Dependence of output on input in DES: Small avalanche characteristics

Yvo Desmedt[2], Jean-Jacques Quisquater[1] and Marc Davio[1,3]

[1] Philips Research Laboratory Brussels,
Avenue Van Becelaere, 2; Box 8; B–1170 Brussels, Belgium;

[2] Katholieke Universiteit Leuven, Laboratorium ESAT,
Kardinaal Mercierlaan, 94, B–3030 Heverlee, Belgium;

[3] Université Catholique de Louvain, Batiment Maxwell,
Place du Levant, 3, B–1348 Louvain-la-Neuve, Belgium.

Abstract. New general properties in the S–boxes were found. Techniques and theorems are presented which allow to evaluate the non-substitution effect in f and the key clustering in DES. Examples are given. Its importance related to the security of DES is discussed.

1. Introduction

The Data Encryption Standard, in short the DES, is the NBS cryptographic standard for the protection of commercial computer data (FIPS, 1977). Since 1981, it is also an ANSI standard. In the meantime, it is called DEA by ANSI (ANSI, 1980), and it is yet in use in many industrial applications. Recently it has been proposed to become an ISO (International Standard Organisation) standard under the name of DEA1 (ISO, 1983).

There exist several reasons to explore the internal structure and the functional properties in the DES.

1. It can help to understand the DES. Remark that the design criteria of the DES are still classified (Bernhard, 1982).
2. A better understanding of the DES can have two consequences: on the one hand, the detection of weaknesses can speed up a cryptanalysis attack. The detection of inherent strengths will on the other hand simplify the task of defining new standards when they will be needed.
3. The structure can be used in order to simplify or to speed up hardware and software implementations.

G.R. Blakley and D. Chaum (Eds.): Advances in Cryptology - CRYPTO '84, LNCS 196, pp. 359-376, 1985.

To achieve the proposed goals, we first survey (section 2) the technical description of the DES as it appeared in the NBS publication. The reader, who knows the NBS description of the DES, can skip section 2. As the full description of all functions in the DES is very long, we refer to the literature (FIPS, 1977; Konheim, 1981; Meyer & Matyas, 1982; Morris & al., 1977) for these functions.

In section 3 general properties in the S–boxes and in the key scheduling will be combined.

We analyze several functions in order to combine their properties. As a consequence this can be used to find different cleartexts for which the function f in the DES gives the same output. These results can also be used to analyze the key clustering in the DES. It means to verify if there exists different keys which gave for most cleartext the same ciphertext.

2. NBS description of the DES

The DES algorithm, as described by NBS (FIPS, 1977), consists of three fundamental parts: *enciphering computation*, *calculation of* $f(R, K)$ and *key scheduling calculation*. They are are briefly described below.

First observe that several boxes are used in the DES algorithm. It would be a too long explanation to give the details of all these boxes; it can be found in the NBS description. The kind of boxes (e.g. permutation) will be mentioned. Remark that the input numbering starts from 0 for some boxes and from 1 for the other ones.

In the *enciphering computation*, the input is first permuted by a fixed permutation IP from 64 bits into 64 bits. The result is split up into the 32 left bits and the 32 right bits, respectively L and R. Then a bitwise modulo 2 sum of the left part L and of $f(R, K)$ is carried out. After this transformation, the left and right 32 bit blocks are interchanged. Observe that the encryption operation continues iteratively for 16 steps or rounds. In the last round, no interchange of the last obtained left and right parts is performed; the output is obtained by applying the inverse of the initial permutation IP to the result of the 16^{th} round.

In the *calculation of* $f(R, K)$ the 32 right bits are first expanded to 48 bits in the box E, by taking some input bits twice, others only once. Then a bitwise modulo 2 sum of the expanded right bits and of 48 key bits is performed. These 48 key bits are obtained in the *key scheduling calculation*, which will be explained later on. The results of the modulo 2 sum go to the eight S–boxes; each of these boxes has six inputs and four outputs. The S–boxes are nonlinear functions. The output bits of the S–boxes are permuted in the box P.

Let us finally describe the *key scheduling calculation*. The key consists of 64 bits, of which 56 bits only are used. The other 8 bits are not used in the algorithm. The selection of the 56 bits is performed in box PC_1, together with a permutation. The result is split into two 28 bit words C and D. To obtain the 48 key bits for each iteration, the words C and D are first left shifted once or twice. A selection and a permutation PC_2 are then applied to the result. The output of PC_2 is the 48 bit key word which is used in $f(R, K)$. An additional table tells the user how many shifts must be performed to obtain the next 48 key bits of the key for the following round. The DES can be used in four modes (FIPS, 1980; Konheim, 1981).

3. Propagation characteristics

We first analyze the new properties, which we observed in the expansion phase, the S–boxes and the key scheduling. We combine our results with older ones (Davio, Desmedt & al., 1983) in order to discuss the non–substitution property in f and the key clustering in the DES. Let us first discuss the importance of the fact that f is not a substitution and of the key clustering.

3.1. The importance of the propagation characteristics

If f is not a substitution, for fixed key, the cardinality of the image plays an important role in the evaluation of the security of the DES. Indeed if the image of f contains only one element, the DES is completely linear. More generally, if the cardinality of the image of f is small the DES may be insecure.

If there is a key clustering present in the DES, it may be possible that for many cleartexts the effect of modifying the key in a special way does not affect the ciphertext. If this is true for the DES it simplifies enormously an exhaustive attack.

3.2. The expansion phase

The expansion phase plays a very important role in this section.

