Contemporary Engineering Sciences, Vol. 1, 2008, no. 4, 147 - 168

Heuristic S-box Design

Anthony Lineham

Dept. of Electrical and Computer Engineering
University of Canterbury
Private Bag 4800
Christchurch, New Zealand

T. Aaron Gulliver

Dept. of Electrical and Computer Engineering
University of Victoria
PO Box 3055, STN CSC
Victoria, B.C., Canada V8W 3P6
agullive@ece.uvic.ca

Abstract

This paper presents the construction of substitution boxes (s-boxes) using heuristic techniques. The objective was to generate s-boxes that meet the Strict Avalanche Criteria (SAC), are non-linear, and have a high degree of resistance to differential cryptanalysis. It was found that it is possible to produce s-boxes which exhibit up to 85% compliance with the SAC, and several were found to give over 90% compliance with the DES-type SAC.

Keywords: Cryptography; S-box; Heuristic design; Strict avalanche criteria

1 Introduction

The concept of concealing the meaning of a message from all but the intended recipients is an old one. Julius Caeser is known to have used cryptography in the form of a simple letter substitution cipher when sending important messages. The development of digital computers allowed the complexity of encryption algorithms to increase and to be used in a wide range of applications, including civilian uses. In 1976 the Data Encryption Standard (DES) was established and up until the late 1980s was the preferred method of encryption for non-classified and private-sector material. Today companies need to protect

commercially sensitive information from competitors, banks need to provide secure electronic commerce, and individuals need to send e-mail and communicate over the Internet without fear of personal details being intercepted and exploited.

Encryption algorithms can be divided into two different types: symmetric and asymmetric. Symmetric algorithms are those that use the same key for both encryption and decryption, and can be separated into block ciphers and stream ciphers. Asymmetric algorithms use one key for encryption and another for decryption. Block ciphers are the most common form of encryption algorithms. DES, IDEA, and Blowfish are all block ciphers. The term block cipher comes from the fact that the plaintext is broken up into blocks of equal length. Each block is encrypted separately and will always result in the same ciphertext when the same key is used. Most algorithms use relatively simple operations but repeat these operations multiple times. Each repetition is referred to as a round and the operation performed on the input is called the round function. Block ciphers are generally well suited to implementation in software [5]. They have the advantage that if an error occurs in the ciphertext, it will only affect the block in which it is located.

A desirable feature of an encryption algorithm is that the output ciphertext be dependent on every input bit and every key bit. Hence if one bit of the plaintext changes it should result in a ciphertext that is unrelated to and cannot be associated with the original ciphertext. This small change at the input resulting in a large change at the output is referred to as the *avalanche effect*.

1.1 Feistel Networks

A Feistel network forms the basis for most block encryption algorithms. A block of data consisting of an even number of bits is divided into a left (L) and a right (R) half. The output of the i^{th} round is given by

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1},\ K_i) \end{aligned} \tag{1}$$

K_i is the sub-key, derived from the main key, that is used in round i and f is the round function. The round function must be repeatable and produce an output of a size that can be XORed with the left half from the previous round. Note that in the final round there is no swapping of the left and right halves so that in round n we have

$$\begin{aligned} R_n &= R_{n-1} \\ L_n &= L_{n-1} \oplus f(R_{n-1},\ K_n) \end{aligned} \tag{2}$$

An important feature of a Feistel network is that the same structure can be used for both encryption and decryption. All that is required is a reversal of the key schedule. Suppose a block of data has been encrypted by a Feistel cipher and the resulting ciphertext is used as the input to the cipher but this time with the key schedule reversed. Hence the input ciphertext, C, is made up of L_0 and R_0 which are equal to L_n and R_n from the previous encryption, and K_1 is equal to K_n from the encryption process. Substituting L_0 and R_0 into (1) gives

$$\begin{aligned} L_1 &= R_0 \\ R_1 &= L_0 \oplus f(R_0,\ K_1) \end{aligned} \tag{3}$$

K_1, L_0 and R_0 from the encryption process can then be substituted into (3) from (2), giving

$$\begin{aligned} L_i &= R_n = R_{n-1} \\ R_i &= (L_{n-1} \oplus f(R_{n-1},\ K_n)) \oplus f(R_{n-1},\ K_n) = L_{n-1} \end{aligned} \tag{4}$$

Equation (4) shows that when the ciphertext is input into the algorithm with the key schedule reversed, the last round of the encryption process is inverted. Each subsequent round reverses another encryption round until the plaintext has been restored.

1.2 Substitution Boxes (S-boxes)

S-boxes are found in the round function of many Feistel network based block encryption algorithms. Typically they are one or two-dimensional arrays of numbers. DES is a symmetric block encryption algorithm based on a Feistel network. It employs two-dimensional s-boxes with four rows and sixteen columns with each row containing a permutation of the numbers 0-15. Blowfish has one-dimensional s-boxes with 256 elements, but the elements are 32-bit numbers. The input to an s-box constitutes a reference to one of its elements. In DES the s-box input is a 6-bit number made up of 2 bits to select a row and 4 bits to select a column. DES s-boxes are said to be '6×4' because of the 6-bit input and 4-bit output. The Blowfish s-boxes require an 8-bit input and give a 32-bit output and so are '8×32'.

