DES S-box Generator

Lauren De Meyer[1] and Serge Vaudenay[2]

lauren.demeyer@student.kuleuven.be

serge.vaudenay@epfl.ch

[1] KU Leuven, Belgium

[2] EPFL, Switzerland

Abstract. The Data Encryption Standard (DES) is a cryptographic algorithm, designed by IBM, that was selected to be the national standard in 1977 by the National Bureau of Standards. The algorithm itself was entirely published but the design criteria were kept secret until 1994 when Coppersmith, one of the designers of DES, published them. He states that the IBM team already knew about the attack called Differential cryptanalysis during the design of the algorithm and that it had an effect on choosing the S-boxes. To be more specific, he mentions eight design criteria that all the S-boxes of DES are based on. How the S-boxes were generated is a mystery, as the legend says this was outsourced to the NSA. Indeed, building a set of S-boxes respecting these criteria is a non-trivial task.

In this paper we present an efficient S-box generator respecting all criteria and even more. Coppersmith's design criteria served as a basis but were strengthened for better resistance to Linear Cryptanalysis.

While other researchers have already proposed S-box generators for DES satisfying either non-linearity or good diffusion, our generator offers both. Moreover, apart from suggesting a new set of 8 S-boxes, it can also very quickly produce a large pool of S-boxes to be used in further research.

Keywords: DES, S-box, Nonlinearity, Differential Property, Linear Property

1 Introduction

When IBM published the DES-algorithm [5] without revealing the design considerations, many people speculated there to be a hidden weakness in the algorithm. This was mostly due to the presence of mysterious S-boxes without any reference on how they were generated. However, when Biham and Shamir [1] demonstrated an attack against DES in 1989 using a technique called differential cryptanalysis, IBM claimed that this attack was known to the designers of DES and that the design criteria for the DES S-boxes contributed to the defense against ‘differential cryptanalysis’. Biham and Shamir further noticed that any variation in the set of S-boxes (even the same ones in a different order) led to a much lower attack complexity.

In 1994, one of the designers of the algorithm, Coppersmith, released a paper [3, p.247], in which he presented a list of eight criteria for the S-boxes, claiming that these criteria were used for the creation of the eight original DES S-boxes. An S-box is a substitution box and it is the only non-linear component in the cipher. Its main purpose is to obscure the relationship between the key, the plaintext, and the ciphertext.

In related works, other sets of DES-like S-boxes have been proposed. In [6, p.71-72], Kim incrementally constructs each output bit of an S-box as a new Boolean function. The proposed S-boxes in his set s^2DES have good differential and linear properties but don’t satisfy the other DES diffusion criteria, even though they are mentioned in the paper. For instance, $S_1(001000) = B$ and $S_1(001001) = 9$ therein criterion (S-4) (as later defined) is not satisfied: The Hamming distance between both the outputs and the inputs is 1. The other proposed set s^5DES in [7, p.157] does comply with the diffusion criteria but instead, the differential and linear properties are not satisfactory. For instance, $DP^{S_1}(20,6) = \frac{9}{32}$ therein criterion (S-7) (as later defined) is not satisfied. Indeed, both the diffusion and non-linearity criteria are quite demanding and it is difficult to satisfy both at the same time.

This paper describes in detail the development of an algorithm that can produce S-boxes satisfying all design criteria of IBM and even more. The main idea is to use graphs with adjacencies based on the DES criteria. Building each graph based on the result from a previous one, we repeatedly combine smaller components to produce bigger ones. Typically, we construct classes of functions mapping 2, 4, then 6 bits to 4 bits. This approach is substantially different from that of Kim in [6, p.66] but makes the algorithm very efficient. Our method produces a complete set of S-boxes on a computer in roughly 1 minute, using less than 17MB of memory. The results show how the DES S-boxes may have been generated with computational resources in the 1970’s.

This paper starts with a discussion of Coppersmith's design criteria in Section 2. Here, we also define a set of criteria for the smaller S-boxes. In Section 3, the methodology and structure of the algorithm is described in detail and finally, we mention implementation details in Section 4.

2 On the Data Encryption Standard

The Data Encryption Standard is a Feistel cipher, in which the round function consists of an expansion, a bitwise XOR-operation with the round key, an S-box layer and a permutation. This research concentrates on the S-box layer, which consists of 8 different parallel S-boxes. Every S-box transforms 6 bits of input to an output of 4 bits:

$$S : \{0,1\}^6 \to \{0,1\}^4 : x \to S(x)$$

The 8 Standard DES-Sboxes of IBM were published together with the algorithm in 1977, but the criteria were only disclosed 15 years after.

2.1 The S-box Design Criteria

The design criteria for S-boxes as described in [3, p.247] are as follows:

(S-1) *Each S-box has six bits of input and 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. (That is, if we select any output bit position and any subset of the six input bit positions, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.)*

(S-3) *If we fix the leftmost and rightmost input bits of the S-box and vary the four middle bits, each possible 4-bit output is attained exactly once as the middle four input bits range over their 16 possibilities.*

(S-4) *If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits. (That is, if $h(\Delta I_{i,j}) = 1$, then $h(\Delta O_{i,j}) \geq 2$, where $h(x)$ is the Hamming weight of x.)*

(S-5) *If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits. (If $\Delta I_{i,j} = 001100$, then $h(\Delta O_{i,j}) \geq 2$.)*

(S-6) *If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same. (If $\Delta I_{i,j} = 11xy00$, where x and y are arbitrary bits, then $\Delta O_{i,j} \neq 0$.)*

(S-7) *For any nonzero 6-bit difference between inputs, $\Delta I_{i,j}$, no more than eight of the 32 pairs of inputs exhibiting $\Delta I_{i,j}$ may result in the same output difference $\Delta O_{i,j}$.*

(S-8) *Similar to (S-7), but with stronger restrictions in the case $\Delta O_{i,j} = 0$, for the case of three active S-boxes on round i: Define*

$$\begin{aligned}
q_{0,j} &= \max_{c,d}(\Pr[\Delta O_{i,j} = 0 | \Delta I_{i,j} = 00cd11]),\\
q_{1,j} &= \max_{g,h}(\Pr[\Delta O_{i,j} = 0 | \Delta I_{i,j} = 11gh10]),\\
q_{2,j} &= \max_{k,m}(\Pr[\Delta O_{i,j} = 0 | \Delta I_{i,j} = 10km00]).
\end{aligned}$$

Arrange S-boxes so as to minimize $\max_{j \in \{1,2,\ldots,8\}}(q_{0,j} q_{1,j+1} q_{2,j+2})$.