3.3. The S–boxes

3.3.1. An introduction

We observed several new properties in the S–boxes. Most of our new properties are valid for all S–boxes and are consequently called "general properties". In the following sections some of these properties are used in order to analyze in which measure f is not a substitution and to analyze the key clustering. We did not apply all general properties in the following sections; perhaps in the future one will be able to explain why the S–boxes have these properties or to use them in some deeper analysis of the DES.

Two kinds of properties are discussed. In the first kind we fix some input bits of the S–boxes (1, 2, ... , or 5 of the 6 possible bits). We are interested in what changes are propagated at the output and how? E.g. for the output one can wonder if the four output bits are always distinct if we change the non–fixed input bits, or if for some inputs the output is not affected. Secondly we discuss how the output changes if we complement some input bits of the S–boxes.

We number the inputs of one S–box by $abcdef$ as Davies did (Davies, 1981). We number the S–boxes from 1 to 8 and denote them as S_i. Remark that representations of the S–boxes, other than in the NBS norm, may be useful (Davio, Desmedt & al., 1983).

3.3.2. Properties of the S–boxes if some input bits are fixed

The inputs a, b, e, f of the S–boxes play a special role in the DES. Indeed one half of the message input bits in each round influences two S–boxes. These bits will go to the mentioned input bits. These bits will play an important role in the analysis of the non–substitution property of the function f in the DES. The next properties draw special

attention to the mentioned input bits. The following properties can however easily be generalized. One can easily verify them using a computer program.

We number the properties by a double numbering technique, such that it is easy to refer to them.

1. The observed properties hold for all S–boxes. We analyze if the output of an S–box can or cannot change if one modifies the inputs of an S–box in the following way:
   (a) fix the inputs e and f,
   (b) one is allowed to change c and d to an arbitrary value c' and d',
   (c) one changes the inputs a and b as described in the properties,

   1.1. $\neg(\forall c, d, c', d', e, f : S_i(0,0,c,d,e,f) \neq S_i(1,0,c',d',e,f))$,

   1.2. $\neg(\forall c, d, c', d', e, f : S_i(0,1,c,d,e,f) \neq S_i(1,1,c',d',e,f))$,

   1.3. $\forall c, d, c', d', e, f : S_i(0,1,c,d,e,f) \neq S_i(1,0,c',d',e,f)$,

   1.4. $\forall c, d, c', d', e, f : S_i(0,0,c,d,e,f) \neq S_i(1,1,c',d',e,f)$.

   Remark: One can wonder why e.g. $S_i(0,0,c,d,e,f)$ was not compared with $S_i(0,1,c',d',e,f)$. This property is already known. Indeed it is known (Konheim, 1981) that each row (see NBS notation) of each S–box is a permutation. In other words $S_i(a,b,c,d,e,f) \neq S_i(a,b',c',d',e',f)$ independent of b,c,d,e,b',c',d',e'. The properties described here are in fact a generalization of it.

2. The observed properties hold for all S–boxes, except property 2.4. We analyze if the output of an S–box can or cannot change if one modifies the inputs of an S–box in the following way:
   (a) fix the inputs a and b,
   (b) one is allowed to change c and d to an arbitrary value c' and d',
   (c) one changes the inputs e and f as described in the properties,

   2.1. $\neg(\forall a, b, c, d, c', d' : S_i(a,b,c,d,0,0) \neq S_i(a,b,c',d',0,1))$,

   2.2. $\neg(\forall a, b, c, d, c', d' : S_i(a,b,c,d,1,0) \neq S_i(a,b,c',d',1,1))$,

   2.3. $\neg(\forall a, b, c, d, c', d' : S_i(a,b,c,d,0,1) \neq S_i(a,b,c',d',1,0))$,

   2.4. If $i \neq 4$ then:
   $\neg(\forall a, b, c, d, c', d' : S_i(a,b,c,d,0,0) \neq S_i(a,b,c',d',1,1))$.
   If $i = 4$ then:
   $\forall a, b, c, d, c', d' : S_i(a,b,c,d,0,0) \neq S_i(a,b,c',d',1,1)$.

Remark: The properties 1.3 and 1.4 change if one also allows that the input e changes to the input e'. Then it will be possible to find identical outputs for special inputs. A similar remark is true for property 2.4 ($i = 4$) if one allows that the input b changes.

3.3.3. Complementation properties of the S–boxes

A well known (Hellman & al., 1976) property for the S–boxes is that if one complements one input of an S–box at least two output bits will change. We analyze the effect of complementing two input bits, while leaving the other ones unchanged. It is evident that one can easily generalize our properties for the case that 3 or more bits are

	ab	ac	ad	ae	af	bf	cf	df	ef
S–box 1	0	6	6	5	0	3	5	2	7
S–box 2	0	4	5	2	0	3	7	1	2
S–box 3	0	3	2	6	5	4	5	3	4
S–box 4	0	8	0	4	0	2	4	2	4
S–box 5	0	1	3	7	0	3	4	6	4
S–box 6	0	3	5	8	1	3	5	5	0
S–box 7	0	7	2	5	2	3	5	3	4
S–box 8	0	5	2	4	0	0	4	2	3

Table 1: shows for how many out of 32 inputs a complementation of two bits of the input of an S–box has no effect.

complemented. The first aim was to observe whether it is possible to maintain a constant output if only two bits are complemented. First observe that in order to maintain a fixed output one has to complement bit a or f, otherwise we conflict with the permutation property of the "rows" in the S–boxes. For special $abcdef$ inputs the output of an S–box remains unchanged if one complements two of the input bits. We give now the results of our research in table 1.

It is remarkable for each S–box that if only ab is complemented, the output changes. This is however very easy to prove starting from our properties 1.3 and 1.4 of the previous section.

