PAIRS AND TRIPLETS OF DES S-BOXES

Donald Davies
Sean Murphy*

Information Security Group,
Royal Holloway and Bedford New College,
University of London,
Egham, Surrey TW20 0EX, U.K.

September 20, 1993

Abstract

This paper describes an investigation of a potential weakness in DES which leads to a statistical property observable in plaintext-ciphertext pairs and dependent on the key. However, the number of encryptions of known plaintext needed to exploit this property is comparable to the number of encryptions of an exhaustive key search, so the "weakness" is mainly of theoretical interest.

Key Words. Data Encryption Standard (DES), Cryptanalysis.

*This author was supported by S.E.R.C. research grant GR/E 64640.

1 Introduction

The Data Encryption Standard (DES) [5] is a block cipher that was adopted by the U.S. National Bureau of Standards as the standard cryptosystem for sensitive but unclassified data. The dependence of the financial community on DES for its data security functions makes it desirable to keep under review the strength of this algorithm. Though many interesting properties have been found, none of these is thought to make it less secure, when used judiciously, than its key size would indicate. For example, the differential cryptanalysis of DES given by Biham and Shamir [3] has a lower complexity than an exhaustive search, but requires the encryption of $2^{47.2}$ chosen plaintexts. This paper describes a potential weakness which, in principle, leads to a statistical property observable in plaintext/ciphertext pairs and dependent on the key. However the attack is of comparable complexity to an exhaustive key search, and requires the encryption of enormous amounts of known plaintext, so the "weakness" is mainly of theoretical interest.

The full description of DES is given by [5], but we begin by describing the details of DES that are relevant for this cryptanalysis. DES is a 16-round Feistel cipher acting on a 64-bit message space under the control of a 56 bit key. A description of the Feistel cipher principle is given in [2]. A schematic representation of DES is given in Figure 1. We ignore the initial permutation IP and its inverse IP^{-1} in our description of the attack. It is trivial to alter the attack to allow for them. For each round, the DES key-scheduling produces a 48-bit subkey for each round by permuting the 56-bit key and then selecting a subset of 48 bits. Thus the i^{th} round subkey K_i $(i = 1, \cdots 15)$ consists of 48 of the 56 bits in the key.

In order to perform the encryption, the 64-bit plaintext is split into two 32-bit halves (L_0, R_0). The plaintext is enciphered by performing 16 iterations or rounds of the following encipherment rules,

$$L_{i+1} = R_i, \qquad R_{i+1} = L_i \oplus f(R_i, K_{i+1}), \qquad i = 0, \cdots, 15,$$

where f is the round encryption function. The ciphertext is then given by (R_{16}, L_{16}). Decipherment is performed by applying the same algorithm to ciphertext but using the subkeys in the reverse order.

We regard the plaintext as (R_{-1}, R_0), when the encipherment rule is

$$R_{i+1} = R_{i-1} \oplus f(R_i, K_{i+1}), \qquad i = 0, \cdots, 15,$$

and the ciphertext is (R_{16}, R_{15}). Thus we have

$$\begin{aligned} R_{16} &= R_{14} \oplus f(R_{15}, K_{16}) \\ &= R_{12} \oplus f(R_{13}, K_{14}) \oplus f(R_{15}, K_{16}) \\ &= R_0 \oplus \{\oplus_{i=1}^{8} f(R_{2i-1}, K_{2i})\}, \end{aligned}$$

and so

$$\oplus_{i=1}^{8} f(R_{2i-1}, K_{2i}) = R_0 \oplus R_{16},$$

and similarly

$$\oplus_{i=1}^{8} f(R_{2(i-1)}, K_{2i-1}) = R_{-1} \oplus R_{15}.$$

Hence, in a known plaintext environment, where both plaintext and ciphertext are known, we can observe the value of an 8-fold XOR of the outputs of the f function. In particular, if DES is being used in some naive feedback mode, we may well observe the XOR of plaintext and ciphertext and hence the 8-fold XOR of the outputs of the f-function.

In order to define $f : Z_2^{32} \times Z_2^{48} \rightarrow Z_2^{32}$, we need to define some other functions. These are $P : Z_2^{32} \rightarrow Z_2^{32}$, a fixed permutation of the 32 bits, $E : Z_2^{32} \rightarrow Z_2^{48}$, the expansion phase, a linear function in which half the input bits are replicated, and $S : Z_2^{48} \rightarrow Z_2^{32}$, a non-linear function, consisting of 8 S-boxes, $S_i : Z_2^6 \rightarrow Z_2^4$ $(i = 1, \cdots, 8)$. f is defined as

$$f(R, K) = P\{S[E(R) \oplus K]\},$$

and a diagram of f is given in Figure 2. The expansion phase works by dividing the 32-bit input into 8 blocks of 4-bit inputs and expanding each block into 6 bits by adding the most significant bit of the block on the right as least significant bit and by adding the least significant bit of the block on the left as most significant bit . Each expanded block of 6 bits forms the input to an S-box. A schematic representation of E is given in Figure 3. Each S-box consists of 4 4-bit permutations of the middle 4 bits of the input, the particular 4-bit permutation being determined by the other two input bits. Thus S can be considered as a different permutation of each 4-bit block in which the two neighbouring bits to a 4-bit block determine the permutation from a list of four. In the original description of DES, the S-boxes are given as 4×16 tables so that each row is a permutation.

We can now describe the basic idea of this cryptanalysis. Suppose we observe input I and output O, then, since P is a bit permutation, and thus linear, we have

$$\begin{aligned} P^{-1}(I \oplus O) &= P^{-1}[\oplus_{i=1}^{8} f(R_i, K_i)] \\ &= \oplus_{i=1}^{8} P^{-1}[f(R_i, K_i)] \\ &= \oplus_{i=1}^{8} S[E(R_i) \oplus K_i], \end{aligned}$$

for inputs R_i and appropriate subkeys K_i. Thus, we can observe the joint distribution of the sum of the outputs of one or several S-boxes by observing the appropriate bit positions in the input/output pairs. For this cryptanalysis, we are concerned with the output of either an adjacent pair or triplet of S-boxes. In this case, we show in Appendix 1 that for a known plaintext attack, R_i, the 10-bit (pairs attack) or 14-bit (triplet attack) inputs to the relevant adjacent S-boxes, are approximately uniformly distributed and independent.

The 4-bit output of any one box is statistically uniform, because each row is a permutation. However, because of the expansion phase of DES (the E function), the inputs to neighbouring DES S-boxes are related by certain key bits. For example, on the first round, the XOR of the bit 5 input (counting from the left) to S_1 and the bit 1 input to S_2 is the same as the XOR of key bits 49 and 33 (from the key scheduling). Similarly, XOR of the bit 6 input to S_1 and the bit 2 input to S_2 is the same as the XOR of key bits 17 and 57. More generally, the input to a pair of neighbouring S-boxes is constrained by two bits of information about the key, which we call the "common" key bits. Furthermore, the distribution of the 8-bit output of a pair of S-boxes conditioned on these common key bits can be determined, and it is always non-uniform. In the next section, we show that the distribution of the output of a pair of adjacent S-boxes in fact depends only on the XOR of these two bits of key information. Notice that, under the assumption of independent inputs, the distribution of the XOR of n outputs of pairs of neighbouring S-boxes is the n-fold convolution of the distribution of the output of neighbouring S-boxes, and we show that the distribution of the XOR of outputs depends only the XOR (a linear combination) of all the bits of key information given above. For a large number of messages and ciphertexts, this gives us 16 empirical distributions for the XOR of pairs of S-boxes, and so observations of outputs of the form given above potentially give us probabilistic information about 16 linearly independent combinations of key bits (from the key-scheduling). Thus we could obtain 16 bits of key information. We show how this can be used to give a known plaintext attack on DES that is comparable in complexity to an exhaustive key search.

In Section 3, we extend this method by considering triplets of adjacent S-boxes. As before we obtain 16 empirical distributions for triplets of adjacent S-boxes. However, the common key bits to a pair of S-boxes affect two S-box triplets, for example the key bits "common" to S-boxes 1 and 2 affect the output of the S-box triplets 123 and 812. This dependence may make a DES-type cipher that is invulnerable to the attack based on pairs of S-boxes vulnerable to one based on triplets of S-boxes.