In this description from [3, p.247], $\Delta I_{i,j}$ ($\Delta O_{i,j}$) is the input difference (resp. output difference) of S-box S_j in round i.

We remark that criterion (S-7) hardens DES against Differential Cryptanalysis [1]. We recall the definition of the Differential Property of a function f (with notations from [11, p.56]):

Definition 1 (Differential Property). *Given a function f, we define*

$$DP^f(a,b) = \Pr[f(X \oplus a) \oplus f(X) = b]$$

$$DP^f_{max} = \max_{a \neq 0, b} DP^f(a,b)$$

Therefore, another way to define (S-7) is

(S-7) $DP^S_{max} \leq \frac{16}{64}$

Coppersmith doesn't mention a criterion regarding Linear Cryptanalysis [9]. (As shown below, (S-2) partially covers it.) Therefore, we add an extra criterion (S-9) to make sure our S-boxes' linear properties are satisfactory and to make the program's execution more efficient.

Definition 2 (Linear Property). *Given a function f, we define*

$$LP^f(a,b) = (2\Pr[a \cdot X = b \cdot f(X)] - 1)^2$$
$$LP^f_{max} = \max_{a,b \neq 0} LP^f(a,b)$$

(S-9) $LP^S_{max} \leq (\frac{28}{64})^2$

This bound is the lowest from the existing 8 DES Sboxes' LP_{max} values (see Table 4). Only one of the standard DES S-boxes satisfies this bound.

Given that in Def. 2: $a \cdot X = \oplus_{i=0}^{5} a_i X_i$, note that (S-2) can also be written as follows:

$$\forall a \in \{0,1\}^6, \forall b \in \{0,1\}^6, h(b) = 1: \text{ minimize } |\Pr[a \cdot X = b \cdot S(X)] - \frac{1}{2}|$$

which is equivalent to minimizing $LP^S(a,b)$ when $h(b) = 1$. Thanks to (S-9), we know this value will at least be smaller than $(\frac{28}{64})^2$. However we will try to minimize it further by requiring $\max_{a,h(b)=1} LP^S(a,b) \leq (\frac{26}{64})^2$. Therefore, (S-2) can be rewritten as follows, with the bound explicitely set to $(\frac{26}{64})^2$:

(S-2) $LP^S_{max(1)} \leq (\frac{26}{64})^2$

where

$$LP^S_{max(1)} = \max_{a,h(b)=1} LP^S(a,b) \tag{1}$$

From the standard S-boxes, only 3 satisfy this criterion so it is more severe than the one proposed by Coppersmith (see Table 4).

2.2 The Permutation Design Criteria

Due to (S-3), each 6×4 S-box can be naturally split into four 4×4 S-boxes (rows), where the leftmost and rightmost input bits of the big S-box are used to select one of the 4 smaller S-boxes. Therefore, we can make a distinction between criteria that are already applicable on these smaller S-boxes and those that can only be evaluated for 6×4 S-boxes.

A 4×4 S-box is a 4-bit permutation as prescribed by criterion (S-3). In the further discussion, this criterion will not be given special attention, since creating a 4×4 S-box will imply it being a permutation. As mentioned earlier, the leftmost and rightmost input bits a and b of a 6×4 S-box select one of the 4×4 S-boxes for which the four middle bits x are the input. I.e., $P_{a,b}(x) = S(a\|x\|b)$. Since only the two middle input bits are varied in (S-5), this criterion can be completely verified for 4×4 S-boxes. If all permutations that are used to generate a 6×4 S-box comply with this criterion, then it is not necessary to test the generated 6×4 S-box for this criterion.

We can now try to establish the criteria for 4×4 S-boxes. Some of the criteria (like (S-4) and (S-5)) automatically imply a criterion for a permutation. We will also attempt to find criteria for permutations equivalent to (S-7) and (S-9).

Criterion (S-2) requires that no output bit should be too close to a linear combination of any subset of the six input bits. As the four input bits of a permutation are a subset of the six input bits of the S-box, we can demand the same for any subset of these four input bits for the permutation. Again, the nonlinearity of these output bits is related to the permutation's Linear Property.

To define equivalent criteria to (S-7) and (S-9), we need to choose upper bounds for the permutation's DP_{max} and LP_{max}. Leander and Poschmann [8, p.163] have defined the following conditions for an **Optimal 4-bit Sbox S**:

1. S is a bijection

2. $LP^S_{max} = (\frac{8}{16})^2$
3. $DP^S_{max} = \frac{4}{16}$

Unfortunately, there exist no S-boxes that satisfy these bounds while also suiting the DES criteria. Therefore we choose our own bounds such that there exists a sufficient number of permutations that also satisfy Coppersmith's criteria. The criteria for differential and linear properties are defined in (P-2) and (P-6):

(P-1) Each permutation has four bits of input and four bits of output. (implied by (S-1))

(P-2) $LP_{max} \leq (\frac{12}{16})^2$

(P-3) If we vary the four input bits, each possible 4-bit output is attained exactly once. (implied by (S-3))

(P-4) If two inputs to a permutation differ in exactly one bit, the outputs must differ in at least two bits: If $h(\Delta x) = 1$, then $h(\Delta P(x)) \geq 2$, where $h(x)$ is the Hamming weight of x. (Implied by (S-4))

(P-5) If two inputs to a permutation differ in the two middle bits exactly, the outputs must differ in at least two bits: If $\Delta x = 0110$, then $h(\Delta P(x)) \geq 2$. (Implied by (S-5))

(P-6) For any nonzero 4-bit difference between inputs, Δx, no more than three of the 8 pairs of inputs exhibiting Δx may result in the same output difference $\Delta P(x)$: $DP_{max} \leq \frac{6}{16}$

Note that the standard S-boxes' permutations satisfy (P-2) (see Table 2) but only S_4 satisfies (P-6) (see Table 1).