3.4. The key scheduling

In our analysis of the key clustering we used in detail the key scheduling in the DES. The ideas of Neutjens about the key scheduling in the DES were very useful in this context (Neutjens, 1983). We now survey them and explain them systematically. We number the 56 key bits from 1 to 64 as in the NBS description (FIPS, 1977).

First of all remark that after PC_1 one can split up the key scheduling in the DES completely in two parts. PC_2 does not affect this decomposition (Davio, Desmedt & al., 1983). As a consequence of this decomposition, one can separate for one round in the DES the selection of the key bits which will influence the first four S–boxes and the last four S–boxes. Let us now construct the equivalent scheme. All used notations, e.g. the registers C and D, originate from the NBS representation of the DES.

We represent the register content of C by $(c_1, c_2, \ldots, c_{28})$ and that of D by $(d_1, d_2, \ldots, d_{28})$. Mostly in the key scheduling the registers C and D are shifted *twice* to obtain the K_i of the i^{th} round, e.g. $(c_1, c_2, c_3, \ldots, c_{28})$ is transformed into $(c_3, c_4, c_5, \ldots, c_2)$. This can now be reformulated for the C register as *one* shift on the following *two* registers $(c_1, c_3, c_5, \ldots, c_{27})$ and $(c_2, c_4, c_6, \ldots, c_{28})$. We call them respectively the odd and the even registers. One can then realize the key scheduling with 4 registers instead of two, which shift only once when in the NBS representation the registers shift twice. This reorganization affects the PC_2.

One has now still to discuss what happens if only one shift is performed on C and D as in the iterations 1, 2, 9 and 16 using our equivalent representation. The first shift in the first iteration can be realized together with PC_1. In the other situations we interchange the content of the odd and the even registers, by performing first a shift on the old content

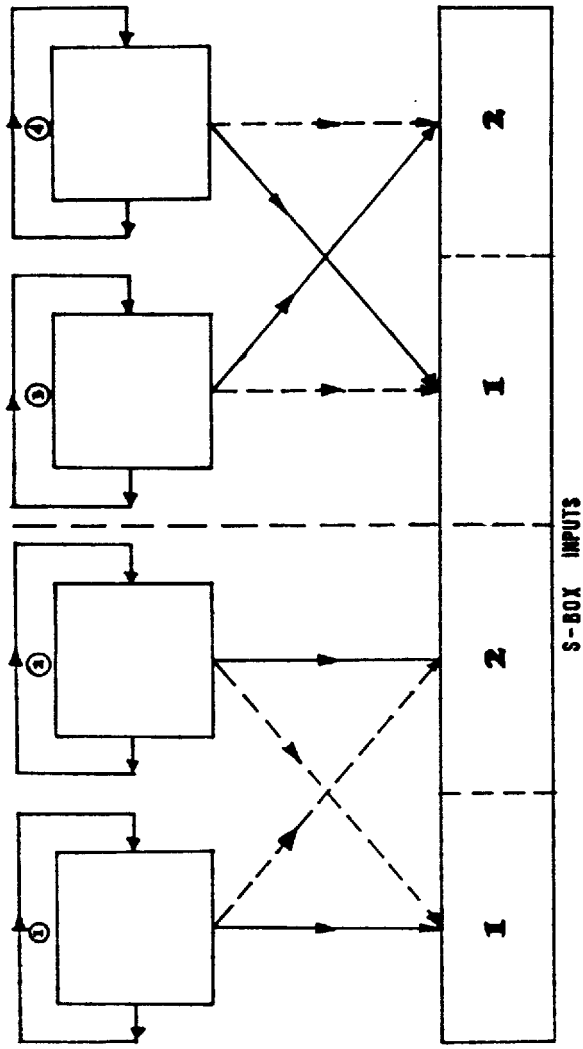

Figure 1: An equivalent key scheduling.

of the odd register and no shift on that of the even register. We then change also the name of each register: odd becomes even, even becomes odd. Indeed $(c_1, c_3, c_5, \ldots, c_{27})$, $(c_2, c_4, c_6, \ldots, c_{28})$ is then changed into $(c_2, c_4, c_6, \ldots, c_{28})$, $(c_3, c_5, c_7, \ldots, c_1)$. One can verify that previous operations are identical to one shift in the NBS notation.

The register D can be treated in a similar way. Remark that it is more difficult to perform one shift in the NBS representation. However we are able to see better which bits of the key affect a particular S–box. We now represent this result in tables 2–5 and fig. 1, where X means that this key bit is not selected by PC_2.

Let us now apply all the described properties.

3.5. The function f is not one–to–one for fixed K

Let us remember here that the function f consists of the expansion box E, of the EXOR–ing with the key bits, of the S–boxes and of the permutation P. It has sometimes been wondered whether the f function is by itself a substitution. The answer to that question is negative (Davio, Desmedt & al., 1983; Konheim, 1981). A more systematic discussion is given in this section.

We will now use the properties described in section 3.3.2. to demonstrate how they can be used in the analysis of the non–substitution of the function f. Evidently we assume that the key K is fixed. We analyze which bits of the message part R (see NBS notation) one must change in order to maintain the same output of the function f. We will progressively increase the number of changed bits. First we only change the inputs (or message part of the input) of one, two and then three S–boxes and generalize afterwards. We will mostly use the new as well as the well known (Hellman & al., 1976; Konheim, 1981) general properties of the S–boxes, together with the structure of E (Davio, Desmedt & al., 1983).

Theorem 1 : If for fixed key, one only changes the input of one S–box the output of the function f will change.

Proof : In order not to affect the inputs of the other S–boxes one can only change the inputs c and d. However if the inputs a and f are not changed an S–box forms a substitution.

■

Theorem 2 : If for fixed key, one changes only the input of two neighbourhood S–boxes the output of the function f will change.