2 Pairs of DES S-Boxes

In this section we give a detailed account of the cryptanalysis as applied to a pair of adjacent S-boxes. Following the result of Appendix 1, we assume that the inputs to a given pair of S-boxes are uniformly and independently distributed. As we mentioned above, the output of such a pair of S-boxes is not necessarily uniform. For a given key, there are 2^{10} inputs and 2^8 outputs. Thus for a uniform distribution of inputs, each output in a uniform distribution of outputs would occur 4 times. Table 1 (at the end of the paper) is a table of the number of times each output of the pair S_1, S_2 occurred with the common key bits $(0, 0)$. Obviously, this distribution of outputs is non-uniform, as is the distribution of any other pair of adjacent S-boxes with any value for the common key bits, and it is this fact we exploit in the cryptanalysis.

Throughout this cryptanalysis, we adopt the following convention. For any n-bit binary number W, we let W_i denote the i^{th} bit from the left (or i^{th} most significant bit). Thus W_1 denotes the left-hand bit of W and W_n the right-hand bit. For a fixed key, let $I, J \in Z_2^6$ denote the inputs to a pair of neighbouring DES S-boxes S_p, S_q, and $X = S_p(I), Y = S_q(J) \in Z_2^4$ the respective outputs. Because of the expansion function of DES, the inputs to adjacent S-boxes are related by certain key bits. Thus, let us define $s = I_5 \oplus J_1$ and $t = I_6 \oplus J_2$, so we can see from Figure 3 that (s, t) are the key bits common to the neighbouring S-boxes. We can now define $S_{X,Y}(s, t)$ to be the number of times output (X, Y) occurs over all 2^{10} possible inputs

(I,J) for the given values of s and t. Thus we have

$$S_{X,Y}(s,t) \;=\; \#\left\{I,J \in Z_2^6 \,|\, I_5 \oplus J_1 = s, I_6 \oplus J_2 = t,\; S_p(I) = X, S_q(J) = Y\right\},$$

so Table 1 is a table of $S_{X,Y}(0,0)$ for S-boxes 1 and 2, and

$$\sum_{X,Y} S_{X,Y}(s,t) = 2^{10}\,.$$

We can define two more functions for the outputs of each S-box, namely

$$\begin{aligned} d_X(i,j) &= \#\{I \in Z_2^6 | I_5 = i, I_6 = j, S_p(I) = X\}, \\ e_Y(i,j) &= \#\{J \in Z_2^6 | J_1 = i, J_2 = j, S_q(J) = Y\}, \end{aligned}$$

so d_X and e_Y are the number of outputs over all possible inputs of the individual S-boxes given that certain inputs are fixed. Each row of a DES S-box is a permutation, so we have the following properties,

$$\sum_i d_X(i,j) = \sum_j e_Y(i,j) = 2 \text{ for all } X,Y \in Z_2^6.$$

We can give an expression for $S_{X,Y}$ in terms of the functions defined above

$$\begin{aligned} S_{X,Y}(s,t) &= \textstyle\sum_{i,j} d_X(i,t\oplus j)e_Y(s\oplus i,j) \\ &= d_X(0,t)e_Y(s,0) + d_X(1,t)e_Y(s',0) \\ &\qquad + d_X(0,t')e_Y(s,1) + d_X(1,t')e_Y(s',1) \\ &= d_X(0,t)e_Y(s,0) + (2 - d_X(0,t))e_Y(s',0) \\ &\qquad + d_X(0,t')(2 - e_Y(s,0)) + (2 - d_X(0,t'))(2 - e_Y(s',0)) \\ &= 4 + (d_X(0,t) - d_X(0,t'))(e_Y(s,0) - e_Y(s',0)) \\ &= 4 + (-1)^{s\oplus t}(d_X(0,0) - d_X(0,1))(e_Y(0,0) - e_Y(1,0)) \\ &= 4 + (-1)^{s\oplus t} d_X e_Y, \end{aligned}$$

where $d_X = (d_X(0,0) - d_X(0,1))$, $e_Y = (e_Y(0,0) - e_Y(1,0))$, and s' denotes $s \oplus 1$, t' denotes $t \oplus 1$. Note that $\sum_X d_X = \sum_Y e_Y = 0$.

We can now calculate an expression for the distribution of the output of the sum of two pairs of S-boxes. If we let $S2_{X,Y}(s_1,s_2,t_1,t_2)$ denote the number of times output (X,Y) occurs over all 2^{20} inputs, where the common key bits to the first box are (s_1,t_1) and the common key bits to the other box are (s_2, t_2), then

$$\begin{aligned} S2_{X,Y}(s,t) &= \textstyle\sum_{x,y} S_{x,y}(s_1,t_1) S_{(x,y)\oplus(X,Y)}(s_2,t_2) \\ &= \textstyle\sum_{x,y}(4+(-1)^{s_1 \oplus t_1} d_x e_y)(4+(-1)^{s_1 \oplus t_1} d_{x\oplus X} e_{y \oplus Y}) \\ &= 2^{12} + (-1)^{(s,t)} \textstyle\sum_x d_x d_{x\oplus X} \sum_y e_y e_{y\oplus Y} \\ &= 2^{12} + (-1)^{(s,t)} d2_X e2_Y, \end{aligned}$$

where $(-1)^{(s,t)}$ denotes $(-1)^{s_1 \oplus s_2 \oplus t_1 \oplus t_2}$ and so forth, and $d2_X = \sum_x d_x d_{x\oplus X}$, $e2_Y = \sum_y e_y e_{y\oplus Y}$. We can extend this result in the obvious way to obtain a result for the distribution of the output of the sum of n pairs of S-boxes, namely,

$$Sn_{X,Y}(s,t) = 2^{10n-8} + (-1)^{(s,t)} dn_X en_Y,$$

where dn_X and en_Y are easily calculable n-fold convolutions.

We can regard the 8-bit output as a number in the range $0, \cdots, 255$, and we have therefore just shown that, for a fixed key, the number of times output i, $(i = 0, \cdots, 255)$ occurs over all 2^{10n} inputs, depends on the value k of the XOR of certain key bits, where $k = (s,t)$, and conditioned on this can be expressed as

$$D_i(k) = 2^{10n-8} + (-1)^k E_i$$

where E_i is easily calculated, and $\sum_i E_i = 0$. Hence, the probability $p_i(k)$ of the output i conditioned on k, is given by

$$p_i(k) = 2^{-8} + 2^{-10n}(-1)^k E_i = d + (-1)^k d_i,$$

where $d_i = 2^{-10n} E_i$.

The problem is to decide whether $k = 0$ or $k = 1$. This is a standard statistical problem of deciding between two simple hypotheses and a fuller explanation of the following technique can be found in any standard statistics textbook, for example Silvey [6]. Let X denote the output of m inputs to the pair of neighbouring S-boxes. Suppose we have a realisation of X, $x = (x_1, \cdots, x_m)$, then let m_i be the number of times output i occurs, so $\sum_i m_i = m$ and we clearly have

$$P(X = x) = \prod_{i=0}^{255} p_i(k)^{m_i}.$$

For given data $X = x$, the above equation defines a function on the set of parameters $(k = 0, 1)$ and this function is known as the likelihood function of k corresponding to the data x. It is intuitively natural that the higher value of the likelihood given data x corresponds to the more likely value of k. Thus, by definition, the likelihood function of k corresponding to the data x is given by

$$L(x;k) \;=\; \prod_{i=0}^{255} p_i(k)^{m_i}.$$

We wish to test whether $k=0$ or $k=1$. Suppose we fix the probability of the error of deciding $k=1$ when $k=0$, then we need to design a test that given this error probability minimises the error of deciding $k=0$ when $k=1$, that is a *most powerful* test. The Neyman-Pearson Lemma [6] tells us that the most powerful test of $k=0$ against $k=1$ is one based on a likelihood ratio statistic. The likelihood ratio statistic, λ, is given by

$$\lambda \;=\; \frac{L(x;k=0)}{L(x;k=1)} \;=\; \prod_{i=0}^{255}\left(\frac{p_i(0)}{p_i(1)}\right)^{m_i}.$$

For symmetric error probabilities, we decide whether $k=0$ or $k=1$ according to whether $\lambda > 1$, that is whether $\log\lambda > 0$. Now,

$$\log\lambda \;=\; \sum_{i=0}^{255} m_i \log\left(\frac{p_i(0)}{p_i(1)}\right) \;=\; \sum_{i=0}^{255} m_i \log\left(\frac{d+d_i}{d-d_i}\right) \;=\; \sum_{i=0}^{255} m_i w_i,$$

$$\text{where } w_i = log\left(\frac{d+d_i}{d-d_i}\right)$$

However, if d_i is small compared to d, as is the case with DES with larger numbers of rounds,

$$w_i \;=\; \log\left(\frac{d+d_i}{d-d_i}\right) \;=\; \log\left(1+\frac{2d_i}{d-d_i}\right) \;\approx\; \frac{2d_i}{d-d_i} \;\approx\; \frac{2d_i}{d}.$$