The elements in an s-box can come from a variety of sources. The DES s-boxes are reported to have been carefully designed to conform to a number of criteria. In other cases the design involves the use of mathematical functions that result in s-boxes with desirable characteristics. Conversely, there are algorithms such as Blowfish, which initialise the s-boxes with bits from the binary representation of π and then modifies them based on the key schedule.

1.3 Cryptanalysis

Cryptanalysis refers to the process of attempting to recover the plaintext or the key that corresponds to a particular ciphertext by a party who is not the intended recipient. Such an attempt is called an 'attack' and the party performing the attack is called an 'attacker'.

The most obvious form of cryptanalysis is the Brute Force Attack, also known as an exhaustive key search. This attack involves repeatedly decrypting the ciphertext trying every possible key until the resulting plaintext is 'meaningful'. Alternatively, this attack is carried out with a plaintext/ciphertext pair with keys being tried until the resulting plaintext is equal to the known plaintext. This method is preferable as it may be difficult to determine what is a meaningful plaintext and it may also be possible to decrypt the ciphertext to many meaningful plaintexts if appropriate keys are used. In order to prevent this kind of attack, it is important that an algorithm has a key space that is so large that it is impractical for a would-be attacker to try all possible keys within the time span over which the plaintext must be protected. When originally put into use, the DES key space was large enough to make such an attack infeasible, but recently an array of parallel computers was able to perform a brute force attack on a DES encrypted message in less than one day [2].

Differential cryptanalysis is a type of attack that involves analysing the changes that occur in the ciphertext when different changes are made to the plaintext. The ciphertext changes are then interpreted with reference to certain exploitable characteristics of the round function. Usually the round function characteristics that are exploited relate to the s-boxes. Typically a very large number of plaintexts/ciphertext pairs are required for this attack. The objective is to reveal either the whole key or enough of the key to make a brute-force attack feasible. Details of how differential cryptanalysis works, and its use against DES and other well known algorithms, can be found in [5].

Linear cryptanalysis attempts to make linear approximations of the operations performed by an algorithm. Probabilities are assigned to each approximation. If enough information is gathered it may be possible to deduce some key bits to within a certain degree of accuracy. A linear cryptanalysis of DES is given in [4].

There are many other methods of cryptanalysis, some of which are general approaches while others are aimed at specific algorithms. differential cryptanalysis and linear cryptanalysis are two of the newest and most powerful methods of cryptanalysis. They can be used on a variety of algorithms.

1.4 Motivation for Research in S-Box Design

Differential and linear cryptanalysis are very powerful methods for attacking encryption algorithms. Both take advantage of weaknesses found in the round

function of an algorithm, usually in the s-box characteristics. It is desirable that there be techniques for generating s-boxes that have characteristics which endow them with a degree of resistance to differential and linear cryptanalysis. These techniques should also have low level of computational complexity so that the generation process can be performed in a relatively short period of time. With such generation techniques available it will be possible to design new encryption algorithms that make use of the ready availability of s-boxes with favourable characteristics.

1.5 The Differential Distribution Table

The Differential Distribution Table (DDT) is a table that indicates how the output of an s-box varies as the input is varied. The number of rows in the s-box is equal to the number of elements in the s-box (2^n where n is the number of input bits), and the number of columns is equal to the number of distinct output values in the s-box (2^m, where m is the number of output bits). Each row corresponds to a change in the input value, i.e., the XOR of the input with some other value of equal bit length. Row 1 corresponds to the trivial case of XOR with zero, while the final row correspond to XOR with n 1's. The columns correspond to the XOR of the original and changed output. Each element in the table indicates how many times a given XOR at the input to the s-box results in a particular change at the output. The sum of each row is equal to 2^n and the sum of each column $2^{2n}/2^m$. The first row of the table is all zeros except for the first element, which is equal to the number of elements in the s-box. Obviously when any input to the s-box is XORed with zero (not changed), the output will not change.

The DDT is useful as it allows for the assessment of how well an s-box conforms to the criteria under which it was generated. The compliance of an s-box with the Strict Avalanche Criteria (SAC) (see Section 2), can be checked by looking at the rows of the table that correspond to an input change of one bit, and the columns that correspond to an output change of half the output bits. In an ideal situation, the sum of all such columns in a one-bit change row would be equal to 2^n. Dividing this sum by 2^n (and multiplying by 100%) gives the percentage compliance with the strict avalanche criteria for that bit change. Averaging this value over all one-bit changes gives an indication of the overall compliance of an s-box with this criterion. The DDT can also be used to check similar properties such as how often a one-bit input change results in only a one-bit output change.

2 Desirable S-box Characteristics

- High level of compliance with the Strict Avalanche Criteria.

It is desirable that the degree to which output bits are dependent on input bits should increase as rapidly as possible through an encryption algorithm. This is primarily an action of the s-boxes. SAC requires that a change in one input bit results in half the output bits of the s-box changing. The ultimate goal is that every output bit of an encryption algorithm be dependent on every input bit. This means that changing one of the input bits should result in a new output that is unrelated to the previous output. A slight variation of the SAC was used in the construction of the DES s-boxes. The requirement is that if 1 input bit changes at least half the output bits must change.

