The Data Encryption Standard (DES) and its strength against attacks

by D. Coppersmith

The Data Encryption Standard (DES) was developed by an IBM team around 1974 and adopted as a national standard in 1977. Since that time, many cryptanalysts have attempted to find shortcuts for breaking the system. In this paper, we examine one such attempt, the method of differential cryptanalysis, published by Biham and Shamir. We show some of the safeguards against differential cryptanalysis that were built into the system from the beginning, with the result that more than 10^{15} bytes of chosen plaintext are required for this attack to succeed.

Introduction

Cryptography has long been in use by governments, particularly in the realms of military and diplomatic communication. It is hard to imagine military communication without cryptography; cryptanalysis, or secretly deciphering the opponent's messages, is perhaps of even greater value. Much has been written about cryptography in the military; see reference [1] for example.

During the early 1970s, it became apparent that the commercial sector also has a legitimate need for cryptography. Corporate secrets must be transmitted between distant sites, without the possibility of eavesdropping by industrial spies. Personal data on databases need to be protected against espionage and alteration.

A familiar example is the communication between an automatic teller machine (ATM) and a central computer. The user inserts a magnetic card and types a few numbers. The ATM sends messages to the computer. The computer checks the account balance and returns a message authorizing the ATM to dispense funds. Obviously, if these messages are unprotected, a thief can tap the wires, find the message authorizing the dispensing of funds, and send multiple copies of that message to the ATM, thereby "cleaning out" the supply of cash from the ATM.

In the early 1970s, a banking customer asked IBM to develop a system for encrypting ATM data. With this problem as a starting point, a team was formed from

Disclaimer

The present author participated in the design and test of DES, particularly in the design of the S-boxes and in strengthening them against differential cryptanalysis. Naturally, this author has strong opinions about DES and its history. Any opinions in this paper are those of the author and are not necessarily shared by IBM.

people at two IBM sites (Kingston and Yorktown Heights, New York). Developers included Roy Adler, Don Coppersmith, Horst Feistel, Edna Grossman, Alan Konheim, Carl Meyer, Bill Notz, Lynn Smith, Walt Tuchman, and Bryant Tuckerman. This team, along with several consultants, developed a cryptographic algorithm. This algorithm was then submitted to the National Bureau of Standards (NBS, which later became the National Institute of Standards and Technology, or NIST) and was adopted in 1977 as a national standard: the Data Encryption Standard, or DES. The National Security Agency (NSA) also provided technical advice to IBM.

The entire algorithm was published in the Federal Register [2], but the design considerations, which we present here, were not published at that time. The design took advantage of knowledge of certain cryptanalytic techniques, most prominently the technique of "differential cryptanalysis," which were not known in the published literature. After discussions with NSA, it was decided that disclosure of the design considerations would reveal the technique of differential cryptanalysis, a powerful technique that can be used against many ciphers. This in turn would weaken the competitive advantage the United States enjoyed over other countries in the field of cryptography.

Many people speculated, however, that the lack of disclosure was due to some "trap door" or hidden weakness in the DES. One of the purposes of the present paper is to dispel this notion and to indicate that, in fact, the reason for not publishing the criteria lay in the hidden strengths of the algorithm, not hidden weaknesses.

Contents of this paper

We begin by describing DES, giving enough detail to understand what follows. We then describe the attack based on differential cryptanalysis. We continue with a disclosure of the design criteria of the S-boxes and permutation, and a discussion of the role of these criteria in defeating differential cryptanalysis.

Description of DES

We give here a brief description of DES, primarily to establish terminology. We do not provide the various tables that are necessary for a full description of the standard; for those, see [2] or [3].

We wish to encipher a 64-bit plaintext *message block* m under the 56-bit *key* k, to produce a 64-bit ciphertext message block $c = E_k(m)$. (The sizes of message blocks and keys, 64 bits and 56 bits respectively, are specified in the standard.) Decipherment, or recovering plaintext from ciphertext, is denoted $m = D_k(c)$.

The plaintext message block m is subjected to an *initial permutation IP*, and the result is broken into two 32-bit *message halves*, m_0 and m_1. Intermediate message halves $m_2, \cdots, m_{17}$ are then created in sixteen *rounds*, according to the procedure described below. Finally, the 64-bit ciphertext c is generated by applying the inverse permutation IP^{-1} to the two message halves m_{17}, m_{16}. (Notice the inversion: m_{17}, m_{16} rather than the natural order. This is to allow decryption and encryption to use the same hardware.)