Thus, we can use the sign of

$$I \;=\; \sum_{i=0}^{255} m_i d_i \;\approx\; \frac{d}{2}\log\lambda$$

as a statistic for deciding whether $k=0$ or $k=1$. Now,

$$\begin{aligned} E(I) &= \textstyle\sum_{i=0}^{255} d_i E(m_i) \\ &\approx \textstyle\sum_{i=0}^{255} d_i m(d+(-1)^k d_i) \\ &= m(-1)^k \textstyle\sum_{i=0}^{255} {d_i}^2 \;=\; (-1)^k mT, \end{aligned}$$

where $T = \sum_{i=0}^{255} d_i^2$, and we also have

$$Var(I) \approx \sum_{i=0}^{255} {d_i}^2 Var(m_i) \approx md \sum_{i=0}^{255} {d_i}^2 = mdT.$$

For large values of m, I is approximately normally distributed, so for a test of size 0.0228, we need to find m such that $|E(I)|$ is twice the standard deviation of I, so

$$m = \frac{4d}{T} = \frac{2^{-6}}{T}.$$

Now,

$$T = \sum_i {d_i}^2 = 2^{-20n} \sum_i {E_i}^2,$$

where $\sum_i E_i^2$ is easily calculable, so we can calculate T for a 16-round DES. The largest value of T occurs for S-boxes 7 and 8, with

$$T = 2^{-160} \times 1.32 \times 2^{97} = 1.32 \times 2^{-63},$$

and so $m = 1.51 \times 2^{56}$, which is larger than an exhaustive key search. A test of size 0.0228 gives a probability of 0.955 of estimating both bits of key information correctly. This may be a higher probability than is needed for a successful cryptanalysis. For example, if we perform a test of size 0.29, so we estimate both bits of key information with probability 0.504, then we require that $E(I)$ is 0.553 standard deviations of I, which happens when $m = 1.85 \times 2^{52}$. Heuristically speaking, in 1.85×2^{52} encryptions, we can, more often than not, determine two bits of key information. We then have to perform a reduced exhaustive key search on 54 bits which takes at most 2^{53} and on average 2^{52} encryptions. Thus, we can determine the DES key in at most $(1.85 \times 2^{52}) + 2^{53} = 1.92 \times 2^{53}$ encryptions more often than not.

DES has the following complementation property,

$$DES(M, K) = DES^c(M^c, K^c) \qquad \text{for all messages } M \text{ and keys } K,$$

where ^{c} denotes complementation, which enables the construction of an exhaustive key search of DES that has an expected running time of 2^{54} encryptions. Thus in some sense this method of cryptanalysis is marginally faster than an exhaustive key search. Of course, as a method of cryptanalysis, it is impractical since it requires a vast number of encrypted pairs whereas an exhaustive key search requires very few known plaintext/ciphertext pairs.

Table 2 (at the end of the paper) gives the number of encryptions for a test of size 0.028 for each S-box pair. These values are inversely proportional to the log-likelihood and so proportional to the complexity of the cryptanalysis for each S-box pair, the constant of proportionality depending on the precision with which we wish estimate the common key bits. It can be seen that a cryptanalysis based on S-boxes 7 and 8 is far simpler than any other pair of S-boxes. We can lessen the amount of data needed for a decision by using a sequential testing procedure [6]. Further details of this sequential procedure can be found in Appendix 2.

It is possible to reduce the number of plaintexts required still further. Recall that we are using equations of the form

$$\oplus_{i=1}^{7} f(R_{2i-1}, K_{2i}) \oplus f(R_{15}, K_{16}) = R_0 \oplus R_{16}.$$

In the known plaintext cryptanalysis given above, we use the known values of R_0 and R_{16}. However we also know the value of R_{15} and it is possible to use this. For a pair of S-boxes, we would now be attempting to estimate the relevant 12 bits of K_{16} as well as the original key bit. When the likelihood framework is applied to this attack, the output counts conditioned on different values of K_{16} are correlated. In a preliminary investigation with Henri Gilbert [4] of this technique, we provisionally estimate the complexity of such an attack as about 2^{52}.

3 Triplets of DES S-Boxes

The obvious extension of a cryptanalysis based on S-box pairs is a cryptanalysis based on S-box triplets, and following Appendix 1, we assume that the inputs to a given triplet of S-boxes triplets are uniformly and independently distributed. For a fixed key, let $I, J, K \in Z_2^6$ denote the inputs to three neighbouring DES S-boxes S_p, S_q, S_r, and $X, Y, Z \in Z_2^4$ the respective outputs. Because of the expansion function of DES, the inputs to adjacent S-boxes are related by certain key bits. Accordingly, let

$$\begin{aligned} S_{X,Y,Z}(s,t,u,v) &= \#\{I,J,K \in Z_2^6 | S_p(I) = X, S_q(J) = Y, S_r(K) = Z, \\ &\qquad (I_5, I_6, J_5, J_6) \oplus (J_1, J_2, K_1, K_2) = (s,t,u,v)\}, \end{aligned}$$

where s, t, u, v are completely determined by the key. $S_{X,Y,Z}(s,t,u,v)$ is the number of times output (X, Y, Z) occurs over all 2^{14} different inputs, so

$$\sum_{X,Y,Z} S_{X,Y,Z}(s,t,u,v) = 2^{14}\,.$$

In Appendix 3 it is shown that

$$\begin{array}{rl} S_{X,Y,Z}(s,t,u,v) = 4 & +2(-1)^{s\oplus t} D_X P_Y^0(v) + 2(-1)^{u\oplus v} F_Z R_Y^0(s) \\ & +(-1)^{s\oplus t\oplus u\oplus v} Q_Y D_X F_Z \\ & +(-1)^{s\oplus t\oplus v} P_Y D_X f_Z^0(u') + (-1)^{s\oplus u\oplus v} R_Y F_X d_X^0(t'), \end{array}$$

where the various functions are defined there. Equivalently we can write

$$\begin{array}{rcl} S_{X,Y,Z}(s,t,u,v) & = & 4 \quad +(-1)^{s\oplus t\oplus u\oplus v} Q_Y D_X F_Z \\ & & \quad\;\; +(-1)^{s\oplus t} D_X [P_Y(v) f_Z^0(u',1) + P_Y(v') f_Z^0(u',0)] \\ & & \quad\;\; +(-1)^{u\oplus v} D_X [R_Y(s) d_X^0(1,t') + R_Y(s') d_X^0(0,t')] \\ & = & 4 \quad +(-1)^{s\oplus t\oplus u\oplus v} Q_Y D_X F_Z \\ & & \quad\;\; +(-1)^{s\oplus t} D_X \alpha_{Y,Z}(u,v) + (-1)^{u\oplus v} F_Z \beta_{X,Y}(s,t), \end{array}$$

where

$$\begin{array}{rcl} \alpha_{Y,Z}(u,v) & = & [P_Y(v) f_Z^0(u',1) + P_Y(v') f_Z^0(u',0)] \\ \beta_{X,Y}(s,t) & = & [R_Y(s) d_X^0(1,t') + R_Y(s') d_X^0(0,t')]. \end{array}$$

In order to perform the cryptanalysis we now need to calculate the XOR of two outputs of three neighbouring S-boxes. We can denote by

$$S2_{X,Y,Z}(s,t,u,v)$$

the number of times output (X, Y, Z) occurs over all 2^{28} inputs in the XOR of the two triplets of S-boxes that give rise to

$$S_{X,Y,Z}(s_1,t_1,u_1,v_1) \quad \text{and} \quad S_{X,Y,Z}(s_2,t_2,u_2,v_2).$$

Thus $S2$ is a convolution, that is

$$\begin{array}{rcl} S2_{X,Y,Z}(s_1,s_2,t_1,t_2,u_1,u_2,v_1,v_2) & = & (S(s_1,t_1,u_1,v_1) * S(s_2,t_2,u_2,v_2))_{X,Y,Z} \\ & = & \sum_{x,y,z} S_{x,y,z}(s_1,t_1,u_1,v_1) S_{X\oplus x, Y\oplus y, Z\oplus z}(s_2,t_2,u_2,v_2). \end{array}$$

In order to calculate this convolution, note first that we can write

$$S_{X,Y,Z}(s,t,u,v) = 4 + T_{X,Y,Z}(s,t,u,v)\,,$$

where

$$\begin{aligned} T_{X,Y,Z}(s,t,u,v) = & \quad 2(-1)^{s\oplus t} D_X P_Y^0(v) + 2(-1)^{u\oplus v} F_Z R_Y^0(s) \\ & + \quad (-1)^{s\oplus t\oplus u\oplus v} Q_Y D_X F_Z \\ & + \quad (-1)^{s\oplus t\oplus v} P_Y D_X f_Z^0(u') + (-1)^{s\oplus u\oplus v} R_Y F_X d_X^0(t'), \end{aligned}$$

so $\sum_{X,Y,Z} T_{X,Y,Z}(s,t,u,v) = 0$. Thus,

$$S2_{X,Y,Z}(s,t,u,v) \;=\; 2^{16} + T2_{X,Y,Z}(s,t,u,v)\,,$$

where $\sum_{X,Y,Z} T2_{X,Y,Z} = 0$ and $T2$ denotes the similar convolution for T that $S2$ denotes for S. We can extend this result to an n-fold convolution, so

$$Sn_{X,Y,Z}(s,t,u,v) \;=\; 2^{14n-12} + Tn_{X,Y,Z}(s,t,u,v)\,,$$

and $\sum_{X,Y,Z} Tn_{X,Y,Z} = 0$. Thus, in order to calculate the convolutions Sn, we need only calculate Tn.