- Non-linear

  It should not be possible to express the operation of the s-box as a linear function of the inputs. This would allow the encryption algorithm to be broken through a process of solving a set of equations for a set of unknowns. Typically as s-boxes become larger the probability of them containing a linearity decreases rapidly.

- Resistance to Differential Cryptanalysis [1] Cryptosystems,"

  Differential Cryptanalysis is a method of breaking encryption algorithms that are based on Feistel networks. It is a statistical attack that uses the characteristics of an s-box given by its DDT to determine either all the key bits or enough of them to reduce the complexity of a brute force attack to a manageable level. The two main features of an s-box that differential cryptanalysis exploits are:

  – large value entries in the DDT,
  – entries in the first column of the DDT, particularly those that have large values.

  Hence the DDT of an s-box should have a low maximum value (excluding the (1,1) entry), which is not significantly greater than the other entries, and a low number of entries in the first column but not so few that they have large values. These last two requirements are conflicting. A small number of entries presents a restricted number of opportunities for differential cryptanalysis to be performed, while small value elements mean that when the cryptanalysis is performed it will be more difficult to get a conclusive result.

2.1 Robustness

Seberry [6] '94," defines an expression for the robustness, R, of an s-box. This is a measure of the resistance of an s-box to differential cryptanalysis.

Robustness is based on two features of the DDT. The first is the number of non-zero elements, N, in the first column of the DDT (excluding the first element). These denote instances when a change in the input results in no change in the output. Such occurrences are a weakness as they reduce the complexity of an algorithm and play an important part in differential cryptanalysis. The other feature is the largest value found in the DDT, L, other than the (1,1) element. The robustness is given by

$$R = \left(1 - \frac{N}{2^n}\right)\left(1 - \frac{L}{2^n}\right) \tag{5}$$

where n is the number of input bits. The higher R is, the more difficult differential cryptanalysis is to perform.

3 The Original S-box Algorithm

In the previous random s-box algorithm [3], the elements of the s-box are chosen by randomly selecting an entry from an array of allowable values. In this case, the rows are required to contain each of the values 0 to 255. When a value has been chosen it is tested to see if it meets certain criteria with respect to the values that are already in the s-box. If it is found to be acceptable, it is added to the s-box. If it is not acceptable, another value is chosen at random and tested. This continues until either a suitable value is found or the iteration limit is reached. If the iteration limit is reached, the criteria which a selected value must meet are relaxed slightly and the search resumes with all subsequent values being tested against the relaxed criteria. This continues until the row has been filled. Then the criteria are reset and the next row is started. The criteria used to test the values are described in Table 1. The hierarchy in which the tests were used is given in Table 2.

4 The New Algorithm

before The size of the s-boxes was chosen to be 9-bit input/6-bit output. This was done to keep the time taken to generate s-boxes reasonable, increase as a large number of s-boxes is required to investigate all the design parameters. The time to test the s-boxes was also a consideration. construct

Test Name	Description
Ch1_4	1-bit change in input column index must give 4-bit change in output.
Chr1_4	1-bit change in input row index must give 4-bit change in output.
Ch1_3	1-bit change in input column index must give $\geq$ 3, $\leq$ 5-bit change in output.
Chr1_3	1-bit change in input row index must give $\geq$ 3, $\leq$ 5-bit change in output.
Ch1_2	1-bit change in input column index must give $\geq$ 2, $\leq$ 6-bit change in output.
Chr1_2	1-bit change in input row index must give $\geq$ 2, $\leq$ 6-bit change in output.
Ch1_1	1-bit change in input column index must give $\geq$ 1, $\leq$ 7-bit change in output.
Chr1_1	1-bit change in input row index must give $\geq$ 1, $\leq$ 7-bit change in output.
Ch2_3	2-bit change in input column index must give $\geq$ 3, $\leq$ 5-bit change in output.
Chr2_3	2-bit change in input row index must give $\geq$ 3, $\leq$ 5-bit change in output.
Ch2_2	2-bit change in input column index must give $\geq$ 2, $\leq$ 6-bit change in output.
Chr2_2	2-bit change in input row index must give $\geq$ 2, $\leq$ 6-bit change in output.

Table 1: Tests Used in the Previous S-box Algorithm

Level	Tests used in level
1	Ch1_4, Chr1_4, Ch2_3, Chr2_3
2	Ch1_4, Chr1_4, Ch2_2, Chr2_2
3	Ch1_4, Chr1_4
4	Ch1_3, Chr1_3
5	Ch1_2, Chr1_2
6	Chr1_1
7	Ch1_1

Table 2: Test Hierarchy

The 2-bit input change tests were removed from the previous algorithm. This was done because tests showed that selecting values to satisfy these criteria had an adverse affect on the robustness of the s-boxes. In addition, such criteria are not found in the literature concerning the SAC.

A requirement used in the design of the DES s-boxes is that a change in the two middle input bits should change at least half of the output bits. This criterion (denoted the 'mid' criteria), was used in the generation of most s-boxes in this paper. Because the number of input bits is odd, the 'middle two bits' are taken to be the middle two bits of the six bits that reference the s-box column.