Proof : Let us call the two affected S–boxes, S_i and S_{i+1} and let us define S_9 as being S_1 (this again shows that it can be more interesting to start the numbering from 0, see (Davio, Desmedt & al., 1983)). In order not to affect the input of S_{i-1} the inputs a and b of S_i may not change and similarly for the inputs e and f of S_{i+1} in order not to affect the inputs of S_{i+2}. In order not to conflict with the permutation properties of the "rows" of the S–boxes and using the previous remark, at least the input f in S_i must be complemented in order to maintain a fixed output. A similar remark is true for the input a of S_{i+1}. As consequence of the expansion box E a complementation of the input e (respectively f) of S_i is equal to a complementation of the input of a (respectively b) of S_{i+1}. *So in order to produce a same output we have at least to complement* a and b in S_{i+1}. Remark that the inputs c and d in S_{i+1} do not influence the proof. In other words

	3	23	9	2	14	11	13	X2	21	5	7	6	20	X3
1	34	18	2	51	35	19	3	52	36	49	33	17	1	50
2	26	10	59	43	27	11	60	44	57	41	25	9	58	42
3	10	59	43	27	11	60	44	57	41	25	9	58	42	26
4	59	43	27	11	60	44	57	41	25	9	58	42	26	10
5	43	27	11	60	44	57	41	25	9	58	42	26	10	59
6	27	11	60	44	57	41	25	9	58	42	26	10	59	43
7	11	60	44	57	41	25	9	58	42	26	10	59	43	27
8	60	44	57	41	25	9	58	42	26	10	59	43	27	11
9	52	36	49	33	17	1	50	34	18	2	51	35	19	3
10	36	49	33	17	1	50	34	18	2	51	35	19	3	52
11	49	33	17	1	50	34	18	2	51	35	19	3	52	36
12	33	17	1	50	34	18	2	51	35	19	3	52	36	49
13	17	1	50	34	18	2	51	35	19	3	52	36	49	33
14	1	50	34	18	2	51	35	19	3	52	36	49	33	17
15	50	34	18	2	51	35	19	3	52	36	49	33	17	1
16	42	26	10	59	43	27	11	60	44	57	41	25	9	58

Table 2: The effect of the selection of the key bits (1–64) by PC_1 and PC_2. The first row of the table indicates to which input of the S boxes the key bits go. (Neutjens, 1983)

	4	17	8	24	16	10	18	12	15	1	19	X1	22	X4
1	60	44	57	41	25	9	58	42	26	10	59	43	27	11
2	52	36	49	33	17	1	50	34	18	2	51	35	19	3
3	36	49	33	17	1	50	34	18	2	51	35	19	3	52
4	49	33	17	1	50	34	18	2	51	35	19	3	52	36
5	33	17	1	50	34	18	2	51	35	19	3	52	36	49
6	17	1	50	34	18	2	51	35	19	3	52	36	49	33
7	1	50	34	18	2	51	35	19	3	52	36	49	33	17
8	50	34	18	2	51	35	19	3	52	36	49	33	17	1
9	42	26	10	59	43	27	11	60	44	57	41	25	9	58
10	26	10	59	43	27	11	60	44	57	41	25	9	58	42
11	10	59	43	27	11	60	44	57	41	25	9	58	42	26
12	59	43	27	11	60	44	57	41	25	9	58	42	26	10
13	43	27	11	60	44	57	41	25	9	58	42	26	10	59
14	27	11	60	44	57	41	25	9	58	42	26	10	59	43
15	11	60	44	57	41	25	9	58	42	26	10	59	43	27
16	3	52	36	49	33	17	1	50	34	18	2	51	35	19

Table 3: Similar as table 2. (Neutjens, 1983)

	34	29	38	33	42	30	47	27	35	X5	28	39	25	X8
1	53	37	21	5	20	4	55	39	23	7	54	38	22	6
2	45	29	13	28	12	63	47	31	15	62	46	30	14	61
3	29	13	28	12	63	47	31	15	62	46	30	14	61	45
4	13	28	12	63	47	31	15	62	46	30	14	61	45	29
5	28	12	63	47	31	15	62	46	30	14	61	45	29	13
6	12	63	47	31	15	62	46	30	14	61	45	29	13	28
7	63	47	31	15	62	46	30	14	61	45	29	13	28	12
8	47	31	15	62	46	30	14	61	45	29	13	28	12	63
9	39	23	7	54	38	22	6	53	37	21	5	20	4	55
10	23	7	54	38	22	6	53	37	21	5	20	4	55	39
11	7	54	38	22	6	53	37	21	5	20	4	55	39	23
12	54	38	22	6	53	37	21	5	20	4	55	39	23	7
13	38	22	6	53	37	21	5	20	4	55	39	23	7	54
14	22	6	53	37	21	5	20	4	55	39	23	7	54	38
15	6	53	37	21	5	20	4	55	39	23	7	54	38	22
16	61	45	29	13	28	12	63	47	31	15	62	46	30	14

Table 4: Similar as table 2. (Neutjens, 1983)

	32	44	37	43	36	45	26	X6	40	31	48	41	46	X7
1	30	14	61	45	29	13	28	12	63	47	31	15	62	46
2	22	6	53	37	21	5	20	4	55	39	23	7	54	38
3	6	53	37	21	5	20	4	55	39	23	7	54	38	22
4	53	37	21	5	20	4	55	39	23	7	54	38	22	6
5	37	21	5	20	4	55	39	23	7	54	38	22	6	53
6	21	5	20	4	55	39	23	7	54	38	22	6	53	37
7	5	20	4	55	39	23	7	54	38	22	6	53	37	21
8	20	4	55	39	23	7	54	38	22	6	53	37	21	5
9	12	63	47	31	15	62	46	30	14	61	45	29	13	28
10	63	47	31	15	62	46	30	14	61	45	29	13	28	12
11	47	31	15	62	46	30	14	61	45	29	13	28	12	63
12	31	15	62	46	30	14	61	45	29	13	28	12	63	47
13	15	62	46	30	14	61	45	29	13	28	12	63	47	31
14	62	46	30	14	61	45	29	13	28	12	63	47	31	15
15	46	30	14	61	45	29	13	28	12	63	47	31	15	62
16	38	22	6	53	37	21	5	20	4	55	39	23	7	54

Table 5: Similar as table 2. (Neutjens, 1983)

even if one additionally changes the inputs c and d in S_{i+1} or does not, the output of S_{i+1} will change, by virtue of property 1.3 and 1.4 of the S–boxes.

