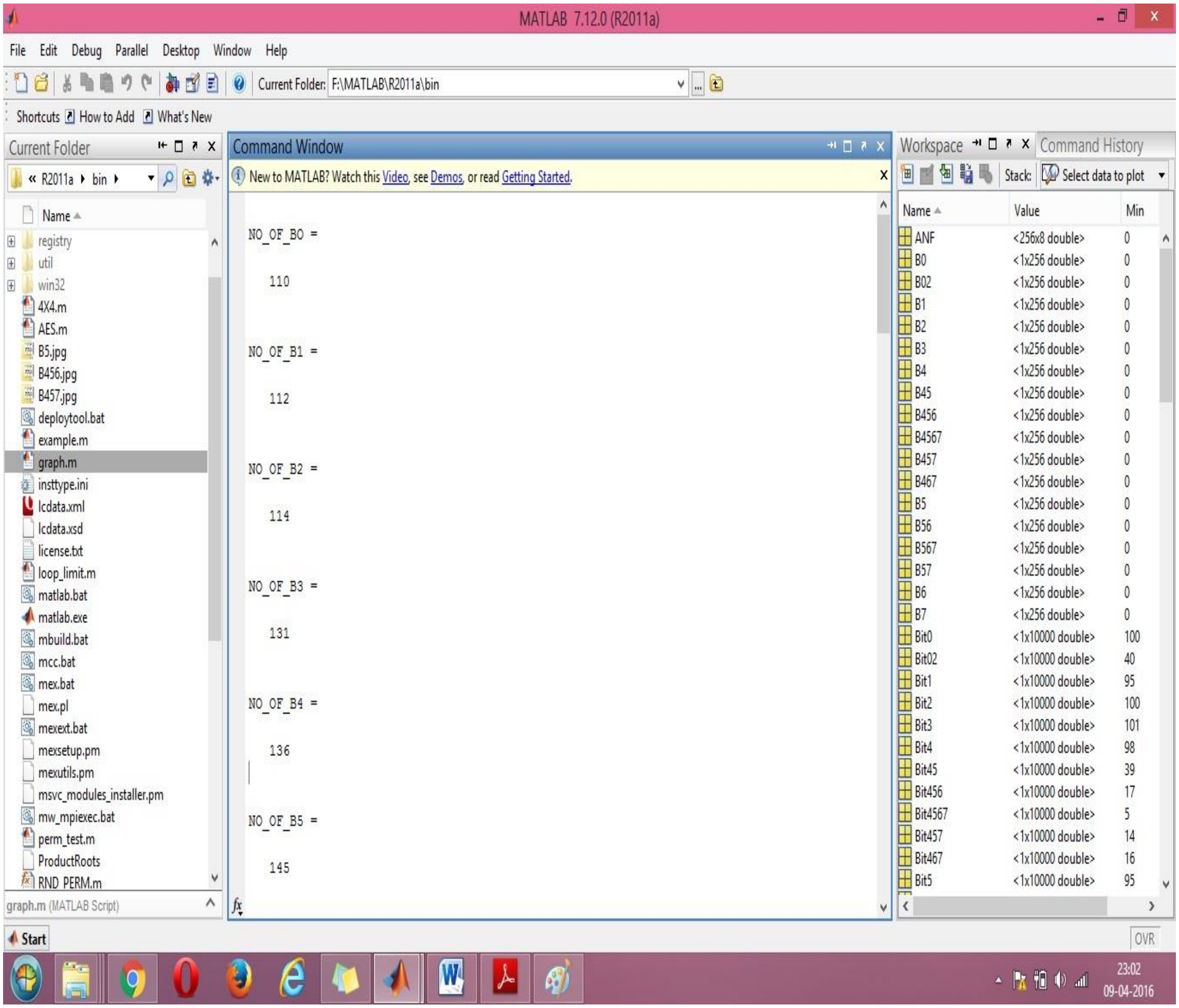
8.1.2 Implementation of Deriving Least Deviated 4-by-4 Substitution Box Among all possible permutation of 16 elements:

1. One by One, generate all the possible permutation of 4 by 4 S-Boxes.
2. Each generated permutation will be converted in Binary Form.
3. It is then changed into ANF form to get eight dimensional vector valued Boolean function of permuted 4 by 4 S-Box.
4. Side by Side Random Permutation of 16 elements are generated 10000 times and stored in array for greater efficiency.
5. Each Random Permutation of Step – 4 are converted into bits and then into its ANF form.
6. Each random generated ANF permutation are compared with that of original 4 by 4 S-Box (ANF Form) bitwise.
7. Number of bit wise matching of all S-Box are stored in arrays e.g. B0, B1 etc.
8. Similarly, bit wise matching of their intersection e.g. B0 and B1 are also stored for analysis purpose.
9. Mean of bit wise matching of 10000 random generated ANF permutations are compared with that of permuted 4 by 4 S-Boxes.
10. Comparison (difference in mean) of intersection of all bits is stored in an array.
11. Simultaneously, it heuristically searches for the best S-Box with least mean difference.

It has been found through mathematical calculation that greater is the intersection of the bits, larger is the deviation. Hence, in above algorithm instead of finding deviation in each bit separately and then comparing them, we have found out mean deviation of all deviation and then heuristically searches for best S-Box. This algorithm not only reduces the complexity of the program but also found out the result in less time. Next we will show about experiment result achieved after implementing above algorithms in Matlab.

8.2 Experiment Result

8.2.1 Experiment Result of AES S-Box deviation


MATLAB 7.12.0 (R2011a)
File Edit Debug Parallel Desktop Window Help
Current Folder: F:\MATLAB\R2011a\bin
Shortcuts How to Add What's New
Current Folder
Command Window
Workspace
Command History
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Stack:
Select data to plot
Start
OVR
23:02
09-04-2016

```
NO_OF_B0 =

   110


NO_OF_B1 =

   112


NO_OF_B2 =

   114


NO_OF_B3 =

   131


NO_OF_B4 =

   136


NO_OF_B5 =

   145
```

