Methods of Cryptanalysis Final Project
"Good" S-box generation

Jason Friedman

Contents

1 Introduction **2**

2 Definitions **2**

3 Design criteria of S-boxes **2**

4 Effect of S-box criteria on Differential Cryptanalysis **3**

5 Effect of S-box criteria on Linear Cryptanalysis **3**

6 Additional potential design criteria and their effect **4**

7 Algorithm implemented **4**

8 Results **5**

9 Discussion **8**

10 Source Code **9**

1 Introduction

The strength of ciphers such as DES lie in the fact that that the S-boxes are non-linear, the rest of the cipher can be described by linear functions. Hence the design of the S-boxes is crucial for the strength of DES and variants. Even small changes to DES can make it much more susceptible to certain attacks (as explained in [2]).

2 Definitions

- $\Delta I_{i,j}$ is the input difference pattern to S-box between inputs i and j, $|\Delta I_{i,j}|$ is the number of different input bits.
- $\Delta O_{i,j}$ is the output difference pattern to S-box between inputs i and j, $|\Delta O_{i,j}|$ is the number of different output bits.
- NS is a measure of linearity for S-boxes. The definition in [4] states that for a given input XOR difference α and output XOR difference β:

  $$NS(\alpha,\beta) = \#\{x|0 \le x < 64, \bigoplus_{s=0}^{5}(x[s] \bullet \alpha[s])) = (\bigoplus_{t=0}^{3}(S(x)[t] \bullet \beta[t]))\}$$

  where the symbol $\bullet$ stands for a bitwise AND operation.

3 Design criteria of S-boxes

Some of the Design criteria for the S-boxes, stated in [1] are as follows:

- (S-1) Each S-box has six bits of input, four bits of output
- (S-2) No output bit of an S-box should be too close to a linear function of the input bits
- (S-3) Fixing the outer two bits at any two values, the resulting map is a permutation.
- (S-4) If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits
- (S-5) If two inputs to an S-box differ in exactly the two middle bits (ie $\Delta I_{i,j} = 001100$), then the output must differ in two bits ($|\Delta O_{i,j}| \geq 2$).
- (S-6) If two inputs to an S-box differ in the pattern $\Delta I_{i,j} = 11xy00$ then the outputs must not be the same
- (S-7) For every 6-bit difference $\Delta I_{i,j}$, there may be no more than 8 of the outputs that have any given 4-bit difference $\Delta O_{i,j}$.
- (S-8) Where $\Delta O_{i,j} = 0$, the input pattern must cause one of the neighboring S-boxes also to be active (ie not be of the form $00xy00$) and have low probabilities for those where only one neighbor must be active.

4 Effect of S-box criteria on Differential Cryptanalysis

The design criteria had Differential Cryptanalysis in mind, hence they have a number of effects on potential cryptanalysis of DES:

- *One active S-box* It is not possible to create a one-round iterative pattern only one active S-box. If there is one active S-box, then the only the middle two bits may have a difference (this is due to the extension function, which means that the outer two bits on either side of an S-box are also inputs to a neighboring S-box). Rule (S-5) states that if both of these are different, then the output must have differ in at least two bits. Also if only one of the two middle bits has a difference, then by rule (S-4), the output must also differ in at least 2 bits. Hence the output difference cannot be zero, and as it is a property of the permutation function that these two bits will go to two different S-boxes, this could not be used to build a pattern with one active S-box.

- *Two active S-boxes* The design criteria also prevent having two active S-boxes that lead to zero output difference. From above, it is clear that the two active S-boxes need to be adjacent. Hence the first S-box must have zero difference for the two left-hand bits, and the second S-box zero difference for the two right-hand bits. If the first S-box had zero output difference, then by (S-3), the right-most bit must be 1, hence the second will have a non-zero output difference. Similarly if the right S-box had zero output difference. Hence either one or both must have non-zero output, preventing building an iterative characteristic of this form.

- *Three active S-boxes* It is possible to build a two round iterative characteristic with 3 active S-boxes that lead to zero output difference, whereby even second round will have no active active S-boxes. As with two active S-boxes, we know that the left most bit of the 3rd S-box must be one, and the rightmost bit of the left S-box must be one. Clearly, by (S-6), the input pattern 11xy00 cannot be used, hence the second left-most bit of the third S-box must be zero. The middle S-box is now of the form $a1bc10$. By (S-3), the left-most bit must be a one, hence we get a permissible characteristic. (S-8) is included to ensure that this characteristic will have a low probability.