∎

Theorem 9: Assume that for fixed key one changes only the input of three neighbouring S–boxes, the output of the function f will for some inputs remain identical only if at least all of the following conditions are satisfied together:

1. one complements the inputs a, b and e of the middle of the three S–boxes,
2. one complements the input c or d of the last S–box,
3. one does not complement the input f of the middle of the three S–boxes.

Proof: We call the three S–boxes S_{i-1}, S_i and S_{i+1} where S_0 is equal to S_8 and S_9 equals S_1. The proof is for a large part similar to that of theorem 2. Let us first give the similar part of the proof.

We must fix the inputs a and b of S_{i-1}, and e and f of S_{i+1}. The input f of S_{i-1} must be complemented and similarly for the input a of S_{i+1}. This last condition is equivalent to say that the inputs b and e of S_i must be complemented. Now we apply the consequences of theorem 2 to continue our proof.

If a and b are both complemented in S_{i+1}, the output will change (see proof of theorem 2 or properties 1.3 and 1.4 of the S–boxes). Using previous observations the input b in S_{i+1} may not be complemented, or equivalently the input f in S_i. At this moment we already know that for S_i the inputs b and e must be complemented and f may not. Because each row in the S–boxes is a permutation and because the input f may not be complemented in S_i, the input a must be complemented in S_i. Remark that in fact one must still complement input c or d in S_{i+1}. Indeed if only one input bit in an S–box is complemented, the output changes.

∎

We have now proven the theorem. It is now very easy to generate in a systematic way several examples for which the function f remains constant even if some bits are complemented.

3.6. The key clustering

We analyze the clustering from the point of view that the DES contains j rounds, where j is between 1 and 16. The input for these j rounds is fixed, while we complement or change some bits of the key. So if we speak now about an input of an S–box, this input is related to a modification of the key.

We first prove some general theorems for the key clustering, and afterwards we give some examples.

3.6.1. A general approach

First of all for a fixed input the permutation IP has no influence on the key clustering. We can start the analysis from L_s and R_s. This means that if we are interested in a complete DES analysis $s = 0$ and $j = 16$. Let us now apply the DES with the key K and

K' and call the subkeys K_1 till K_{16} and K'_1 till K'_{16}. The key K will produce some L and R register content, while K' produces L' and R'. The effect of the first of the j rounds is that in the case we use the key K we have $L_{s+1} = R_s$ and $R_{s+1} = L_s \oplus f(R_s, K_{s+1})$. Applying the key K' we obtain $L'_{s+1} = R_s$ and $R'_{s+1} = L_s \oplus f(R_s, K'_{s+1})$. After t rounds we obtain using key K the register content $L_{s+t} = R_{s+t-1}$ and $R_{s+t} = L_{s+t-1} \oplus f(R_{s+t-1}, K_{s+t})$. Using the key K' we have: $L'_{s+t} = R'_{s+t-1}$ and $R'_{s+t} = L'_{s+t-1} \oplus f(R'_{s+t-1}, K'_{s+t})$. Remark that in general by changing the key the content of the registers L and R change too. Let us now call $H_{s+t} = f(R_{s+t-1}, K_{s+t}) \oplus f(R'_{s+t-1}, K'_{s+t})$. It is now easy to see using (Davio, Desmedt & al., 1983) that the global effect of a change in the key has no final effect on the ciphertext if the two following conditions are satisfied together.

1. $H_{s+1} \oplus H_{s+3} \oplus H_{s+5} \oplus \cdots \oplus H_t = 0$, where $t = s + j$ if j is odd, else $t = s + j - 1$.
2. $H_{s+2} \oplus H_{s+4} \oplus H_{s+6} \oplus \cdots \oplus H_u = 0$, where $u = s + j$ if j is even, else $u = s + j - 1$.

Using previous conditions it is now easy to analyze the conditions necessary for key clustering if one analyzes only 1, 2, 3 or 4 rounds. The analyze of more rounds seems to be more difficult.

3.6.2. An analysis of the key clustering in a DES with 1, 2, 3 or 4 rounds

In the case one round is considered we must have $H_{s+1} = 0$. This means $f(R_s, K_{s+1}) = f(R_s, K'_{s+1})$. Using previous knowledge on the S-boxes this means that the input of an S-box is not changed or that at least two bits change. It is very easy to generate several examples for this case. Using the fact that E is an expansion of 32 bits to 48 bits and its structure (Davio, Desmedt & al., 1983) and because PC_2 selects only 48 bits out of the 56 bits of the key we have the following result. *For each (cleartext, ciphertext) pair in a one round DES there exist exactly* 2^{24} *keys which generate the same (cleartext, ciphertext) pair starting from a fixed cleartext.* If a similar remark remains true for the complete DES algorithm (16 rounds), the DES is very easy to break using a simplified exhaustive attack. Let us therefore start to analyze more rounds.

