SEL 76-042

RESULTS OF AN INITIAL ATTEMPT TO CRYPTANALYZE THE NBS DATA ENCRYPTION STANDARD

M. Hellman, R. Merkle, R. Schroeppel,
L. Washington, W. Diffie, S. Pohlig,
P. Schweitzer

September 9, 1976

Revised: November 10, 1976

CONTENTS

Page

I. INTRODUCTION 1

II. SUMMARY OF POTENTIAL WEAKNESSES 3

III. SYMMETRY UNDER COMPLEMENTATION 5

IV. DANGEROUS S-BOX STRUCTURE 12

V. DES's STRUCTURE 15

A. Observed DES Structure 15

B. Lack of Structure 25

VI. STATISTICAL TESTS 32

VII. TWO ROUND CRYPTANALYSIS 40

VIII. POSSIBLE WAYS TO STRENGTHEN DES 41

IX. FUTURE TASKS 43

X. CONCLUSION 45

Appendix. BINARY VERSIONS OF S-BOXES 46

REFERENCES 50

TABLES

Number Page

1. The effect of the * operation 8

2. Number of 0000 entries in all S-boxes for each value of $\underline{c}$ = c1c2c3c4c5c6 21

3. V_{ij}/σ for DES truncated to first two rounds (N = 8000, σ = 0.0112) 35

4. Values of α_i and β_i for two-round version of DES 36

I. INTRODUCTION

The National Bureau of Standards has proposed a data encryption standard (DES) for nonmilitary governmental and commercial use. Such a cryptosystem can be used to provide both privacy and authentication to messages [1,2]. In many applications, authentication (a form of digital signature) is of primary interest.

For reasons of national security, NBS, IBM (which designed DES), and the National Security Agency (which participated in its evaluation), will not divulge the results of their tests concerning the strength of the system. Because they will also not explain the rationale behind certain critical elements or adequately quantify the level of security, we undertook an evaluation of DES in an attempt to determine its strengths and weaknesses. This evaluation lasted one month and involved approximately ten person weeks of effort. Our results are therefore only preliminary, and it must be assumed that an adversary with greater resources would make greater progress.

It is thus somewhat disturbing that we achieved a 50 percent savings in computation over an exhaustive search. We also found suspicious structure in the critical elements of the system (the S-boxes). This structure could be the result of an accidental weakness, a deliberately set trap door (i.e., hidden structure which allows those who know of it to break the system), or no weakness at all. An explanation and further study are needed before trust can be placed in DES. This need is enhanced because NSA does not want a genuinely strong system to frustrate its cryptanalytic intelligence operations. As a result, DES is mildly suspect *a priori*.

The types of attack to which a cryptosystem can be subjected are as follows.

- *Ciphertext Only Attack*: The cryptanalyst has a substantial quantity of ciphertext (enciphered data) but only knows the statistical structure of the plaintext (such as, E occurs 13 percent of the time, Z occurs 0.1 percent of the time). This attack can always be mounted, and any system that succumbs to it is totally useless.

- Known Plaintext Attack: The cryptanalyst has a quantity of corresponding plaintext and ciphertext. This attack occurs if enciphered messages are later declassified (press releases, product announcements) or if some of the enciphered data are known to the cryptanalyst (his portion of a set of financial transactions). Consequently, any system which succumbs to this attack is not secure.

- Chosen Plaintext Attack: The cryptanalyst has access to corresponding plaintext and ciphertext and can select the plaintext to be enciphered. This type of attack is more difficult to mount but not impossible. By proposing a sensitive agreement to a competitor, we might intercept an enciphered version sent to another location for evaluation, and it is easier for an employee to send a simple message (such as 0000...0) than it is for him to learn the key in use.

In all three attacks, it is assumed the cryptanalyst can obtain a cryptosystem (either honestly or in other ways) and hence knows how it operates. All security resides in the secrecy of the key. No faith is placed in secret design principles because they are too easily compromised. The cryptanalyst tries to determine the key so that he can decipher cryptograms to which he does not know the plaintext (to violate the privacy of messages) and encipher messages to which he does not know the ciphertext (to forge authentic looking messages).

The rule followed by NBS in selecting DES was that it must resist a known plaintext attack [3]. A chosen plaintext attack is often used in this report because it can occur in practice and certification of a cryptosystem should utilize conservative estimates of strength.

II. SUMMARY OF POTENTIAL WEAKNESSES

1. DES is invariant under complementation of P, K, and C (plaintext, key, and ciphertext). Section III describes how this symmetry can be used to reduce search effort by 50 percent under a partially chosen plaintext attack. With carefully chosen S-boxes, it is possible to save another factor of two, for a total savings of 75 percent over exhaustive search. Although this exact structure is not present in DES's S-boxes, portions of it appear and require explanation. Section III also elaborates on this approach.

2. If the S-boxes are linear, the system would yield to a known plaintext attack with less than $1.00 worth of computation time on a minicomputer. Although the S-boxes are not linear, as discussed in Sections IV and V, they are much closer to linear than one would expect. This structure is surprisingly similar to a type that can be used to build a trap door into the system.

   Other structure was also found in the S-boxes. For example, 75 percent of S4 is redundant; that is, three of its rows can be derived from the fourth.

3. Diffie and Hellman [4] dispute NBS' claim that "trying all possible keys is not economically feasible" [3]. They estimate that a DES key can be recovered by exhaustive search for approximately $5000 worth of computation time on a special purpose machine. Fortunately, the current price of this machine is in the $20 million range and is out of reach of all groups with the possible exception of governmental security agencies. Cost trends, however, predict that, in 10 years, this machine will cost approximately $200,000 and recovery of one key will be $50.

   Since the arguments concerning the above mentioned costs are detailed in [4], we do not elaborate on them here. These estimates indicate, however, that even a small weakness will shatter the security of DES. By contrast, doubling the key size to 112 bits would increase the cost of exhaustive search to $\$4 \times 10^{20}$ and the cost of the special purpose machine to $\$10^{24}$. This would ensure a much more comfortable margin against unforeseen weaknesses.

In the sections that follow, we use NBS' terminology for DES [6].

III. SYMMETRY UNDER COMPLEMENTATION

Claim. If P,K yields C, then $\bar{P},\bar{K}$ yields $\bar{C}$, where overbar denotes bit by bit complementation.

Proof.

Note that

$$f(R_i,K_{i+1}) = f'(ER_i + K_{i+1}) \ , \tag{1}$$

where E represents the expansion operation and $+$ represents XOR or mod-2 addition. Complementing both R_i and K_{i+1} therefore does not change the value of f

$$f(R_i,K_{i+1}) = f(\bar{R}_i,\bar{K}_{i+1}) \ . \tag{2}$$

Because

$$L_1 = R_0 \tag{3}$$

$$R_1 = L_0 + f(R_0,K_1) \ ,$$

complementing P (equivalently L_0,R_0) and K (equivalently K_1 as well as $K_2,\ldots,K_{16}$) complements L_1 and R_1 and, by induction, L_2, $R_2,\ldots,L_{16},R_{16}$, and hence C.

This property can reduce the search effort by half if two P-C pairs, P_1-C_1 and P_2-C_2, are available with $P_1 = \bar{P}_2$. The search enciphers P_1 with all keys K that start with a 0. The resultant ciphertext C is compared with C_1 and $\bar{C}_2$. If $C \neq C_1$, the key in use was not K; and, if $\bar{C} \neq C_2$, the key in use was not $\bar{K}$ (which starts with a 1) because

$$\bar{C} = \overline{S_K(P_1)} = S_{\bar{K}}(\bar{P}_1) \tag{5}$$

$$= S_{\bar{K}}(P_2) \ ,$$

where $S_K(P)$ denotes the cryptogram resulting when P is enciphered under K.

In a chosen plaintext attack, two plaintext-ciphertext pairs can be obtained with $P_1 = \bar{P}_2$. This may be possible even in a known plaintext attack. If $P = 0101\ldots$ is sent in idle periods to maintain synchronization, the two phases $0101\ldots$ and $1010\ldots$ would suffice. Even if the P_i are chosen uniformly and at random, approximately $2^{32} = 4 \times 10^9$ blocks (not 2^{64}) are needed before finding a pair of complementary plaintexts.

A second symmetry almost exists which would allow an additional factor of two reduction in the search effort, for a total savings of 75 percent over exhaustive search. Although the exact structure necessary for this symmetry is not present in DES, it is worthwhile examining for several reasons. Before describing this symmetry, an operation (denoted *) that complements half of P, K, and C must be defined.

P^* = plaintext P with the 32 bits that comprise the first 16 bits of L_0 and R_0 complemented (complement the first and third fourths of P after applying the initial permutation)

C^* = ciphertext C with the 32 bits that comprise the first 16 bits of L_{16} and R_{16} complemented (complement the first and third fourths of C before applying IP^{-1})

K^* = key K with the 28 bits that comprise the contents of the C_0 register complemented (complement the first half of K after applying PC-1). Note that there is a notational conflict between C_i as the ith ciphertext block and C_i as the contents of the 28-bit C register after the ith round. Where context does not suffice, this will be clarified "the ciphertext C_i" or "the C register contents C_i"

With these definitions, the symmetry of interest can now be stated. If

$$C = S_K(P) \tag{6}$$

then

$$C^* = S_{K^*}(P^*) \ . \tag{7}$$

The reason for this symmetry almost existing is that PC-2 selects the first 24 bits of K_{i+1} (which are XOR'd with the first half of R_i, the third fourth of L_iR_i) solely from the C register, and PC-2 selects the second 24 bits of K_{i+1} (which are XOR'd with the second half of R_i) solely from the P register. With reference to NBS' PC-2 table, this property manifests itself in that the first 24 entries are all ≤ 28 and the last 24 entries are all >28.