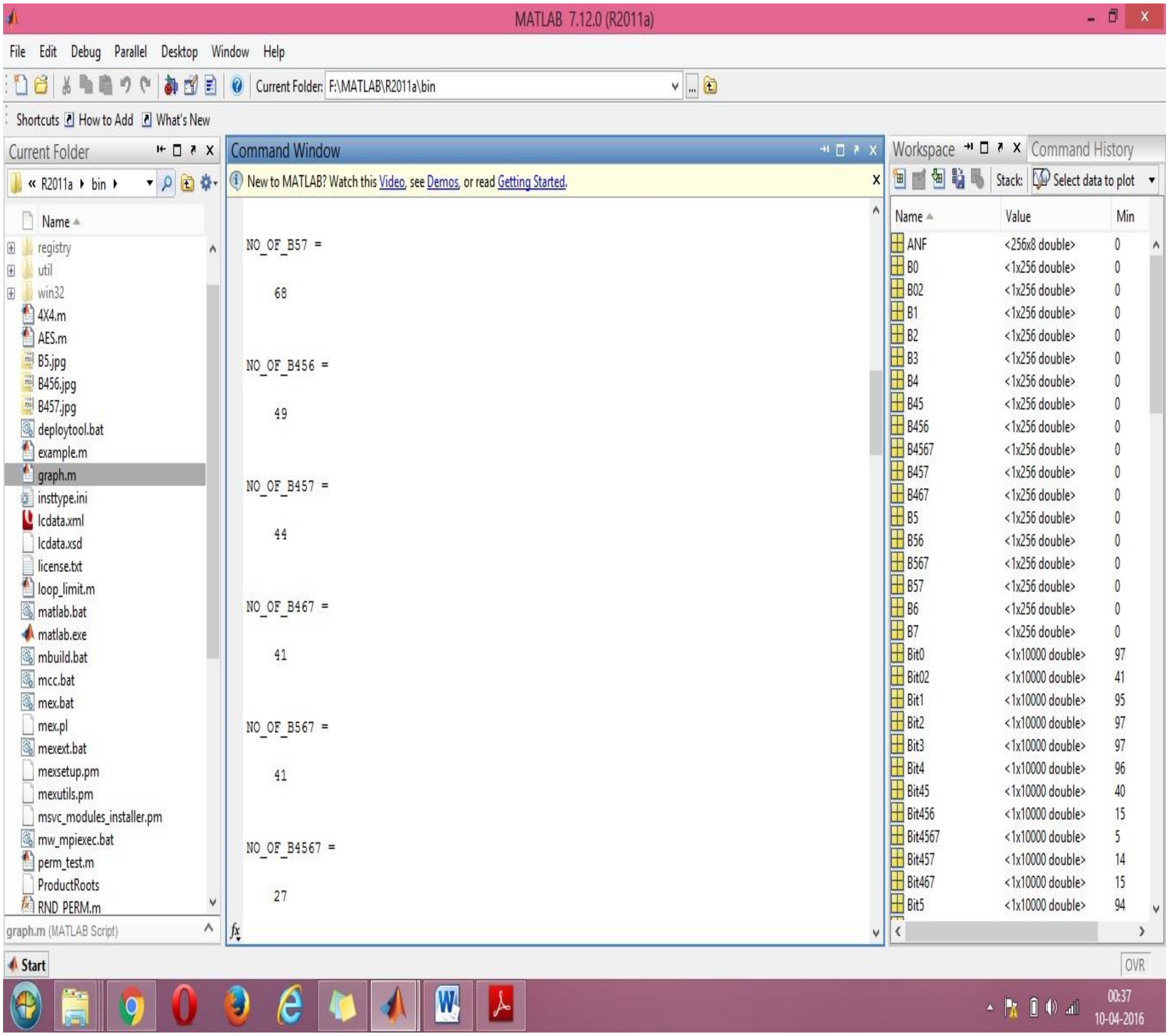
R2011a ▸ bin ▸

Name ▲
registry
util
win32
4X4.m
AES.m
B5.jpg
B456.jpg
B457.jpg
deploytool.bat
example.m
graph.m
insttype.ini
lcdata.xml
lcdata.xsd
license.txt
loop_limit.m
matlab.bat
matlab.exe
mbuild.bat
mcc.bat
mex.bat
mex.pl
mexext.bat
mexsetup.pm
mexutils.pm
msvc_modules_installer.pm
mw_mpiexec.bat
perm_test.m
ProductRoots
RND_PERM.m

graph.m (MATLAB Script)

Name ▲	Value	Min
ANF	<256x8 double>	0
B0	<1x256 double>	0
B02	<1x256 double>	0
B1	<1x256 double>	0
B2	<1x256 double>	0
B3	<1x256 double>	0
B4	<1x256 double>	0
B45	<1x256 double>	0
B456	<1x256 double>	0
B4567	<1x256 double>	0
B457	<1x256 double>	0
B467	<1x256 double>	0
B5	<1x256 double>	0
B56	<1x256 double>	0
B567	<1x256 double>	0
B57	<1x256 double>	0
B6	<1x256 double>	0
B7	<1x256 double>	0
Bit0	<1x10000 double>	100
Bit02	<1x10000 double>	40
Bit1	<1x10000 double>	95
Bit2	<1x10000 double>	100
Bit3	<1x10000 double>	101
Bit4	<1x10000 double>	98
Bit45	<1x10000 double>	39
Bit456	<1x10000 double>	17
Bit4567	<1x10000 double>	5
Bit457	<1x10000 double>	14
Bit467	<1x10000 double>	16
Bit5	<1x10000 double>	95

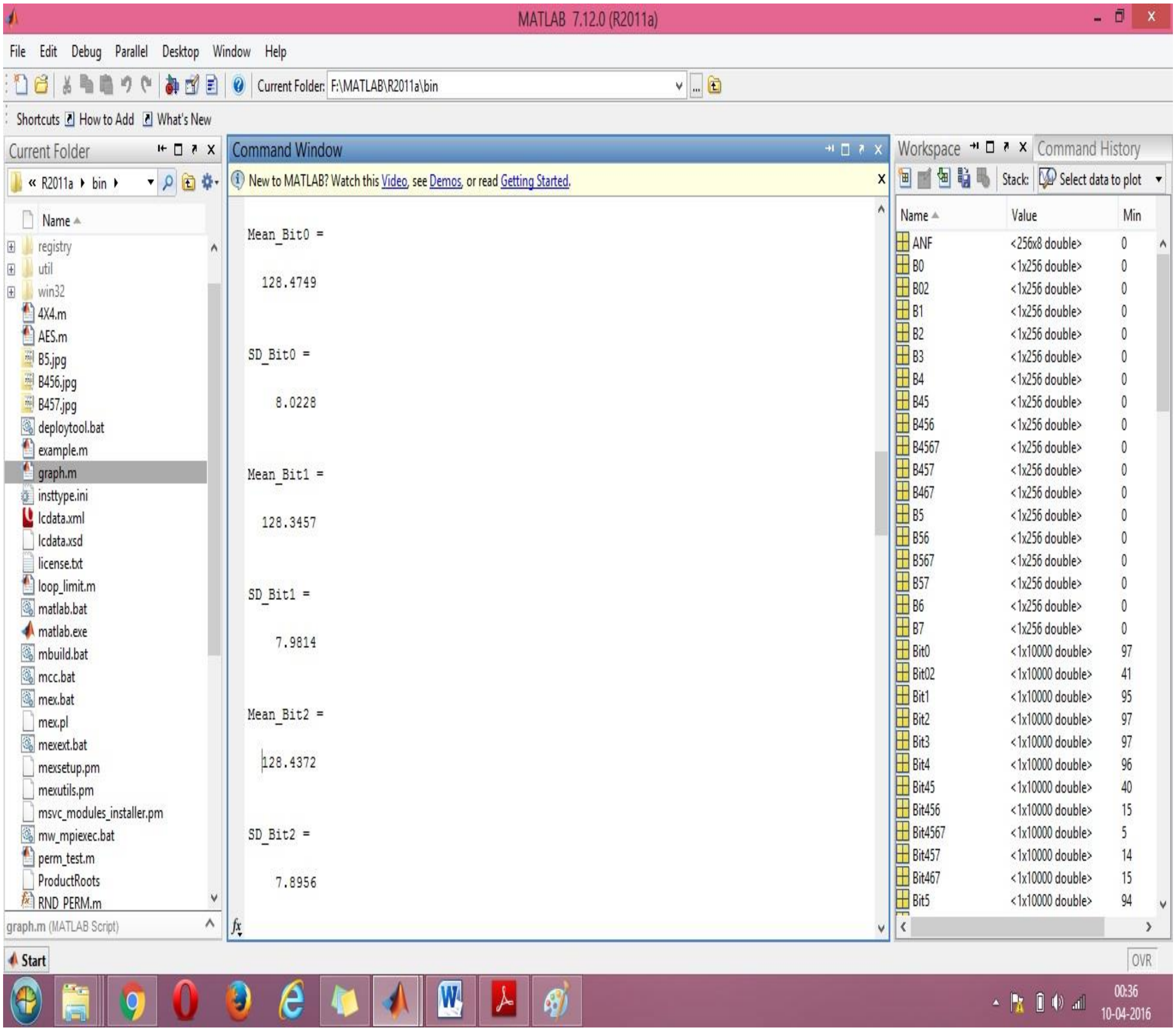
Figure 1 : No. of AES used terms in B0, B1, B2 etc.



```
NO_OF_B57 =

    68


NO_OF_B456 =

    49


NO_OF_B457 =

    44


NO_OF_B467 =

    41


NO_OF_B567 =

    41


NO_OF_B4567 =

    27
```