The plaintext message halves and intermediate message halves $m_0, m_1, m_2, \cdots, m_{17}$ are related as follows:

$$m_{i+1} = m_{i-1} \oplus f(k_{(i)}, m_i) \qquad i = 1, 2, \cdots, 16.$$

Here k is the secret 56-bit key, and i is the number of the round (from 1 through 16). Also, $k_{(i)}$ is a selection of 48 bits from the 56 bits of k; this selection, or *key schedule* (described in [2]), depends on the round number, i. The symbol $\oplus$ denotes bit-by-bit "exclusive OR" (addition modulo 2), which we call "XOR" in the text.

Now we describe the function f. There are eight *S-boxes*, $S_1, \cdots, S_8$, described in the standard. Each S-box is a table lookup, using six bits as input and providing four bits as output. For each S-box, say S_j, six consecutive bits are selected from the 48 bits of $k_{(i)}$, namely bits $6j - 5, 6j - 4, \cdots, 6j$. Also, six consecutive bits are selected from m_i, namely bits $4j - 4, 4j - 3, \cdots, 4j + 1 \pmod{32}$. The "mod 32" is shorthand for the convention that for $j = 1$ the bits are 32, 1, 2, 3, 4, 5, and for $j = 8$ the bits are 28, 29, 30, 31, 32, 1. Two adjacent S-boxes share two message bits; for instance, S_1 uses message bits 32, 1, 2, 3, 4, 5, while S_2 uses message bits 4, 5, 6, 7, 8, 9, and they share bits 4 and 5. (Key bits are not shared among S-boxes on one round.) S_8 and S_1 are considered to be "adjacent" because they share message bits 32 and 1.

The six key bits and the six message bits are XORed together bitwise, and the resulting six bits are used as input for a table lookup. That is, the six inputs to S-box S_j at round i are

$$m_i[4j - 4] \oplus k_{(i)}[6j - 5],$$

$$m_i[4j - 3] \oplus k_{(i)}[6j - 4],$$

$$\ldots$$

$$m_i[4j + 1] \oplus k_{(i)}[6j],$$

or, written another way,

$$m_i[4j - 4, 4j - 3, 4j - 2, 4j - 1, 4j, 4j + 1] \oplus k_{(i)}[6j - 5, 6j - 4, 6j - 3, 6j - 2, 6j - 1, 6j].$$

Each of the eight S-boxes implements a different table, each with 2^6 entries of four bits each. These tables are described in the standard.

The eight S-boxes together put out $8 \times 4 = 32$ bits. These bits are permuted according to a permutation P that

is fixed for all rounds i. The resulting 32-bit quantity is the value of $f(k_{(i)}, m_i)$.

In summary, the 64-bit message undergoes a permutation IP to produce two 32-bit message halves m_0 and m_1. Then we compute the 32-bit quantity $f(k_{(1)}, m_1)$, and XOR that quantity with m_0 to produce m_2. We use this new quantity m_2 to compute $f(k_{(2)}, m_2)$, and XOR that quantity with m_1 to produce m_3. We continue in a like fashion until m_{16} and m_{17} have been computed. These two message halves are interchanged and then subjected to the permutation IP^{-1}, to produce the ciphertext c.

Decryption is easily accomplished by a user in possession of the same key k. First, one applies the permutation IP to c to produce the message halves m_{17}, m_{16}. Next, one computes $f(k_{(16)}, m_{16})$ and XORs that quantity with m_{17} to recover m_{15}. Recalling that

$$m_{17} = m_{15} \oplus f(k_{(16)}, m_{16}),$$

we have

$$m_{17} \oplus f(k_{(16)}, m_{16}) = [m_{15} \oplus f(k_{(16)}, m_{16})] \oplus f(k_{(16)}, m_{16}) = m_{15},$$

because of the identity $(A \oplus B) \oplus B = A$. Similarly, one computes $m_{14} = m_{16} \oplus f(k_{(15)}, m_{15})$ and continues in like fashion until one has computed m_1 and m_0. Applying IP^{-1} to the pair (m_0, m_1), one recovers the plaintext message m.

Any function could be used in place of f, and we would still have a reversible encryption method. Different choices of f, however, yield different levels of security in the overall algorithm. The function f used in DES was designed to provide a high level of security.