5 Effect of S-box criteria on Linear Cryptanalysis

The second criteria (S-2) is designed to deter Linear cryptanalysis, although the wording is particularly vague.

6 Additional potential design criteria and their effect

Some possible additional design criteria is as follows:

- *(S-9) If two inputs to an S-box differ in the pattern* $11ef10$*, then the two outputs must not be the same* This design criteria is to ensure that the only two round iterative characteristic of the form described before will require all S-boxes to be active, hence making the probability too small for it to be useful in cryptanalysis. As explained before, the only possible input pattern for such an iterative pattern with 3 active S-boxes requires the middle S-box to be of the pattern $11ef10$, this is not possible with this design characteristic hence only one with 8 active S-boxes (where each is of the form $10xy10$) is possible.

- *(S-10) If two inputs to an S-box differ in the pattern* $10ef10$*, then the two outputs must not be the same* This will prevent any two round iterative patters of the type described above.

7 Algorithm implemented

I implemented an algorithm to search for S-boxes that satisfy the design criteria. It is clear that an exhaustive search of all possible S-boxes, even assuming (S-3) is not practical. There are: $(16!)^4 \approx 2^{177}$ such possible S-boxes. I aimed to instead build S-boxes bit by bit, checking after adding each bit that certain criteria were satisfied. After some experimentation, I realized that the order of checking is highly important for an efficient algorithm. Hence I checked for the least probably criteria first so that more partial S-boxes could be 'thrown away' at an earlier stage. The algorithm was implemented as follows, similar to that found in [3]. First a database was created of pairs of rows:

- A 'row' of the S-box (that is, with the outer two bits fixed, so that the row is a permutation) was generated, and each bit added was checked, compared to the previously added bits that:

  1. (S-3) Each bit was added (randomly selected) so that the row remained a permutation
  2. (S-4) This design criteria is checked by ensuring that if the two inputs have a one bit input difference, the output difference will be at least 2 bits.
  3. (S-5) This check is performed by ensuring that if the input has the pattern 001100, then the output difference will be at least 2 bits.

- Now a second row is built, that can be paired with the first row, to be either rows 1 and 3, or 2 and 4. It needs to be one of these pairs so that S-6 can be tested (and S-9 and S-10).

4. (S-6) This check is performed by ensuring that if the input has the pattern 11xy00, then the output difference will not be zero.

  From the database, two pairs were combined.

5. (S-4) This was checked now for the combined rows

- For the remainder of the tests, a difference table is built, and this is used to perform the following tests:

6. (S-7) Here the table is checked that no entry is greater than 20. (Note that this is a relaxation of the original DES criteria)
7. (S-8) This checks the table that the input differences that make up the two round iterative pattern (namely 00xy11, 11xy10, 10xy00) have a sufficiently low probability of zero output difference.

- The last test is performed on the inputs (without using the table)

8. (S-2) This is checked by using the measure from Matsui [4], and making sure that the NS[1] value is no worse than that for DES S-boxes. The magnitude of $|NS(\alpha, \beta) - 32|$ represents the effectiveness of a particular correlation, that is, the higher the value, the more effective it is. In DES S-boxes, the highest value is found for S-box 5, where $NS_5(16, 15) = 12$, ie $|NS(16, 15) - 32| = 20$, and the program ensures that no generated S-box has a higher value than this.

8 Results

The program produces S-boxes in a reasonable amount of time, running in about 10 minutes to half an hour. Even with a fairly small database (10000 pairs), it is possible to find a number of S-boxes that satisfy the stated rules as explained above. Here are 8 such S-boxes produced by the program:

-

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

  - Maximum NS value = 16
  - Non-zero percentage = 47
  - Maximum difference between 0s/1s for a fixed bit = 8

-

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

[1] NS is defined on page 2

- Maximum NS value = 16
- Non-zero percentage = 48
- Maximum difference between 0s/1s for a fixed bit = 8

•

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

- Maximum NS value = 14
- Non-zero percentage = 47
- Maximum difference between 0s/1s for a fixed bit = 8