5 Summary of the S-Boxes Produced

5.1 Purely Random S-Boxes

As a 'control' against which to compare the quality of the s-boxes generated using the various algorithms, 8 purely random s-boxes were generated. Values for each position were selected randomly so that the only intentional structure in the resulting s-box was that each row contained the values 0 to 63. A summary of the characteristics of these s-boxes is given in Table 3. The de-

S-box	Robust-ness (R)	Zero Column Elements (N)	Largest DDT Value (L)	SAC Compliance (%)	DES-type SAC Compliance (%)	Compliance with 'Mid' (%)
1	0.1250	444	30	31.34	66.88	70.31
2	0.1324	440	30	32.07	64.97	67.97
3	0.1361	438	30	32.03	66.62	67.58
4	0.1287	442	30	30.08	66.49	60.55
5	0.1311	441	28	31.77	66.84	69.92
6	0.1453	433	30	31.77	66.57	67.97
7	0.1305	441	30	31.64	66.80	55.47
8	0.1298	442	26	32.51	66.23	63.28

Table 3: Characteristics of the Randomly Generated S-boxes

gree of normal and DES-type SAC compliance are consistent with the weight distributions of 6-bit numbers. The 'mid' compliance is slightly higher than expected. The L values are low compared to the maximum (512) and quite consistent, while the N values are high and dominate the robustness levels.

5.2 The Original Random Algorithm

A set of s-boxes was generated using a slightly modified version of the original algorithm as described in Section 4. Although the mode of operation is the same, the test hierarchy used is given in Table 4. The characteristics of the resulting s-boxes are summarized in Table 5. The SAC compliance has

Level	Tests used in level
1	Ch1_3, Chmid, Chr1_3
2	Ch1_3, Chr1_3
3	Ch1_3, Chr1_2
4	Ch1_3, Chr1_1
5	Ch1_2, Chr1_1
6	Ch1_1, Chr1_1

Table 4: Test Hierarchy

S-box	Robust-ness (R)	Zero Column Elements (N)	Largest DDT Value (L)	SAC Compliance (%)	DES-type SAC Compliance (%)	Compliance with 'Mid' (%)
1	0.1452	431	42	67.66	83.68	47.66
2	0.1653	419	46	73.31	86.63	48.83
3	0.1501	429	38	69.84	84.59	45.70
4	0.1416	432	48	65.28	83.51	55.86
5	0.1422	433	40	61.45	80.82	62.11
6	0.1489	430	36	65.49	83.51	66.80
7	0.1493	428	46	67.06	83.64	62.11
8	0.1560	425	42	72.14	86.46	55.08

Table 5: Characteristics of the S-boxes Generated with the Original Algorithm

improved considerably, indicating that the algorithm is producing the desired effect. It is interesting that the 'mid' compliance levels have actually dropped. This seems strange given that the algorithm encourages this characteristic. The N values have decreased while the L values have increased, giving a slight overall improvement in robustness.

5.3 Improved Version of the Algorithm

One major problem with the original algorithm is that once no suitable values can be found that meet the primary criteria, a less strict set of criteria are

adopted and used for the rest of the row. It seems more appropriate to return to the strictest level when searching for subsequent values. Even if the criteria needs to be relaxed again in order to find a value further along the row, at least all the values in between meet the primary criteria. With the original algorithm, this would not be the case and consequently the resulting s-boxes have a low probability of compliance with the strict avalanche criteria. The algorithm was modified to generate s-boxes using this new method. As in the original version, multiple attempts are made to obtain values complying with the primary criteria. If a value is not found, the criteria are reduced until a value is found. The criteria then reverts to the highest level. The characteristics of the s-boxes generated with this method are shown in Table 6. These results show that the average SAC compliance has increased to around

S-box	Robust-ness (R)	Zero Column Elements (N)	Largest DDT Value (L)	SAC Complian-ce (%)	DES-type SAC Complian-ce (%)	Percentage Complian-ce with 'Mid' (%)
1	0.1727	414	50	72.74	86.07	41.80
2	0.1529	426	46	68.36	83.94	41.80
3	0.1517	427	44	69.61	84.41	46.09
4	0.1468	428	54	71.09	85.24	43.75
5	0.1404	434	40	72.74	85.76	44.53
6	0.1569	423	50	66.88	83.72	59.77
7	0.1475	429	46	68.71	83.81	62.89
8	0.1667	419	42	69.70	84.90	63.67

Table 6: Characteristics of the S-boxes Generated with the New Algorithm

70% from 68%. The average DES-type SAC level has increased by less than 1%. The average robustness has increased from 0.1498 to 0.1545. This is due to a drop in the average N value, the effect of which has been reduced by an increase in the average L value.

5.4 The Effect of the "Mid" Criteria

The 'mid' condition is described in [5]. It requires that when the two middle bits of the input are changed, half the output bits will change. This condition is said to have been used in the design of the DES s-boxes. It was not included in the original algorithm, but is in the new version. As the number of input bits to the s-box is odd, the 'middle two bits' were taken to be the middle two bits that reference the s-box column. It was required that three of the output bits should change when these two bits changed. In the generation

of the s-boxes in Table 6, this condition was broken quite frequently and so contributed to a long generation time. Thus it was decided to generate s-boxes without this condition in order to investigate the effect of "chmid." The characteristics of these s-boxes are given in Table 7. The average SAC