Without the expansion operation E,

$$f(R_i,K_{i+1}) = f(R_i^*,K_{i+1}^*) \tag{8}$$

and the complementation in the first and third fourths of L_0R_0 would carry over to L_1R_1 and, by induction, to $L_{16}R_{16}$; however, E produces a slight mixing between the first and second halves of R. The first half of ER includes bits 32 and 17 which are in the second half of R. Similarly, bits 1 and 16 from the first half of R move into the second half of ER.

Even with the expansion operation included, symmetry (7) can exist if the S-boxes are properly, or rather improperly, chosen. It can be seen from Table 1 that use of P^* and K^*, instead of P and K, causes ER + K to be complemented (denoted by c) in only four positions--the first input to S1, the sixth input to S4, the first input to S5, and the sixth input to S8. If the output of S1 did not depend on its first input bit, etc. then (8) would hold and (7) would follow. Such a trap door, however, would be extremely obvious. The first and third rows of S1 would match, as would its second and fourth rows; this same structure would appear in S5. And S4, as well as S8, would have their first two rows match and their last two rows match. Below is an example of S8 used in DES modified to be invariant to its sixth input. The structure is glaring, especially when S1, S4, and S5 exhibit similar structure.

S8

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

Table 1

THE EFFECT OF THE * OPERATION

Inputs to	ER_i^*	$PC2(K_{i+1}^*)$	$ER_i^* + PC2(K_{i+1}^*)$ Complemented?	Inputs to	ER_i^*	$PC2(K_{i+1}^*)$	$ER_i^* + PC2(K_{i+1}^*)$ Complemented?
S1	r32	k01c	yes	S5	r16c	k25	yes
	r01c	k02c			r17	k26	
	r02c	k03c			r18	k27	
	r03c	k04c			r19	k28	
	r04c	k05c			r20	k29	
	r05c	k06c			r21	k30	
S2	r04c	k07c		S6	r20	k31	
	r05c	k08c			r21	k32	
	r06c	k09c			r22	k33	
	r07c	k10c			r23	k34	
	r08c	k11c			r24	k35	
	r09c	k12c			r25	k36	
S3	r08c	k13c		S7	r24	k37	
	r09c	k14c			r25	k38	
	r10c	k15c			r26	k39	
	r11c	k16c			r27	k40	
	r12c	k17c			r28	k41	
	r13c	k18c			r29	k42	
S4	r12c	k19c		S8	r28	k43	
	r13c	k20c			r29	k44	
	r14c	k21c			r30	k45	
	r15c	k22c			r31	k46	
	r16c	k23c			r32	k47	
	r17	k24c	yes		r01c	k48	yes

The following versions of S5 and S8 possess a much less glaring trap door which allows the other six S-boxes to remain unchanged, and yet allows an extra degree of symmetry similar to (7).

S5

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

S8

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

These S-boxes were chosen so that, for all 64 inputs $\underline{x}$,

$$S5(\underline{x}) = S5(\underline{x} + 110100) \tag{9}$$

and

$$S8(\underline{x}) = S8(\underline{x} + 001011) . \tag{10}$$

The symmetry present within the new S5 and S8 is

$$C^{+} = S_{K*}(P^{+}) \tag{11}$$

where the complementation pattern of C^{+} and P^{+} is the same as C^{*} and R^{*} except bits $\ell 17$, r17, $\ell 19$, r19, $\ell 30$, r30, $\ell 32$, and r32 are also complemented. Referring to Table 1, complementing r32 resets the input to S1 so that its output remains unchanged under the K^{*}, P^{+} operations, and complementing r17 resets the input to S4. Now, not only is S5's first input complemented (because r16c spills over), but its second and fourth inputs are complemented as well (because r17 and r19 are now complemented); however, because S5 was chosen to satisfy (9), its output is unchanged! Similarly, with r30 and r32 now complemented, the third and fifth as well as the sixth inputs of S8 are complemented. But S8 was chosen so its output is unaffected by these changes. For these new S5 and S8, therefore,

$$f(R_i, K_{i+1}) = f(R_i^+, K_{i+1}^*) \tag{12}$$

and L_1R_1 will be changed to $L_1^+R_1^+$ by changing P to P^+ and K to K^*. By induction, $L_{16}R_{16}$ will be changed to $L_{16}^+R_{16}^+$ and C to C^+, completing the proof of our claim (11).

Any patterns of the form 11XX00 and 00XX11 would work in (9) and (10), respectively, by redefining the P^+ and C^+ operations. The X's indicate bits which can be set at either 0 or 1. The two 1's at the beginning of 11XX00 for S_5 allow r17 to be complemented, thereby fixing S4. The two 0's at the end ensure that r20 and r21 are not complemented because these spill over to S6; they could be complemented, but then it would be necessary to make S6 invariant to the changes. We could also have modified S1 instead of S8, or S4 instead of S5.

In light of the above remarks, one should test to see if, for any $1 \leq i \leq 8$ and any binary 6- vector $\underline{c}$,

$$Si(\underline{x}) = Si(\underline{x} + \underline{c}) \tag{13}$$

for all inputs $\underline{x}$. Although no such i and $\underline{c}$ were found, complex structure of the form (13) was observed, and is described in Section V. The simplest forms found are as follows.

- The mod-2 sum of the four output bits of S4 does not depend on its sixth input bit. Letting $\underline{1}$ denote the binary 4-vector of all 1's,

  $$\underline{1} \cdot S4(\underline{x}) = \underline{1} \cdot S4(\underline{x} + 000001) \ , \tag{14}$$

  which is very similar to (13). This structure is extremely suspicious, as described in Sections IV and V.

- As can be seen on page 9, the permutation going from row 1 to row 2 of the trap door S5 is the same as from row 3 to row 4 (2→14, 12→11, etc.). In the trap door S8, the row 1→3 permutation matches the row 2→4 permutation. This is a result of the structure in (9) and (10). The S4 used in DES exhibits this property going from row 1→2 and row 3→4 (7→13, 13→8, etc.).

Although the above structure does not yield a symmetry of the form (11), more general symmetries may hold.

These symmetries illustrate that the S-box structure is crucial to DES's security, and how a careful choice of S-boxes can lead to hidden structures that facilitate cryptanalysis for those who know its presence and form.

Most symmetries related to (11) would be destroyed if PC-2 were to mix the C- and D-register contents. We therefore suggest modifying PC-2 to strengthen the algorithm.

IV. DANGEROUS S-BOX STRUCTURE

The last section showed that carefully chosen S-boxes can introduce additional degrees of symmetry and shorten the key search. This section describes even more dangerous structures, and the next section says to what extent such structures were found in DES.

Essentially, all of DES's strength lies in the S-boxes because all other operations (XOR, expansion and permutation) are linear in binary arithmetic. If the S-boxes were also linear, the overall algorithm would be linear, and the net effect of enciphering and deciphering would be the

$$\underline{C} = A\underline{P} + B\underline{K} \tag{15}$$

$$\underline{P} = A^{-1}(\underline{C} - B\underline{K}) \ . \tag{16}$$

The invertible 64×64 binary matrix A and the 64×56 binary matrix B depend only on the parameters of the S-boxes, and can be easily computed. Cryptanalysis can be accomplished by

$$\underline{K} = B^{-1}(\underline{C} - A\underline{P}) \ , \tag{17}$$

where B^{-1} is the pseudo-inverse of B. The number of operations required to compute K is approximately $64^3 = 262{,}144$ and takes only seconds on a minicomputer. If B does not have full rank, the computed key is equivalent, although not necessarily equal to, the true key.

A linear S-box would signal its presence because it must have a 0 in the first position of its first row. Although the S-box shown below is not linear, it is as weak as one that is linear.

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

This S-box is affine, and its output $\underline{y}$ can be written as

$$\underline{y} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \underline{x} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} . \qquad (18)$$

The general affine relation is

$$\underline{y} = G\underline{x} + \underline{h} \qquad (19)$$

and, if each S-box is in this form, then (15), (16), and (17) become

$$\underline{C} = A\underline{P} + B\underline{K} + \underline{H} \qquad (20)$$

$$P = A^{-1}(C - B\underline{K} - H) \qquad (21)$$

$$K = B^{-1}(C - A\underline{P} - \underline{H}) \ . \qquad (22)$$

Cryptanalysis is no more difficult than before.

The S-boxes used in DES were examined, and none were found to be affine; however, even nonaffine S-boxes can cause problems. The above S-box is no longer affine if four of its entries are changed, but the "probability" that its output will match the output of the original affine S-box is 60/64 = 15/16 (assuming that random plaintext and key are used and that each round of DES is a good pseudo-random number generator). If eight affine S-boxes are so modified, the probability that the modifications will not affect the output in any of the 16 rounds is $(15/16)^{8\times16} \doteq 1/3870$. With a large number of P-C pairs, (22) could be used to determine a possible solution $\hat{K}$ from each pair. Each is checked by enciphering P_i with $\hat{K}$ and watching for C_i to result. Approximately 3870 attempts are needed. These computations require less than one hour on a standard computer.