•

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

- Maximum NS value = 14
- Non-zero percentage = 47
- Maximum difference between 0s/1s for a fixed bit = 8

•

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

- Maximum NS value = 14
- Non-zero percentage = 47
- Maximum difference between 0s/1s for a fixed bit = 8

•

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

- Maximum NS value = 12
- Non-zero percentage = 48
- Maximum difference between 0s/1s for a fixed bit = 8

•

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

- Maximum NS value = 14

– Non-zero percentage = 48
– Maximum difference between 0s/1s for a fixed bit = 8

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

  – Maximum NS value = 14
  – Non-zero percentage = 70
  – Maximum difference between 0s/1s for a fixed bit = 8

The following is the difference distribution table for the last S-box shown:

	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	a_x	b_x	c_x	d_x	e_x	f_x
0_x	64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	4	0	2	10	4	0	8	8	4	12	2	2	8
2_x	0	0	0	6	0	10	14	2	0	4	14	4	4	2	0	4
3_x	4	0	0	10	2	6	8	2	0	6	2	2	8	6	0	8
4_x	0	0	0	8	0	10	6	0	0	6	4	2	6	2	4	16
5_x	8	0	2	10	4	8	4	0	4	6	6	0	8	2	0	2
6_x	0	0	2	4	0	12	2	0	4	10	6	2	6	0	4	12
7_x	6	0	2	12	4	0	14	2	2	10	4	0	2	0	2	4
8_x	0	0	0	16	0	8	8	0	0	16	4	0	6	2	2	2
9_x	8	0	4	6	0	8	2	0	2	4	6	2	10	0	4	8
a_x	0	0	2	8	0	6	8	0	6	12	8	0	4	0	0	10
b_x	4	0	2	2	2	12	10	0	4	6	6	0	4	0	4	8
c_x	0	0	0	10	0	6	14	2	0	2	12	0	8	4	2	4
d_x	8	8	0	4	0	6	2	0	0	8	6	2	8	2	0	10
e_x	0	4	0	8	0	8	8	0	2	2	4	4	18	2	0	4
f_x	14	4	2	4	0	10	2	4	0	4	14	2	0	0	0	4
10_x	0	0	0	0	0	0	2	10	0	2	0	10	6	12	20	2
11_x	6	8	12	2	10	0	2	4	6	0	0	6	2	4	2	0
12_x	0	6	0	0	10	0	0	8	10	2	4	10	2	10	2	0
13_x	4	8	6	0	4	0	2	8	10	0	6	6	0	6	4	0
14_x	0	8	16	0	8	4	0	0	10	2	0	4	0	4	6	2
15_x	0	6	6	6	6	0	0	8	2	0	0	8	0	6	10	6
16_x	0	6	18	2	10	0	0	12	4	2	0	0	0	2	8	0
17_x	0	6	0	0	6	4	0	4	12	6	2	2	0	14	8	0
18_x	0	2	12	2	6	0	0	6	10	0	0	6	2	4	10	4
19_x	2	8	16	0	8	8	0	2	0	2	0	4	0	8	6	0
$1a_x$	0	8	10	0	16	0	0	6	6	4	2	2	0	2	6	2
$1b_x$	0	4	4	4	2	0	2	8	14	2	0	12	0	2	6	4
$1c_x$	0	18	0	0	6	0	2	10	8	0	4	6	0	8	0	2
$1d_x$	0	6	8	0	8	0	4	6	6	0	2	6	4	4	8	2
$1e_x$	0	12	4	0	8	0	0	8	4	0	2	14	2	10	0	0
$1f_x$	0	6	0	0	8	0	2	12	2	2	2	8	6	8	8	0
20_x	0	0	0	14	0	6	12	8	0	4	2	4	6	0	0	8
21_x	4	2	0	4	0	8	4	2	2	12	10	2	6	2	2	4
22_x	6	4	2	6	0	10	4	0	0	8	6	0	8	4	2	4
23_x	14	2	4	8	2	2	4	0	2	4	6	0	6	0	2	8
24_x	12	2	2	4	2	2	8	0	4	14	6	0	4	0	2	2
25_x	4	2	0	2	0	12	12	0	0	2	8	6	6	2	2	6
26_x	6	0	2	4	2	8	10	0	4	4	8	0	6	2	2	6
27_x	8	2	0	8	0	10	4	4	0	4	6	0	2	6	0	10
28_x	4	0	4	4	4	14	10	4	0	2	10	0	4	0	0	4
29_x	4	2	0	4	2	6	2	0	4	8	8	2	12	2	0	8
$2a_x$	8	2	2	4	2	8	2	0	0	2	4	2	14	0	4	10
$2b_x$	6	2	2	10	0	4	4	4	0	16	6	2	0	0	2	6
$2c_x$	12	4	0	2	2	4	4	0	2	14	8	2	4	2	0	4
$2d_x$	6	0	4	10	0	4	10	2	4	0	8	0	10	0	2	4
$2e_x$	4	0	0	14	0	0	2	0	2	4	8	4	6	4	2	14
$2f_x$	6	0	2	6	8	6	12	0	0	6	0	0	10	0	2	6
30_x	0	6	4	0	6	4	0	4	14	2	6	8	0	4	6	0
31_x	0	4	4	4	8	4	0	16	6	2	0	4	0	4	6	2
32_x	4	4	8	2	6	0	0	8	4	0	0	10	6	4	8	0
33_x	0	8	4	2	0	2	0	12	8	2	0	4	2	10	6	4
34_x	0	4	4	2	10	2	2	8	4	0	2	4	2	6	12	2
35_x	6	2	8	2	8	2	0	4	16	0	2	8	0	6	0	0
36_x	0	12	2	4	6	4	0	4	6	0	2	10	0	8	4	2
37_x	0	10	8	2	2	0	4	2	6	2	0	4	2	6	14	2
38_x	0	10	2	0	6	0	0	2	12	8	0	8	2	6	6	2
39_x	0	8	12	0	6	2	6	10	0	0	4	4	0	4	8	0
$3a_x$	2	10	14	2	0	2	0	6	2	2	0	8	0	10	2	4
$3b_x$	4	8	2	0	14	0	0	4	2	0	2	6	6	6	10	0
$3c_x$	0	2	8	2	6	0	4	14	4	0	2	2	2	6	10	2
$3d_x$	0	4	4	2	12	2	2	2	4	4	0	10	2	12	4	0
$3e_x$	6	4	10	0	12	0	6	6	6	0	0	2	0	8	4	0
$3f_x$	2	8	10	0	2	0	0	2	10	2	4	12	0	4	4	4