Differential cryptanalysis

We present here an overview of the cryptanalytic attack known as "differential cryptanalysis." The terminology and notation are as presented by Biham and Shamir [4] (within IBM, the attack was formerly known as the "T attack"). Our purpose in presenting this is to show how the criteria for the S-boxes and the permutation were developed to thwart such attacks.

A cryptanalyst trying to break the system may be in possession of large amounts of plaintext and corresponding ciphertext, but not the secret key under which the text was enciphered. He knows the complete specification of the system (IP, S-boxes, P, key schedule); he would like to deduce the key.

As one can imagine, if he starts with a known plaintext m and unknown key k and tries to trace the encipherment through 16 rounds of DES encryption, he soon becomes hopelessly entangled, because bits of the unknown key k are XORed with the message at the input of every S-box.

In differential cryptanalysis, however, he starts with *two* messages, m and m', differing by a known difference Δm. That is,

$$\Delta m = m \oplus m'.$$

He considers the difference between the intermediate message halves:

$$\Delta m_i = m_i \oplus m_i'.$$

The input to S-box S_1, for example, at round i of the encipherment of message m is

$$m_i[32,1,2,3,4,5] \oplus k_{(i)}[1,2,3,4,5,6],$$

and the input to S_1 at round i of the encipherment of message m' is

$$m_i'[32,1,2,3,4,5] \oplus k_{(i)}[1,2,3,4,5,6].$$

From the identity $(a \oplus c) \oplus (b \oplus c) = a \oplus b$, we see that the XOR of these two inputs is

$$\begin{aligned}(m_i[32,1,2,3,4,5] &\oplus k_{(i)}[1,2,3,4,5,6]) \\ &\oplus (m_i'[32,1,2,3,4,5] \oplus k_{(i)}[1,2,3,4,5,6]) \\ &= m_i[32,1,2,3,4,5] \oplus m_i'[32,1,2,3,4,5] \\ &= \Delta m_i[32,1,2,3,4,5].\end{aligned}$$

The dependence on k has disappeared.

Now suppose that there is a relation between input differences and output differences for some S-box. That is, the 64 possible 6-bit inputs to S_1 can be divided into 32 pairs, so that the XOR of the two inputs in each pair is the given nonzero value $\Delta m_i[32,1,2,3,4,5]$. We call this difference $\Delta I_{i,1}$, because this is the change of inputs on the ith round for S_1. For each such pair of inputs, consider the pair of 4-bit outputs, and consider their XOR, called $\Delta O_{i,1}$. Differential cryptanalysis depends on the fact that many input pairs with a given input difference $\Delta I_{i,j}$ give rise to the same output difference $\Delta O_{i,j}$. For example, if $\Delta I_{i,1}$ is 110100, only eight of the 16 possible values of $\Delta O_{i,1}$ can occur, and one value of $\Delta O_{i,1}$ (0010) occurs for eight of the 32 input pairs sharing the difference $\Delta I_{i,1}$ = 110100. (This example is from Table 27 of [4].)

The quantities $f(k_{(i)}, m_i)$ and $f(k_{(i)}, m_i')$ are the permuted outputs of the S-boxes. Recall that

$$m_{i+1} = m_{i-1} \oplus f(k_{(i)}, m_i),$$

$$m_{i+1}' = m_{i-1}' \oplus f(k_{(i)}, m_i').$$

Taking the XOR of these two equations, we find

$$\begin{aligned}\Delta m_{i+1} &= m_{i+1} \oplus m_{i+1}' \\ &= [m_{i-1} \oplus f(k_{(i)}, m_i)] \oplus [m_{i-1}' \oplus f(k_{(i)}, m_i')] \\ &= \Delta m_{i-1} \oplus [f(k_{(i)}, m_i) \oplus f(k_{(i)}, m_i')].\end{aligned}$$

So if Δm_{i-1} and Δm_i are known with high probability, and if Δm_i gives rise to any difference, $f(k_{(i)}, m_i) \oplus f(k_{(i)}, m_i')$,

with high probability, we know Δm_{i+1} with high probability.