In the case two rounds are considered we must have $H_{s+1} = 0$ and $H_{s+2} = 0$. This means $f(R_s, K_{s+1}) = f(R_s, K'_{s+1})$, as in previous case, and additionally $f(R_{s+1}, K_{s+2}) = f(R_{s+1}, K'_{s+2})$, because from the first equality we have $R'_{s+1} = R_{s+1}$. Remark that the S-boxes must satisfy similar conditions as in the case only one round was considered. However to satisfy it for the two rounds together we must take the key scheduling in the DES into consideration. This is now easy to do if one uses the tables explained earlier. We now give a simple example of it.

Example 1. If one complements the bits 3 and 44 (in the NBS notation) of any 64 bit key, then there exists $6 \cdot 2^{59}$ pairs of (cleartext, ciphertext) which remain identical during round 1 and 2 in the DES. In other words, about 1/5 of all pairs (cleartext, ciphertext) are not affected by the complementation of 2 bits of the key, during round 1 and 2.

Let us now explain using fig. 2 what happens and how one can calculate the (cleartext, ciphertext) pairs. The bits 3 and 44 go both after the key scheduling in the first round to S_3 and become there the inputs a and e. Using table 1 we know that for 6 out of 32 (or 12 out of 64) possible inputs a complementation of a and e in S_3 does not change the output. This means that the possible inputs for which the above property is true are restricted from 2^{64} to $6 \cdot 2^{59}$. The *cardinality of the set of cleartexts* for which the *explained clustering is satisfied is independent of the used key.* However the *set of cleartexts* for which the above clustering is satisfied, *changes if other keys* are considered. This is a consequence of the

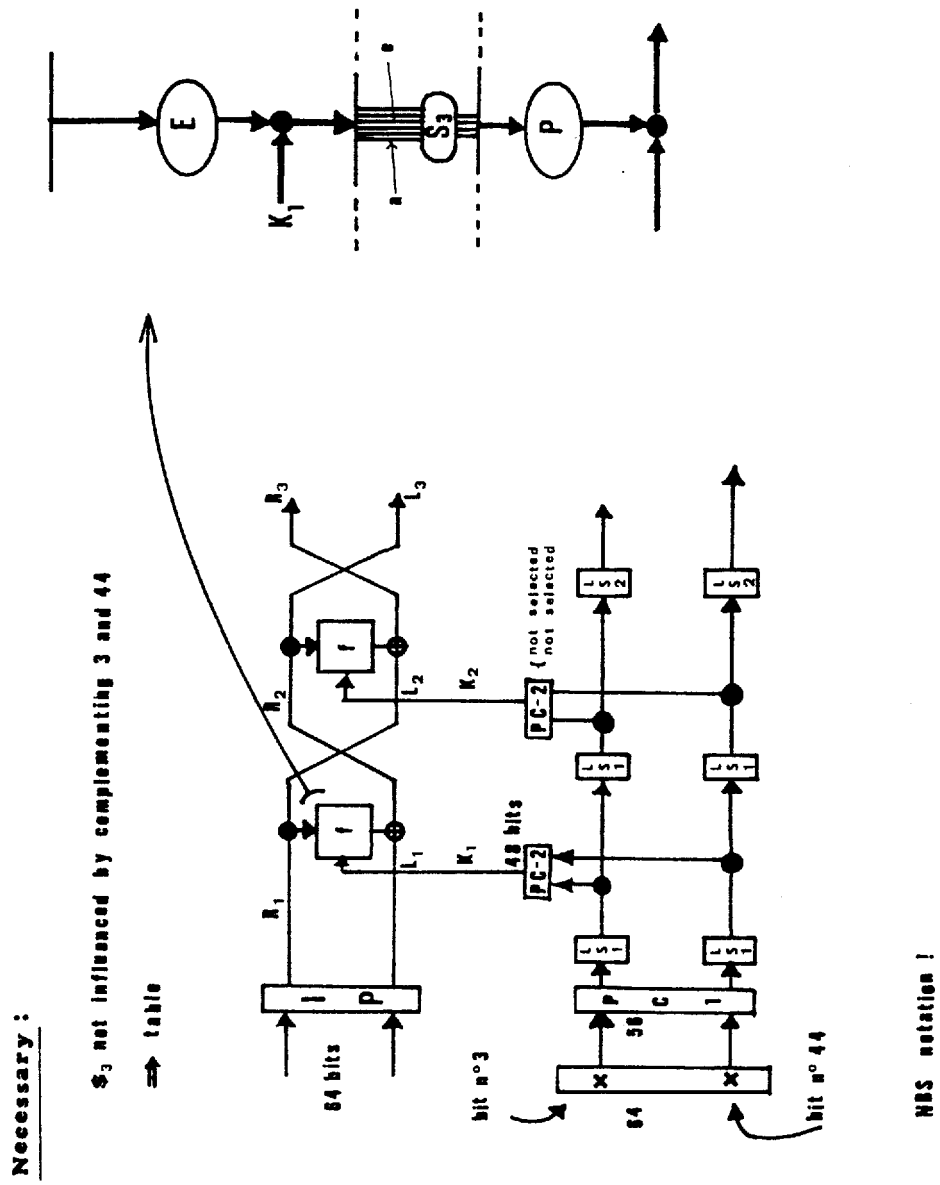

Figure 2: Example 1 on the key clustering in a two round DES.

input S_3 $abcdef$	output S_3
000100	1001
100110	idem
000101	0000
100111	idem
010000	0001
110010	idem
010100	1100
110110	idem
010111	1110
110101	idem
011011	1011
111001	idem

Table 6: Inputs (in binary form) for S_3 which generate the same output if the bits a and e are complemented.

exor of the subkey with the expanded R register in the function f. Let us now analyze which *input for the S-boxes we must force* in order to satisfy the key clustering. The input for S_3, in the first round, must be one of those collected in table 6, in order to satisfy the key clustering. Now we must still analyze which restrictions the second round imposes on the possible cleartext. The analysis in this example is straightforward because the key bits 3 and 44 are not selected in the second round, so no extra condition is necessary.