This last S-box is one that also satisfies (S-9)

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

  – Maximum NS value = 16
  – Non-zero percentage = 47
  – Maximum difference between 0s/1s for a fixed bit = 8

It was not possible to generate an S-box that also satisfied (S-10) after attempting this several times over large databases.

9 Discussion

It has been shown that it is possible to generate S-boxes that fulfill the specified design criteria in a reasonable amount of time. These S-boxes could be used for a DES-like cipher. Generating a large set of such DES-like S-boxes could also be used for a key-dependent S-box cipher.

The only criteria that was relaxed was that of (S-7), allowing values of up to 20 in the DDT, rather than 16 as is specified in [1]. When this was left at 20, the program was not able to find a solution. This leads me to believe that to generate such S-boxes it would be necessary to further constrain the database generated with some form of (S-7) applicable to the rows (although upon examining the DES S-boxes, individual rows can cause entries up to 14, so it was difficult to see how this could be done). In [3], they also used a value of 20, although this may have been necessary to accommodate the extra design criteria (S-9) that they used. It was interesting to note that (S-10) could not be fulfilled with (S-7) set at a maximum of 20 - the trade-off it seems needs to be made between lowering the probability of a two round iterative characteristic and the maximum probability in the table.

10 Source Code

References

[1] Coppersmith Don, *The Data Encrpytion Standard (DES) and its strength against attacks.*

[2] Biham Eli and Shamir Adi, *Differential cryptanalysis of the data encryption standard*, Springer-Verlag, 1993.

[3] Kim Kwangjo, Park Sangjun, and Lee Sangjin, *Reconstruction of s^2DES S-boxes and their immunity to differential cryptanalysis*, JW-ISC'93 (1993).

[4] Mitsuru Matsui, *Linear cryptanalysis of DES cipher (i)*, Class handout.