The distribution Tn is parameterised by 2^{4n} parameters, but there are less than 2^{4n} Tn distributions, because we show in Appendix 4 that we can express Tn in the following way,

$$Tn_{X,Y,Z}(s,t,u,v) \;=\; Tn_{X,Y,Z}(\oplus s, t\pi, u\pi, \oplus v)\,,$$

for any $\pi \in S_n$, so there are many identical distributions. This result for Tn means we can write the components of s and v in any order we please. We can now calculate the number of essentially different distributions Sn. Let $W_i = (t_i, u_i)$, so we can regard $W_i \in \{0,1,2,3\}$. The number of different distributions is then four times the number of different arrangements of $W = (W_1, \cdots, W_n)$. These arrangements for $n = 4$ and $n = 8$ are given in Tables 3 and 4 (at the end of the paper). Type 4 0 0 0 means that all of the W_i take one value, type 3 1 0 0 means that three of the W_i take one value and the other W_i takes a different value and so forth, where for example $W = (0,0,0,0), (1,1,1,1)$ are forms of the type 4 0 0 0. We therefore have 140 essentially different $S4$ distributions and 660 different $S8$ distributions. The efficient calculation of the distributions Tn and thus Sn depends on the fast calculation of discrete convolutions. We can do this efficiently by using Walsh transforms, a discrete form of Fourier transforms, see Beauchamp [1]. Details are given in Appendix 5.

We have thus shown the XOR of a number of S-box triplet outputs belongs to one of a manageable number of easily calculable distributions that depend on common key bits. The cryptanalytic decision problem is to decide from which of these distributions (or which subset) the XOR arose. This is a similar, but more difficult, problem to the one faced in a cryptanalysis based on an adjacent pair of S-boxes, where we had to choose between two distributions. Thus, consider a $2n$-round DES, and let

$$\Psi^i \;=\; (s^i, t^i, u^i, v^i) \;=\; (\oplus s^i, W^i, \oplus v^i)$$

denote the key bits concerned with the triplet of S-boxes centred on S_i, for the left half of the output. Note that $W^i = (t^i, u^i)$, where there is no loss of generality in assuming $W^i_1 \leq \cdots \leq W^i_n$, when $W^i_{\cdot}$ is regarded as a 2-bit number. We also trivially have

$$(s^i, t^i) = (u^{i-1}, v^{i-1}) \;, \quad (u^i, v^i) = (s^{i+1}, t^{i+1}) \;.$$

We have to find a statistical procedure that gives us candidates for the key, that is the most likely values in some sense for Ψ^i or W^i. We can do this using Bayes' Theorem, of which more details can be found in any standard statistics textbook [6]. Suppose we wish to estimate some parameter θ, which we have reason to believe comes from some distribution $p(\theta)$, a prior distribution. Suppose we now have some data x, then by using the likelihood function of θ corresponding to the data x, $l(x|\theta)$, we can then obtain a posterior distribution for θ, that is a distribution for θ given x by using Bayes' Theorem. Thus we have

$$p(\theta|x) \;\propto\; l(x|\theta)\, p(\theta) \;,$$

or equivalently, up to an additive constant.

$$\log[p(\theta|x)] \;=\; \log[l(x|\theta)] \,+\, \log[p(\theta)] \;=\; L(x|\theta) \,+\, \log[p(\theta)] \;.$$

We do have however a natural prior distribution for W^i, and thus a natural prior for the common key bits. Since we can calculate a likelihood, we can calculate a posterior distribution for the common key bits. Thus, it may be possible to obtain candidates for the true common key bits.

The initial distribution of W^i,

$$q_i(w) \;=\; P[W^i = w]$$

is given by the relevant entry in Tables 3 and 4 (at the end of the paper). For example, for a 4-round DES, $P[W^i = (0,0,0,0)] = 1/256$ and $P[W^i = (0,0,1,3)] = 12/256$. We can also define

$$f_i(c) = P[\oplus t^i = c], \qquad g_i(d) = P[\oplus u^i = d]$$

and thus obtain

$$p_i(c,w,d) \;=\; P[\Psi^i = (c,w,d)] \;=\; g_{i-1}(c)q_i(w)f_{i+1}(d) \;,$$

where initially $f_i(c) = g_i(d) = \frac{1}{2}$.

In order to calculate the likelihood function, note that we can represent (X,Y,Z) as a number in the range $0,\cdots,4095$, and so we have

$$Sn^i_j(\Psi^i) \;=\; 2^{14n-12} \;+\; Tn^i_j(\Psi^i) \qquad j = 0,\cdots,2^{12}-1.$$

The probability of any particular outcome j is D^i_j, where $D^i_j = 2^{-14n}Sn^i_j$. D^i_j is thus given by

$$D^i_j(\Psi^i) \;=\; d \;+\; d^i_j(\Psi^i),$$

where $d = 2^{-12}$ and $d^i_j = 2^{-14n}Tn_j$. Let X^i denote the output of m inputs to three neighbouring S-boxes. Suppose we have a realisation of X^i, $x^i = (x^i_1,\cdots,x^i_m)$, then let m^i_j be the number of times output j occurs, so $\sum m^i_j = m$, and we have

$$P(X^i = x^i) \;=\; \prod_{j=0}^{k} D^i_j(\Psi^i)^{m^i_j}.$$

Thus, the likelihood function of Ψ^i is given by

$$l(x^i;\Psi^i) \;=\; \prod_{j=0}^{k} D^i_j(\Psi^i)^{m^i_j},$$

and so the log-likelihood is given by, up to an additive constant,

$$\begin{aligned} L(x^i;\Psi^i) \;&=\; \textstyle\sum_{j=0}^{k} m^i_j \log[D^i_j(\Psi^i)] \;=\; \sum_{j=0}^{k} m^i_j \log[d + d^i_j(\Psi^i)] \\ &=\; \textstyle\sum_{j=0}^{k} m^i_j \log[1 + d^i_j(\Psi^i)/d] \\ &\approx\; \textstyle\frac{1}{d}\sum_{j=0}^{k} m_j d^i_j(\Psi^i) \;, \end{aligned}$$

since $d^i_j(\Psi^i) \ll d$. Hence let us define, for permissible θ,

$$I^i(\theta) \;=\; \sum_{j=0}^{k} m^i_j d^i_j(\theta) \;\approx\; dL(x^i;\Psi^i).$$

Having calculated the log-likelihood of the data for each permissible key, we can use Bayes' Theorem to calculate a posterior distribution for permissible keys. By repeated use of Bayes' Theorem (taking the old posterior as the new prior), we may find candidates for the true common key bits.

The application of Bayes' Theorem in effect alters the probability of our belief that a particular key is the true key. The amount we alter it by is given by realisations of the random variables $I^i(\theta)$. Suppose that the correct key is ψ^i, then we have

$$\begin{array}{ll} E[I^i(\psi^i)] & = \sum_{j=0}^{k} d^i_j(\psi^i)E[m^i_j] \\ & \approx \sum_{j=0}^{k} d^i_j(\psi^i)m[d + d^i_j(\psi^i)] \\ & = m\sum_{j=0}^{k}\left[d^i_j(\psi^i)\right]^2 = m\Phi^i(\psi^i), \end{array}$$

where $\Phi^i(\psi^i) = \sum_{j=0}^{k}\left[d^i_j(\psi^i)\right]^2$. However for incorrect keys $\theta \neq \psi^i$ we have

$$\begin{array}{lll} E[I^i(\theta)] & = & \sum_{j=0}^{k} d^i_j(\theta)E[m^i_j] \\ & \approx & \sum_{j=0}^{k} d^i_j(\theta)m[d + d^i_j(\psi^i)] \\ & = & m\sum_{j=0}^{k} d^i_j(\theta)d^i_j(\psi^i) \end{array}$$

However, for any value of θ, we have

$$Var[I^i(\theta)] \;\approx\; md\Phi^i(\theta)\ .$$

Hence, on average, the change in the value of the posterior distribution at θ is proportional to the correlation between $I^i(\theta)$ and $I^i(\psi^i)$. Thus, all we have to do is to choose m large enough, so that the value of $E[I^i(\psi^i)]$ is highly likely to be positive. Essentially we are then looking at the keys which are likely to give the largest value for $E[I^i(\theta)]$ as candidate keys. These keys are those for which the distribution $\{d^i_j(\theta)\}_j$ is highly correlated with the distribution of the true key $\{d^i_j(\Psi^i)\}_j$.