Name	Value	Min
ANF	<256x8 double>	0
B0	<1x256 double>	0
B02	<1x256 double>	0
B1	<1x256 double>	0
B2	<1x256 double>	0
B3	<1x256 double>	0
B4	<1x256 double>	0
B45	<1x256 double>	0
B456	<1x256 double>	0
B4567	<1x256 double>	0
B457	<1x256 double>	0
B467	<1x256 double>	0
B5	<1x256 double>	0
B56	<1x256 double>	0
B567	<1x256 double>	0
B57	<1x256 double>	0
B6	<1x256 double>	0
B7	<1x256 double>	0
Bit0	<1x10000 double>	97
Bit02	<1x10000 double>	41
Bit1	<1x10000 double>	95
Bit2	<1x10000 double>	97
Bit3	<1x10000 double>	97
Bit4	<1x10000 double>	96
Bit45	<1x10000 double>	40
Bit456	<1x10000 double>	15
Bit4567	<1x10000 double>	5
Bit457	<1x10000 double>	14
Bit467	<1x10000 double>	15
Bit5	<1x10000 double>	94

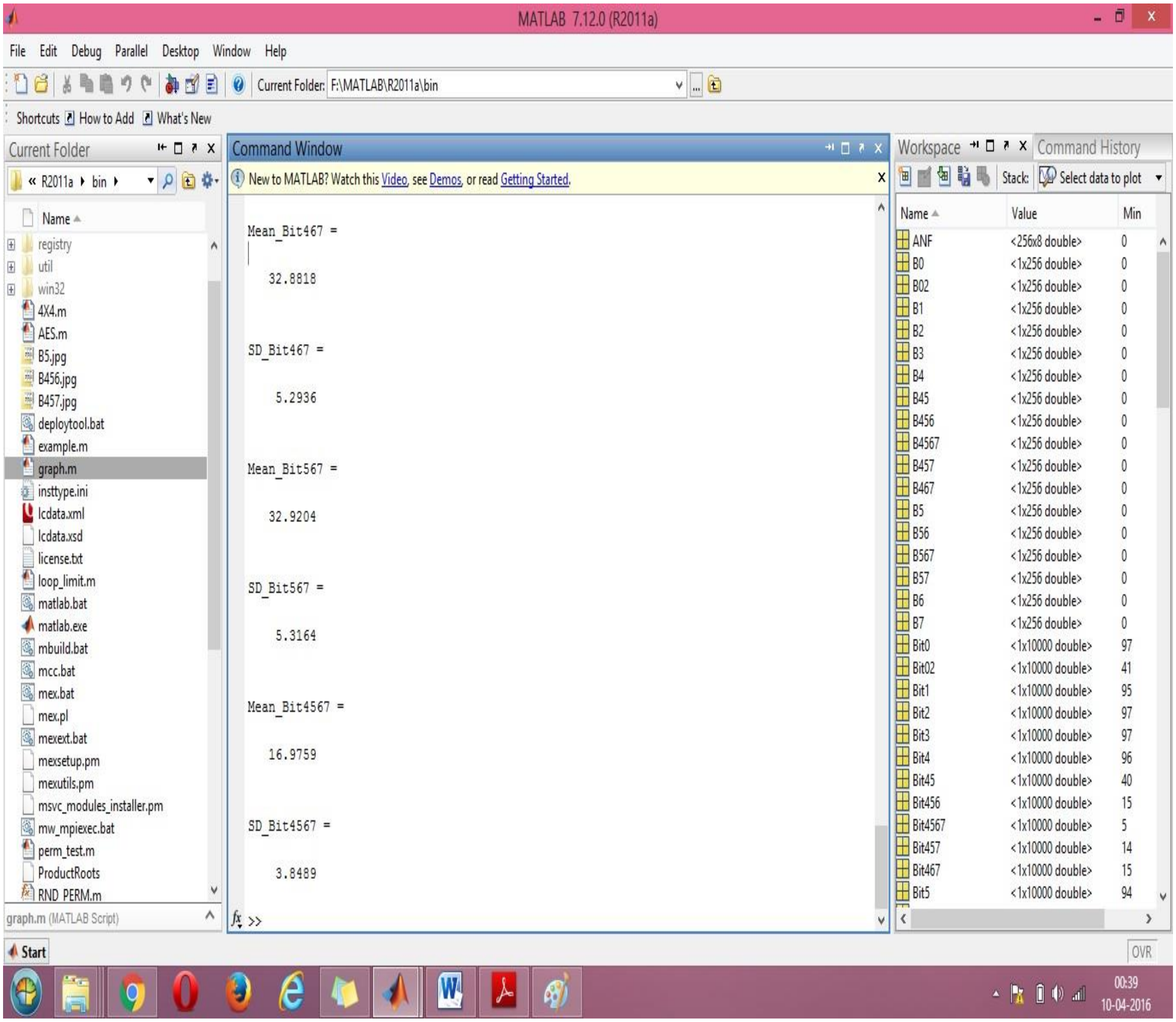
Figure 2 : No. of AES used terms in intersection of bits i.e. B01, B57 etc.



```
Mean_Bit0 =

  128.4749


SD_Bit0 =

    8.0228


Mean_Bit1 =

  128.3457


SD_Bit1 =

    7.9814


Mean_Bit2 =

  128.4372


SD_Bit2 =

    7.8956
```

Name	Value	Min
ANF	<256x8 double>	0
B0	<1x256 double>	0
B02	<1x256 double>	0
B1	<1x256 double>	0
B2	<1x256 double>	0
B3	<1x256 double>	0
B4	<1x256 double>	0
B45	<1x256 double>	0
B456	<1x256 double>	0
B4567	<1x256 double>	0
B457	<1x256 double>	0
B467	<1x256 double>	0
B5	<1x256 double>	0
B56	<1x256 double>	0
B567	<1x256 double>	0
B57	<1x256 double>	0
B6	<1x256 double>	0
B7	<1x256 double>	0
Bit0	<1x10000 double>	97
Bit02	<1x10000 double>	41
Bit1	<1x10000 double>	95
Bit2	<1x10000 double>	97
Bit3	<1x10000 double>	97
Bit4	<1x10000 double>	96
Bit45	<1x10000 double>	40
Bit456	<1x10000 double>	15
Bit4567	<1x10000 double>	5
Bit457	<1x10000 double>	14
Bit467	<1x10000 double>	15
Bit5	<1x10000 double>	94