S-box	Robust-ness (R)	Zero Column Elements (N)	Largest DDT Value (L)	SAC Complian-ce (%)	DES-type SAC Complian-ce (%)	Percentage Complian-ce with 'Mid' (%)
1	0.1475	429	46	70.57	86.06	55.47
2	0.1433	430	54	71.70	85.33	62.11
3	0.1485	427	54	73.74	87.11	60.55
4	0.1519	428	38	72.48	86.63	60.16
5	0.1600	422	46	75.65	87.98	61.33
6	0.1529	426	46	72.01	85.24	62.89
7	0.1709	412	64	76.00	88.15	54.69
8	0.1607	422	44	72.57	86.59	53.91

Table 7: Characteristics of the S-boxes Generated without the Chmid Condition

compliance level has increased to approximately 73%. The average robustness is virtually unchanged although this is achieved through slight increases and reductions in the N and L values, respectively. It is interesting that the 'mid' levels are better.

5.5 Random Starting Point Algorithm

This version of the algorithm starts generating each row at a randomly selected column, moving left to right in circular fashion. The 'mid' condition was not used for these s-boxes. S-boxes 1 to 4 were generated with the usual random element selection and testing method. S-boxes 5 and 6 were generated in the usual way for the Ch1_3 criteria elements, but all lower criteria elements were selected by moving sequentially through the remaining candidate values until a suitable one was found. This was done in the hope of improving the speed of the algorithm. The time taken to produce an s-box was reduced from approximately 90 miutes to 60 minutes. As shown in Table 8, the robustness of the resulting s-boxes was low compared to the previous ones. The algorithm was modified to randomly try candidate values up to an iteration limit equal to the number of remaining values before resorting to the sequential search. This method was used for s-boxes 7 and 8, and increased the generation time back up to about 90 minutes. Table 8 shows the characteristics of the s-

boxes that were generated. The SAC compliance levels of these s-boxes are

S-box	Robust-ness (R)	Zero Column Elements (N)	Largest DDT Value (L)	SAC Compliance (%)	DES-type SAC Compliance (%)	Percentage Compliance with 'Mid' (%)
1	0.1392	435	38	64.62	82.81	63.67
2	0.1506	428	42	64.71	81.16	64.45
3	0.1560	425	42	65.63	82.55	61.72
4	0.1447	432	38	60.58	80.82	61.72
5	0.1374	436	38	67.75	82.94	57.81
6	0.1483	430	38	65.93	83.33	58.20
7	0.1501	429	38	63.93	82.16	67.58
8	0.1362	437	36	62.08	81.08	64.06

Table 8: Characteristics of the S-boxes Generated with a Random Starting Point

significantly lower than those generated previously, and there is no significant improvement in the average robustness. It is interesting to note that the L values are significantly lower than any obtained previously (except the purely random s-boxes), but the effect of this has been offset by an increase in the N values.

6 Comparison with Other S-boxes

As the Data Encryption Standard (DES) is considered to have very good s-boxes, a comparison of DES s-boxes with those generated here provides an indication of how well the algorithm works. The constraints that the DES s-boxes are reported to have been designed to meet are:

1. Two inputs differing in exactly one bit must differ in their outputs by at least 2 bits (SAC).

2. If two inputs differ in the two middle bits the outputs must differ in at least 2 bits.

3. If two inputs differ in their first two bits and are identical in their last 2 bits, the two outputs must not be the same.

4. For any non-zero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference, i.e., $L = 16$.

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)	1-bit ip/op change (%)
1	0.3047	38	16	59.90	100	100	0
2	0.3516	34	16	54.69	100	100	0
3	0.3047	38	16	45.83	100	100	0
4	0.4570	25	16	58.33	100	100	0
5	0.3750	32	16	55.21	100	100	0
6	0.3516	34	16	47.40	100	100	0
7	0.3281	36	16	49.48	100	100	0
8	0.3164	37	16	59.90	100	100	0

Table 9: Characteristics of the DES S-boxes

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)	1-bit ip/op change (%)
1	0.2461	43	16	36.46	67.19	68.75	29.69
2	0.2930	39	16	30.21	69.27	81.25	30.73
3	0.2813	40	16	46.35	73.44	71.88	24.48
4	0.2471	41	20	40.10	69.79	78.13	28.65
5	0.2563	43	14	40.63	70.31	81.25	28.13
6	0.2808	39	18	39.06	76.56	78.13	21.35
7	0.2578	42	16	35.42	71.88	84.38	25.00
8	0.2363	42	20	39.06	75.00	78.13	23.96

Table 10: Characteristics of the DES Size Random S-boxes

There are a few other requirements, but they are concerned with the size and usage of the s-boxes rather than their properties. Note that constraint 1 is a slight variation of the SAC. Generally the requirement is that a 1-bit input change results in exactly half of the output bits changing. Both of these SAC variations are included in the tables.