Table 5 (at the end of the paper) gives the number of encryptions for a test of size 0.028 for each S-box triplet if there are essentially two key classes. The values in Table 5 are inversely proportional to the log-likelihood, so these values are proportional to the complexity of the various attacks. The constant of proportionality depends on the precision of estimation of the true key class, and the correlations of the key classes. In particular, if there are essentially two key classes, these values can be directly compared with the values in Table 2. However, as we mentioned in the introduction, common key bits to a pair of S-boxes affect two S-box triplets, and for some choices of S-boxes it may be possible to use this fact together with Bayes' Theorem by allowing a posterior distribution of triplet key classes to alter the prior distribution of overlapping triplet key classes. An algorithm to do this is given in Appendix 6.

Clearly, the analysis of triplets of S-boxes is an extension of the analysis of pairs of S-boxes. Thus, any result on the complexity of an attack based on pairs of S-boxes is an upper bound for the complexity of an attack based on triplets of S-boxes. Hence, a cryptanalysis of DES based on triplets of S-boxes has at most the same complexity as an exhaustive key search. As noted above, an attack based on different triplets of S-boxes may be quicker since the distributions we observe are clearly correlated, and it may be possible to use this fact. Heuristically, after each iteration of the algorithm given above, the probability assigned to the true key bits is increased slightly and the probability assigned to the false key bits remain roughly unchanged. We then have to perform enough iterations so that the probability assigned to the true key becomes much larger than any other probability. Table 5 essentially gives the average value of $\Phi^i(.)$ for true keys each different i, that is each triplet of S-boxes. It can be seen from this list that triplets [678] and [781] give a much larger value for $\Phi(.)$ than any other triplets, so a cryptanalysis of DES based on triplets would cycle between $i = 7$ and $i = 8$. However, this large value occurs because in the analysis of pairs of S-boxes, the pair [78] is very much weaker than any other pair (Table 2). In fact, any pair of the 660 distributions for the triplets [678] has a correlation co-efficient of ± 1 to 3 decimal places (10 binary places), depending on whether the pair have the same XOR of the key bits "common" to S-boxes 7 and 8. This is, of course, the criterion used in the analysis of pairs of S-boxes for dividing the possible key inputs "common" to boxes 7 and 8 into two subsets. The same result is true of the triplet [781]. Thus, the S-box pair [78] dominate any triplet cryptanalysis and there is no saving in this cryptanalysis over a pair cryptanalysis.

4 Conclusions

We have exhibited a property of the DES algorithm that leads to a known plaintext attack, an S-box pair attack, that requires a vast amount of data. The natural extension of the S-box pair attack is the S-box triplet attack, but we have shown that there in no saving for an S-box triplet cryptanalysis of DES over an S-box pair cryptanalysis. However, a DES-type cipher with different S-boxes may be vulnerable to an S-box pair or an S-box triplet triplet cryptanalysis. One way to avoid the triplet attack would be to design the S-boxes so that one S-box pair is very much "weaker" than all the other S-box pairs. The security of the cryptosystem against these types of attack would rest solely with its security against a pairs attack. This is certainly the case with DES.

Acknowledgements

The authors would like to thank Henri Gilbert of CNET, France Telecom, for some interesting ideas about this cryptanalysis. The authors would also like to thank the referees for their helpful comments about this paper.

References

[1] K.G. Beauchamp. *Walsh Transforms and their Applications.* Academic Press, 1975.

[2] H. Beker and F.Piper. *Cipher Systems.* Northwood Books, 1982.

[3] E. Biham and A. Shamir. Differential Cryptanalysis of the full 16-round DES. Presented at CRYPTO 92, 1992.

[4] H. Gilbert. A Note on Davies and Murphy's Cryptanalysis of DES. Personal Communication, 1992.

[5] National Bureau of Standards. Data Encryption Standard. *U.S. Department of Commerce*, FIPS pub. 46, 1977.

[6] S.D. Silvey. *Statistical Inference.* Chapman and Hall, 1975.

Table 1

Joint Distribution of the Outputs of S_1, S_2

S_2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S_1																
0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
1	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
2	2	2	4	6	4	4	6	4	6	4	0	4	4	2	6	6
3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
6	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
7	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
8	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
9	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
A	6	6	4	2	4	4	2	4	2	4	8	4	4	6	2	2
B	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
C	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
D	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
E	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
F	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Table 2

No. of Encryptions for a Pairs Test of Size 0.028

S-Boxes	No. of Encryptions
12	1.94×2^{65}
23	1.24×2^{69}
34	1.56×2^{85}
45	1.55×2^{70}
56	1.54×2^{71}
67	1.96×2^{65}
78	1.51×2^{56}
81	1.21×2^{77}

Table 3

S4 : W Distribution

	W Type	*No. of Forms*	*No. of each Form*	*No. of Type*
(1)	4000	4	1	4
(2)	3100	12	4	48
(3)	2200	6	6	36
(4)	2110	12	12	144
(5)	1111	1	24	24
Total		35		256

Table 4

S8 : W Distribution

	W Type	*No. of Forms*	*No. of each Form*	*No. of Type*
(1)	8000	4	1	4
(2)	7100	12	8	96
(3)	6200	12	28	336
(4)	6110	12	56	672
(5)	5300	12	56	672
(6)	5210	24	168	4032
(7)	5111	4	336	1344
(8)	4400	6	70	420
(9)	4310	24	280	6720
(10)	4220	12	420	5040
(11)	4211	12	840	10080
(12)	3320	12	560	6720
(13)	3311	6	1120	6720
(14)	3221	12	1680	20160
(15)	2222	1	2520	2520
Total		165		65536

Table 5

No. of Encryptions for a Triplets Test of Size 0.028

S-Boxes	No. of Encryptions
812	1.94×2^{65}
123	1.77×2^{65}
234	1.24×2^{69}
345	1.55×2^{70}
456	1.03×2^{70}
567	1.92×2^{65}
678	1.51×2^{56}
781	1.51×2^{56}

Appendix 1
S-Box Inputs are Approximately Uniform and Independent

In the pairs and triplets cryptanalysis we observe the 8-fold XOR of the outputs of two or three neighbouring S-boxes respectively. In this appendix, we show that for the pairs cryptanalysis, under the assumption of a known-plaintext attack, the inputs to the neighbouring S-boxes are uniform and approximately independent. A similar argument holds for the triplets attack, though the approximation is less accurate. We denote the function $f(., K_i) : Z_2^{32} \to Z_2^{32}$ by f_i, where K_i is the i^{th} round subkey. Under the assumption of a known-plaintext attack, the left and right halves of the plaintext are independent uniformly distributed 32-bit numbers, that is for plaintext (Y, X), $X, Y \sim Uni(Z_2^{32})$ independently.