Theorem 1. *Let S be a* 6×4 *S-box and let* $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$, *be the corresponding permutations defined by* $P_{a,b}(x) = S(a||x||b)$.

- *S satisfies (S-1) is equivalent to* $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, *and* $P_{1,1}$ *satisfy (P-1).*
- *S satisfies (S-3) is equivalent to* $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, *and* $P_{1,1}$ *satisfy (P-3).*
- *S satisfies (S-4) implies that* $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, *and* $P_{1,1}$ *satisfy (P-4).*

- *S satisfies (S-5) is equivalent to $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, and $P_{1,1}$ satisfy (P-5).*

There is no equivalence between (S-4) and (P-4), but (P-4) is a necessary condition. There is no direct link between (S-2) (or (S-9)) and (P-2), and neither between (S-7) and (P-6). However, (P-4) and (P-6) increase the chances to build S-boxes satisfying these criteria. In what follows, we present an algorithm to give an exhaustive list of permutations satisfying (P-1), (P-3), (P-4), (P-5). Then, it is easy to filter this list based on (P-2) and (P-6). The remaining task consists of assembling these permutations by quadruplet in order to build an S-box.

Proof. The properties for (P-1) and (P-3) are trivial.

For (P-4), we observe that with a and b fix, if $h(\Delta x)) = 1$, then $h(\Delta(a\|x\|b)) = 1$. So, we must have $h(\Delta S(a\|x\|b)) \geq 2$ due to (S-4). This can be written $h(\Delta P_{a,b}(x)) \geq 2$. So, (P-4) is satisfied.

The proof for (P-5) is similar. □

3 Our Generator

3.1 Finding 4×4 S-boxes

In order to create permutations in an efficient way we make use of 3 graphs. When vertices are connected by an edge, we say they are *compatible*. The compatibility criteria are based on Permutation design criteria (P-4) and (P-5). These criteria imply that for a valid permutation P:

$$\forall x \in \{0,1\}^4 \; \forall \Delta \in \{1,2,4,6,8\} \; h(P(x) \oplus P(x \oplus \Delta)) \geq 2. \tag{2}$$

Step 1. G_1 is a graph of size $2^4 = 16$, containing all nibbles who are compatible if their distance is 1. I.e., the hypercube of dimension 4 in which we additionally connect all pairs of nibbles that have a xor equal to 6 (which is important for criterion (P-5)). The result is a graph shown in Table 1. Note that nibbles connected by borders are also compatible. For example, 0 is connected to $\{1,2,4,6,8\}$ as 15 is connected to $\{7,9,11,13,14\}$. So, each row of G_1 is a 4-clique and each column is a 4-cycle.

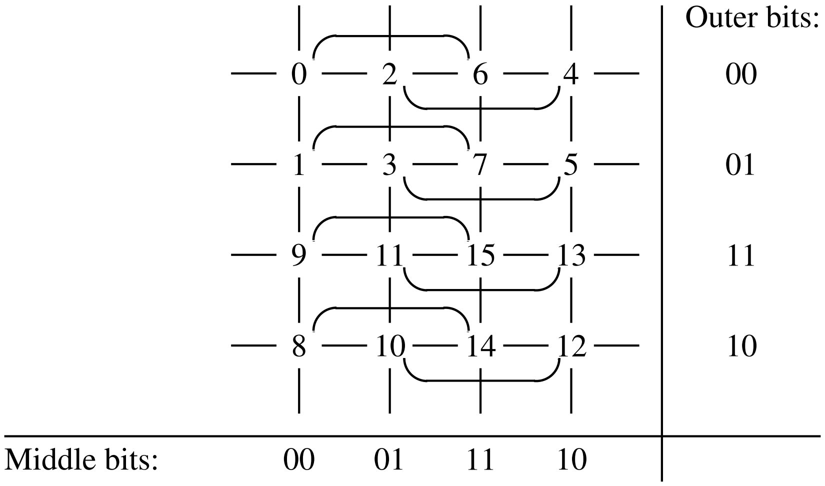

Fig. 1: Graph G_1

Step 2. To find a permutation P we create a second graph, G_2. The vertices of this graph are also nibbles (so, G_2 has 16 vertices) but in this case, they are connected if their Hamming distance is at least 2. We recall the definition of a graph homomorphism.

Definition 3 (Graph homomorphism). *A graph homomorphism $f : G \to G'$ from a graph $G = (V,E)$ to a graph $G' = (V',E')$ is a mapping $f : V \to V'$ from the vertex set of G to the vertex set of G' such that $(u,v) \in E$ implies $(f(u), f(v)) \in E'$.*

Due to (2), the permutations are graph homomorphisms from G_1 to G_2. Since they are 1-to-1 functions, they map each row of G_1 (corresponding to 2 outer bits) to a 4-vertex clique of G_2. By analyzing G_2, we find 228 such 4-cliques.

Given a permutation P, for $a, b \in \{0,1\}$, we introduce the mapping f_{ab} :

$$f_{ab} : \{0,1\}^2 \mapsto C_{ab} : f_{ab}(x,y) = P(a,x,y,b) \tag{3}$$

Note that C_{ab} is a 4-clique of G_2. A 4-clique of G_2 is thus a class of functions from $\{0,1\}^2$ to $\{0,1\}^4$ having the same output set and to which f_{ab} belongs: we map the two middle bits of the S-box' input to the output.