The characteristics of the 8 DES s-boxes are given in Table 9. The final column refers to the percentage of 1-bit input changes that result in 1-bit output changes. For comparison, 8 purely random DES-size s-boxes were generated. The characteristics of these s-boxes are given in Table 10. Table 11 shows the characteristics of a set of s-boxes that are the same size as the DES s-boxes. The first 8 of these were generated with the algorithm in Section 5.3. This version was chosen as it exhibited more favourable characteristics than the others, and therefore is the most logical to use in a comparison with the DES

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)	1-bit ip/op change (%)
1	0.2969	32	26	75.00	89.58	90.63	9.38
2	0.3711	24	26	82.29	89.58	100	9.38
3	0.3071	27	30	78.13	89.58	96.88	10.42
4	0.4102	22	24	82.81	91.15	100	8.85
5	0.3525	26	26	78.13	89.06	96.88	10.94
6	0.3779	21	28	82.29	90.63	100	9.38
7	0.3149	21	34	82.29	90.10	100	9.90
8	0.3604	23	28	82.29	91.67	100	7.81
9	0.3896	22	26	77.60	99.48	100	0.52
10	0.4004	23	24	77.60	97.40	100	1.56
11	0.3281	22	32	80.21	99.48	100	0.52
12	0.3691	22	28	80.73	99.44	100	1.04
13	0.3989	21	26	79.17	98.96	100	1.04
14	0.3691	22	28	80.73	98.96	100	1.04
15	0.2729	21	38	79.69	98.96	100	0.52
16	0.4004	23	24	79.17	97.40	100	2.08

Table 11: Characteristics of the DES Size S-boxes Using the New Algorithm

s-boxes. The R values are of a similar order to those of the DES s-boxes but it is interesting to note the means by which this was achieved. The DES s-boxes all have $L = 16$, while the N values are mostly in the mid to high thirties. The random s-boxes have L values ranging from the mid 20's up to mid 30's but have N values in the low 20's to low 30's. From the description of differential cryptanalysis, a high L value is more likely to make an s-box susceptible to this kind of attack than a high N value. Note the high level of consistency of the DES s-boxes.

A point in favour of the random s-boxes is that they meet the strict avalanche criteria more closely than the DES s-boxes. However, the compliance levels are significantly lower than those for the larger s-boxes discussed earlier. This reinforces the proposition that larger s-boxes are better than smaller ones. The first 4 random s-boxes were generated with the 'mid' condition enforced while the rest were generated without it. There are corresponding differences in the percentage compliance with this condition in the resulting s-boxes.

A strong point in favour of the DES s-boxes is the consistent zeros in the right most column of Table 9. The random s-boxes perform poorly in this regard. It is worth considering a change in some of the algorithm constraints in order to rectify this problem. At present the first level constraint for the selection of an s-box element is that changing one bit must change half the

output bits ($b/2$). The next level constraint requires that a 1 bit input change requires that the output change be from $(b/2-1)$ to $(b/2+1)$, i.e., the 1-bit input/1-bit output change is permitted. Given the small number of values to choose from with this size of s-box, it may be worthwhile to alter the level 2 criteria to require greater than $(b/2-1)$ of the output bits to change with a 1 bit input change. A level 3 constraint could then be introduced that allows a 1-bit change in the output. S-boxes 9 to 12 were generated with the new level 2 constraint, but do not have the level 3 constraint implemented. It was found that some s-boxes were unable to be completed so the level 3 constraint was added for s-boxes 13 to 16. Note the dramatic improvement that these changes produce. The level of 1-bit input/output differences drops to less than 2%, while the compliance with the DES-type SAC increase to 97-99%. Compliance with the 'mid' condition is 100%. With such improvements due to the above change, the level 1 criteria could be replaced by the level 2 criteria.

Table 12 contains s-boxes generated with the primary criteria being that a 1-bit change in the input must produce a change in 2-bits or more in the output. The results in Table 12 show several effects due to this change in

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)	one2one (%)
1	0.2344	44	16	53.65	95.31	87.50	4.17
2	0.2813	40	16	56.77	93.75	90.63	6.25
3	0.2793	38	20	60.42	98.44	87.50	1.56
4	0.2686	39	20	49.48	96.88	81.25	3.13
5	0.3257	35	18	61.98	96.35	84.38	3.13
6	0.2227	40	26	45.83	94.79	96.88	4.69
7	0.2734	36	24	61.46	93.75	93.75	5.21
8	0.3027	33	24	66.67	96.35	100	3.65

Table 12: Characteristics of the S-boxes with Primary Criteria: 1-bit Input/Output Change Gives 2-bits or More Output Change

criteria. First, almost all R values are lower than in Table 11. This is caused by large increases in the N values, but the effect of this is lessened by the decrease in the L values. Seven of the eight L values are as low or lower than the lowest value found in Table 11. This is due to permitting a wider range of output difference patterns while previously a high proportion of these output differences would have been restricted to patterns with only two 1's. The SAC compliance has dropped to much lower levels than those found in Table 11, while the DES-type SAC compliance is somewhere between those of s-boxes 1 to 8 and 9 to 16 in Table 11. The level of 'mid' compliance has also dropped to as low as 84%. These drops are counter-intuitive, as the reason for the change

in criteria was the hope that these levels would increase. It is likely that the more rigid high level criteria of the previous version allows for the selection of better values towards the end of the row. Also, the 1-bit input/output change level has increased.