We first note that after any number of rounds, the left and right registers are independent uniformly distributed 32-bit numbers, since the joint entropy of the left and right registers is 64 bits. We are concerned with the inputs to the f-function in alternate rounds. Suppose we have (Y, X) as register contents before the i^{th} round, where we have shown that X and Y are independent randomly distributed 32-bit numbers. Before the $(i+2)^{th}$ round we have register contents

$$(Y \oplus f_i(X), X \oplus f_{i+1}(Y \oplus f_i(X)))\,.$$

Let Z_D denote the restriction of the 32-bit number Z to the ten input bits to a pair of neighbouring S-boxes, and k and l be the twelve bits of the i^{th} and $(i+2)^{th}$ round subkeys K_i and K_{i+2} that form the input to the pair of neighbouring S-boxes on the i^{th} and $(i+2)^{th}$ round respectively. Then the input bits to the pair of S-boxes on round i are $E_D(X_D) \oplus k$ and on round $(i+2)$ are

$$E_D(X_D) \oplus E_D(f_{i+1}(Y \oplus f_i(X))_D) \oplus l,$$

where $E_D : Z^{10} \to Z^{12}$ is the restriction of the expansion phase E to the inputs to two boxes and E_D^{-1} its inverse where this is well-defined. The conditional probability of the input on round $(i+2)$ given the input on round i is given, for any a and b, by

$$\begin{aligned} P\ &[\ (E_D(X_D) \oplus E_D(f_{i+1}(Y \oplus f_i(X))_D) \oplus l = b) \mid (E_D(X_D) \oplus k) = a)\ ] \\ &= P[f_{i+1}(W)_D = E_D^{-1}(c)], \end{aligned}$$

where $W = Y \oplus f_i(X|X_D)$ is a uniformly distributed 32-bit number and $E^{-1}(c) = E^{-1}(a \oplus b \oplus k \oplus l)$ is a fixed but unknown 10-bit number. Thus the conditional probability is given by the distribution of certain output bits of the DES f-function under $(i+1)^{th}$ round subkey K_{i+1}. In fact, the distribution only depends on the "overlapping" key bits. It is of course intractable to calculate the distribution of $f_{i+1}(W)_D$ since it would require 2^{32} evaluations of the function f_{i+1} for each relevant subkey, but we can calculate an approximation to it by simulation. In a number of simulations, each based on a million random values for W and different overlapping key bits, we have found that that this distribution is approximately uniform, with the most non-uniform distribution corresponding to the inputs to S-boxes 3 and 4. In this case, the conditional probability of a particular input is approximately $2^{-10} \pm 2^{-12}$ (depending on the XOR of the outer input bits), though for other S-boxes it is much nearer 2^{-10}. We can therefore calculate the distribution of the output of a pair of S-boxes given the inputs to (and hence the outputs of) the S-boxes two rounds earlier. This corresponds to perturbing the values of $S_{X,Y}$, as given for example in Table 1, by a small but unknown amount, with most values of $S_{X,Y}$ for most S-box pairs almost exactly correct.

In order to perform the cryptanalysis, we have to calculate $Sn_{X,Y}$, an n-fold convolution of $S_{X,Y}$ with itself. If we allow for the conditional distributions above, we have to calculate the n-fold convolution of small unknown perturbations of $S_{X,Y}$ with itself and this is very nearly $Sn_{X,Y}$. Thus it is a valid assumption that the S-box inputs in different rounds are uniformly and independently distributed.

Appendix 2
Sequential Procedure for S-box Pairs

The idea of a sequential procedure is to calculate the log-likelihood ratio, $\log \lambda$, sequentially for every output, and stop if it is too large or too small. A fuller explanation of sequential testing can be found in Silvey [6]. We can write I_l, the log-likelihood ratio after l outputs, as

$$I_l = \sum_{j=0}^{m} d_{x_j}$$

To construct a sequential test of size approximately 0.0228, we have to calculate I_l according to the following stopping rule

$$\begin{array}{cc} I_l > a & \mathit{Accept}\ K = 0 \\ a \le I_l \le -a & \mathit{Continue} \\ I_l < -a & \mathit{Accept}\ K = 1, \end{array}$$

where $a = -2^{-9} \log(0.0228/0.9772) = 0.00734$. The expected value of l, that is how many outputs are needed before a decision can be reached, is given by

$$\begin{aligned} E(l) &= \frac{a^2}{E({d_{x_j}}^2)} = \frac{2^{-18}\log(0.9772/0.0228)^2}{2^{-8}T^2} \\ &= [\log(42.85)]^2 2^{-4}\frac{2^{-6}}{T} = 0.882\frac{2^{-6}}{T}. \end{aligned}$$

Thus, a sequential approach on average needs 88% of the data of the non-sequential approach given above to reach a decision about the key in a test of size 0.0228. For the S-box pair [78], this means (1.33×2^{56}) encryptions. Thus we can guess 2 bits of key information with probability 0.95 in this time. We can of course perform a sequential tests of a larger size than this, but as the size of the test increases, the calculation of the stopping rule becomes more complicated.

Appendix 3
Derivation of $S_{X,Y,Z}(s,t,u,v)$

Recall that

$$\begin{array}{rcl} S_{X,Y,Z}(s,t,u,v) & = & \#\{I,J,K \in Z_2^6 | S_p(I)=X, S_q(J)=Y, S_r(K)=Z, \\ & & (I_5,I_6,J_5,J_6) \oplus (J_1,J_2,K_1,K_2) = (s,t,u,v)\}, \end{array}$$

where s,t,u,v are completely determined by the key. $S_{X,Y,Z}(s,t,u,v)$ is the number of times output (X,Y,Z) occurs over all 2^{14} different inputs, so

$$\sum_{X,Y,Z} S_{X,Y,Z}(s,t,u,v) = 2^{14}\,.$$

We can define three other functions, related to S, for the outputs of each S-box,

$$\begin{array}{rcl} d_X(i,j) & = & \#\{I \in Z_2^6 | S_p(I) = X, I_5 = i, I_6 = j\}, \\ e_Y(i,j,k,l) & = & \#\{J \in Z_2^6 | S_q(J) = 1, J_1 = i, J_2 = j, J_5 = k, J_6 = l\}, \\ f_Z(i,j) & = & \#\{Z | K_2 = k, K_2 = l\}. \end{array}$$

Since each row of a DES S-box is a permutation, we have the following properties,

$$\sum_i d_X(i,j) = 2, \qquad \sum_{j,k} e_Y(i,j,k,l) = 1, \qquad \sum_l f_Z(k,l) = 2,$$

and hence we can define

$$d_X^0(j) = d_X(0,j) = 2 - d_X(1,j)\,, \quad f_z^0(k) = f_Z(k,0) = 2 - f_Z(k,1)\,.$$

We can give an expression for $S_{X,Y,Z}$ in terms of the functions defined above

$$\begin{array}{rcl} S_{X,Y,Z}(s,t,u,v) & = & \displaystyle\sum_{i,j,k,l} d_X(s\oplus i, t\oplus j) e_Y(i,j,k,l) f_Y(u\oplus k, v\oplus l) \\ & = & \displaystyle\sum_{j,k} \left(\begin{array}{rc} & d_X(0,t\oplus j) e_Y(s,j,k,v) f_Z(u\oplus k,0) \\ + & d_X(0,t\oplus j) e_Y(s,j,k,v') f_Z(u\oplus k,1) \\ + & d_X(1,t\oplus j) e_Y(s',j,k,v) f_Z(u\oplus k,0) \\ + & d_X(1,t\oplus j) e_Y(s',j,k,v') f_Z(u\oplus k,1) \end{array} \right), \end{array}$$

where s' denotes $s \oplus 1$ and t' denotes $t \oplus 1$. Thus,

$$S_{X,Y,Z}(s,t,u,v) = \sum_{j,k} \left(\begin{array}{cc} & d_X^0(t \oplus j) e_Y(s,j,k,v) f_Z^0(u \oplus k) \\ + & d_X^0(t \oplus j) e_Y(s,j,k,v')[2 - f_Z^0(u \oplus k)] \\ + & [2 - d_X^0(t \oplus j)] e_Y(s',j,k,v) f_Z^0(u \oplus k) \\ + & [2 - d_X^0(t \oplus j)] e_Y(s',j,k,v')[2 - f_Z^0(u \oplus k)] \end{array} \right).$$

Suppose we define

$$\begin{aligned} p_Y(j,v) &= e_Y(0,j,0,v') - e_Y(1,j,0,v') + e_Y(0,j,1,v') - e_Y(1,j,1,v') \\ r_Y(s,k) &= e_Y(s',0,k,0) - e_Y(s',0,k,1) + e_Y(s',1,k,0) - e_Y(s',1,k,1) \\ q_Y(j,k) &= e_Y(0,j,k,0) - e_Y(0,j,0,1) - e_Y(1,j,k,0) + e_Y(1,j,k,1), \end{aligned}$$

so $p_Y(0,v) = -p_Y(1,v)$ and $r(s,0) = -r(s,1)$. We then have

$$\begin{aligned} (-1)^s p_Y(j,v) &= e_Y(s,j,0,v') - e_Y(s',j,0,v') \\ &\quad + e_Y(s,j,1,v') - e_Y(s',j,1,v') \end{aligned}$$

$$\begin{aligned} (-1)^v r_Y(s,k) &= e_Y(s',0,k,v) - e_Y(s',0,k,v') \\ &\quad + e_Y(s',1,k,v) - e_Y(s',1,k,v') \end{aligned}$$

$$\begin{aligned} (-1)^{s \oplus v} q_Y(j,k) &= e_Y(s,j,k,v) - e_Y(s,j,k,v') \\ &\quad - e_Y(s',j,k,v) + e_Y(s',j,k,v). \end{aligned}$$