Step 3. We create G_3, a graph with the 4-cliques of G_2 as vertices. This graph has thus 228 vertices. We define two vertices C and C' to be compatible if and only if they are disjoint and there exists a one-to-one mapping $\pi : C \mapsto C'$ such that $\forall x \in C$ the Hamming distance between x and $\pi(x)$ is at least 2. I.e., there is a perfect matching between C and C' in G_2. Note that given a permutation P, the permutation $\pi_{aba'b'} = f_{a'b'} \circ f_{ab}^{-1}$ from C_{ab} to $C_{a'b'}$ is such a one-to-one mapping from C_{ab} to $C_{a'b'}$ when $h(ab \oplus a'b') = 1$ due to (2):

$$\forall x \in \{0,1\}^2 \ \forall a,b,a',b' \in \{0,1\} \ h(x \oplus \pi_{aba'b'}(x)) \geq 2 \text{ when } h(ab \oplus a'b') = 1$$

As the existence of such a mapping indicates adjacency in the graph, a permutation P defines a 4-vertex cycle in G_3 as in Fig. 2. A 4-cycle of G_3 is thus a class of permutations. This is summarized as follows.

Theorem 2. *If P is a 4×4-permutation satisfying (P-1), (P-3), (P-4), and (P-5), then $(C_{00}, C_{01}, C_{11}, C_{10})$ is a 4-cycle of G_3, where $C_{ab} = \{P(a\|x\|b); x \in \{0,1\}^2\}$.*

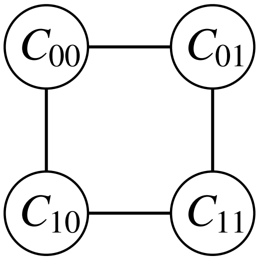

Fig. 2: DES Permutation as a cycle in G_3

By analyzing the graph G_3, we find 6 281 4-vertex cycles $\{C_1, C_2, C_3, C_4\}$ that can represent a 4-cycle of a permutation $P(a,x,y,b) = C_{ab}(x,y)$. To find these permutations, the 4-cycles still have to be mapped in the right way. Each cycle can correspond to 8 assignments of $\{C_{00}, C_{01}, C_{11}, C_{10}\}$. Each vertex $C_{ab} = \{x_0, x_1, x_2, x_3\}$ in the cycle can be permuted $4! = 24$ times. Therefore, every

4-cycle in G_3 can define $8 \times (4!)^4$ 4-bit permutations, but not all of them satisfy (P-4) and (P-5). We exhaustively check all arrangements and all permutations. If there exists a valid sequence of mappings $C_{00} \to C_{01} \to C_{11} \to C_{10} \to C_{00}$, we construct the permutation P defining this cycle. This way, we find 60 834 432 4-bit permutations that comply with criteria (P-1), (P-3), (P-4) and (P-5). This is an exhaustive list.

Step 4. For the resulting permutations, criteria (P-2) and (P-6) still need to be verified.

First, criterion (P-6) limits the number of times one input difference can lead to the same output difference for differential properties. To verify this we create a permutation's XOR table (or Differential Distribution table) and check its maximum value. For the permutations, we derived an initial DP_{max} value $\frac{10}{16}$ from the DES permutations' DP_{max} values. (Table 1). However, we decided to make the criterion more severe and to require $DP_{max} \leq \frac{6}{16}$. [3]

Permutations in	DP_{max}	Permutations in	DP_{max}
S_1	$\frac{8}{16}, \frac{8}{16}, \frac{8}{16}, \frac{8}{16}$	S_5	$\frac{8}{16}, \frac{6}{16}, \frac{6}{16}, \frac{6}{16}$
S_2	$\frac{6}{16}, \frac{8}{16}, \frac{8}{16}, \frac{6}{16}$	S_6	$\frac{4}{16}, \frac{8}{16}, \frac{6}{16}, \frac{6}{16}$
S_3	$\frac{8}{16}, \frac{8}{16}, \frac{8}{16}, \frac{8}{16}$	S_7	$\frac{8}{16}, \frac{8}{16}, \frac{6}{16}, \frac{8}{16}$
S_4	$\frac{6}{16}, \frac{6}{16}, \frac{6}{16}, \frac{6}{16}$	S_8	$\frac{6}{16}, \frac{10}{16}, \frac{6}{16}, \frac{8}{16}$

Table 1: DP_{max} values of the standard DES permutations

Finally, we use the permutations' Linear Approximation Tables to check their Linear Property (P-2). Since we already constructed the Difference Distribution Tables and because we only need the magnitude of the entries of the Linear Approximation Table, we may use the Walsh-Hadamard Transform to obtain it [2, p.359]. The permutation is deemed valid if the maximum in this table doesn't exceed $(\frac{12}{16})^2$. This value was again derived from the LP_{max} values of the DES

[3] We tried using the optimal DP_{max} bound $\frac{4}{16}$ and while it is possible to generate S-boxes this way, the number of resulting S-boxes is much lower and their differential properties are not better. For more information, see appendix B

permutations, as for all 32 standard permutations we have $LP_{max} = (\frac{12}{16})^2$ (see Table 2). Moreover, there are no permutations found with a lower bound that also satisfy the other criteria.

Permutations in	LP_{max}	Permutations in	LP_{max}
S_1	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$	S_5	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$
S_2	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$	S_6	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$
S_3	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$	S_7	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$
S_4	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$	S_8	$(\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2, (\frac{12}{16})^2$

Table 2: LP_{max} values of the standard DES permutations

We can classify the 60 834 432 permutations satisfying (P-1),(P-3),(P-4) and (P-5) according to their DP_{max} and LP_{max} values. Table 3 shows the number of permutations that can be found satisfying each combination of DP_{max} and LP_{max}. The permutations used in regular DES S-boxes are all situated in the first four rows of column one. Our generator will only consider the permutations from the first two rows. We then obtain an exhaustive list $\mathcal{P}$ of 1 069 056 permutations satisfying (P-1) to (P-6).