Because of the dangerously small 56-bit key size used in DES, presolving for even a fraction of the key bits results in a fairly rapid solution on a special-purpose computer. Solving for 10 key bits or equivalent information would reduce the search time and cost by a factor of 1024.

The S-box shown below

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

is neither affine nor close to affine in a probabilistic sense, yet it too possesses a dangerous structural flaw since

$$y1 + y2 + y4 = x1 + x3 + x4 \tag{23}$$

where x1 to x6 and y1 to y4 are the inputs to and the outputs from the S-box. This structure is dangerous because each xi is the sum of a bit from R and a bit from K. If the permutation P and the expansion operation E return the selected output bits from each S-box back to the selected input bits, then, by computing the corresponding sum on C and subtracting the sum on P, a linear combination of the bits of K is obtained. Each independent linear combination narrows the key search by a factor of 2.

Because a complete set of eight S-boxes having this property was not devised, it is not known how successfully such a trap door can be hidden. It appears much more difficult to find, especially if it only holds probabilistically. The probability of its holding true must be significantly greater than 1 divided by the key search saving factor. For example, if 20 independent linear combinations are valid with probability 10^{-2}, then the average savings factor is 10^{4}, when compared to an exhaustive search.

V. DES's STRUCTURE

The preceding section demonstrates the importance of determining the structure, if any, present in the S-boxes used in DES. NBS has stated that each row of each S-box is a permutation of the integers 0 through 15, but has refused to reveal whether there is any additional structure. Our first objective, therefore, was to determine if the DES S-boxes (subject to the permutation constraint) were randomly selected or if they were generated to possess structures which are extremely improbable in randomly chosen boxes. The problem is complicated by the ability of the human mind to find apparent structure in random data, which is really not structure at all.

Despite an initial division of opinion, our group is now convinced that the DES S-boxes were carefully chosen with certain structures in mind.

This makes DES suspect on two grounds. First, it appears that structure should be avoided and that a "typical" randomly chosen S-box is at least as good as one selected from a probabilistically small, specially structured set. Second, it is a well-established cryptographic principle that a cryptosystem should be secure even against an opponent who has full knowledge of the system's structure. Until NBS makes such structural information available, very little faith can be placed in DES' security. If someone involved in the design of DES were to turn against it, he would have a significant advantage.

The Japanese "PURPLE" cryptosystem, broken just prior to World War II, is a good analogy. Two years of intensive cryptanalysis was required to discover the structure of the system; however, after the structure became known, keys could usually be recovered in a matter of hours [5, pp. 18 and 22].

A. Observed DES Structure

1. The least significant bit (y4) in row 1 of S3 is the complement of the LSB in row 2 of S3. The LSB is determined by the parity (odd or even) of the entry; however, it becomes more apparent in the Appendix which gives the binary forms for the S-boxes, with the four rows converted

into four columns for reasons of space. The first column indicates x2, x3,x4,x5, the middle four bits of the input to the S box. (Recall that x1 and x6 determine the "row" with 00 = row 1,...,11 = row 4.)

Other complements can also be seen in the Appendix. In S2, y2 is complemented by going from row 2 to 4. In S7, y4 is complemented by going from row 1 to 3. And in S8, y2 is complemented by going from row 1 to 2.

Neglecting the permutation structure, the probability of any one such behavior is 2^{-16}. The expected number of occurrences, therefore, is

$$2^{-16} \cdot \binom{4}{2} \cdot 4 \cdot 8 = 0.0029 \equiv \lambda \ . \tag{24}$$

The $\binom{4}{2} = 6$ factor is the number of ways in which rows can be paired within an S-box; the factor of 4 accounts for the four output possibilities; and 8 is the number of S-boxes. Using the Poisson approximation, the probability of finding four such patterns is

$$e^{-\lambda}\lambda^4/4! = 3.1\,\mathrm{E}-12 \ . \tag{25}$$

If the permutation constraint is taken into account,

$$\lambda = \left[1\Big/\binom{16}{8}\right] \cdot \binom{4}{2} \cdot 4 \cdot 8 = 0.015 \ , \tag{26}$$

and the probability of finding four such patterns becomes 2.0E - 9. This is incontrovertible evidence for the presence of structure.

The possible danger associated with this behavior becomes more obvious when it is restated as a 50 percent XOR. For example, in S3, when x1 = 0 (i.e., in rows 1 and 2)

$$y4 = f(x2,x3,x4,x5) + x6 \tag{27}$$

because changing x6 causes a move from row 1 to 2 (or from row 3 to 4 when x1 = 1). Since x1 = 0 50 percent of the time, the name's [illegible] derivation (50 percent XOR) is clear. A 50 percent XOR is different from but related to a half XOR which will be of importance in later discussions. The XOR of two binary variables (say x1 and x6) can be written as

$$x1 + x6 = x1\overline{x6} + \overline{x1}x6 \,. \tag{28}$$

Either term on the RHS of this equation will be referred to as half of an XOR. Note that x1x6 and $\overline{x1x6}$ are each half of an XOR complement $(\overline{x1 + x6})$.

2. To generalize the above structure, a program was written to find 25 and 50 percent XORs involving bits other than x1 or x6. A 25 percent XOR means that an output bit is toggled by an input bit (complemented when it is, with the other five inputs arbitrary but constant) when two other inputs are held constant. A 17 page printout revealed that the S-boxes have a much closer relationship to linear structure than would appear safe. By way of comparison, a randomly generated set of S-boxes had less than one page of output.

The number of 25 and 50 percent XORs found are as follows.

S-box	S1	S2	S3	S4	S5	S6	S7	S8
no. of 25 percent XORs	30	61	68	12	32	64	53	57
no. of 50 percent XORs	0	4	4	0	2	1	2	1

Here, S4 appears to be further from linear than the other seven S-boxes. Surprisingly, this is not so (see item 3 on page 18).

The 14 50-percent XORs are

S-box no.	2	2	2	2	3	3	3	3	5	5	6	7	7	8
output no. is toggled by	1	2	2	2	1	2	4	4	2	2	3	2	4	2
input no. when	4	1	1	1	4	1	2	6	2	6	4	5	1	6
input no.	2	2	5	6	1	3	1	1	4	4	5	4	6	1
equals	1	0	0	1	1	1	0	0	1	1	0	0	0	0
				*				*					*	*

where * denotes a bit being complemented going from one row to another within an S-box.

A program was also written to find "nontogglings" or invariances. An output bit is "25 percent invariant to an input bit" if its value does not depend on that input when two other inputs are held constant. The small number of these in comparison to 25 percent togglings is striking.

In S-box	1	2	2	2	2	2	3	3	3	3	5	6	6	6	7	7	8	8
output no. is invariant	1	1	2	2	2	4	1	2	4	4	3	1	1	4	3	4	3	3
to input no. when	1	4	2	3	6	4	2	3	3	4	6	2	3	6	5	2	6	6
input no.	3	5	6	6	3	5	3	5	5	5	4	6	5	4	6	6	2	4
equals and	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	0	0	1
input no.	2	2	3	2	2	2	1	4	1	1	1	3	2	1	4	3	1	1
equals	1	0	1	0	0	1	1	0	0	0	1	1	0	0	0	1	0	0

3. Examination of the above lists reveals that S4 possesses a very different structure from the other S boxes and very few partial XORs. As a result, we searched for other forms of structure. A complex line of reasoning finally led us to observe that, in the binary form of S4, going from row 1 to 2 causes y1 to become $\overline{y2}$, y2 to become y1, y3 to become y4, and y4 to become $\overline{y3}$; this is also true going from row 3 to 4.

The second and fourth rows of S4 are thus entirely redundant. This relation can be compactly expressed as

$$S4(\underline{x} + 000001) = (2,1)(3,4)\ S4(\underline{x}) + (x6,\overline{x6},\overline{x6},x6)\ , \qquad (29)$$

where (2,1)(3,4) means to interchange the first and second bits in addition to the third and fourth bits of the quantity that follows.

Actually, S4 is 75 percent redundant because the fourth row can also be obtained from the first. Again referring to the binary version of S4, note that the y1 values in the first half of row 1 are 01100011, which exactly match the y4 values in the first half of row 4 when read backward. Similarly, y1 in the second half of row 1 matches y4 in the second half of row 4 when read backward (00101101). The same is true of y2 in row 1 and $\overline{y3}$ in row 4 (11100100 and 00010111). This pattern continues; y3 in row 1 equals $\overline{y2}$ in row 4 read backward, half at a time; and y4 in row 1 equals y1 in row 4 read in the same manner. This relationship can be expressed as

$$S4(\underline{x}) + (1,4)(2,3)\ S4(\underline{x} + 101111) = 0110 \ . \tag{30}$$

Note that (1,4)(2,3) reverses the four bits that follow.

Another property that follows from (30) is that, in S4,

$$y1 + y2 = g(x1,x2,x3,x4,x5) + x6 \tag{31}$$

$$y3 + y4 = h(x1,x2,x3,x4,x5) + x6 \ . \tag{32}$$

The sums $y1 + y2$ and $y3 + y4$ are toggled 100 percent of the time by x6. This is especially surprising because S4 had the fewest 25 percent XORs of any S-box on individual output bits, but it has a closer resemblance to a linear S-box than any behavior found in the others. The resemblance to the trap door S-box on page 14 is striking. Note that (31) and (32) imply that $y1 + y2 + y3 + y4$ does not depend on x6, as was noted in (14).