Thus,

$$\begin{aligned} S_{X,Y,Z}(s,t,u,v) = 4 \quad & +2(-1)^s \textstyle\sum_j d_X^0(t \oplus j) p_Y(j,v) \\ & +2(-1)^v \textstyle\sum_j f_Z^0(u \oplus k) r_Y(s,k) \\ & +(-1)^{s \oplus v} \textstyle\sum_{j,k} d_X^0(t \oplus j) f_Z^0(u \oplus k) q_Y(j,k), \end{aligned}$$

since $\sum_{j,k} e_Y(s',j,k,v') = 1$. Now,

$$\begin{aligned} \textstyle\sum_j d_X^0(t \oplus j) p_Y(j,v) &= d_X^0(0) p_Y(t,v) + d_X^0(1) p_Y(t',v) \\ &= [d_X^0(0) - d_X^0(1)] p_Y(t,v) \\ &= (-1)^t [d_X^0(0) - d_X^0(1)] p_Y(0,v) \\ &= (-1)^t D_X^0 P_Y^0(v), \end{aligned}$$

where

$$D_X^0 = d_X^0(0) - d_X^0(1)\,, \qquad P_Y^0(v) = p_Y(0,v)\,.$$

Similarly,

$$\sum_k f_Z^0(u \oplus k) r_Y(s,k) \ = \ (-1)^u F_Z^0 R_Y^0(s)\,,$$

where

$$F_Z^0 = f_Z^0(0) - f_Z^0(1)\,, \qquad R_Y^0(s) = r_Y(0,s)\,.$$

Now,

$$\begin{aligned} P_Y &= P_Y^0(1) - P_Y^0(0) = q_Y(0,0) + q_Y(0,1) = q_Y(1,1) + q_Y(1,0) \\ R_Y &= R_Y^0(1) - R_Y^0(0) = q_Y(0,0) + q_Y(1,0) = q_Y(1,1) + q_Y(0,1)\,, \end{aligned}$$

and so, if we let $Q_Y = q_Y(0,0)$, we have

$$\begin{aligned} \textstyle\sum_{j,k} \quad & d_X^0(t \oplus j) f_Z^0(u \oplus k) \ q_Y(j,k) \\ = \quad & \ \ d_X^0(t) f_Z^0(u) q_Y(0,0) \ + \ d_X^0(t) f_Z^0(u') q_Y(0,1) \\ & + d_X^0(t') f_Z^0(u) q_Y 1,0) \ + \ d_X^0(t') f_Z^0(u') q_Y(1,1) \\ = \quad & \ \ d_X^0(t) f_Z^0(u) Q_Y \ + \ d_X^0(t) f_Z^0(u')[P_Y - Q_Y] \\ & + d_X^0(t') f_Z^0(u)[R_Y - Q_Y] \ + \ d_X^0(t') f_Z^0(u')[Q_Y - P_Y - R_Y] \\ = \quad & \ \ Q_Y[d_X^0(t) - d_X^0(t')][f_Z^0(u) - f_Z^0(u')] \\ & + P_Y[d_X^0(t) - d_X^0(t')] f_Z^0(u') \ + \ R_Y[f_Z^0(u) - f_Z^0(u')] d_X^0(t') \\ = \quad & (-1)^{t \oplus u} Q_Y D_X F_Z \ + \ (-1)^t P_Y D_X f_Z^0(u') \ + \ (-1)^u R_Y F_Z d_X^0(t'). \end{aligned}$$

We are now able to give an expression, conditioned on certain key bits, for the number of inputs to three adjacent DES S-boxes that give a particular output value. Thus we have

$$\begin{aligned} S_{X,Y,Z}(s,t,u,v) = 4 \quad & +2(-1)^{s \oplus t} D_X P_Y^0(v) + 2(-1)^{u \oplus v} F_Z R_Y^0(s) \\ & +(-1)^{s \oplus t \oplus u \oplus v} Q_Y D_X F_Z \\ & +(-1)^{s \oplus t \oplus v} P_Y D_X f_Z^0(u') + (-1)^{s \oplus u \oplus v} R_Y F_X d_X^0(t'), \end{aligned}$$

or equivalently,

$$\begin{aligned} S_{X,Y,Z}(s,t,u,v) = 4 \quad & +(-1)^{s \oplus t \oplus u \oplus v} Q_Y D_X F_Z \\ & +(-1)^{s \oplus t} D_X [P_Y(v) f_Z^0(u',1) + P_Y(v') f_Z^0(u',0)] \\ & +(-1)^{u \oplus v} D_X [R_Y(s) d_X^0(1,t') + R_Y(s') d_X^0(0,t')] \\ = 4 \quad & +(-1)^{s \oplus t \oplus u \oplus v} Q_Y D_X F_Z \\ & +(-1)^{s \oplus t} D_X \alpha_{Y,Z}(u,v) + (-1)^{u \oplus v} F_Z \beta_{X,Y}(s,t), \end{aligned}$$

where

$$\begin{aligned}\alpha_{Y,Z}(u,v) &= [P_Y(v)f_Z^0(u',1)+P_Y(v')f_Z^0(u',0)]\\ \beta_{X,Y}(s,t) &= [R_Y(s)d_X^0(1,t')+R_Y(s')d_X^0(0,t')].\end{aligned}$$

Appendix 4
Derivation of $Tn_{X,Y,Z}(s,t,u,v)$

In order to simplify the calculation of $T2$, we introduce some notation for the various convolutions :

$$\begin{gathered}
D2_X = (D * D)_X, \quad F2_Z = (F * F)_Z, \quad Q2_Y = (Q * Q)_Y, \\
PQ_Y = (P * Q)_Y, \quad PR_Y = (P * R)_Y, \quad QR_Y = (Q * R)_Y, \\
d2^{00}_X(t_1, t_2) = (d^0(t_1) * d^0(t_1))_X, \quad f2^{00}_Z(u_1, u_2) = (f^0(u_1) * f^0(u_1))_Z, \\
P2^{00}_Y(v_1, v_2) = (P^0(v_1) * P^0(v_2))_Y, \quad R2^{00}_Y(s_1, s_2) = (R^0(s_1) * R^0(s_2))_Y, \\
P2^0_Y(v) = (P * P^0(v))_Y, \quad R2^0_Y(s) = (R * R^0(s))_Y, \\
Dd^0_X(t) = (D * d^0(t))_X, \quad Ff^0_Z(u) = (F * f^0(u))_Z.
\end{gathered}$$

Recall that $\sum_X D_X = \sum_Z F_Z = 0$, so if we denote an expression of the form $(-1)^{s_1 \oplus s_2 \oplus t_1 \oplus t_2}$ by $(-1)^{(s,t)}$ and so forth, then we the following expression for $T2$,

$$\begin{aligned}
T2_{X,Y,Z}(s,t,u,v) \; = & \\
& (-1)^{(s,t,u,v)} Q2_Y D2_X F2_Z \\
& +64(-1)^{(s,t)} D2_X P2^{00}_Y(v_1, v_2) + 64(-1)^{(u,v)} F2_Z R2^{00}_Y(s_1, s_2) \\
& +(-1)^{(s,t,v)} P2_Y D2_X f2^{00}_Z(u'_1, u'_2) + (-1)^{(s,u,v)} R2_Y F2_X d2^{00}_Z(t'_1, t'_2) \\
& +32(-1)^{(s,t)} D2_X [(-1)^{v_2} P2^0_Y(v_1) + (-1)^{v_1} P2^0_Y(v_2)] \\
& +32(-1)^{(u,v)} F2_Z [(-1)^{s_2} P2^0_Y(s_1) + (-1)^{s_1} P2^0_Y(s_2)] \\
& +(-1)^{(s,t,u)} D2_X PQ_Y [(-1)^{u_2} Ff^0_Z(u'_1) + (-1)^{u_1} Ff^0_Z(u'_2)] \\
& +(-1)^{(s,u,v)} QR_Y F2_Z [(-1)^{t_2} Dd^0_X(t'_1) + (-1)^{t_1} Dd^0_X(t'_2)] \\
& +(-1)^{(s,v)} PR_Y [(-1)^{t_1 \oplus u_2} Dd^0_X(t'_2) Ff^0_Z(u'_1) + (-1)^{t_2 \oplus u_1} Dd^0_X(t'_1) Ff^0_Z(u'_2)].
\end{aligned}$$