One may observe that we were lucky in the construction of the previous example. First the non–selection of the key bits in the second iteration seems to be lucky. Secondly example 1 is only valid for rounds 1 and 2 in the DES. In the following example the reader can observe that similar examples can be given for all rounds and that it is not necessary that some key bits are not selected in the second or first round.

Example 2. This example is true for most consecutive rounds. As a consequence of the ideas of Neutjens on the key scheduling (see section 3.4), two consecutive rounds can mostly be analyzed systematically (Neutjens, 1983). This is true if one uses two shifts in the key scheduling, as represented by the NBS, to move to the next round. This means the rounds 2–3, 3–4, 4–5, 5–6, 6–7, 7–8, 9–10, 10–11, 11–12, 12–13, 13–14 and 14–15. In order not to affect the generality we will use a more general descriptions of the property. If one complements the two bits of the key which will "arrive" in S–box 4 at locations a and e during the first of the two above rounds, then for every key there exists $24 \cdot 2^{54}$ (or about $1/43$ of all possible) pairs (cleartext, ciphertext) which remain identical during two consecutive rounds mentioned earlier. This can be easily analyzed (similar as in example 1) using tables 2–5, and using our properties of the S–boxes (table 1).

Let us now consider three consecutive rounds. First more restrictions on the cleartext are then imposed in order not to affect the ciphertext if one modifies the key. This is a consequence of the key scheduling. However the output of the function f in the first and last (of the three) rounds must no longer be constant (see section 3.6.1). This relaxes the imposed restrictions. Let us give a short example to illustrate it.

Example 3. The three consecutive rounds may be 2–3–4, 3–4–5, 4–5–6, 5–6–7, 6–7–8,

9–10–11, 10–11–12, 11–12–13, 12–13–14 and 13–14–15. Hereto one complements (e.g.) three bits of the key (fig. 3). In our example the three key bits must "arrive" at location a and d in S–box 8 in the first round (of the three consecutive) and at location d in S–box 4 in the second round (of the three consecutive). We call these three key bits respectively k_1, k_2 and k_3. By analyzing the box P (see (Davies, 1981)) and using section 3.6.1 two cases can be distinguished.

1. The third output bit of S_8 is complemented in the first and third iteration (of the three consecutive) as a consequence of the previous modification of the key. In other words bit 15 of the output of f (after the box P) must be complemented in the first and last round. The modification of the previous bit will have no influence at all in the second round of the three. Indeed after the expansion phase it is exored with key bit k_3 which we complemented too. Remark first that *the set of cleartexts for which the above clustering is satisfied changes if other keys are considered.* This is a consequence of the exor of the subkey with the expand R register in the function f. Let us now analyze which input we must force at *the input of the S–boxes*, in the three rounds, in order to satisfy the above conditions. Remember from Fig. 2 that the input of the S–boxes is equal to the subkey exor the expanded R register. In the first round key bits k_1 and k_2 influence respectively the input a and d in S_8, as a consequence of our choice. k_3 is not selected. The input of S_8 must be chosen from table 7. In the second round (of the three consecutive) we yet discussed the influence of key bit k_3. Using table 2–5 we find that k_1 and k_2 become now the input a and e respectively in S_7. The input of S_7 must be chosen from table 8. In the third round k_1 and k_2 influence respectively the inputs b and f from S_8. The input of S_8 must be chosen from table 9.

2. The second and third output bits of S_8 are complemented in the first and third round as a consequence of the previous modification of the key. We must then choose the inputs of S_8 in the first round out of table 10, the inputs of S_7 in the second round out of table 11 and the input of S_8 in the third round out of table 12. This can be analyzed in a similar way as for the first case.

We can then analyze that for 50% of the keys: For 21 on 16384 (about 1/780) cleartexts, the ciphertext is not modified. For the other 50% of the keys this happens for 1 on 2048 cleartexts. This analysis is involved. The reader can check it using tables 7–12. He must then take into consideration that the tables impose conditions on the cleartext input of the three rounds. Using fig. 3 he can then easily prove that the first round imposes some conditions on the right input of the cleartext. Similarly the second round imposes some conditions on the cleartext at the left side input of the three rounds. To analyze the restrictions on the input as a consequence of the third round the reader must use the property that each round is a substitution from 2^{64} to 2^{64} elements (Davio, Desmedt & al., 1983) for fixed key. Care should be taken in performing this last step. It is possible that previously imposed conditions influence the new one. Indeed by imposing special conditions on the cleartext, some restrictions can exist on the output of f in previous rounds.

Other examples can easily be generated. It would be interesting to generalize the previous examples to the complete DES with 16 rounds.