$\frac{LP_{max}\rightarrow}{DP_{max}\downarrow}$	$(\frac{12}{16})^2$	$(\frac{16}{16})^2$
$\frac{4}{16}$	36 864	0
$\frac{6}{16}$	1 032 192	0
$\frac{8}{16}$	1 732 608	25 092 096
$\frac{10}{16}$	368 640	11 599 872
$\frac{12}{16}$	73 728	14 991 360
$\frac{16}{16}$	49 152	5 857 920

Table 3: Number of permutations for each combination of (DP_{max}, LP_{max})

3.2 Creating 6×4 S-boxes

In the previous section, we found the set of all possible 4×4 S-boxes $\mathcal{P}$. We proceed by combining compatible permutations to form S-boxes $\{P_{00}, P_{01}, P_{10}, P_{11}\}$ such that $S(a\|x\|b) = P_{ab}(x)$ and verifying the remaining criteria. Therefore, we make other graphs G_4 an G'_4, in which vertices are permutations. We can define two compatibility criteria, namely based on (S-4) (we have seen in Th. 1 that (P-4) is a necessary but not sufficient condition for (S-4), so (S-4) is not fully guaranteed so far) and based on (S-6) (which is independent from (P-1)–(P-6)).

In G'_4, we define compatibility between P and P' as follows:

$$\forall x \in \{0,1\}^4 \; h(P(x) \oplus P'(x)) \geq 2 \tag{4}$$

Pairs of permutations for which this is the case are connected by an edge in G'_4.

We also consider the following property between P and P':

$$\forall x \in \{0,1\}^4 \; \forall y \in \{0,1\}^2 \; P(x) \neq P'(x \oplus (1\|y\|0)) \tag{5}$$

Edges of G'_4 satisfying this property are edges in G_4.

We have the following results.

Lemma 1 *Let S be a 6×4 S-box and let $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$, be the corresponding permutations defined by $P_{a,b}(x) = S(a\|x\|b)$. S satisfies (S-4) is equivalent to $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, and $P_{1,1}$ satisfying (P-4) with $(P_{0,0}, P_{0,1}, P_{1,1}, P_{1,0})$ a 4-cycle of G'_4.*

Proof. Clearly, (S-4) is equivalent to the two following conditions:

- for all a, b, $P_{a,b}$ satisfies (P-4);
- for all a, b, a', b' such that $h(ab \oplus a'b') = 1$, we have that $P_{a,b}$ and $P_{a',b'}$ satisfy (4).

The latter condition is equivalent to $(P_{0,0}, P_{0,1}, P_{1,1}, P_{1,0})$ being a 4-cycle of G'_4.

Lemma 2 *Let S be a* 6×4 *S-box and let* $P_{0,0}$, $P_{0,1}$, $P_{1,0}$, $P_{1,1}$, *be the corresponding permutations defined by* $P_{a,b}(x) = S(a\|x\|b)$. *S satisfies (S-6) is equivalent to the pairs of permutations* $\{P_{0,0}, P_{1,0}\}$ *and* $\{P_{0,1}, P_{1,1}\}$ *satisfying relation* (5).

Proof. Clearly, (S-6) is equivalent to that for all a and b, $P_{a,b}$ and $P_{\bar{a},b}$ satisfy (5). □

We depict edges in G_4 by a double line and edges in G'_4 by a simple line. Fig. 3 represents how the four permutations of an S-box are connected in G_4 and G'_4. So, we conclude as follows.

Theorem 3. *We consider the* $S \leftrightarrow (P_{0,0}, P_{0,1}, P_{1,1}, P_{1,0})$ *correspondence defined by* $P_{a,b}(x) = S(a\|x\|b)$. *The* 6×4*-S-box S satisfies (S-1), (S-3), (S-4), (S-5), and (S-6) if and only if* $P_{0,0}$, $P_{0,1}$, $P_{1,1}$, *and* $P_{1,0}$ *satisfy (P-1), (P-3), (P-4), and (P-5), and are vertices of* G_4 *and* G'_4 *connected as on Fig. 3.*

Proof. If S satisfies (S-1), (S-3), (S-4), (S-5), and (S-6), by Th. 1, every $P_{a,b}$ satisfies (P-1), (P-3), (P-4), and (P-5). Furthermore, (4) and (5) are satisfied due to Th. 1. So, the $P_{a,b}$ are connected as on Fig. 3, by definition of G_4 and G'_4.

If now every $P_{a,b}$ satisfies (P-1), (P-3), (P-4), and (P-5), by Th. 1, S satisfies (S-1), (S-3), and (S-5). If they are connected as on Fig. 3, (4) and (5) are satisfied. So, due to Th. 1, S further satisfies (S-4) and (S-6). □

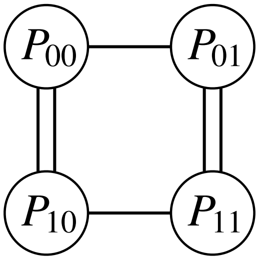

Fig. 3: DES Sbox

Edges of G_4 are pairs of permutations. These pairs represent either $\{P_{00}, P_{10}\}$ or $\{P_{01}, P_{11}\}$ and will be the vertices of the last graph G_5. In this graph, adjacency means the existence of a connection as in Fig. 3. More formally, two vertices τ and τ' of G_5 are connected if and only if τ and τ' are non-intersecting edges of G'_4 and there exists a perfect matching between τ and τ'. Finally, an edge of G_5 defines 4 possible S-boxes due to the possible permutations of the elements $(P_{00}, P_{01}, P_{11}, P_{10})$. Since these 4 S-boxes are quite similar and because given one S-box, the other 3 can easily be constructed (by changing the row order), the generator only returns one of them.

While the preceding methods were deterministic, the remaining part of the algorithm is non-deterministic, as the total number of permutations is too large to put in a graph. Therefore we only create a subgraph Σ_4 of G_4 by choosing random permutations in $\mathcal{P}$ until we find m edges in Σ_4 (i.e., pairs of permutations that are connected with a double line). Then, we use the m edges from Σ_4 to define m vertices of the subgraph Σ_5 of G_5. Note that m is a parameter that can be chosen arbitrarily. The higher this parameter, the more S-boxes you can find and the more time the program needs to complete. We will see that the value $m = 10000$ is a good choice if you want to obtain 8 S-boxes.