In differential cryptanalysis, we begin with two plaintext messages m and m' with a specified difference $\Delta m = (\Delta m_0, \Delta m_1)$ (known with certainty). We trace through a *probable pattern* of round-by-round differences $\Delta m_2, \Delta m_3, \cdots, \Delta m_{16}, \Delta m_{17}$. If the ciphertexts $E_k(m) = IP^{-1}(m_{17}, m_{16})$ and $E_k(m') = IP^{-1}(m'_{17}, m'_{16})$ exhibit the difference of our probable pattern

$$E_k(m) \oplus E_k(m') = IP^{-1}(\Delta m_{17}, \Delta m_{16}),$$

we suspect that this probable pattern is in fact the pattern of round-by-round differences. Assuming this probable pattern to be correct, we can then make deductions about the key bits on the basis of this one plaintext–ciphertext pair, and eventually discover the key. The reader is referred to [4] for further discussion.

Notice that, under the "chosen plaintext" assumption, we can choose the plaintexts m and m' to exhibit a desired difference Δm, chosen to optimize the cryptanalytic process, and we can observe their ciphertexts $E_k(m) = IP^{-1}(m_{17}, m_{16})$ and $E_k(m') = IP^{-1}(m'_{17}, m'_{16})$ and their difference $IP^{-1}(\Delta m_{17}, \Delta m_{16})$, and compare this difference with the difference predicted by the pattern. We cannot observe the intermediate results, m_i and m'_i, or their differences, Δm_i $(2 \le i \le 15)$.

Differential cryptanalysis will succeed if one of these probable patterns, extending over the 16 rounds of the encipherment, has a high enough probability that it will be observed among the ciphertext resulting from the corpus of chosen plaintext that the cryptanalyst is able to have encrypted on his behalf. In fact, a given probable pattern has only a very low probability of matching a given pair of messages for the entire 16 rounds, so that an enormous number of plaintext messages (more than 10^{14}) must be enciphered in order to have a reasonable probability of success.

Biham and Shamir [5] show ways to bypass the requirements of matching the probable pattern on the first one or two rounds and the last one or two rounds of the encipherment, so that the probable pattern need only be matched during twelve rounds of encipherment, rather than sixteen. This is significant, because the probability of existence of a given pattern decreases roughly exponentially with the length of the pattern.

With a particular probable pattern in mind, we say that S-box S_j is *active* on round i if $\Delta I_{i,j}$ (the set of six bits of Δm_i that are input to S_j) is not all zero. For each active S-box on each round, we can calculate the probability that the predicted value $\Delta O_{i,j}$ arises from the input $\Delta I_{i,j}$, given that all input pairs resulting in $\Delta I_{i,j}$ are equally likely. To a first approximation, we can estimate the probability of the entire probable pattern as the multiplicative product of these individual probabilities over all of the active S-boxes on all 12 or 16 rounds. (In doing so, we are treating the events at different S-boxes on different rounds as being statistically independent, while they are in fact dependent; this makes the analysis easier without materially affecting the outcome.) As the number of rounds increases, the total number of active S-boxes on these rounds also increases, and the probability decreases exponentially.

History and discussion

Differential cryptanalysis was not known in the open literature until quite recently. Some of the ideas were present in Bert Den Boer's 1988 cryptanalysis [6] of the four-round FEAL cryptographic scheme proposed by NTT. This cryptanalysis examined the difference between encryptions of two related chosen plaintext messages. At the Securicom meeting in 1989, Adi Shamir demonstrated an attack against an eight-round, shortened version of DES, but without making the techniques known; he and Eli Biham had been investigating differential cryptanalysis since 1988. In 1990, Sean Murphy published the method [7], as used in his cryptanalysis of NTT's newer, eight-round version of FEAL. This was soon followed by several papers of Biham and Shamir, among them [4, 5, 8, 9].

Differential cryptanalysis was well known, however, to the IBM team that designed DES, as early as 1974. Knowledge of this technique, and the necessity to strengthen DES against attacks using it, played a large part in the design of the S-boxes and the permutation P. We list the relevant design criteria employed during the design of the S-boxes and the permutation, and show how they contributed to the defense of DES against differential cryptanalysis. (Many of these criteria have been noted in the open literature; see for example [10], where several of the criteria were discovered by reverse engineering.) Because of this careful design, a differential cryptanalysis attack against DES requires enormous amounts of chosen plaintext. Biham and Shamir [5] estimate the amount of plaintext necessary for their attack as $2^{47.2} \approx 1.6 \times 10^{14}$ chosen plaintext messages (of eight bytes each), or more than 1.2×10^{15} bytes of chosen plaintext.