4. A program was written to search for generalizations of the structure in (13). For $1 \leq i \leq 8$ and all 63 nonzero binary 6-vectors $\underline{c}$, tables of $Si(\underline{x}) + Si(\underline{x} + \underline{c})$ were printed out. If (13) had held for i_0, $\underline{c}_0$ the associated table would have consisted of only 0000 entries because + and - are the same in binary arithmetic. Although no all $\underline{0}$ table was found, this printout was extremely useful in determining patterns and structure. Because not all patterns have yet been related to S-box structure, there is probably much structure still to be discovered.

One discernible pattern was that, if $\underline{c}$ had only a single 1 (e.g., $\underline{c}$ = 001000), all entries in the table would have at least two 1's (e.g., 0110 or 1101). Each S-box was chosen so that complementing only one input bit, with the other five held constant, complements at least two output bits.

We believe that this pattern was built into the system to help avoid a key clustering attack. If the same plaintext enciphered under two similar keys yields two similar ciphertext blocks, one could attempt to find a key near the correct one and then perform a local search to determine it exactly. This is far more computationally efficient than searching all keys. For example, determination of a key $\hat{K}$ that differs from the right one in five or fewer bits requires testing only $2^{40} \doteq 10^{12}$ properly spaced keys. It is then necessary to search the approximately four million keys which differ from $\hat{K}$ in five or fewer places to find the exact key in use. The effort required for this search is five orders of magnitude less than for an exhaustive search and is easily accomplished with currently available hardware.

To prevent this type of attack, the system should have strong error propagation characteristics. The near linearity of the S-boxes may be partially the result of the desire for this 2:1 error expansion. (Or, if the linear structure is part of a deliberately set trap door, the need for error expansion will probably be held responsible, and the linearity claimed to be a good property rather than a weakness.) Although a large rapid avalanche of change is valuable in defeating a key clustering attack, it is not worth the danger of quasi-linear S-boxes. If more randomly chosen S-boxes would yield to a key clustering attack with only 16 rounds, we would encourage an increase in the number of rounds to give the avalanche more time to develop.

Other patterns have been found from the $\underline{c}$-test tables. We have noted that, if $\underline{c}$ has a single 1, the table's entries will contain at least two 1's. If $c1 = c6 = 0$, then we remain in the same row of the table, and there will be at least one change in the output. (Because each row is a permutation of 0 through 15, no value is repeated.) This property appears in Table 2 which lists the number of 0000 entries found for each value of $\underline{c}$, summed over all eight S-boxes. Duplications ($\underline{x}$ and $\underline{x}' = \underline{x} + \underline{c}$) are not counted twice. Each c1c6 column must sum to 256, but columns 01 and 10 show a noticeable difference; the 10 column has four 0 entries besides the obvious one ($\underline{c} = 100000$), while the 01 column has only the obvious 0 entry ($\underline{c} = 000001$). This difference is unexpected because inputs 1 and 6 are structurally equivalent from the

Table 2

NUMBER OF 0000 ENTRIES IN ALL S-BOXES FOR EACH VALUE OF c = c1c2c3c4c5c6

c2c3c4c5	c1c6			
	00	01	10	11
0000	256	0	0	$8^{1,2,4,5,8}$
0001	0	28^{6}	41	$10^{1,6,7}$
0010	0	24	25^{4}	$13^{4,6,8}$
0011	0	15	$15^{1,4,5+}$	25^{3}
0100	0	39	37	$9^{2,3,4}$
0101	0	$10^{4,7}$	15^{4}	33
0110	0^{*}	31	24	20
0111	0	$11^{2,4}$	23^{4+}	34^{4+}
1000	0	21^{8}	0	$6^{2,4,5,7}$
1001	0^{4*}	16^{5}	23	$14^{4,7}$
1010	0	$14^{1,8}$	0	$9^{2,4,5,7}$
1011	0	$12^{3,5,8}$	$12^{4,6}$	17^{4}
1100	0	$10^{5,7}$	0	$9^{2,4,5}$
1101	0	$6^{2,4,6,7}$	$24^{4,5}$	$23^{5,6}$
1110	0	$13^{2,3}$	0	$6^{1,2,5,6}$
1111	0	$6^{3,4,5,6}$	17^{4}	$20^{6,8,2+}$

*c = 001100 is the only entry with more than a single 1 which always causes at least two changes in the output.

Superscripts denote the identity of individual S-boxes having no 0000 entries.

In c = 010010, superscript 4* indicates that S4 had at least two bit changes in its output.

Superscripts followed by a + denote an S-box having no single bit changes, but some 0000 entries.

point of view of error propagation. The pattern of the four 0's (whenever c2 = 1 and c5 = 0) is indicative of still hidden structure.

5. Using the c-test printout as a guide, an attempt was made to find simple expressions for single outputs in terms of XORs; this was done manually and without a theoretical framework. (The Quine-McCluskey minimization works for an inclusive OR of products, but not for an XOR.) These preliminary results are surprisingly encouraging for a cryptanalyst. For example, in S2,

$$\begin{aligned} y2 = {} & x1 + x5 + x2x3 + \overline{x2}\,\overline{x4} + \overline{x3}x6 \\ & + x2x4x6 + \overline{x3}x4x5x6 + x1x2x3x5\overline{x6} \\ & + x1x2x4x5x6\ . \end{aligned} \qquad (33)$$

Dropping the last three terms results in an extremely simple expression for y2 which is correct 57/64 of the time. It is probably possible to find an approximation of comparable complexity with a higher probability of being correct than this simple truncation operation.

When x1 = 1, y2 has the even simpler expression

$$y2 = \overline{x1} + x2 + x3 + x5 + x6 + x1\overline{x3}x5\ . \qquad (34)$$

In S3,

$$\begin{aligned} y4 = {} & \overline{x1} + x2 + x3 + x5 + x6 + \overline{x1}x3\overline{x5} + \overline{x1x4x5} + x1\overline{x4x6} \\ & + x1x2\overline{x3x5} + x1x2\overline{x5}x6 + x1x2x3\overline{x4}x6 \end{aligned} \qquad (35)$$

which, when x1 = 0, can be transformed into

$$y4 = x2 + x6 + x3x5 + x4\overline{x5}\ . \qquad (36)$$

Note the half XOR, $x4\overline{x5}$, and the half XOR complement, x3x5.

These expressions indicate that an attempt to solve for K in terms of P and C (or parity sums thereof) may result in a simpler set of equations than one would expect. Because $x + x = 0$, a significant number of cancellations occur, especially if half XORs combine to form regular XORs. As a result, if approximations to the S-boxes can be found in terms of an XOR of products with not many terms, it may be possible to solve the final set of equations on a large general-purpose computer. Further study is required to find such approximations, to estimate the number of terms after 16 rounds, and to determine the difficulty of solving the resultant set of equations.

6. The rows of S1, S2, S3, and S4 appear to have a different permutation composition from those of S5, S6, S7, and S8. Each of the four rows of S1 represents an even permutation, and the rows of S2, S3, and S4 all represent odd permutations; the remaining S-boxes mix odd and even permutations. To be exact, S5's rows are OEEE, S6's are OEEO, S7's are OOOE, and S8's are OOOE. The same parity permutation in rows 2 and 3 possibly indicates some relationship between these two rows.

The parity of a permutation is calculated by noting its cycle structure. For example, row 1 of S1 has cycles (0,14)(1,4,2,13,9,10,6,11,12,5,15,7,8,3). Each cycle of even length is assigned the value 1 and each cycle of odd length is assigned the value 0. These values are then added modulo-2 to obtain the parity (1→ odd 0→ even) of the permutation.

7. If structure in the S-boxes leads to a trap door, it may have to be coupled to the permutation P which is part of f(R,K). There is a slight indication of such coupling, but it is probably only an example of seeing "patterns" in random data. Obviously, the all-0 input to f(R,K) is special. Surprisingly, its output in a hexadecimal representation is D8D8DBBC. If the permutation P is deleted, $\underline{0}$ maps into a more random looking pattern, EFA72C4D. There is a similar slight indication of coupling to the expansion operation because the expanded form of D8D8DBBC is 27,49,27,49,27,55,55,57. We have separated the expanded version into eight 6-bit bytes (represented as decimal integers 0 to 63) because this is how it enters the eight S-boxes. This shows that the context of each D, 8, and B (the bits on either side of the 4-bit group) is the same.

8. A program was written to count the number of matches between pairs of rows of S-boxes, both within an S-box and between two different ones. The number of agreements varied from 0 to 7, neglecting comparisons of a row to itself which always produces 16 matches. Considering pairs of rows in different S-boxes and then pairs in the same box resulted in the following number of occurrences of i matches.

i	0	1	2	3	4	5	6	7
no. of pairs of rows in different S-boxes with i matches	182	144	73	36	5	3	4	1
no. of pairs of rows in same S-box with i matches	43	3	1	1	0	0	0	0

The noticeable difference between intra-S-box matches and inter-S-box matches is partly attributable to the requirement that any single bit change in the input to an S-box produces at least two bit changes in the output. As a result, there can be no agreement between rows 1 (00) and 2 (01), rows 1 and 3 (10), rows 2 and 4 (11), and rows 3 and 4. Of the 43 zero matches, 32 are due to this effect. Even the remaining i count (11,3,1,1,0,0,0,0) differs moderately from the inter-S-box count when both are normalized, which indicates dependencies (negative correlation) between the rows of an S-box.