In an attempt to obtain good L values in Table 12 without sacrificing good SAC and 'mid' compliance, the criteria were changed again. The level 1 criteria stayed as ch1_2, chr1_2, and chmid, and new level 2 criteria were introduced. A 1-bit input change must result in a 2 or 3-bit change in the output. The level 3 criteria requires 2 or more output bits to change for a 1-bit input change. Finally, the 1-bit input/output change is permitted. The characteristics of 8 s-boxes generated in this way are given in Table 13. As previously, there has been a destabilising effect on the L values, but they have not dropped below the levels found in Table 11. DES-type SAC compliance is slightly below that of Table 11, but the standard SAC compliance is slightly better. The 'mid' levels have been restored to 100%. This method of generation is therefore more preferable to that used in generating Table 11.

S-box	R	N	L	SAC Complian (%)	DES-type SAC Complian (%)	mid (%)	1-bit ip/op (%)
1	0.3779	21	28	81.25	97.92	100	2.08
2	0.3613	27	24	76.04	97.40	100	2.08
3	0.4102	22	24	82.81	97.40	100	2.60
4	0.3604	23	28	81.25	98.96	100	0.52
5	0.2793	20	38	81.25	98.44	100	1.56
6	0.3516	19	32	80.73	99.48	100	0.52
7	0.2930	24	34	82.29	97.92	100	1.04
8	0.3989	21	26	81.25	98.44	100	1.04

Table 13: Characteristics of the DES S-boxes

7 Another Attempt at Generating 8×64 S-boxes

7.1 Improving the Selection Criteria

Using the techniques that were found to be effective in Section 6, 8×64 s-boxes were constructed. The characteristics of the resulting s-boxes are summarised in Table 14. S-boxes 1 and 2 were generated with the same version of the algorithm used in Section 5.3. Subsequent s-boxes were generated with gradual

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)
1	0.1571	424	44	70.36	85.37	64.06
2	0.1607	422	44	73.13	86.68	61.72
3	0.2179	391	40	78.95	93.01	68.75
4	0.2107	395	40	80.56	94.18	68.75
5	0.2357	380	44	81.34	94.62	62.50
6	0.2474	371	52	81.73	94.92	62.23
7	0.2900	346	54	85.37	96.40	64.06
8	0.2844	352	46	83.55	95.40	60.16
9	0.2725	360	42	84.07	96.40	62.11
10	0.2946	347	44	84.38	96.70	68.75

Table 14: Improved 8 x 64 S-boxes

changes to the criteria of the checking functions and rearrangement of their hierarchy.
The new check functions are:

> Ch1_3b, Chr1_3b: require that 3 or more output bits change with a 1-bit input change.
> Chmid: requires that 3 or more output bits change when the middle 2 column determining bits are changed.
> Ch1_2b, Chr1_2b: require that 2 or more output bits change with a 1-bit input change.

The new test hierarchy is given in Table 15, and was used to generate s-boxes 7 to 10. They show a clear improvement in robustness, which is now almost twice that of s-boxes 1 and 2. The generation time has also been improved with an s-box produced on average in about 37 minutes.

7.2 Lowering the L Value

It was noted that a reduction in the N values of the s-boxes was primarily responsible for the improvement in robustness, while the L values were fairly consistent with those of earlier s-boxes. The most recent group of s-boxes was assessed to determine where the L values occur in the DDT. The results are given in Table 16. This shows that slightly more than half of the L values occur in positions corresponding to the primary selection criteria, i.e. a 1-bit input change giving a 3-bit output change. In order to reduce the L value it would seem to be beneficial to spread the output changes corresponding to a 1-bit

Level	Tests used in level
1	Ch1_3, Chr1_3, Chmid
2	Ch1_3, Chr1_3
3	Ch1_3, Chr1_3b
4	Ch1_3b, Chr1_3b
5	Ch1_3b, Chr1_2
6	Ch1_2, Chr1_2
7	Ch1_2b, Chr1_2b
8	Ch1_2b, Chr1_1
9	Ch1_1

Table 15: Revised Test Hierarchy

input change over a greater number of values. The DES-type SAC permits output changes that consist of 3 or more bits. However, when introducing the Ch1_3b and Chr1_3b criteria it was found that having them in level 1 of the hierarchy had a detrimental effect on the resulting s-boxes, giving a typical robustness of approximately 0.15. Some s-boxes were generated with the level 1 criteria permitting a 3 or 4-bit change in the output. All other criteria were the same as those used for s-boxes 7 to 10 in Table 14. The characteristics of the s-boxes produced are given in Table 17. The most striking feature of these s-boxes is that the robustness has dropped back to the levels found in Section 5.3. Closer inspection reveals that this has been caused by an increase in N. However, the desired effect of dropping the L values has been achieved. S-boxes 2 and 4 are particularly notable as their L values are almost as low as those found for the random s-boxes in Section 5.1. The SAC compliance has dropped to disappointing levels, while the DES-type SAC levels are similar to previous examples. The level of 'mid' compliance has improved by about 5%.