It is important to notice that these messages are *chosen plaintext*. The attacker must arrange for this much plaintext to be enciphered by a *target machine*, namely, a machine in possession of the secret key. In general, this is much more difficult to arrange than computations on one's own machine. Biham and Shamir's attack aroused much interest because the number of chosen plaintext messages, $2^{47.2}$, was less than the number of encipherments, $2^{56} \approx 7.2 \times 10^{16}$, required for "key exhaustion," or trying all possible keys until one finds the correct one. But the comparison is between chosen plaintext messages (encipherments on the target machine) and computations on one's own machine, so that a direct one-for-one

comparison is misleading. At any rate, the amount of necessary chosen plaintext is so enormous as to render the attack infeasible.

We remark that iterated encryption enhances the strength of DES against both key exhaustion and differential cryptanalysis. Some installations use triple encryption under two independent keys (encipher under the first key, decipher the result under the second key, and re-encipher the result under the first key). This raises the cost of key exhaustion to $2^{112} \approx 5 \times 10^{35}$ encipherments (but see [11] for a decrease in this estimate when a large corpus of known plaintext and corresponding ciphertext is available), while the cost of differential cryptanalysis suffers an exponential growth to something exceeding 10^{52} chosen plaintexts and corresponding ciphertexts. At this point the size of message space ($2^{64} \approx 1.8 \times 10^{19}$ possible messages of 64 bits) becomes the limiting factor in security.

The IBM team knew about differential cryptanalysis but did not publish any reference to it. That was because the tool can be a very powerful cryptanalytic tool, useful against many schemes, and there was concern that placing such information in the public domain could adversely affect national security.

Design criteria

We list here the criteria for the S-boxes and the permutation P, which were used in the original specifications, and which are satisfied by the design of DES.

The relevant criteria for the S-boxes are as follows:

(S-1) Each S-box has six bits of input and four bits of output. (This was the largest size that we could accommodate and still fit all of DES onto a single chip in 1974 technology.)

(S-2) No output bit of an S-box should be too close to a linear function of the input bits. (That is, if we select any output bit position and any subset of the six input bit positions, the fraction of inputs for which this output bit equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.)

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

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

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

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

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

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

Other criteria dealt with ease of implementation; those presented above are the only cryptographically relevant criteria.

The criteria for the permutation P are the following:

(P-1) The four output bits from each S-box at round i are distributed so that two of them affect (provide input for) "middle bits" of S-boxes at round $i + 1$ (the two middle bits of input to an S-box, not shared with adjacent S-boxes), and the other two affect "end bits" (the two left-hand bits or the two right-hand bits, which are shared with adjacent S-boxes).

(P-2) The four output bits from each S-box affect six different S-boxes; no two affect the same S-box. (Remember that each "end bit" affects two adjacent S-boxes.)

(P-3) For two S-boxes j, k, if an output bit from S_j affects a middle bit of S_k, then an output bit from S_k cannot affect a middle bit of S_j. This implies that in the case $j = k$, an output bit from S_j must not affect a middle bit of S_j.

Discussion of criteria

(S-2) was needed because the S-boxes constitute the only nonlinear part of DES. If they were linear [each output bit being a linear combination of the input bits in the finite field $GF(2)$], the entire algorithm would be linear and thus trivially broken; if they were close to being linear, the entire algorithm would be too close to linearity, and thus susceptible to attacks based on near-linearity.

Most of the criteria are aimed at increasing the number of active S-boxes involved over the 12 or 16 rounds of the probable pattern. If this total number is n, then (S-7), along with the simplifying assumption of independence, puts an upper bound of $(1/4)^n$ on the overall probability of this probable pattern.

At least one of any two consecutive rounds i, $i + 1$ must have a nonzero number of active S-boxes; otherwise the pattern is the trivial one of all 0's over all rounds. Suppose that round i has at least one active S-box. We break our analysis into cases, based on the number of active S-boxes on round i. In each case, we show that if round i has a small nonzero number of active S-boxes, then either round $i - 1$ or round $i + 1$ has at least one active S-box as well. The goal is to show that, summing over the 12 or 16 rounds of the pattern, there will be a large number of active S-boxes, on average at least 1.6 per round.