Sbox	DP_{max}	LP_{max}	$LP_{max(1)}$
S_1	$\frac{16}{64}$	$(\frac{36}{64})^2$	$(\frac{28}{64})^2$
S_2	$\frac{16}{64}$	$(\frac{32}{64})^2$	$(\frac{28}{64})^2$
S_3	$\frac{16}{64}$	$(\frac{32}{64})^2$	$(\frac{28}{64})^2$
S_4	$\frac{16}{64}$	$(\frac{32}{64})^2$	$(\frac{20}{64})^2$
S_5	$\frac{16}{64}$	$(\frac{40}{64})^2$	$(\frac{28}{64})^2$
S_6	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_7	$\frac{16}{64}$	$(\frac{36}{64})^2$	$(\frac{36}{64})^2$
S_8	$\frac{16}{64}$	$(\frac{32}{64})^2$	$(\frac{24}{64})^2$

Table 4: Properties of the standard DES S-boxes

Now all that is left is to verify the resulting S-boxes with (S-2), (S-7) and (S-9) as (S-1) and (S-3) to (S-6) are satisfied by construction, due to Th. 3. Table 4 shows the corresponding properties for the 8 DES S-boxes. According to our criterion (S-9) we decide that an S-box is rejected when its LP_{max} exceeds $(\frac{28}{64})^2$, which is thus more severe than for DES. The differential criterion is identical to that of Coppersmith: We require $DP_{max} \leq \frac{16}{64}$. Finally, we check that $LP_{max(1)} \leq (\frac{26}{64})^2$ for (S-2) as defined by (1).

How many valid S-boxes would we find if we could create G_4 and G_5 completely? We try to find this $\underline{\text{number}}$ by approximating the $\underline{\text{number}}$ of edges in G_4 and G_5.

Firstly, note that $\Pr[h(\Delta) \geq 2 | \Delta \in \{0,\ldots,15\}] = \frac{11}{16}$ and $\Pr[P(x) \neq P'(x')] = \frac{15}{16}$. The probability that two permutations P and $P' \in \mathcal{P}$ form an edge in G'_4 is the probability that the pair satisfies (4):

$$p'_4 = \Pr[(P,P') \in \mathcal{E}(G'_4)] \approx \left(\frac{11}{16}\right)^{2^4} \approx 2^{-8.6}$$

This is for a random function. We can check that it is also correct for random permutations. But P and P' are taken from a special list of permutations and we observe in practice a larger $p'_4 = 2^{-6.74}$. The probability that two permutations P and $P' \in \mathcal{P}$ form an edge in G_4 is the probability that the pair satisfies (4) and (5):

$$p_4 = \Pr[(P,P') \in \mathcal{E}(G_4)] \approx p'_4 \cdot \left(\frac{15}{16}\right)^{2^{4+2}} \approx 2^{-12.7}$$

but we observe a larger $p_4 = 2^{-10.97}$ in practice. The probability that two vertices $\{P_1,P_2\}$ and $\{P'_1,P'_2\}$ in G_5 are adjacent is the probability that either $\{P_1,P'_1\}$ and $\{P_2,P'_2\}$ or $\{P_1,P'_2\}$ and $\{P_2,P'_1\}$ are edges of G'_4:

$$p_5 = \Pr[(\{P_1,P_2\},\{P'_1,P'_2\}) \in \mathcal{E}(G_5)] \approx 2 \cdot (p'_4)^2 \approx 2^{-12.5}$$

but we observe a larger $p_5 = 2^{-10.3}$ in practice. Finally, we experimentally found that an edge in G_5 forms a valid DES S-box with probability around $p_s = 2^{-11.74}$. Table 5 shows the resulting approximations for the number of graph edges and valid DES S-boxes.

# Vertices G_4' = # Vertices G_4	n	1 069 056	
# Edges G_4'	$\frac{n^2}{2} \cdot p_4'$	$\approx 2^{32.32}$	$p_4' = 2^{-6.74}$
# Edges G_4 = # Vertices G_5	$m = \frac{n^2}{2} \cdot p_4$	$\approx 2^{28.09}$	$p_4 = 2^{-10.97}$
# Edges G_5	$e = \frac{m^2}{2} \cdot p_5$	$\approx 2^{44.87}$	$p_5 = 2^{-10.3}$
# DES S-boxes	$\approx e \cdot p_s$	$\approx 2^{33.13}$	$p_s = 2^{-11.74}$

Table 5: Analysis of the total number of valid DES S-boxes

3.3 Ordering the S-boxes

By implementing Coppersmith's last criterion, we can obtain for 8 S-boxes the optimal order to use them in the DES round function. Let's recall criterion (S-8):

(S-8) Define

$$\begin{aligned} q_{0,j} &= \max_{c,d}(\Pr[\Delta O_{i,j} = 0 | \Delta I_{i,j} = 00cd11]), \\ q_{1,j} &= \max_{g,h}(\Pr[\Delta O_{i,j} = 0 | \Delta I_{i,j} = 11gh10]), \\ q_{2,j} &= \max_{k,m}(\Pr[\Delta O_{i,j} = 0 | \Delta I_{i,j} = 10km00]). \end{aligned}$$

Arrange S-boxes so as to minimize $\max_{j \in \{1,2,\ldots,8\}}(q_{0,j} q_{1,j+1} q_{2,j+2})$.

The probabilities in this expression can be found in an S-box' difference distribution table. Therefore, for each valid S-box i, we already store the three quantities $q_{0,i}$, $q_{1,i}$ and $q_{2,i}$ when checking Criterion (S-7).

Given 8 unordered S-boxes, we recursively calculate the above quantity for every order of the S-boxes. We update the current ordering whenever we find a better one.