In the distribution of i in the inter-S-box count, four or more matches occur only 13 times. Eight of these 13 (62 percent) occur between adjacent S-boxes (including S1 as being adjacent to S8), although these account for only 29 percent of the row-pairs counted. This is indicative of dependencies, with a positive correlation, between adjacent S-boxes.

9. With reference to the Appendix, in S8 the fourth entry in each row is the bit by bit complement of the seventh entry; similarly, the eleventh and fifteenth entries in each row are complements. These were the only two cases of this structure found and are probably the result of some higher level of structure in S8. (Similar small observations led

to the simple description of S4 noted in 3 above.) Another anomaly (similar to another initial observation on S4) is that a 6 in row 1 (or 2,3,4) is always paired with a 9 in row 3 (or 4,1,2). Nines and 6's seem to be together much more often than expected in other S-boxes also. A total of 11 pairings (in S1 through S8, the numbers are 2,1,0,0,1,1,2, 4) were found, whereas only six are expected on probabilistic grounds.

Another anomaly similar to the 75 percent redundancy in S4 is exhibited by S8 since $y4$ in rows 1 and 3 equals $\overline{y1}$ in rows 2 and 4, respectively. This could be indicative of a more general and, as yet, unfound structure in S8.

B. <u>Lack of Structure</u>

1. No S box is affine. Further, the function which interchanges the rows is not affine.

Letting $\underline{x}_1 = 000000$, $\underline{x}_2 = 000010$, $\underline{x}_3 = 000100$, and $\underline{x}_4 = 000110$, if Si is affine,

$$Si(\underline{x}_1) + Si(\underline{x}_2) + Si(\underline{x}_3) + Si(\underline{x}_4) = 0000 . \tag{37}$$

This was a simple test because the four values on the LHS constitute the first four entries in the first row of Si. The results were negative.

Using the first four entries in another row tests for affinity of the 4-bit to 4-bit mapping defined by that row. The test results were negative for all 32 rows (four in each S box). This was unexpected because even a nonaffine mapping will pass the test (37) approximately one time in 16 and will require additional tests to reveal its nonaffinity. Because $(15/16)^{32} = 0.13$, the rapidity of our test is mildly surprising. If the fact that each row is a permutation is taken into account, the probability of such a rapid, negative test result drops even further to $(0.9231)^{32} = 0.077$.

A similar affinity test was performed on the 2-bit to 4-bit mappings which result by fixing the middle four inputs to an S-box. The test is to XOR the first entries in each row (e.g., in S1 does $14 + 0 + 4 + 15 = 0$?, where + denotes XOR). All eight tests were negative.

The overall algorithm was also tested for affinity by holding the key fixed and enciphering the plaintexts 0^{64}, $0^{63}1$, $0^{62}10$, $0^{62}11$ and XORing the resultant ciphertexts. The result was negative.

2. As noted above, no S-box is invariant under $\underline{x} \leftarrow \underline{x} + \underline{c}$ for any $\underline{c} \neq \underline{0}$.

3. No S-box is equivalent to any other S-box under permutation and partial complementation of inputs and outputs. Further, except within S4, whose structure has already been discussed, no output of any S-box is equivalent to another output of that or another S-box under complementation and permutation of inputs and possible complementation of the output. The same lack of equivalences applies to inputs.

4. Each row of an S-box defines a function from the set of integers 0 through 15 to itself. The method of finite differences was used to determine if these functions can be represented as low degree polynomials, modulo some integer $N \geq 16$. No such structure was found for any N; for example, when $N = 17$, no row can be described by a polynomial of degree less than 14.

5. Adding 1 to each element of a row of an S-box produces the numbers 1 through 16, which are the nonzero residue classes modulo-17. Each row, therefore, can represent a function $f(x)$ from the integers 0 through 15 to the nonzero residue classes modulo-17. Let g be a primitive root modulo-17 (for example, $g = 5$). If $f(x) = g^{ax+b}$, mod-17 for some integers a and b, then $f(0) \cdot f(2) = f(1)^2$ mod-17. This test proved that no row of any S-box can be described by a function with the above form.

6. Polynomial structure for finite fields of order 16: The numbers 0 through 15 can be identified with polynomials mod-2 of degree less than 4; for example, 13 = 1101 in binary corresponds to $x^3 + x^2 + 1$. These polynomials of degree less than four form a field with 16 elements; multiplication is performed modulo an irreducible, degree 4 polynomial $g(x)$.

Any map from a finite field to itself is a polynomial function, but the degree of this polynomial depends on the choice of $g(x)$. For example, there is at most one (and usually not any) $g(x)$ for which the function has the form $F(y) = Ay + B$, where A and B are elements of the finite field. It was necessary, therefore, to develop a test for low degree polynomial structure which was invariant to $g(x)$.

Suppose there is a function $F(y)$ such that, for some choice of $g(x)$, $F(y)$ can be represented in the form $F(y) = Ay^4 + By^3 + Cy^2 + Dy + E$, where A,B,C,D,E are field elements. The following identity then follows.

$$F(0) + F(1) + F(x) + F(x+1) + F(x^2) + F(x^2+1) + F(x^2+x) + F(x^2+x+1) = 0 \qquad (38)$$

In terms of the original S-box notation, this means that, if there is some $g(x)$ for which the function F is a pōlynomial of degree ≤ 4, then

$$F(0) + F(1) + F(2) + F(3) + F(4) + F(5) + F(6) + F(7) = 0 \; . \qquad (39)$$

Similarly,

$$F(0) + F(2) + F(4) + F(6) + F(8) + F(10) + F(12) + F(14) = 0 \qquad (40)$$

must hold. Since no row passed both tests, there is no polynomial structure of degree ≤ 4 in any row of any S-box, regardless of the choice for $g(x)$.

The inverses of the maps given by a few of the rows were also checked for polynomial structure by the above test and, again, the results were negative.

7. A rank test was performed on $f(R,K)$ to determine whether the mod-2 sum of any subset of the 32 outputs was equal to the sum of any subset of the 48 inputs. This was done by taking 80-vectors whose

first 48 and last 32 components were the inputs to and the outputs from the eight S-boxes. Because a matrix of rank 80 could be generated with such vectors as its columns, no such linear relationship held between input and output.

By adding a 1 to each of the 80-vectors, a test was performed to determine whether the input and output sums are always complements. Again, no such relationship holds.

The probabilistic linearity discussed following (22), however, would not appear in these tests. More finely tuned tests would detect this linearity but there was no time to implement them.

8. In another test, $P = 0$ was enciphered under each of the 56 unit vector keys (such as $K = 0^{55}1$), and $K = 0$ was used to encipher each unit vector plaintext. Also, $P = (01)^{32}$, $(10)^{32}$, and 1^{64}, and $K = (01)^{28}$, $(10)^{28}$, and 1^{56}, were used in various pairings. The resultant ciphertexts were examined for regularity or relationships. Aside from the P,K,C to $\bar{P},\bar{K},\bar{C}$ effect noted in Section III, none was found by visual examination.

The most unusual finding (which led to the discovery of the complementation symmetry) was that $P = 0^{64}$, $K = 1^{56}$ results in ciphertext = CAAAAF4DEAF1DBAE (in hex representation). The four consecutive A's are somewhat surprising, but are probably a random pattern.

9. Schroeppel proved the following theorem which was used to show that exact expressions for any output of S1 or S4, as an XOR of products, will involve at least one product of five variables. A program should be written to check the manual calculations for S1 and S4 and to extend them to the other S-boxes.

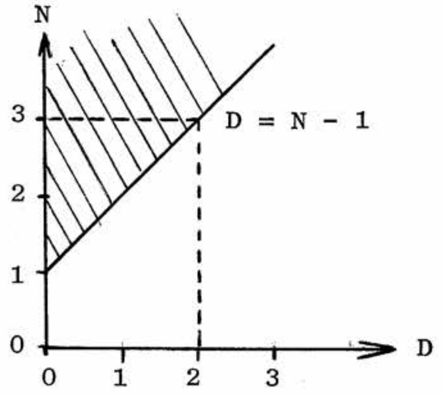
Theorem 5.1. A Boolean function of N variables can be expressed as an XOR of product terms, each of degree $\leq D$ (i.e., with D or fewer literals) if and only if for all restrictions of the function to $D+1$ variables (i.e., $N-D-1$ variables are held constant), y takes on the value 1 an even number of times in the 2^{D+1} points of the restricted domain.

Example. $y = x1x2 + \overline{x2}x3 + x4$ takes on 16 values, eight of which are 1 and eight of which are 0; therefore, $D+1 \leq 4$ or $D \leq 3$. Restricting the function by holding x1 constant yields a {1,0} count of 4,4 both when $x1 = 0$ and when $x1 = 1$. Similarly, holding x2,x3 or x4 at 0 or at 1 yields a count of 4,4; therefore, $D \leq 2$. Holding $x3 = 0$, $x4 = 0$ yields a count of 1,3, so $D > 1$.

Proof of Theorem 5.1.

If $y = f(x1,x2,\ldots,xN)$ has terms of degree at most D, each term has an even number of 1's on any $D+1$ restriction because the active (nonconstant) part of each term repeats 2^k times, where $k = D+1-M \geq 1$ and $M \leq D$ is the number of active variables in the term. But then y has an even number of 1's on any $D+1$ restriction since it is the "parity check" on all terms.