• *Two active S-boxes*

Suppose first that round i has exactly two active S-boxes and that they are adjacent, S_j and S_{j+1}. (The nonadjacent case is similar to the case of one active S-box, which is treated below.) Because S_{j-1} is inactive on this round, we know that the two left-hand bits of $\Delta I_{i,j}$ are zero; because S_{j+2} is inactive on this round, the two right-hand bits of $\Delta I_{i,j+1}$ are zero. We claim that either $\Delta O_{i,j} \neq 0$ or $\Delta O_{i,j+1} \neq 0$ (or both); the proof is by contradiction. If $\Delta O_{i,j} = 0$, then (S-3) and the fact that the left-hand two bits of $\Delta I_{i,j}$ are 0 together imply that the rightmost bit of $\Delta I_{i,j}$ is 1. [We know that the leftmost bit is 0. If the rightmost bit were also 0, this would imply that for the two inputs to S_j on round i in the two encipherments, m and m', the leftmost and rightmost input bits of S_j would be fixed. Some of the other four bits are varied, however. (S-3) implies that the two outputs must be different, so $\Delta O_{i,j} \neq 0$. This contradicts our assumption, so we conclude the rightmost bit is 1.] Because of the sharing of message bits, the rightmost bit of $\Delta I_{i,j}$ is also the second bit from the left of $\Delta I_{i,j+1}$. Similarly, if $\Delta O_{i,j+1} = 0$, then (S-3) and the fact that the two right-hand bits of $\Delta I_{i,j+1}$ are 0 imply that the leftmost bit of $\Delta I_{i,j+1}$ is 1. Combining these facts: if $\Delta O_{i,j} = \Delta O_{i,j+1} = 0$, then $\Delta I_{i,j+1}$ is of the form **11xy00**. In this case, (S-6) implies that $\Delta O_{i,j+1} \neq 0$. The conclusion is that we cannot have $\Delta O_{i,j} = \Delta O_{i,j+1} = 0$.

Remembering that the bits of $\Delta O_{i,j}$, $\Delta O_{i,j+1}$ are part of

$$f(k_{(i)}, m_i) \oplus f(k_{(i)}, m_i') = \Delta m_{i+1} \oplus \Delta m_{i-1},$$

we see that each of the nonzero bits of $\Delta O_{i,j}$, $\Delta O_{i,j+1}$ (there is at least one such nonzero bit) forces the corresponding bit of either Δm_{i-1} or Δm_{i+1} to be nonzero. Thus, either round $i - 1$ or round $i + 1$ (or both) has at least one active S-box.

This contributes to our conclusion that there will be a large number of active S-boxes over the course of the 12-round or 16-round pattern.

• *One active S-box*

Suppose next that round i has only one active S-box, namely S_j. Because S_{j-1} is inactive on this round, we know that the two left-hand bits of $\Delta I_{i,j}$ are zero; because S_{j+1} is inactive on this round, the two right-hand bits of $\Delta I_{i,j}$ are zero. Now either (S-4) (if only one of the middle bits of $\Delta I_{i,j}$ is 1) or (S-5) (if they are both 1) implies that $|\Delta O_{i,j}| \geq 2$. As stated before, $\Delta O_{i,j}$ is part of

$$\Delta m_{i+1} \oplus \Delta m_{i-1},$$

and each of the (at least two) nonzero bits of $\Delta O_{i,j}$ forces an active S-box either in round $i - 1$ or in round $i + 1$. Because of (P-2), there are at least two different active S-boxes included in rounds $i - 1$ and $i + 1$ together.

As before, this helps assure us that there will be a large number of active S-boxes over the course of the 12-round or 16-round pattern.

Consider the possibility of exactly one active S-box per round. Suppose $|\Delta O_{i,j}| = 2$, its minimum possible value. Of the two 1-bits in $\Delta O_{i,j}$, one cancels the 1-bit that had activated some S-box, say S_a, on round $i - 1$, and the other activates S_b on round $i + 1$. Because only one box is activated by each bit, the bit must be one of the two middle inputs to the S-box in each instance. That is, an output bit from S_j affects a middle input bit of S_a, and an output bit from S_a affects a middle input bit of S-box j. But (P-3) outlaws this situation, so a pattern of exactly one active S-box per round is impossible.

By these and similar arguments, we find a lower bound on the average number of active S-boxes per round. Except in the case of "three active S-boxes" (see below), this lower bound works out to an average of 1.6 active S-boxes per round. By (S-7) and the simplifying assumptions of statistical independence, each active S-box contributes a multiplicative factor of at most 1/4 to the probability of a given probable pattern. Thus, we have effectively constrained the probability of such a probable pattern to be less than some minuscule upper bound, of the order of 10^{-14}.