4 Implementation

A summary of the algorithm can be seen below and a recap of the structure of all graphs in Table 6.

```
Construct $G_2$
Find all 4-cliques in $G_2$
Construct $G_3$
Find all 4-cycles in $G_3$
for each 4-cycle in $G_3$: do
    for each permutation that maps to this 4-cycle: do
        % By construction, the permutation satisfies (P-1), (P-3), (P-4) and (P-5)
        if criteria (P-2) and (P-6) satisfied: then
            Store permutation in $\mathcal{P}$
        end
    end
end
Construct a subgraph $\Sigma_4$ of $G_4$ as follows:
while # Edges in $\Sigma_4 < 10\,000$: do
    Pick random pair in $\mathcal{P}$
    if pair adjacent in $G_4$: then
        Add pair as a new edge in $\Sigma_4$
    end
end
Construct the subgraph $\Sigma_5$ of $G_5$ from $\Sigma_4$
for each edge in $\Sigma_5$: do
    Build S-box corresponding to edge
    % By construction, the S-box satisfies (S-1), (S-3), (S-4), (S-5) and (S-6)
    Verify criteria (S-2), (S-7) and (S-9)
end
if Goal = a set of 8: then
    Pick 8 S-boxes and order them to satisfy (S-8)
end
```

Algorithm 1: Summary of the algorithm

Although finding large cliques in a graph is a hard problem, finding 4-cliques in G_2 can be done in time $O(n^4)$, where n is the number of vertices in G_2. (Here, $n = 16$.)

Finding cycles is done in polynomial time, but the trivial algorithm to find 4-cycles in G_3 with complexity $O(n^4)$ (where n is the number of vertices in G_3) is good as well. (Here, $n = 228$.)

Graph	**Vertices**	**Edge**
G_1	v_i = nibble	E_{ij}: $h(v_i \oplus v_j) = 1$ or $v_i \oplus v_j = 6$
G_2	v_i = nibble	E_{ij}: $h(v_i \oplus v_j) \geq 2$
G_3	C_i = 4-Clique from G2	E_{ij}: $\exists \pi : C_i \to C_j : \forall x : h(x \oplus \pi(x)) \geq 2$
G'_4	P_i = permutation	E_{ij}: (4) satisfied for (P_i, P_j)
G_4	P_i = permutation	E_{ij}: (4) and (5) satisfied for (P_i, P_j)
G_5	$\{P_i^1, P_i^2\}$ = Edge from G4	E_{ij}: (4) satisfied for (P_i^1, P_j^1) and (P_i^2, P_j^2)

Table 6: Summary of graphs

The iterations are of course pruned by disconnected vertices, so the complexity might sooner be $O(nd^3)$ with d the degree of the graph. In G_3, the mean degree is about 77.5.

Given a 4-cycle in G_3, we have to explore $8 \times (4!)^4$ possible permutations. Not all of them satisfy criteria (S-4) and (S-5). Given that we have 6 281 cycles, this gives about 2^{34} permutations to explore. Again, the exact number of iterations is much less because there are often less than 4! possible mappings π between two cycles. After this, we are left with an exhaustive list of $60\,834\,432 \approx 2^{25.8}$ permutations that satisfy (P-1), (P-3), (P-4), and (P-5).

To check (P-6), we build a table of differences with a loop of $(2^4)^2$ steps. Then, to check (P-2), the Walsh transform takes 4×2^4 more steps. Therefore, to obtain our final list of valid permutations, we need another $2^{26} \times 2^8 = 2^{34}$ iterations.

To find edges in G_4, we observe experimentally that a random pair of permutation from $\mathcal{P}$ is an edge of G_4 with probability p_4 as given in Table 5. Therefore, to obtain $\underline{m}$ edges in Σ_4, we must iterate over approximately $\frac{m}{p_4}$ pairs.

Then, from m edges in G_4, constructing G_5 can be done in $O(m^2)$. Experimentally, we observe that two random edges in Σ_4 form an edge of G_5 with probability p_5 as given in Table 5. So, with $\underline{m}$ edges in Σ_4 we obtain a graph Σ_5 of m vertices and approximately $\underline{e = \frac{m^2}{2} p_5}$ edges.

Finally, we observe that we can obtain $ep_s = \frac{m^2}{2} p_5 p_s$ S-boxes from the edges in Σ_5 satisfying the non-linearity criteria (see Table 5). Our algorithm uses $m = 10\,000$ edges in Σ_4. On the one hand, this choice always resulted in at least 8 S-boxes in our experiments. On the other hand, it is justified by our probability analysis that predicts around $ep_s = 11$ S-boxes at the output. An additional $O(8! = 2^{15.3})$ iterations for (S-8) ensures optimal ordering.

This algorithm was implemented in C, compiled with gcc with the optimization option -O3 and executed on a 2,2GHz Intel Core i7 processor running OS X. Generating all S-boxes from 10 000 pairs once (executing Algorithm 1 completely) takes approximately 1 minute 5 seconds with a memory usage of 16.7MB. More precisely, generating the list of permutations $\mathcal{P}$ takes roughly 1 minute, constructing Σ_4 lasts about 5 seconds and the time to construct Σ_5 is negligible. The number of resulting S-boxes varies around 10.

To get more S-boxes, we can generate a pool of permutation pairs multiple times, without repeating the generation of the permutations $\mathcal{P}$. This way we can for example get around 350 S-boxes in 4 minutes with the same memory usage by generating $\mathcal{P}$ once and constructing Σ_4 and Σ_5 25 times.

We also implemented Coppersmith's last criterion (S-8), to obtain for 8 S-boxes the ideal order that they should be used in. Generating 8 S-boxes and printing them in the ideal order takes 1 minute 5 seconds and 16.3MB of memory. This duration includes the making of $\mathcal{P}$. An example of an S-box set generated with our method can be found in Appendix A. Non-linearity measures are shown in Table 7 and Table 8 provides the best differential and linear characteristics, calculated with Matsui's algorithm [10]. Note that these are all smaller than the best characteristics obtained with the standard DES S-boxes as reported in [10] and [9].

Sbox	DP_{max}	LP_{max}	$LP_{max(1)}$
S_1^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_2^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_3^*	$\frac{14}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_4^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_5^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{20}{64})^2$
S_6^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_7^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$
S_8^*	$\frac{16}{64}$	$(\frac{28}{64})^2$	$(\frac{24}{64})^2$

Table 7: Properties of the new S^*-boxes

	DP_{max}^{DES}		LP_{max}^{DES}	
	Standard	New	Standard	New
13 Rounds	$2^{-47.22}$	$2^{-52.98}$	$2^{-34.85}$	$2^{-40.42}$
14 Rounds	$2^{-54.10}$	$2^{-60.49}$	$2^{-39.49}$	$2^{-44.42}$
15 Rounds	$2^{-55.10}$	$2^{-61.81}$	$2^{-41.49}$	$2^{-47.25}$
16 Rounds	$2^{-61.97}$	$2^{-69.32}$	$2^{-44.85}$	$2^{-50.80}$

Table 8: Best characteristics of the standard S-boxes and new S^*-boxes

5 Conclusion

We now have an algorithm that can generate either a large pool of DES-like S-boxes or a group of 8 S-boxes in the order in which they should be used for DES. Thanks to the use of several graphs, the generator is very efficient. Moreover, it generates very quickly *all* 4×4 S-boxes that we want to start from. Therefore, the methodology can serve as a basis for other S-box generators.