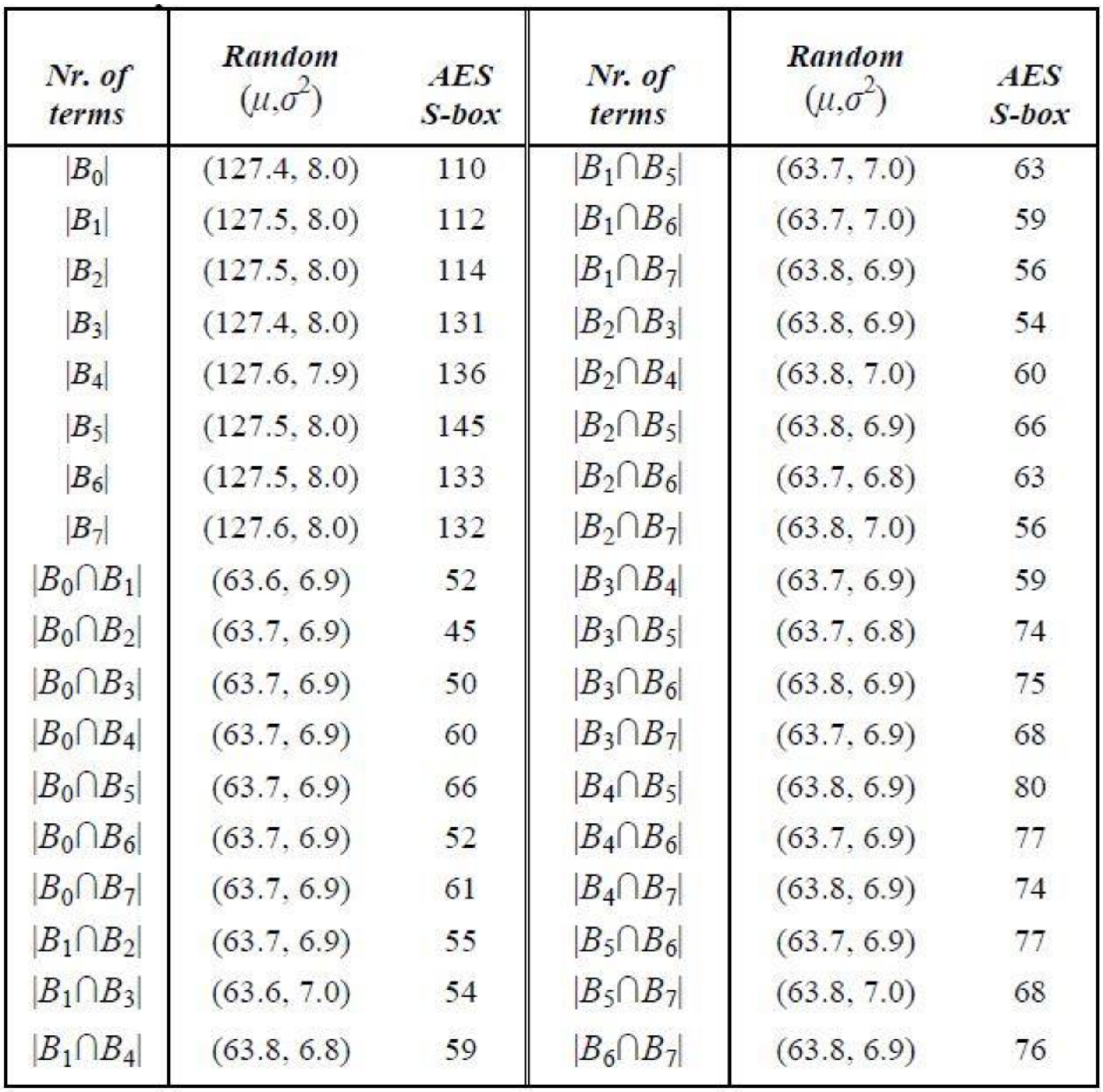
Figure 3 : Mean and SD of Random Permutations when compared with AES S-Box (single bit)



```
Mean_Bit467 =

   32.8818


SD_Bit467 =

    5.2936


Mean_Bit567 =

   32.9204


SD_Bit567 =

    5.3164


Mean_Bit4567 =

   16.9759


SD_Bit4567 =

    3.8489

fx >>
```

Name	Value	Min
ANF	<256x8 double>	0
B0	<1x256 double>	0
B02	<1x256 double>	0
B1	<1x256 double>	0
B2	<1x256 double>	0
B3	<1x256 double>	0
B4	<1x256 double>	0
B45	<1x256 double>	0
B456	<1x256 double>	0
B4567	<1x256 double>	0
B457	<1x256 double>	0
B467	<1x256 double>	0
B5	<1x256 double>	0
B56	<1x256 double>	0
B567	<1x256 double>	0
B57	<1x256 double>	0
B6	<1x256 double>	0
B7	<1x256 double>	0
Bit0	<1x10000 double>	97
Bit02	<1x10000 double>	41
Bit1	<1x10000 double>	95
Bit2	<1x10000 double>	97
Bit3	<1x10000 double>	97
Bit4	<1x10000 double>	96
Bit45	<1x10000 double>	40
Bit456	<1x10000 double>	15
Bit4567	<1x10000 double>	5
Bit457	<1x10000 double>	14
Bit467	<1x10000 double>	15
Bit5	<1x10000 double>	94

Figure 4 : Mean and SD of Random Permutations when compared with AES S-Box (intersection of bits)



Nr. of terms	*Random* (μ,σ^2)	*AES S-box*	*Nr. of terms*	*Random* (μ,σ^2)	*AES S-box*
$\lvert B_0\rvert$	(127.4, 8.0)	110	$\lvert B_1\cap B_5\rvert$	(63.7, 7.0)	63
$\lvert B_1\rvert$	(127.5, 8.0)	112	$\lvert B_1\cap B_6\rvert$	(63.7, 7.0)	59
$\lvert B_2\rvert$	(127.5, 8.0)	114	$\lvert B_1\cap B_7\rvert$	(63.8, 6.9)	56
$\lvert B_3\rvert$	(127.4, 8.0)	131	$\lvert B_2\cap B_3\rvert$	(63.8, 6.9)	54
$\lvert B_4\rvert$	(127.6, 7.9)	136	$\lvert B_2\cap B_4\rvert$	(63.8, 7.0)	60
$\lvert B_5\rvert$	(127.5, 8.0)	145	$\lvert B_2\cap B_5\rvert$	(63.8, 6.9)	66
$\lvert B_6\rvert$	(127.5, 8.0)	133	$\lvert B_2\cap B_6\rvert$	(63.7, 6.8)	63
$\lvert B_7\rvert$	(127.6, 8.0)	132	$\lvert B_2\cap B_7\rvert$	(63.8, 7.0)	56
$\lvert B_0\cap B_1\rvert$	(63.6, 6.9)	52	$\lvert B_3\cap B_4\rvert$	(63.7, 6.9)	59
$\lvert B_0\cap B_2\rvert$	(63.7, 6.9)	45	$\lvert B_3\cap B_5\rvert$	(63.7, 6.8)	74
$\lvert B_0\cap B_3\rvert$	(63.7, 6.9)	50	$\lvert B_3\cap B_6\rvert$	(63.8, 6.9)	75
$\lvert B_0\cap B_4\rvert$	(63.7, 6.9)	60	$\lvert B_3\cap B_7\rvert$	(63.7, 6.9)	68
$\lvert B_0\cap B_5\rvert$	(63.7, 6.9)	66	$\lvert B_4\cap B_5\rvert$	(63.8, 6.9)	80
$\lvert B_0\cap B_6\rvert$	(63.7, 6.9)	52	$\lvert B_4\cap B_6\rvert$	(63.7, 6.9)	77
$\lvert B_0\cap B_7\rvert$	(63.7, 6.9)	61	$\lvert B_4\cap B_7\rvert$	(63.8, 6.9)	74
$\lvert B_1\cap B_2\rvert$	(63.7, 6.9)	55	$\lvert B_5\cap B_6\rvert$	(63.7, 6.9)	77
$\lvert B_1\cap B_3\rvert$	(63.6, 7.0)	54	$\lvert B_5\cap B_7\rvert$	(63.8, 7.0)	68
$\lvert B_1\cap B_4\rvert$	(63.8, 6.8)	59	$\lvert B_6\cap B_7\rvert$	(63.8, 6.9)	76

Table 3 : Summary of Mean and SD with comparison to AES used terms.