• *Three active S-boxes*

The most promising case for the cryptanalyst (thus the most difficult case for the designers of the system) turns out to be three adjacent active S-boxes S_j, S_{j+1}, and S_{j+2} on one round i, and no active S-boxes on adjacent rounds $i - 1$ and $i + 1$. Extending this pattern, we have three adjacent active S-boxes on even-numbered rounds and none on odd-numbered rounds.

A feature of this type of pattern is that, for i even, we have $\Delta m_{i+1} = 0$, so that $\Delta O_{i+1,j} = 0$ for all j, and

$$f(k_{(i+1)}, m_{i+1}) \oplus f(k_{(i+1)}, m_{i+1}') = \Delta m_i \oplus \Delta m_{i+2} = 0.$$

This implies that

$$\Delta m_i = \Delta m_{i+2} = \Delta m_{i+4} = \cdots.$$

We need only examine one nonzero value of Δm_i, and the same analysis holds for rounds $i + 2$, $i + 4$, $\cdots$.

The analysis is similar to the case of two active S-boxes previously discussed. The three S-boxes have a total of 14 input bits, which we label as in **Figure 1(a)**. The labeled bits such as **a** and **b** are understood to be XORs of input bits, i.e., part of Δm_i. Since $\Delta m_{i-1} = \Delta m_{i+1} = 0$, we must have $\Delta O_{i,k} = 0$ for all k. Since $j - 1$ is inactive, the leftmost two bits of $\Delta I_{i,j}$ are 0. That fact, together with $\Delta O_{i,j} = 0$ and criterion (S-3), enable us to conclude (as we did in the case of two active S-boxes) that the rightmost bit of $\Delta I_{i,j}$ is 1. Similarly, the rightmost two bits of $\Delta I_{i,j+2}$ are 0, and the leftmost bit is 1. So far, our knowledge of the input bits is as summarized in **Figure 1(b)**.

Applying (S-6) to S_{j+2}, we find that **j = 0**. Then, applying (S-3) to S_{j+1}, we find that **e = 1**, so our information is as in **Figure 1(c)**. That is,

$\Delta I_{i,j}$ = **0 0 c d 1 1**,

$\Delta I_{i,j+1}$ = **1 1 g h 1 0**,

$\Delta I_{i,j+2}$ = **1 0 k m 0 0**.

The unknown bits (**c**, **d**, **g**, h, **k**, **m**) are the middle two bits of each S-box, not shared with adjacent S-boxes. This simplifies the subsequent analysis.

To find the attack with the highest probability of success, we choose bits **c**, **d** so as to maximize the probability that $\Delta O_{i,j} = 0$ given that $\Delta I_{i,j}$ = **0 0 c d 1 1**. Making similar choices of bits **g**, **h** and **k**, **m**, we can estimate the probability that $\Delta O_i = 0$, given the choice of location of the S-boxes S_j, S_{j+1}, and S_{j+2}. We maximize this over the eight choices of $j = 1, 2, \cdots, 8$ (S_8 and S_1 are considered to be adjacent) to find the most likely pattern of this form, which occurs when $j = 1$ and Δm_i = 19600000 (hex). This choice leads to the following probabilities:

for $\Delta I_{i,1}$ = **0 0 0 0 1 1**, prob($\Delta O_{i,1} = 0$) = 7/32,

for $\Delta I_{i,2}$ = **1 1 0 0 1 0**, prob($\Delta O_{i,2} = 0$) = 4/32,

for $\Delta I_{i,3}$ = **1 0 1 1 0 0**, prob($\Delta O_{i,3} = 0$) = 5/32.

Thus,

$$\text{prob}[\Delta O_i = 0 | \Delta m_i = 19600000 \text{ (hex)}]$$

$$\simeq \frac{7}{32} \times \frac{4}{32} \times \frac{5}{32} = \frac{35}{8192}$$

$$\simeq 0.004272.$$

This is the pattern investigated in [4, 5].