Since the results in Table 17 are poor in terms of overall robustness, it is useful to determine why this has occurred. Towards this end, it is useful to determine the results when a 3 or 5-bit output change is permitted for a 1-bit input change. Several s-boxes were generated with this condition as the primary criteria. All other criteria were kept the same. The results are given in Table 18. The L values in these s-boxes are not as low as those in Table 17, but are lower than those of boxes 7 to 10 in Table 14. DES-type SAC compliance and 'mid' compliance fall into similar ranges to those of the last four s-boxes in Table 14. Normal SAC compliance levels are an improvement on those given in Table 17, but are lower than in Table 14. The robustness of the s-boxes in Table 18 covers a wider range than the final four s-boxes in Table 14, with a higher upper limit. The average generation time was approximately 14 minutes for these s-boxes. This is a significant improvement on the earlier

Box	L value	Position in the DDT	
		Input Change (bits)	Output Change (bits)
1	50	1	3
2	46	4	2
3	44	2	0
		1	3
4	48	2	0
5	44	1	3
6	42	3	1
		1	3
		1	3
		2	6
7	46	1	3
		1	3
8	54	1	3
9	52	2	0
10	44	3	3

Table 16: Location of the L values

s-boxes, which were taking over 30 minutes to generate.

8 Conclusions

An investigation has been conducted into the generation of substitution boxes (s-boxes) using heuristic techniques. They involve randomly choosing a value and then testing it against a set of criteria to determine if it is suitable for inclusion in the s-box. The objective is to generate s-boxes that meet the strict avalanche criteria (SAC), are non-linear, and have a high degree of resistance to differential cryptanalysis. The primary focus of this investigation was on s-boxes with 9-bit inputs and 6-bit outputs. A large size (compared to other s-boxes in use, e.g. DES) is preferable as it reduces the probability of the s-box containing linearities. This size was chosen, as it is the largest size that can be conveniently tested. Adding another input bit to the s-box increases the testing time requirements by at least a factor of 4.

It was found that the probability of obtaining an s-box that fully complies with the SAC using random methods is very low. However, a method was found that produces s-boxes which exhibit up to 85% compliance with the SAC and several were found to give over 90% compliance with the DES-type SAC. Controlling the robustness of the s-boxes was found to be considerably more difficult. Although the potential influence of an individual element on

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)
1	0.1338	438	38	52.86	94.92	71.48
2	0.1410	435	32	51.22	94.40	76.56
3	0.1537	427	38	51.43	95.31	76.95
4	0.1428	434	32	51.30	94.92	72.27

Table 17: Characteristics of the S-boxes using a Level 1 Criteria Permitting 3 or 4-bit Output Change

S-box	R	N	L	SAC Compliance (%)	DES-type SAC Compliance (%)	mid (%)
1	0.2689	362	42	69.44	95.62	62.50
2	0.2690	360	48	70.88	95.79	62.11
3	0.3223	333	40	72.14	96.53	63.28
4	0.3084	340	42	72.01	95.96	67.58
5	0.2905	350	42	70.75	96.22	52.34
6	0.3385	324	40	72.44	97.66	69.92
7	0.2773	358	40	70.66	96.61	66.41

Table 18: S-boxes with a Level 1 Criteria Permitting a 3 or 5-bit Output Change

s-box behaviour can be estimated when testing for SAC compliance, there is no easy way to determine the cumulative effects of all the elements during the construction process. Hence, although a large percentage of the elements may meet the SAC it is possible that undesirable characteristics may come about that are not detectable until the s-box is tested. For example, there may be a large number of instances where if a particular input bit is changed the same 3-bit output change will result. Although it is desirable that a 1-bit input change results in a 3-bit output change, if enough of these input/output pairs group together in the same position in the DDT, it will have a detrimental effect on robustness. Some methods were found that had a positive effect on robustness. These were generally based on adjusting the criteria used to select an element, unlike improvements to the SAC compliance, which were largely based on changes to the overall selection strategy. For example, Section 7.2 showed that by allowing a 1-bit input change to result in either a 3 or 4-bit output change, the L value could be lowered significantly. This result can be easily explained by the fact that this criteria allows many more output

change patterns, hence reducing the DDT values corresponding to 3-bit output changes. Such positions often contain the L value. Unfortunately, this was accompanied by a significant increase in the N, value resulting in an overall detrimental effect. This effect cannot easily be explained. Similarly, by allowing 3 or more output bits to change for a 1-bit input change as a second level criteria, the N value was lowered. This effect also cannot be explained easily. This destabilised the L values, which in some instances resulted in lower values, giving an overall positive effect.

There are other desirable s-box features that have not been considered here, many of which are application dependent. However this investigation has shown that it is possible to generate s-boxes, in a random manner, that can meet desirable criteria to a high degree.

References

[1] E. Biham and A. Shamir, Differential cryptanalysis of DES-like cryptosystems, in *Advances in Cryptology – Crypto '90*, Springer-Verlag Lecture Notes in Computer Science, 537 (1991), 2–21.

[2] Electronic Frontier Foundation, Press release January 19, 1999, `http://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19990119_deschallenge3.html`.

[3] F. Hendessi, T.A. Gulliver and A.U.H. Sheikh, Large s-box design using a converging method, in *Proc. IEEE Int. Symp. on Inform. Theory*, (1997), 177.

[4] M. Matsui, Linear cryptanalysis method for DES cipher, in *Advances in Cryptology – Eurocrypt '93*, Springer-Verlag Lecture Notes in Computer Science, 765 (1994), 386–397.

[5] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Ed.*, Wiley, New York, 1996.

[6] J. Seberry, X. Zhang and Y. Zheng, Pitfalls in designing substitution boxes, in *Advances in Cryptology – Crypto '94*, Springer-Verlag Lecture Notes in Computer Science, 839 (1994), 383–396.

Received: January 9, 2008