The proof in the other direction uses mathematical induction. The shaded area in the figure below indicates the region $N \geq 1$, $D \leq N-1$ for which the result must be proved. The first step is to prove the result for the N axis $(D = 0)$ and then to induct on N and D to show that, if it is true for N and D, it is also true for $N+1$ and $D+1$. This will prove it for the entire region.



To establish the base for the induction $(D = 0)$, it must be shown that, if $y = f(x1,x2,\ldots,xN)$ has an even number of 1's in all restrictions to $D+1 = 1$ variables, then y is either constantly 0 or constantly 1 for all 2^N values of its argument. An example with $N = 3$ illustrates why the existence of Gray codes (which "count" through all 2^N binary N-vectors in such a manner that adjacent counts differ in only one bit) implies the result.

For $N = 3$, the set {000,001,011,010,110,111,101,100} is a Gray code. Let $y = f(x1,x2,x3)$ and hold x1,x2 constant at 0 to restrict $\underline{x}$ to 000 and 001, the first entries in the Gray code. By assumption, y has an even number of 1's in this range; therefore, f(000) = f(001). Now, restricting x1 = 0, x3 = 1, it is found that f(001) = f(011). In a similar manner, x1,x2,x3 is taken through all 2^N values to show that f(000) = f(001) = f(011) = ... = f(101) = f(100). Equivalently, f(x1,x2,x3) ≡ 0 or ≡ 1 and is of degree 0.

Having established the base for the induction, we now show that, if the result is true for N and D, it is also true for N + 1 and D + 1. Let y = f(x1,x2,...,xN, x N+1) be a function of N + 1 variables such that all restrictions to (N + 1) - (D + 1) - 1 = N - D - 1 variables contain an even number of 1's. We must show it can be expressed as an XOR of products, each of degree $\leq$ D + 1.

Holding x N+1 = 0 or 1 produces functions f'(x1,x2,...,xN) and f"(x1,x2,...,xN). Because all restrictions of f(·) to N - D - 1 variables contain an even number of 1's, so do all restrictions of f'(·) and f"(·) to N - D - 1 variables. By assumption, therefore, f'(·) and f"(·) can be written as XORs of products, each of degree $\leq$D, and

$$f(x1,x2,\ \ldots,\ x\ N+1) = x\ \overline{N+1}\ f'(x1,x2,\ \ldots,\ xN)$$

$$+\ x\ N+1\ f''(x1,x2,\ \ldots,\ xN) \qquad (41)$$

is an XOR of products, each of degree $\leq$ D + 1. This follows from the distributivity of AND over XOR; that is,

$$a(b + c) = ab + ac\ . \qquad (42)$$

This completes the proof of Theorem 5.1.

<u>Corollary</u>. Each output of each S-box is expressible as an XOR of products, with each term having degree at most five.

<u>Proof</u>.

Taking D = N - 1, Theorem 5.1 states that a function with an even number of 1's in its truth table need not involve terms of degree N. Because each row of an S-box is a permutation of the integers 0 through 15, any output bit has eight 0's and 1's in each row and 32 0's and 1'S in the complete truth table.

It is a good sign that the two S-boxes tested (S1 and S4) each required terms of the maximum possible degree (five).

As a final comment, note that taking $c = 1$ in (42) shows that $a\bar{b} = ab + a$. The maximal degree required is therefore not affected by whether or not complementation of literals is allowed.

10. As discussed in the next section, the S-boxes were probably chosen to minimize the difference between the number of 1's and the number of 0's in any S-box output when any input bit is held constant. Holding one input bit constant results in 32 values for each output bit and DES' S-boxes are much closer to an even 16,16 distribution on 0 and 1 than would be expected. For reasons explained in the next section, this probably strengthens the algorithm.

VI. STATISTICAL TESTS

Several statistical tests were performed to determine whether P-C pairs could be used to predict bits of K. If P is variable, then we have a mapping from 128 to 56 bits; setting $P = 0^{64}$ reduces the mapping to one from 64 to 56 bits, and this reduced size makes it more likely that correlations will be found. These tests are of obvious use in a chosen plaintext attack, but are useful even in a known plaintext attack. If P_1-C_1 is the known plaintext-ciphertext pair, our statistical tests could be duplicated with $P = P_1$ instead of $P = 0$.

A set of $N = 8000$ K's were chosen at random and $P = 0$ was enciphered with each, resulting in 8000 ciphertext blocks. We then searched for correlations between C and k_i (the i^{th} bit of K, $1 \leq i \leq 56$) by computing $\underline{RS}_i(0)$ and $\underline{RS}_i(1)$, the running (vector) sums of all C's corresponding to $k_i = 0$ and $k_i = 1$. We then let

$$\underline{V}_i = [\underline{RS}_i(0)/n_i(0)] - [\underline{RS}_i(1)/n_i(1)] \tag{43}$$

where $n_i(0)$ was the number of terms contributing to $\underline{RS}_i(0)$ (the number of times $k_i = 0$), and $n_i(1) = 8000 - n_i(0)$.

If DES is a good pseudo-random number generation (PRNG), each component of $\underline{V}_i$ will have a normal distribution with mean 0 and variance $\sigma^2 = 1/8000$. For all $1 \leq i \leq 56$ and $1 \leq j \leq 64$, values of $V_{ij} > 3\sigma$ were printed as an indication of correlation between k_i and C_j. (Note that $V_{ij} < -3\sigma$ is as significant but, as a result of a programming oversight, they were not printed.)

Because the probability that $V_{ij} > 3\sigma$ for fixed i,j is 0.0014, approximately five false alarms can be expected on 56 × 64 values. If $V_{ij} > 4\sigma$, however, the false alarm rate drops to 0.11 over all i,j. At 5σ, it is 0.0005 and is truly indicative of correlation. The 3σ cutoff was chosen to allow later examination of the special cases found. Our sample size of 8000 was small and, if larger runs are made, special attention should be paid to those i,j values found here. Six values of $V_{ij} > 3\sigma$ were found.

i	3	13	27	37	41	46
j	46	40	28	6	15	16
V_{ij}	0.0337	0.0372	0.0347	0.0431	0.0450	0.0348
V_{ij}/σ	3.0	3.3	3.1	3.9	4.0	3.1

The χ^2 statistic was also computed

$$\alpha_i = \sum_{j=1}^{64} (V_{ij}/\sigma)^2 \tag{44}$$

to determine whether any key bit was particularly well correlated with C. Values below the 1 percent and above the 99 percent values were specially marked, as were those below 5 percent or above 95 percent. Random fluctuations will cause approximately (0.02)(56) = 1.1 occurrences in the first category and (0.1)(56) = 5.6 in the first or second category category even if there is no real correlation. A total of five values were found, all in the second category. This is indicative of no real correlations, although larger tests are required to make a stronger case.

We also let

$$\underline{W}_i = \underline{V}_i / \|\underline{V}_i\| \tag{45}$$

and used the decision rule

$$\text{decide } k_i = 0 \text{ if } \underline{W}_i \cdot (2\underline{C} - 1) > 0$$

$$\text{decide } k_i = 1 \text{ if } \underline{W}_i \cdot (2\underline{C} - 1) \leq 0$$

to predict k_i on 8000 new ciphertext blocks. If DES is a good PRNG, the expected probability of error on each bit is 0.50 and the standard deviation is $(1/4 \cdot 1/8000)^{1/2} = 0.0056$. Probabilities of error of 0.489 or less are at the 2σ level, and approximately one false alarm can be expected in 56 tries; 3σ (0.483) or 4σ (0.472) are more surprising. Only one value, close to 2σ, was found. Thus far, DES gets good grades on statistical regularity.

A similar set of tests, also with $N = 8000$, was run on a two-round version of DES with the thought that weak statistical structure in the 16-round algorithm may be more pronounced. Studying the correlations in the two-round version may lead to the discovery of correlations in the full algorithm.

As an analogy, let

$$Y_j = \sum_{i=1}^{16} X_{ij} \tag{46}$$

where each X_{ij} is an independent 0-1 random variable with bias (probability $X = 1$) of 0.25, and the sum is mod-2. Then, Y has bias 0.499992, and a sample of over 10^{10} Y's are required to validate the statistical structure of Y. By comparison, a sample of only 100 X's would suffice to validate the structure in X and, by implication, in Y.

Table 3 lists V_{ij} values more than $+3\sigma$ from the mean of 0 and is impressive both in its length and in the V_{ij}/σ values. Random fluctuations will cause V_{ij}/σ to exceed 5 only once in a 1000 tries, yet 25 such values were observed. The values of V_{ij}/σ which exceed 10 are even more indicative of statistical structure. It is apparent that the two-round version of DES has significant statistical structure that can be used in cryptanalysis.

A large number of α_i were also observed in the 99 percent range, which indicates that the above, linear decision rule can successfully predict k_i. Table 4 lists those values of i for which α_i was above the 99 percent value or for which the probability of correct prediction was above 51.5 percent. The large number of double entries ($P(\alpha_i) > 0.99$ and $Pr(\text{correct}) > 0.515$) indicates the success of the linear decision rule. Further, when one entry is missing, the other is small. The quantity β_i is defined as

$$\beta_i = 100\ [Pr(k_i \text{ correctly predicted}) - 0.5]\ . \tag{47}$$

Thus, $\beta_i = 2.5$ corresponds to a 52.5 percent correct prediction.