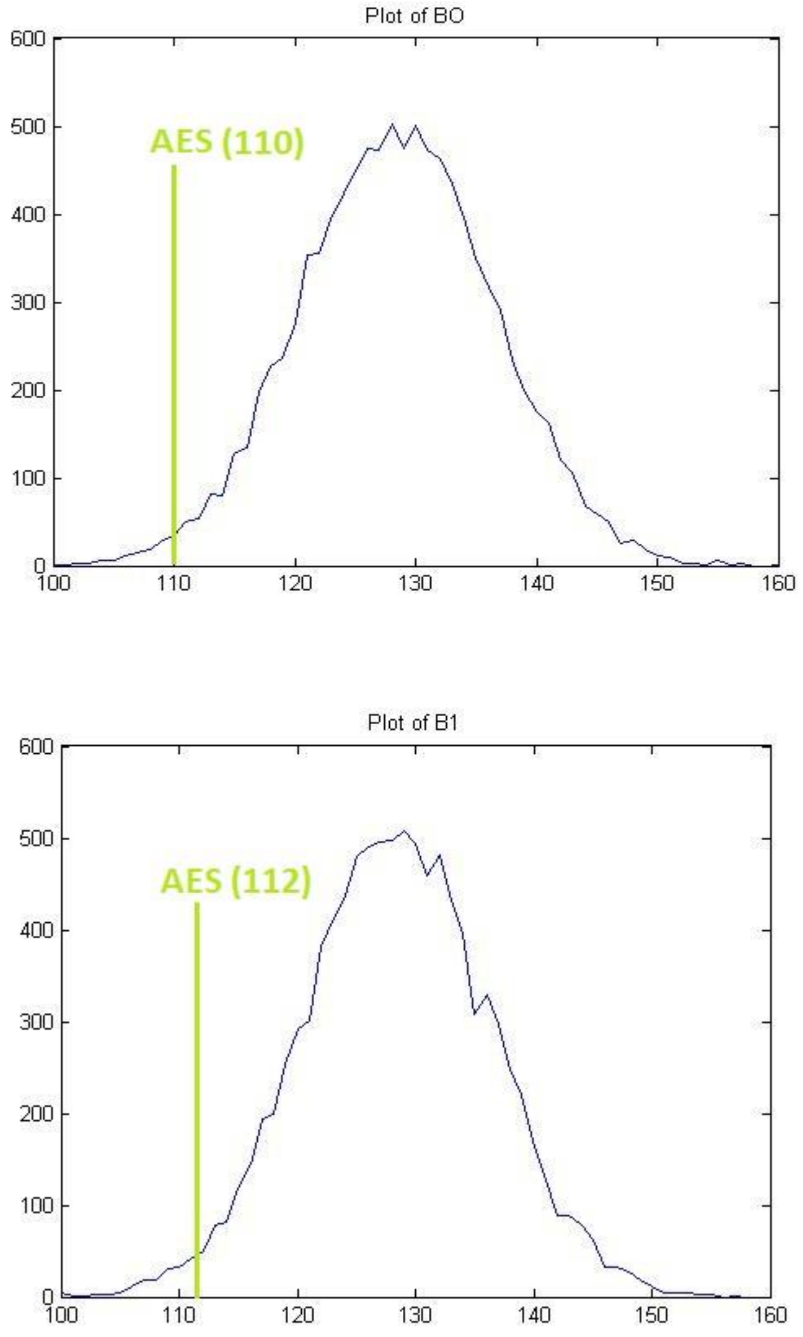


Figure 5: Distribution of matches in B (i). X-axes represent number of matches in B (i). Green line depicts AES actual no. of terms. Here i = 0, 1

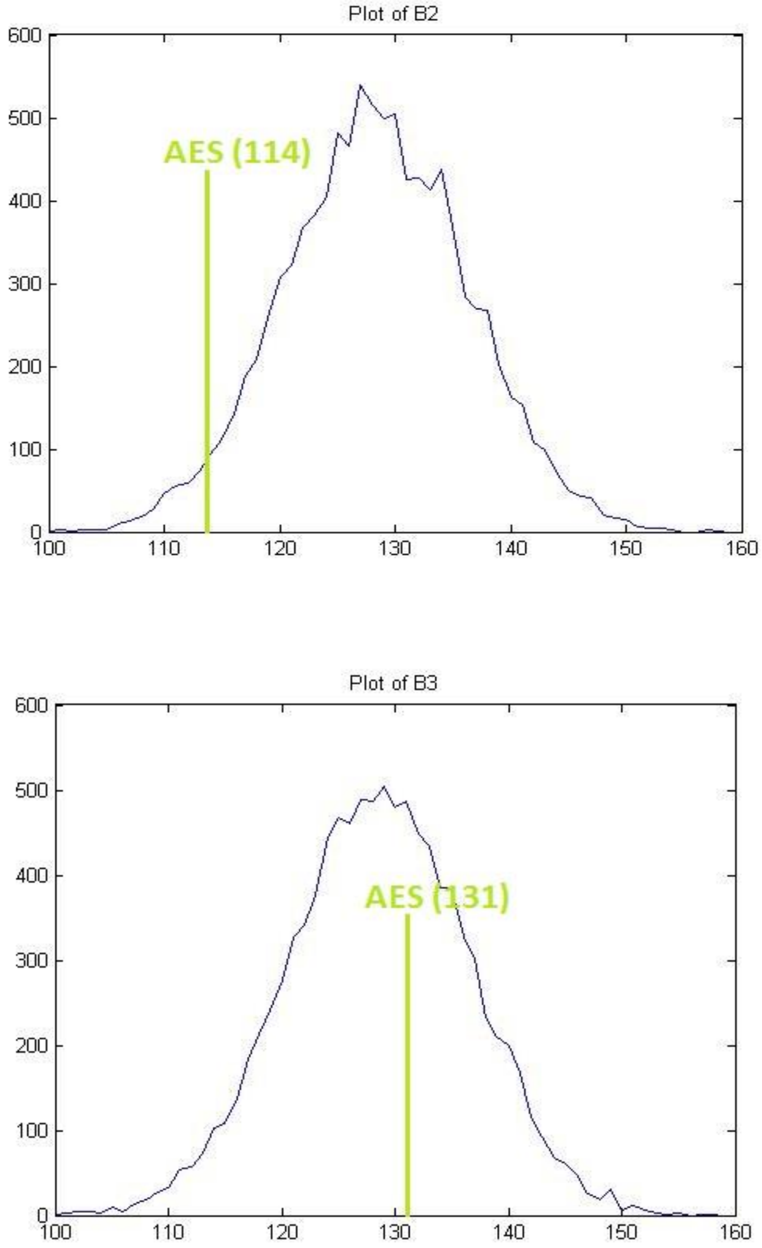


Figure 6 : Distribution of matches in B (i). X-axes represent number of matches in B (i). Green line depicts AES actual no. of terms. Here i = 2, 3

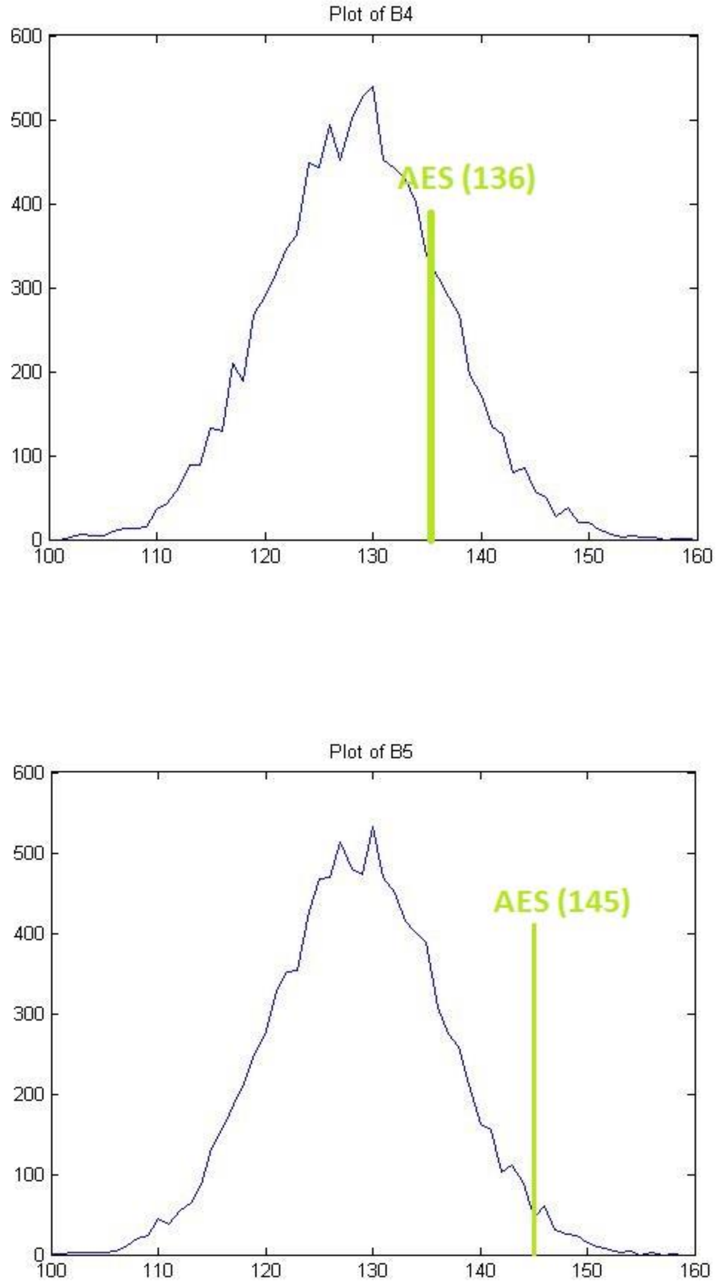
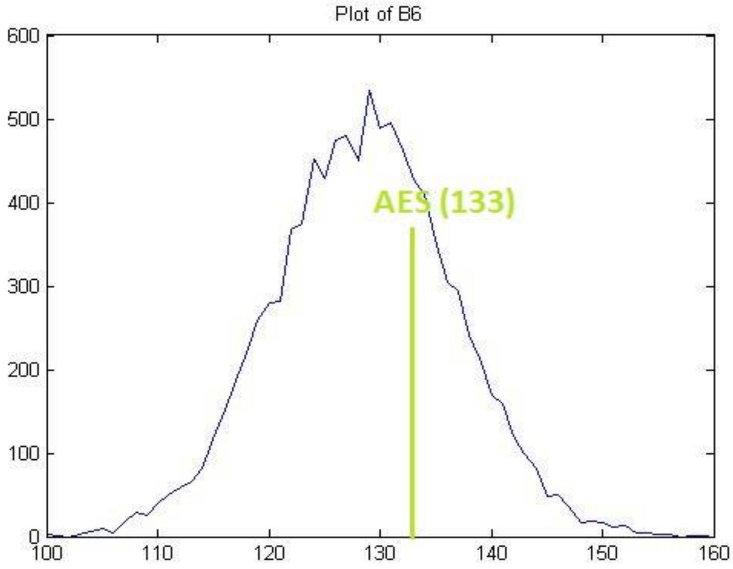