The algorithm can be extended to include criteria that protect against other attacks such as Murphy's attack [4]. Finally, those who still use DES, could generate their own set of S-boxes.

References

1. Biham, E., and Shamir, A. Differential cryptanalysis of DES-like cryptosystems. In *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology* (London, UK, UK, 1991), CRYPTO '90, Springer-Verlag, pp. 2–21.

2. Chabaud, F., and Vaudenay, S. Links between differential and linear cryptanalysis. In *Advances in Cryptology EUROCRYPT'94*, A. Santis, Ed., vol. 950 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1995, pp. 356–365.

3. Coppersmith, D. The data encryption standard (DES) and its strength against attacks. *IBM J. Res. Dev. 38*, 3 (May 1994), 243–250.

4. Davies, D., and Murphy, S. Pairs and triplets of DES S-boxes. *Journal of Cryptology 8*, 1 (1995), 1–25.

5. DES. Data encryption standard. In *In FIPS PUB 46, Federal Information Processing Standards Publication* (1977), pp. 46–2.

6. Kim, K. Construction of DES-like S-boxes based on boolean functions satisfying the SAC. In *Advances in Cryptology ASIACRYPT '91*, H. Imai, R. Rivest, and T. Matsumoto, Eds., vol. 739 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1993, pp. 59–72.

7. Kim, K., Park, S., Park, S., and Lee, D. Securing DES S-boxes against three robust cryptanalysis. In *Proceedings of the Workshop on Selected Areas in Cryptography SAC '95* (1995), pp. 145–157.

8. Leander, G., and Poschmann, A. On the classification of 4 bit S-boxes. In *Arithmetic of Finite Fields*, C. Carlet and B. Sunar, Eds., vol. 4547 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 159–176.

9. Matsui, M. Linear cryptanalysis method for DES cipher. In *Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology* (Secaucus, NJ, USA, 1994), EUROCRYPT '93, Springer-Verlag New York, Inc., pp. 386–397.

10. Matsui, M. *Advances in Cryptology — EUROCRYPT'94: Workshop on the Theory and Application of Cryptographic Techniques Perugia, Italy, May 9–12, 1994 Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1995, ch. On correlation between the order of S-boxes and the strength of DES, pp. 366–375.

11. Matsui, M. New block encryption algorithm MISTY. In *Fast Software Encryption*, E. Biham, Ed., vol. 1267 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1997, pp. 54–68.

A Example of 8 DES-like S-boxes

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

Table 9: S_1^*

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

Table 10: S_2^*

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

Table 11: S_3^*

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

Table 12: S_4^*

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

Table 13: S_5^*

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

Table 14: S_6^*

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

Table 15: S_7^*

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

Table 16: S_8^*

B Working with optimal 4×4 S-boxes

As the classification in Table 3 shows, we can find 36 864 *optimal* permutations that satisfy (P-1), (P-3), (P-4), and (P-5), with $DP^P_{max} = \frac{4}{16}$ and $LP^P_{max} = (\frac{12}{16})^2$. Performing Algorithm 1 using this smaller list of permutations leads to a smaller number of S-boxes in the end. This can be resolved by increasing the number of edges in Σ_4. However, since the differential and linear properties of these S-boxes are not different from those obtained with the complete list $\mathcal{P}$ of permutations, we decided to keep criterion (P-6) as described in section 2.2.

As an alternative, we perform the second part of the algorithm deterministically, by generating graphs G'_4 and G_4 completely. We observe that G'_4 and G_4 have respectively 10 321 920 and 1 483 776 edges. Using these totals and with $n = 36\,864$, we derive new probabilities p'_4 and p_4:

$$p'_4 = \Pr[(P,P') \in \mathcal{E}(G'_4)] \approx \frac{10\,321\,920}{(n^2)/2} = 2^{-6.04}$$

$$p_4 = \Pr[(P,P') \in \mathcal{E}(G_4)] \approx \frac{1\,483\,776}{(n^2)/2} = 2^{-8.84}$$

To find all edges in G_5, we would have to iterate over $O(m^2)$ pairs. Instead, we predict the number of edges using the same procedure as before. Experimentally, we observe that

$$p_5 = \Pr[(\{P_1,P_2\},\{P'_1,P'_2\}) \in \mathcal{E}(G_5)] \approx 2^{-8.91}$$

and that an edge from G_5 is a valid S-box with probability $p_s = 2^{-13.76}$.

Interestingly, p_4 and p'_4 are higher than the ones we had with $DP^P_{max} \leq \frac{6}{16}$ but p_s is lower. So, the non-linearity criteria have an important impact on these probabilities.

Table 17 shows the resulting analysis of the number of valid S-boxes.

# Vertices G'_4 = # Vertices G_4	n	36 864	
# Edges G'_4	$\frac{n^2}{2} \cdot p'_4$	$2^{23.30}$	$p'_4 = 2^{-6.04}$
# Edges G_4 = # Vertices G_5	$m = \frac{n^2}{2} \cdot p_4$	$2^{20.50}$	$p_4 = 2^{-8.84}$
# Edges G_5	$e = \frac{m^2}{2} \cdot p_5$	$\approx 2^{31.09}$	$p_5 = 2^{-8.91}$
# DES S-boxes	$\approx e \cdot p_s$	$\approx 2^{17.33}$	$p_s = 2^{-13.76}$

Table 17: Analysis of the amount of valid DES S-boxes when $DP^P_{max} = \frac{4}{16}$.