However,

$$\begin{aligned}
& 2.P2^{00}_Y(v_1, v_2) + (-1)^{v_2} P2^0_Y(v_1) + (-1)^{v_1} P2^0_Y(v_2) \\
& \quad = 2[P^0(v_1) * P^0(v_2)]_Y \\
& \qquad + [P^0(v_1) * (P^0(v'_2) - P^0(v_2))]_Y + [P^0(v_2) * (P^0(v'_1) - P^0(v_1))]_Y \\
& \quad = [P^0(v_1) * P^0(v'_2)]_Y + [P^0(v_2) * P^0(v'_1)]_Y \\
& \quad = P2^{00}_Y(v_1, v'_2) + P2^{00}_Y(v'_1, v_2),
\end{aligned}$$

and similarly,

$$2.R2^{00}_Y(s_1, s_2) + (-1)^{s_2} R2^0_Y(s_1) + (-1)^{s_1} R2^0_Y(s_2) = R2^{00}_Y(s_1, s'_2) + R2^{00}_Y(s'_1, s_2).$$

Thus

$$\begin{gathered}
T2_{X,Y,Z}(s,t,u,v) \;=\; (-1)^{(s,t,u,v)} Q2_Y D2_X F2_Z \\
+32(-1)^{(s,t)} D2_X [P2^{00}_Y(v_1, v'_2) + P2^{00}_Y(v'_1, v_2)] \\
+32(-1)^{(u,v)} F2_Z [R2^{00}_Y(s_1, s'_2) + R2^{00}_Y(s'_1, s_2)] \\
+(-1)^{(s,t,v)} P2_Y D2_X f2^{00}_Z(u'_1, u'_2) + (-1)^{(s,u,v)} R2_Y F2_X d2^{00}_Z(t'_1, t'_2) \\
+(-1)^{(s,t,u)} D2_X PQ_Y [(-1)^{u_2} F f^0_Z(u'_1) + (-1)^{u_1} F f^0_Z(u'_2)] \\
+(-1)^{(s,u,v)} QR_Y F2_Z [(-1)^{t_2} D d^0_X(t'_1) + (-1)^{t_1} D d^0_X(t'_2)] \\
+(-1)^{(s,v)} PR_Y [(-1)^{t_1 \oplus u_2} D d^0_X(t'_2) F f^0_Z(u'_1) + (-1)^{t_2 \oplus u_1} D d^0_X(t'_1) F f^0_Z(u'_2)].
\end{gathered}$$

It can now be easily seen that, for any $c = 0, 1$,

$$\begin{aligned}
T2_{X,Y,Z}(s_1, s_1 \oplus c, t, u, v) &= T2_{X,Y,Z}(s'_1, s'_1 \oplus c, t, u, v)\ , \\
T2_{X,Y,Z}(s, t, u, v_1, v_1 \oplus c) &= T2_{X,Y,Z}(s, t, u, v'_1, v'_1 \oplus c)\ .
\end{aligned}$$

We can therefore write $T2$ as a function with fewer than 8 arguments as

$$T2_{X,Y,Z}(s,t,u,v) \;=\; T2_{X,Y,Z}(\oplus s, t, u, \oplus v)\ ,$$

where $\oplus s$ denotes the XOR of the elements of s. It is also clear from the form of $T2$ given above, and intuitively obvious, that $T2$ is invariant if (t_1, u_1) and (t_2, u_2) are swapped, so for any permutation $\pi \in S_2$, we can express $T2$ in the following way,

$$T2_{X,Y,Z}(s,t,u,v) \;=\; T2_{X,Y,Z}(\oplus s, t\pi, u\pi, \oplus v)\ .$$

As with the result for the distribution of S at the end of the previous appendix, the result for $T2$ was verified by direct computer calculation. This result for $T2$ essentially means we can write the pairs (s_i, s_j) and (v_k, v_l) in any order we please. Therefore, the general n-fold convolution can be written, for some $\pi \in S_n$, as

$$Tn_{X,Y,Z}(s,t,u,v) \;=\; Tn_{X,Y,Z}(\oplus s, t\pi, u\pi, \oplus v)\ .$$

Appendix 5
Fast Calculation of Tn by Walsh Transform

The efficient calculation of the distributions Tn and thus Sn depends on the fast calculation of discrete convolutions, which can be done by using Walsh Transforms, a discrete form of Fourier transforms, see Beauchamp [1]. Suppose $f, g\colon Z_2^n \to R$, then the Walsh transform of f, $\overline{f}$, is given by

$$\overline{f}(x) \;=\; \sum_{y \in Z_2^n} (-1)^{x \cdot y} f(y)\;.$$

The usual inversion and convolution theorems apply, so

$$\begin{aligned} \overline{\overline{f}}(x) &= 2^n f(x), \\ \overline{(f * g)}(x) &= \overline{f}(x)\overline{g}(x). \end{aligned}$$

We thus have a method of calculating convolutions by taking the Walsh transforms, multiplying and inverting. We can however calculate Walsh transforms extremely quickly by a method similar to the fast Fourier transform. For any $z \in Z_2^n$, let $z' \in Z_2^{n-1}$ denote the first $(n-1)$-bits of z. Thus $z' = (z_1, \cdots, z_{n-1})$. We can now define two functions $h_0, h_1\colon Z_2^{n-1} \to R$ by

$$h_0(z') = f[(z', 0)], \qquad h_1(z') = f[(z', 1)]$$

Let $x, w \in Z_2^n$, then

$$\begin{aligned} \overline{f}(x) &= \sum_{w \in Z_2^n} (-1)^{x.w} f(w) \\ &= \sum_{w' \in Z_2^{n-1}} (-1)^{x'.w'} f[(w', 0)] \;+\; \sum_{w' \in Z_2^{n-1}} (-1)^{x'.w'} (-1)^{x_n} f[(w', 1)] \\ &= \sum_{w' \in Z_2^{n-1}} (-1)^{x'.w'} h_0(w') \;+\; (-1)^{x_n} \sum_{w' \in Z_2^{n-1}} (-1)^{x'.w'} h_1(w') \\ &= \overline{h_0}(x') \;+\; (-1)^{x_n} \overline{h_1}(x'), \end{aligned}$$

We have thus reduced the calculation of two n-fold transforms to two $(n-1)$-fold transforms, and we can of course calculate both of these in terms of $(n-2)$-fold transforms and so on.

Appendix 6
An Algorithm for Overlapping Triplets Cryptanalysis

Pre-Calculation :
Calculate the values $I^i(\theta)$ for all permissible values of θ ;
Calculate the initial distribution of W^i, q^i, by the method given above.

Calculation (cyclically for some ordering of i) :

1.) The prior distribution p_i by

$$p_i(c,w,d) \;=\; P[\Psi^i = (c,w,d)] \;=\; g_{i-1}(c)q_i(w)f_{i+1}(d) \;;$$

2.) The log posterior distribution, $\log p_i$, given up to a constant by

$$\log[p_i(c,w,d)|x^i] \;=\; I^i(c,w,d) \;+\; \log[p_i(c,w,d)] \;;$$

3.) The renormalised posterior distribution, p_i ;

4.) The posterior distribution, q_i, given by

$$q_i(w|x^i) \;=\; \sum_{c,d} p_i[(c,w,d)|x^i] \;;$$

5.) The posterior distributions, f_i and g_i, given by

$$\begin{aligned} f_i(c|x^i) &= \sum_{t^i=c} q_i[(t^i,u^j)|x^i] \\ g_i(d|x^i) &= \sum_{u^i=d} q_i[(t^i,u^j)|x^i] \;; \end{aligned}$$

6.) The posterior distributions, g_{i-1} and f_{i+1}, given by

$$\begin{aligned} g_{i-1}(d|x^i) &= \sum_{c,w} p_i[(c,w,d)|x^i] \\ f_{i+1}(c|x^i) &= \sum_{w,d} p_i[(c,w,d)|x^i] \;; \end{aligned}$$

7.) The posterior distributions, q_{i-1} and q_{i+1}, given by

$$\begin{aligned} q_{i-1}[(t,u)|x^i] &= \frac{g_{i-1}[\oplus t|x^i]}{g_{i-1}(\oplus t)}\, q_{i-1}[(t,u)] \\ q_{i+1}[(t,u)|x^i] &= \frac{f_{i+1}[\oplus u|x^i]}{f_{i+1}(\oplus u)}\, q_{i+1}[(t,u)] \;. \end{aligned}$$

We continue with this algorithm until we have sufficient information about some of the key bits, that is f_i and g_i take values sufficiently near to 0 and 1. We can then repeat the process with the right half of the output. The critical stage is step 2. At this stage, we are able to alter the probability of a particular key occurring. The amount we alter it by is given by realisations of the random variables $I^i(\theta)$, given by Table 5.

Figure 1 : Enciphering Computation

Figure 2 : Schematic Diagram of $f(R, K)$

Figure 3 : The Expansion Box E