Figure 7 : Distribution of matches in B (i). X-axes represent number of matches in B (i). Green line depicts AES actual no. of terms. Here i = 4, 5



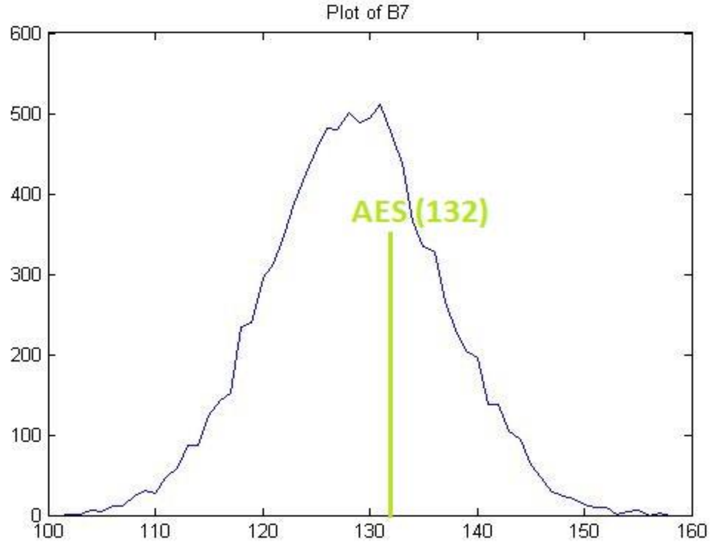
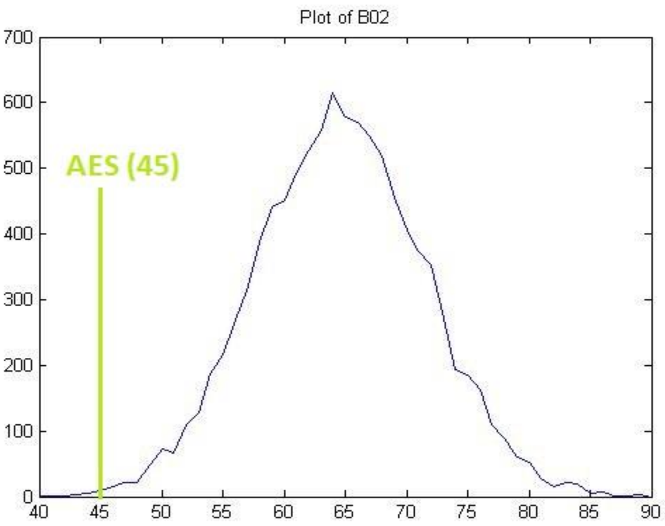


Figure 8 : Distribution of matches in B (i). X-axes represent number of matches in B (i). Green line depicts AES actual no. of terms. Here i = 6, 7



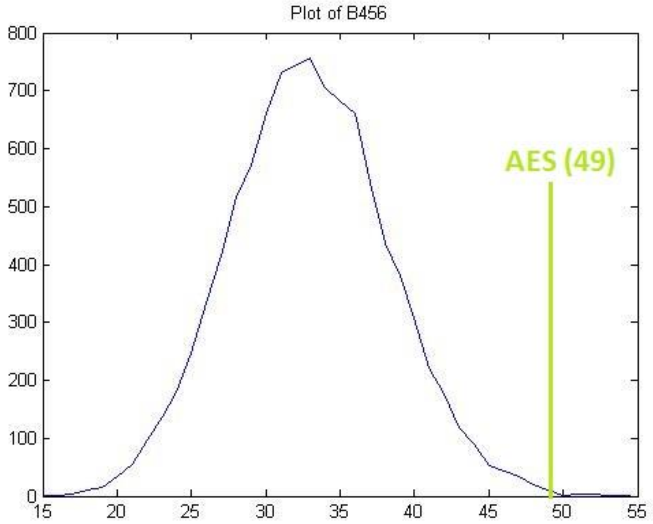


Figure 9 : Distribution of matches in B02, B456. X-axes represents number of matches in Bi. Green line depicts AES actual no. of terms.