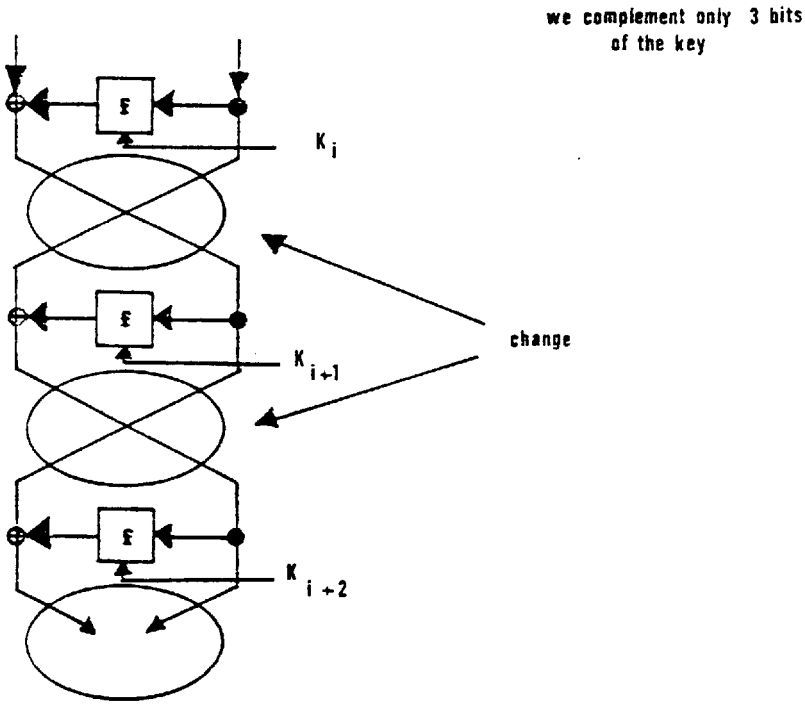

Figure 3: The key clustering in a three round DES.

input S_8 $abcdef$	output S_8
001001	1010
101101	1000
001100	1011
101000	1001

Table 7: Inputs (in binary form) for S_8 which generate outputs in which the third output bit is complemented if the bits a and d of the input are complemented.

input S_7 $abcdef$	output S_7
000000	0100
100010	idem
001001	0100
101011	idem
001111	1010
101101	idem
011000	0101
111010	idem
011001	0010
111011	idem

Table 8: Inputs (in binary form) for S_7 which generate the same output if the bits a and e are complemented.

input S_8 $abcdef$	output S_8
001100	1011
011101	1001
110000	0000
100001	0010

Table 9: Inputs (in binary form) for S_8 which generate outputs in which the third output bit is complemented if the bits b and f of the input are complemented.

input S_8 $abcdef$	output S_8
011001	0000
111101	0110

Table 10: Inputs (in binary form) for S_8 which generate outputs in which the second and third output bit is complemented if the bits a and d of the input are complemented.

input S_7 $abcdef$	output S_7
000010	1011
100100	idem
000101	1011
100011	idem
010101	0101
110011	idem

Table 11: Inputs (in binary form) for S_7 which generate the same output if the bits a, d and e are complemented.

input S_8 $abcdef$	output S_8
001000	0110
011001	0000
000011	1111
010010	1001
000111	1000
010110	1110

Table 12: Inputs (in binary form) for S_8 which generate outputs in which the second and third output bits are complemented if the bits b and f of the input are complemented.

4. Conclusions and perspectives

A cryptographic system can only be considered secure if a small modification in the cleartext and/or in the key strongly affect on a non–linear way the ciphertext. We described techniques for analyzing this constraint for the DES. We found that if the DES had only a few rounds it would be a weak system. Our analysis demonstrated at the same time that the known probabilistic test done on the DES are insufficient to conclude that the scheme is secure. Were it possible to work out on a 16–round DES the techniques presented here one could possibly prove the so often alleged existence of a key clustering in the DES.

References

ANSI X3.92-1981, "Data Encryption Algorithm,", American National Standards Institute, New York (December 31, 1980).

Ayoub, F., "On the design of SP-networks," *presented at Eurocrypt '83, 21-23 March 83, Udine, Italy.*

Bernhard, R., "Breaching system security," *Spectrum*, vol. **19**, pp. 24-31 (1982).

Davies, D. W., "Some regular properties of the Data Encryption Standard algorithm," NPL note, *presented at Crypto '81* (1981).

Davio, M., Desmedt, Y., Fosséprez, M., Govaerts, R., Hulsbosch, J., Neutjens, P., Piret, P., Quisquater, J. J., Vandewalle, J. & Wouters, P., "Analytical characteristics of the DES," pp. 171-202, in *Advances in cryptology: Proc. of CRYPTO '83, Santa Barbara*, ed. D. Chaum, Plenum Publishing Corp., New York (1984).

Denning, D. E., *Cryptography and data security*, Addison Wesley, Reading (Mass.) (1982).

Diffie, W. & Hellman, M. E., "Exhaustive cryptanalysis of the NBS Data Encryption Standard," *Computer*, vol. **10**, n° 6, pp. 74-84 (1977).

FIPS publication 46, "Data Encryption Standard," Federal Information Processing Standard, National Bureau of Standards, U.S. Department of Commerce, Washington, D.C. (January 1977).

Fosséprez, M. & Wouters, P., "Cryptanalyse et matérialisation des réseaux de chiffrement," Final work, Université Catholique de Louvain, Belgium (1983).

Hellman, M. E., Merkle, R., Schroeppel, R., Washington, L., Diffie, W., Pohlig, S. & Schweitzer, P., "Results of an initial attempt to cryptanalyze the NBS data encryption standard," SEL 76-042, Stanford University (1976).

Hulsbosch, J., "Analyse van de zwakheden van het DES-algoritme door middel van formele codering," Final work, Katholieke Universiteit Leuven, Belgium (1982).

ISO/DP 8227 (Draft proposal), "Data encipherment, specification of algorithm DEA1," (1983).

Konheim, A. G., *Cryptography: A primer*, J. Wiley, New York (1981).

Morris, R., Sloane, N. J. A. & Wyner, A. D, "Assessment of the NBS proposed Data Encryption Standard," *Cryptologia*, vol. 1, pp. 301-306 (1977).

Neutjens, P., "Diepere inzichten en eenvoudige hardware voor DES cryptografisch algoritme aan de hand van equivalente strukturen," Final work, Katholieke Universiteit Leuven, Belgium (1983).