Table 3

V_{ij}/σ FOR DES TRUNCATED TO FIRST TWO ROUNDS (N = 8000, σ = 0.0112)

i	1	2	5	5	7	9	9	10	13
j	37	49	33	45	7	39	49	39	11
V_{ij}/α	6.1	5.5	4.6	5.1	3.1	6.8	4.8	3.1	5.5
i	15	15	15	18	18	21	23	25	26
j	7	21	35	37	51	21	31	3	23
V_{ij}/α	7.2	5.5	11.1	12.5	10.6	3.1	7.6	7.8	16.5
i	27	28	30	34	35	35	35	36	36
j	55	41	45	15	3	5	23	19	55
V_{ij}/α	3.4	6.1	5.5	11.9	4.3	3.6	5.5	3.7	11.1
i	37	37	39	40	42	44	49	49	49
j	1	41	63	25	38	17	25	50	59
V_{ij}/α	10.0	11.9	4.7	3.1	3.5	5.1	3.3	3.2	5.5
i	51	51	53	53	53	54	55		
j	13	15	31	33	43	41	29		
V_{ij}/α	5.3	5.3	4.4	4.8	4.7	4.4	5.5		

Table 4

VALUES OF α_i AND β_i FOR TWO-ROUND VERSION OF DES

i	1	2	5	6	9	13	15	16	18	19
α_i	200	169	270	93	172	151	269		309	
β_i	3.5		6.9		5.1	3.1	4.6	1.6	5.7	1.6
i	21	23	25	26	27	28	29	30	34	35
α_i	144	159	211	375	142	185		244	649	119
β_i	3.0		6.7	4.5	2.1	2.9	1.9	2.7	5.1	4.1
i	36	37	38	39	41	42	44	45	46	47
α_i	331	352	362	120		98	136			105
β_i	3.8	3.5	6.5	2.4	4.6	2.9	3.0	1.7	1.9	
i	49	50	51	52	53	54	55			
α_i	214		149		124	141	148			
β_i	2.2	4.0	3.3	2.2						

The large statistical fluctuations present in the two-round version of DES may not be indicative of a weakness in the full 16-round algorithm because, as noted below, the S-boxes were chosen to minimize these fluctuations, and random S-boxes would have even larger fluctuations in a two-round implementation.

Since $P = 0$, $L_0 = 0$ and $R_0 = 0$; then

$$L_1 = R_0 = 0 \tag{48}$$

$$R_1 = L_0 + f(ER_0 + K_1) = f(K_1) \tag{49}$$

and

$$L_2 = R_1 = f(K_1) \tag{50}$$

$$\begin{aligned} R_2 &= L_1 + f(ER_1 + K_2) \\ &= f\left(Ef(K_1) + K_2\right) . \end{aligned} \tag{51}$$

As a result, the ciphertext

$$C = IP^{-1}(R_2, L_2) \tag{52}$$

includes the 32 bits of $f(K_1)$. Examination of IP^{-1} reveals that these bits occupy the 32 odd positions (1,3,...,63) of C, and not too surprisingly 41 of the 43 j values listed in Table 3 are odd. The two exceptions ($i = 42$, $j = 38$, and $i = 49$, $j = 50$) are near the borderline of V_{ij}/σ values, 3.5σ and 3.2σ, respectively.

The reason for this behavior and for the clustering of V_{ij} values is that each of the 32 bits in $f(K_1)$ is the output of an S-box whose inputs are six key bits in some permuted order. Holding one of these input bits constant and letting the other five vary generates 32 outputs. If the number of 1's in these 32 outputs is not exactly 16, a correlation exists between that pair of input-output bits. The unrestricted 64 values of the output had exactly half 1's because of the permutation property within each row; this is not true, however, when the input bit held constant is one of the middle four (x2,x3,x4 or x5). Probability theory indicates that the V_{ij} value can be used to predict the number of 1's in the 32 value restriction. If there are 16 1's, the expected value of V_{ij} is 0. If there are 15 1's,

$$EV_{ij} = \frac{17}{32} - \frac{15}{32} = 0.063 = 5.6\sigma \qquad (53)$$

where $\sigma = 0.11$ is not the actual standard deviation of V_{ij}, but is the value used in Table 3 which assumed independent samples. Similarly, if there are 14 1's,

$$EV_{ij} = \frac{18}{32} - \frac{14}{32} = 0.125 = 11.2\sigma \qquad (54)$$

etc. Choosing decision boundaries midway between the expected values, we estimate 35 occurrences of 15 1's, seven occurrences of 14 1's, one occurrence of 13 1's, and no occurences of 12 or fewer 1's. Because negative values of V_{ij} were not available, the number of occurrences of 17 or more 1's could not be predicted.

If each permutation in each row is chosen randomly,

$$\Pr(k\ 1\text{'s}) = \frac{\binom{32}{k}\binom{32}{32-k}}{\binom{64}{32}}$$

$$= \frac{(32!)^4}{64!\ (k!)^2\ \left((32-k)!\right)^2} \qquad (55)$$

which is obviously symmetric about $k = 16$. A short table of Pr(k) and the expected number of occurrences of k 1's in 128 samples, 128Pr(k), are

k	16	17	18	19	20	21	22
Pr(k)	0.197	0.175	0.121	0.066	0.028	0.009	0.002
128Pr(k)	25	22	15	8.5	3.6	1.2	0.3

It is seen that DES' S-boxes have many fewer occurrences of 18 or more 0's (15 or fewer 1's) than would be expected if the S-boxes were chosen randomly. This is a good characteristic, if it does not include other, weakening structure.

To validate the use of V_{ij} to predict the number of 1's, we selected the $i = 18$, $j = 37$ and 51 and the $i = 37$, $j = 1$ and 41 values; these are all in the 8.4σ to 14σ range which indicates 14 1's and 18 0's in the restricted S-box. Tracing $i = 18$ through PC-1, a left shift and PC-2, k_{18} becomes x5 in S4 during round 1. Tracing C_{37} and C_{51} back through IP^{-1} and P, $C_{37} = y2$ and $C_{51} = y3$ from S4 in round 1. Holding $x5 = 0$ in S4 yields 14 1's in y2 and in y3, in agreement with our prediction. Similarly, $i = 37$, $j = 1$ and 41 corresponds to x5, y1, and y4, all in S5 during round 1. Again, holding $x5 = 0$ in S5 yields 14 1's in both y1 and y4.

It must be remembered that IP^{-1} operates on $R_{16}L_{16}$ not $L_{16}R_{16}$; in the two-round version, this corresponds to IP^{-1} operating on R_2L_2.

VII. TWO ROUND CRYPTANALYSIS

Cryptanalysis of the two-round version of DES was undertaken with the hope of finding methods for attacking the full 16-round algorithm. We were successful in developing relatively simple techniques for breaking the two-round version but did not have time to investigate how these might be used in the 16-round algorithm.

Knowing P is equivalent to knowing L_0 and R_0, and knowing C is equivalent to knowing L_2 and R_2. Since

$$L_1 = R_0 \qquad R_1 = L_0 + f(R_0,K_1) \tag{56}$$

and

$$L_2 = R_1 \qquad R_2 = L_1 + f(R_1,K_2) \ , \tag{57}$$

we have

$$L_2 + L_0 = f(R_0,K_1) \tag{58}$$

and

$$R_2 + R_0 = f(L_2,K_2) \ . \tag{59}$$

Breaking the two-round version therefore reduces to solving two equations of the form $Z = f(R,K)$ for K, with both Z and R known. Because there are four inverse images for each S-box, there are $4^8 = 65,536$ solutions to each of the two equations (58) and (59). Using the constraint that 40 bits of K_1 must match 40 bits of K_2, in fixed permuted positions, allows determination of the one correct solution K. One method deletes the eight bits of K_1 that are not in K_2 and orders the resultant 40-bit quantities. It then selects the corresponding 40 bits of each possible K_2 solution, with the bits permuted to match those of the ordered K_1's, and watches for a match. The number of operations involved is approximately $65536 \log_2(65536) = 1.05$ million, and can be done on a minicomputer.

VIII. POSSIBLE WAYS TO STRENGTHEN DES

We recommend the following techniques for possible strengthening of DES. Further study is obviously needed on most.

1. Increase the number of rounds from 16 to 32, 64, or even more.

2. Use less structured S-boxes. Making an algorithm more complex tends to make it more secure, and low complexity S-boxes appear to be a weakening influence (S4 is particularly poor, being 75 percent redundant).

Even if certain structures are required (such as the 2:1 error expansion and small V_{ij} values), the S-boxes should be chosen randomly from the set which satisfies these properties. If the set becomes too small, there is probably a fault in the basic structure of DES. The increased number of rounds suggested will probably ease the restrictions on the S-boxes.

3. PC-2 should mix the C- and D-register contents.

4. Introduce the effect of the key in a more complex way than just XORing it with ER. The most general way to map six bits of ER and six bits of K into a 6-bit S-box input is through a 12-bit to 6-bit ROM (4098 × 6 organization). This is too expensive but indicates that the problem is solvable. For example, two 6-bit to 3-bit mappings could be used.

Having one or more key dependent S-boxes may even be better.