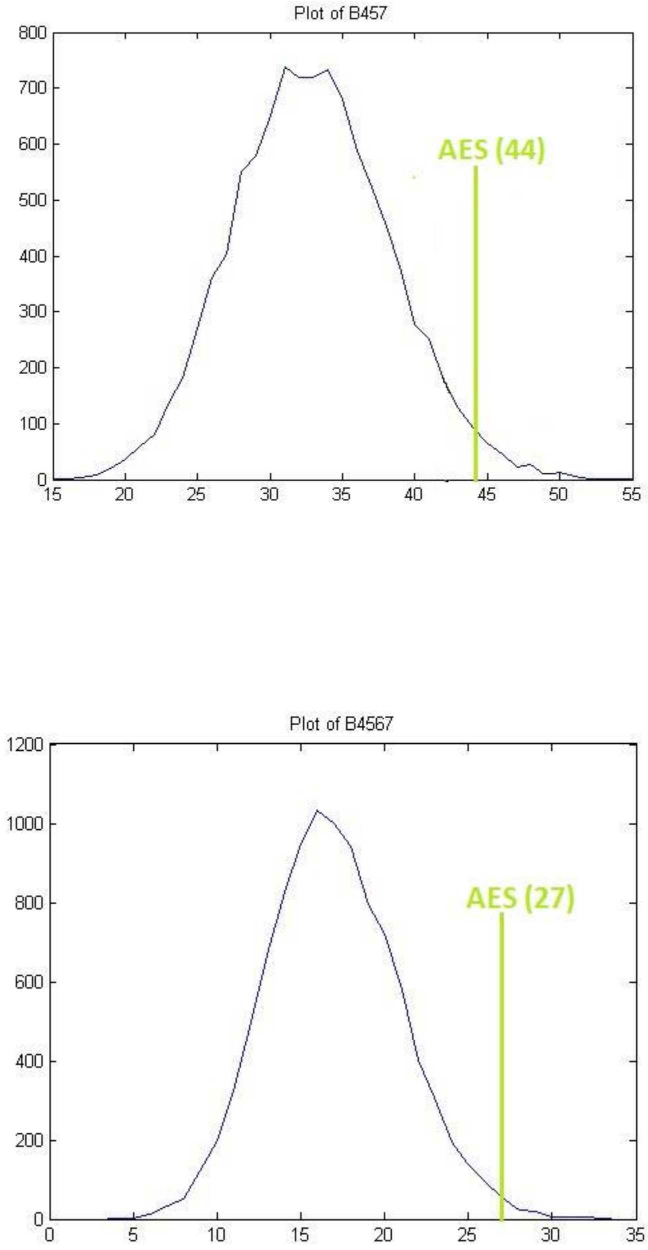
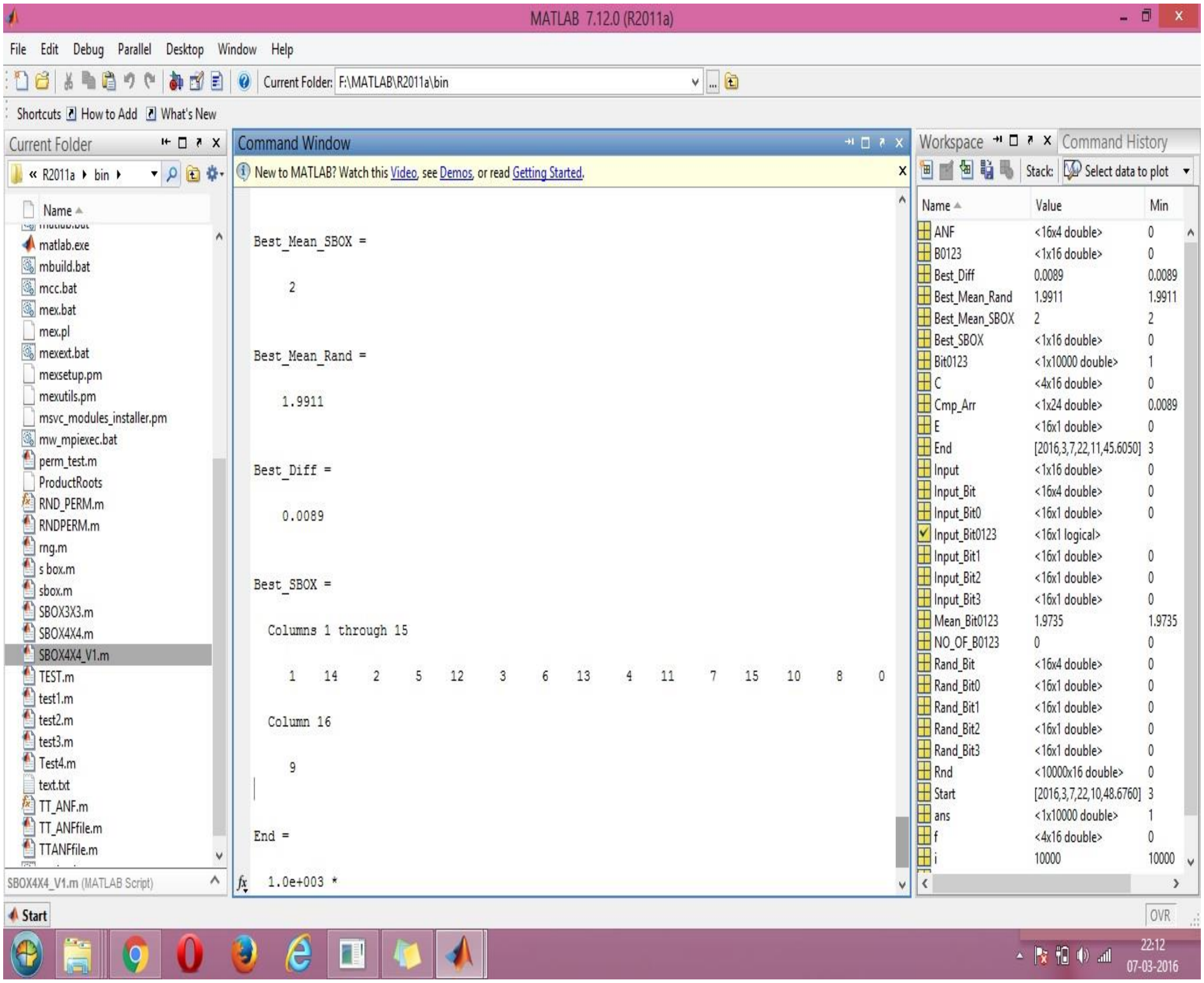


Figure 10 : Distribution of matches in B457, B4567. X-axes represent number of matches in Bi. Green line depicts AES actual no. of terms.

8.2.2 Experiment Result of least deviated 4-by-4 S-Box

After applying above algorithm to get least deviated 4-by-4 S-Box. Below are the result snapshots.

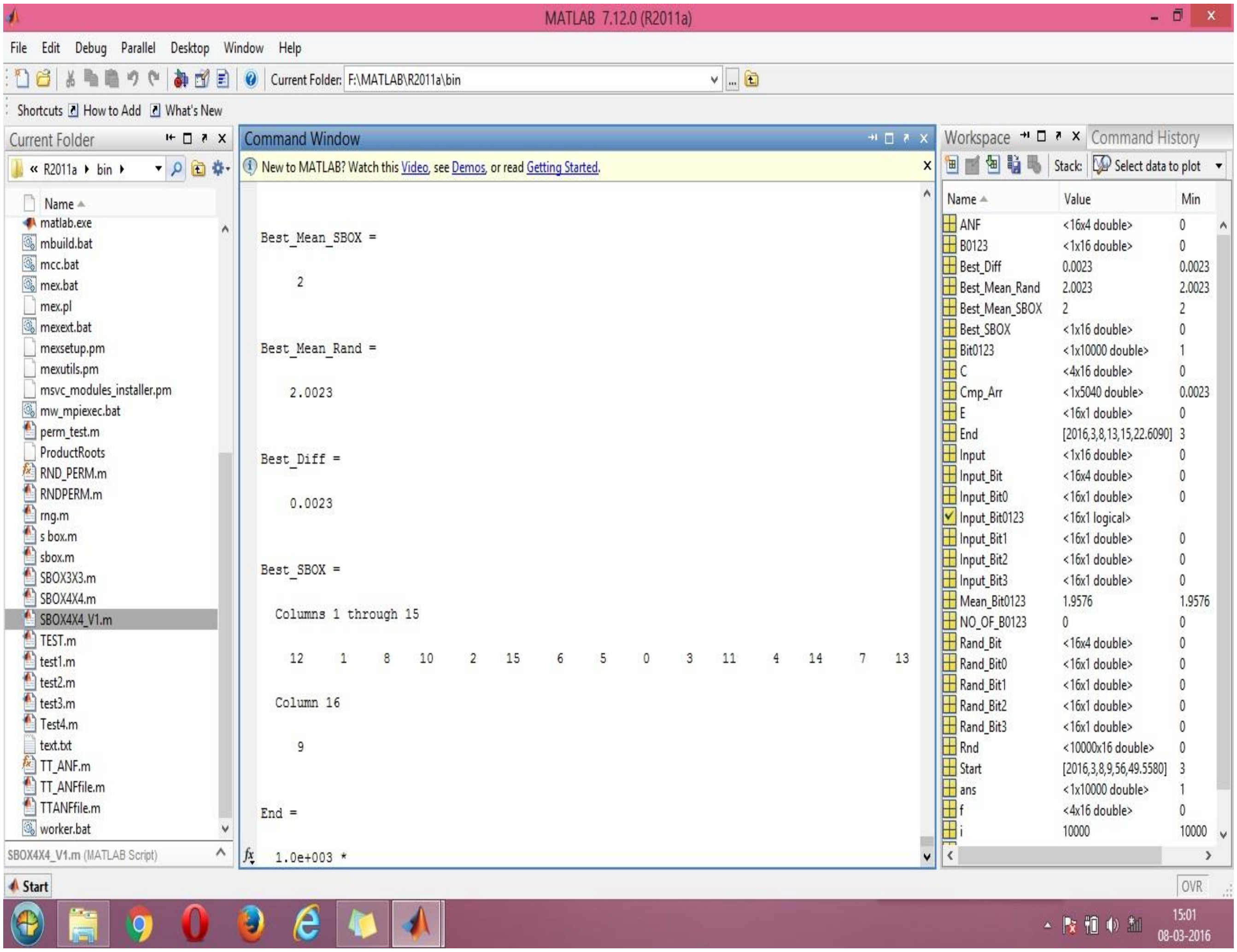


Current Folder: F:\MATLAB\R2011a\bin

```
Best_Mean_SBOX =

     2


Best_Mean_Rand =

    1.9911


Best_Diff =

    0.0089


Best_SBOX =

  Columns 1 through 15

     1    14     2     5    12     3     6    13     4    11     7    15    10     8     0

  Column 16

     9


End =

  1.0e+003 *
```

Name ▲	Value	Min
ANF	<16x4 double>	0
B0123	<1x16 double>	0
Best_Diff	0.0089	0.0089
Best_Mean_Rand	1.9911	1.9911
Best_Mean_SBOX	2	2
Best_SBOX	<1x16 double>	0
Bit0123	<1x10000 double>	1
C	<4x16 double>	0
Cmp_Arr	<1x24 double>	0.0089
E	<16x1 double>	0
End	[2016,3,7,22,11,45.6050]	3
Input	<1x16 double>	0
Input_Bit	<16x4 double>	0
Input_Bit0	<16x1 double>	0
Input_Bit0123	<16x1 logical>	
Input_Bit1	<16x1 double>	0
Input_Bit2	<16x1 double>	0
Input_Bit3	<16x1 double>	0
Mean_Bit0123	1.9735	1.9735
NO_OF_B0123	0	0
Rand_Bit	<16x4 double>	0
Rand_Bit0	<16x1 double>	0
Rand_Bit1	<16x1 double>	0
Rand_Bit2	<16x1 double>	0
Rand_Bit3	<16x1 double>	0
Rnd	<10000x16 double>	0
Start	[2016,3,7,22,10,48.6760]	3
ans	<1x10000 double>	1
f	<4x16 double>	0
i	10000	10000

Figure 11 : Result Snapshot of 4X4 S-Box when no. of permutation is factorial 8



```
Best_Mean_SBOX =

     2


Best_Mean_Rand =

    2.0023


Best_Diff =

    0.0023


Best_SBOX =

  Columns 1 through 15

    12     1     8    10     2    15     6     5     0     3    11     4    14     7    13

  Column 16

     9


End =

  1.0e+003 *
```

Name	Value	Min
ANF	<16x4 double>	0
B0123	<1x16 double>	0
Best_Diff	0.0023	0.0023
Best_Mean_Rand	2.0023	2.0023
Best_Mean_SBOX	2	2
Best_SBOX	<1x16 double>	0
Bit0123	<1x10000 double>	1
C	<4x16 double>	0
Cmp_Arr	<1x5040 double>	0.0023
E	<16x1 double>	0
End	[2016,3,8,13,15,22.6090]	3
Input	<1x16 double>	0
Input_Bit	<16x4 double>	0
Input_Bit0	<16x1 double>	0
✓ Input_Bit0123	<16x1 logical>	
Input_Bit1	<16x1 double>	0
Input_Bit2	<16x1 double>	0
Input_Bit3	<16x1 double>	0
Mean_Bit0123	1.9576	1.9576
NO_OF_B0123	0	0
Rand_Bit	<16x4 double>	0
Rand_Bit0	<16x1 double>	0
Rand_Bit1	<16x1 double>	0
Rand_Bit2	<16x1 double>	0
Rand_Bit3	<16x1 double>	0
Rnd	<10000x16 double>	0
Start	[2016,3,8,9,56,49.5580]	3
ans	<1x10000 double>	1
f	<4x16 double>	0
i	10000	10000

Figure 12 : Result Snapshot of 4X4 S-Box when no. of permutation is factorial 16

12	1	8	10
2	15	6	5
0	3	11	4
14	7	13	9

Table 4 : Least Deviated 4-by-4 S Box

Hence, the desired 4 by 4 Substitution Box among all possible permutation of 16 elememts is obtained as shown in Table 3 with least deviation from a set of truly random permutation. This S-Box has deviation of only 0.0023 from a set of truly random permutation. We can’t have better pseudo random S-Box than above with a very small difference in mean from a set of truly random permutation. It is very difficult for an attacker to do cryptanalysis on above S-Box on the basis of randomness. Thus, above S-Box is least susceptible to cryptanalysis. Hence, using the above S-Box in any cryptographic algorithm will give it more security against attacks.

CHAPTER 9

CONCLUSION

Main aim of this thesis work is to study about Substitution Box and do analysis of the Substitution Box in depth so that the shortcoming in the S-Box can be find out and eliminate it for making strong and improved cryptographic system which is prone less to attacks. In introductory phase of this thesis, background, representations of Substitution Box were introduced. Then crucial properties of substitution box is described. We found out AES S-Box deviation from truly random source and then we have got least deviated 4-by-4 S-Box from truly random source. For further research, these findings (several significant deviations from a randomly generated permutation) could be used to build a good distinguisher.

REFERENCES

[1] K. Mohamed, M. N. Mohammed Pauzi, F. H. HjMohd Ali, S. Ariffin, N. H. NikZulkipli (September, 2014). Study of S-box properties in block cipher in IEEE International Conference on Computer, Communications, and Control Technology (I4CT), 362 – 366.

[2] Anthony Lineham, Jacob, T. Aaron Gulliver (2008). Heuristic S-box Design in *Contemporary Engineering Sciences, Vol. 1, 2008, no. 4, 147 – 168.*

[3] Rashi Kohli, Divya Sharma and Manoj Kr. Baliyan (December 2012). S-Box Design Analysis and Parameter Variation in AES Algorithm in International Journal of Computer Applications 60(2), 42-45
.

[4] Sakthi Vignesh Radhakrishnan, S. Subramanian (April, 2012).An analytical approach to s-box generation in IEEE International Conference on Communications and Signal Processing (ICCSP), 1-5

[5] J. Daemen, V. Rijmen (2002).The Design of Rijndael : AES - The Advanced Encryption Standard in ISBN 3-540-42580-2, Springer-Verlag Berlin Heidelberg.

[6] K.Rahimunnisa, Dr. S. Sureshkumar, K.Rajeshkumar (2011). Implementation of AES with New S-Box and Performance Analysis with the Modified S-Box in IJCA Proceedings on International Conference on VLSI, Communications and Instrumentation (ICVCI) (2), 5–8

[7] Danilo Gligoroski and Marie Elisabeth Gaup Moe (April, 2007). On Deviations of the AES S-box when Represented as Vector Valued Boolean Function *in IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.4* .

[8] N. Ahmad, R. Hasan, W. M. Jubadi (October, 2010). Design of AES S-box using combinational logic optimization in IEEE Symposium on Industrial Electronics & Applications (ISIEA), 696 – 699

[9] M. M. Alani (June, 2010). DES96 - improved DES security in IEEE 7th International Multi-Conference on Systems Signals and Devices (SSD), 1 - 4

[10] Linda Burnett (2005). Heuristic Optimization of Boolean functions and substitution boxes in cryptography.

[11] Third edition Cryptography Theory and Practice Third Edition Discrete Mathematics and Its Applications.

[12] Converting truth table into ANF: www.ii.uib.no/~mohamedaa/odbf/help/ttanf.pdf

[13] Converting ANF into Truth table: www.metu.edu.tr/~ccalik/anfweight_slides.pdf