5. Make the key-scheduling algorithm more nonlinear and lengthen the distance between expanded keys. Currently, a 56-bit key is expanded into a $768 = 16 \times 48$ bit key by repeating each bit between 12 and 15 times. Changing one bit of the 56-bit key can thus change as few as 12 positions in the expanded key. Simple techniques from coding theory produce larger "minimum distances" which would help combat a key-clustering attack.

A BCH code would yield a minimum distance of over 140, and larger values are possible. These key expansion codes could be implemented as a 56-stage feedback shift register and are therefore no more complex than the current C-D registers and shift-schedule memory. Use of a nonlinear code would also be useful.

6. Enlarge the key. We understand that NSA, through the Munitions Control Board, allows 56-bit (and perhaps 64-bit) key systems to be exported but almost always rejects applications for export licenses on larger key systems. This adequately supports our recommendation for a larger key.

IX. FUTURE TASKS

1. Search for additional structure in the S-boxes and relate it to potential weaknesses and strengths. Relationships between sums of inputs and sums of outputs (as in S4) are of particular interest.

2. Use the Quine-McCluskey minimization algorithm and possible extensions to an XOR of products to find minimal expressions for each S-box. Joint minimization of the four outputs should also be considered.

3. Investigate the degree of degeneracy of the 32-bit to 32-bit function from R to y defined by $y = f(R,K)$, with K fixed. Determine whether this can be used in cryptanalysis and how the choice of K affects the degeneracy.

4. Generate S-boxes with various trap doors (such as parity) that allow recovery of a number of key bits. Determine how well hidden they can be and whether any related structures are present in DES.

5. Run longer and more varied statistical tests. Attempt to relate any anomalies to structure in the S-boxes, P, etc., and determine whether they can be used in cryptanalysis.

6. Generate S-boxes at random within the constraints found thus far in DES (such as the 2:1 error expansion), and note whether they avoid the potential weaknesses found in DES' S-boxes.

7. Study the use of approximations to the S-boxes in cryptanalysis. Determine how difficult the equations are to solve as a function of the number of terms present.

8. Check if any row of any S-box is equivalent to a different row of the same or another S-box under input and output complementation and permutation.

9. Run Schroeppel's D test on all 32 S-box outputs.

10. Search for a sum of input bits that toggle a sum of output bits in an S-box.

11. Run a modified rank test searching for overlaps in the null space that indicate probabilistic trap doors.

12. Run a program to determine the number of occurrences of i 1's and 32-i 0's in an S-box output when one input is held constant. Compare to the V_{ij} prediction.

13. Extend the two-round cryptanalysis to a larger number of rounds.

14. Seek ways to reduce cryptanalysis of the full 16 round algorithm to solving two simultaneous equations of the form $f(R_1,K_1) = Z_1$ and $f(R_2,K_2) = Z_2$ for K_1 and K_2, as in the two-round cryptanalysis. Determine whether the birthday problem can be used. (If there are n days in the year, it only takes about $n^{1/2}$ people before two will have a birthday in common. In DES, it only takes about $2^{16} = 65,536$ P-C pairs before two will share a common L_{15}. If a test for finding this pair can be found, approximately 32 key bits could probably be recovered.)

X. CONCLUSION

Structures have been found in DES that were undoubtedly inserted to strengthen the system against certain types of attack. Structures have also been found that appear to weaken the system.

We believe that the potential weaknesses require greater attention because it only takes one successful avenue of attack to break a system. In addition, it is poor security practice to trust a system whose design and certification will not be described. We, therefore, encourage the public release of DES' design principles and the results of IBM's 17 man-year certificational effort.

Appendix

Binary Versions of S-Boxes

SBOX NUMBER 1

0000	1110	0000	0100	1111
0001	0100	1111	0001	1100
0010	1101	0111	1110	1000
0011	0001	0100	1000	0010
0100	0010	1110	1101	0100
0101	1111	0010	0110	1001
0110	1011	1101	0010	0001
0111	1000	0001	1011	0111
1000	0011	1010	1111	0101
1001	1010	0110	1100	1011
1010	0110	1100	1001	0011
1011	1100	1011	0111	1110
1100	0101	1001	0011	1010
1101	1001	0101	1010	0000
1110	0000	0011	0101	0110
1111	0111	1000	0000	1101

SBOX NUMBER 2

C

0000	1111	0011	0000	1101
0001	0001	1101	1110	1000
0010	1000	0100	0111	1010
0011	1110	0111	1011	0001
0100	0110	1111	1010	0011
0101	1011	0010	0100	1111
0110	0011	1000	1101	0100
0111	0100	1110	0001	0010
1000	1001	1100	0101	1011
1001	0111	0000	1000	0110
1010	0010	0001	1100	0111
1011	1101	1010	0110	1100
1100	1100	0110	1001	0000
1101	0000	1001	0011	0101
1110	0101	1011	0010	1110
1111	1010	0101	1111	1001

SBOX NUMBER 3 c

0000	1010	1101	1101	0001
0001	0000	0111	0110	1010
0010	1001	0000	0100	1101
0011	1110	1001	1001	0000
0100	0110	0011	1000	0110
0101	0011	0100	1111	1001
0110	1111	0110	0011	1000
0111	0101	1010	0000	0111
1000	0001	0010	1011	0100
1001	1101	1000	0001	1111
1010	1100	0101	0010	1110
1011	0111	1110	1100	0011
1100	1011	1100	0101	1011
1101	0100	1011	1010	0101
1110	0010	1111	1110	0010
1111	1000	0001	0111	1100

SBOX NUMBER 4 c c c c

0000	0111	1101	1010	001$\boxed{1}$
0001	1101	1000	0110	1111
0010	1110	1011	1001	0000
0011	0011	0101	0000	0110
0100	0000	0110	1100	1010
0101	0110	1111	1011	0001
0110	1001	0000	0111	1101
0111	$\boxed{1}$010	0011	1101	1000
1000	0001	0100	1111	100$\boxed{1}$
1001	0010	0111	0001	0100
1010	1000	0010	0011	0101
1011	0101	1100	1110	1011
1100	1011	0001	0101	1100
1101	1100	1010	0010	0111
1110	0100	1110	1000	0010
1111	$\boxed{1}$111	1001	0100	1110

SBOX NUMBER 5

0000	0010	1110	0100	1011
0001	1100	1011	0010	1000
0010	0100	0010	0001	1100
0011	0001	1100	1011	0111
0100	0111	0100	1010	0001
0101	1010	0111	1101	1110
0110	1011	1101	0111	0010
0111	0110	0001	1000	1101
1000	1000	0101	1111	0110
1001	0101	0000	1001	1111
1010	0011	1111	1100	0000
1011	1111	1010	0101	1001
1100	1101	0011	0110	1010
1101	0000	1001	0011	0100
1110	1110	1000	0000	0101
1111	1001	0110	1110	0011

SBOX NUMBER 6

0000	1100	1010	1001	0100
0001	0001	1111	1110	0011
0010	1010	0100	1111	0010
0011	1111	0010	0101	1100
0100	1001	0111	0010	1001
0101	0010	1100	1000	0101
0110	0110	1001	1100	1111
0111	1000	0101	0011	1010
1000	0000	0110	0111	1011
1001	1101	0001	0000	1110
1010	0011	1101	0100	0001
1011	0100	1110	1010	0111
1100	1110	0000	0001	0110
1101	0111	1011	1101	0000
1110	0101	0011	1011	1000
1111	1011	1000	0110	1101

SBOX NUMBER 7

C

0000	0100	1101	0001	0110
0001	1011	0000	0100	1011
0010	0010	1011	1011	1101
0011	1110	0111	1101	1000
0100	1111	0100	1100	0001
0101	0000	1001	0011	0100
0110	1000	0001	0111	1010
0111	1101	1010	1110	0111
1000	0011	1110	1010	1001
1001	1100	0011	1111	0101
1010	1001	0101	0110	0000
1011	0111	1100	1000	1111
1100	0101	0010	0000	1110
1101	1010	1111	0101	0010
1110	0110	1000	1001	0011
1111	0001	0110	0010	1100

SBOX NUMBER 8

C

C

C

0000	1101	0001	0111	0010
0001	0010	1111	1011	0001
0010	1000	1101	0100	1110
0011	0100	1000	0001	0111
0100	0110	1010	1001	0100
0101	1111	0011	1100	1010
0110	1011	0111	1110	1000
0111	0001	0100	0010	1101
1000	1010	1100	0000	1111
1001	1001	0101	0110	1100
1010	0011	0110	1010	1001
1011	1110	1011	1101	0000
1100	0101	0000	1111	0011
1101	0000	1110	0011	0101
1110	1100	1001	0101	0110
1111	0111	0010	1000	1011

C C

REFERENCES

1. Horst Feistel, "Cryptography and Computer Privacy," Scientific American, Vol. 228, pp. 15-23, May 1973.

2. Whitfield Diffie and Martin Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol. IT-22, November 1976.

3. Seymour Jeffery (Chief, Systems and Software Division, NBS), letter dated January 6, 1975 (error in date--it was sent in January 1976) to Prof. Martin Hellman.

4. Whitfield Diffie and Martin Hellman, "Cryptanalysis of the NBS Data Encryption Standard," submitted to Computer magazine, May 1976.

5. David Kahn, The Codebreakers, New York: Macmillan, 1967.

6. National Bureau of Standards, "Notice of a Proposed Federal Information Processing Data Encryption Standard," Federal Register, Vol. 40, No. 12134, March 17, 1975.