Because this situation (three active S-boxes on even-numbered rounds, alternating with 0 active S-boxes on odd-numbered rounds) is so attractive to the cryptanalyst, the design team instituted condition (S-8) to lower the probability of success with such a pattern. With this background, we can now state (S-8):

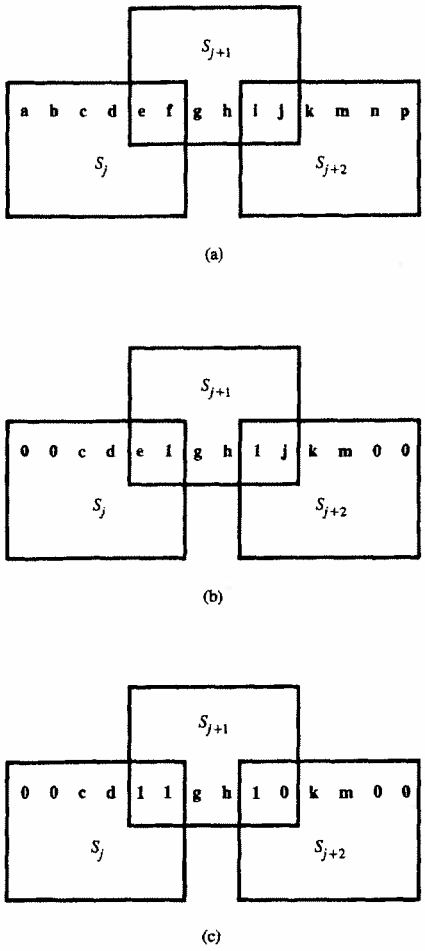

Figure 1

Three stages of knowledge about inputs to three S-boxes: (a) initially; (b) partial knowledge; (c) with eight bits determined.

(S-8) Define

$$q_{0,j} = \max_{\mathbf{c},\mathbf{d}} \text{prob}(\Delta O_{i,j} = 0 | \Delta I_{i,j} = \mathbf{0\ 0\ c\ d\ 1\ 1}),$$

$$q_{1,j} = \max_{\mathbf{g},\mathbf{h}} \text{prob}(\Delta O_{i,j} = 0 | \Delta I_{i,j} = \mathbf{1\ 1\ g\ h\ 1\ 0}),$$

$$q_{2,j} = \max_{\mathbf{k},\mathbf{m}} \text{prob}(\Delta O_{i,j} = 0 | \Delta I_{i,j} = \mathbf{1\ 0\ k\ m\ 0\ 0}).$$

Arrange S-boxes so as to minimize

$$\max_{j=1,2,\cdots,8} q_{0,j} q_{1,j+1} q_{2,j+2}.$$

As stated above, the objective function we achieved was

$$\frac{7}{32} \times \frac{4}{32} \times \frac{5}{32} = \frac{35}{8192},$$

obtained when $j = 1$.

Linear cryptanalysis

Very recently, Mitsuru Matsui has developed a related attack (not yet published), known as "linear cryptanalysis." This attack is stronger than differential cryptanalysis on two counts: It uses less text (about 10^{14} rather than 10^{15} characters), and it requires *known* plaintext and corresponding ciphertext, rather than *chosen* plaintext. Text must still be enciphered on a machine containing the secret key, but the cryptanalyst can use any such text, without needing to specify it himself. Of course, collecting this amount of known plaintext and corresponding ciphertext from the attacked machine is still a huge logistical problem, and this attack does not represent a viable threat against DES; it is still much more difficult than simple key exhaustion.

The design criterion related to this attack is (S-2). The following stronger criterion (S-2′) would be more useful, but to the best of my recollection it was not part of the design criteria:

(S-2′) No linear combination of output bits of an S-box should be too close to a linear function of the input bits. (That is, if we select any subset of the four output bit positions and any subset of the six input bit positions, the fraction of inputs for which the XOR of these output bits equals the XOR of these input bits should not be close to 0 or 1, but rather should be near 1/2.)

Neither (S-2) nor (S-2′) can be achieved perfectly, with all probabilities being exactly equal to 1/2. However, the fact that (S-2) was a design criterion and was almost achieved helped DES to resist this new attack. Even higher resistance could have been achieved by including (S-2′). One could achieve tighter controls [probabilities much closer to 1/2 for both (S-2) and (S-2′)] by using larger S-boxes. Using a larger number of rounds would also blunt this attack. Future cryptographic systems should take these modifications into consideration.

Summary: Design criteria and differential cryptanalysis

We have summarized Biham and Shamir's attack. We have outlined the criteria that IBM used to design the S-boxes and permutation. These criteria were developed specifically to thwart attacks based on differential cryptanalysis; we have shown here the relationship between these criteria and these attacks.

A measure of the success of IBM's approach to the design of S-boxes and permutation is the enormous amount of chosen plaintext (in excess of 10^{15} bytes) required by Biham and Shamir's attack.

References

1. D. Kahn, *The Codebreakers*, MacMillan Publishing Co., New York, 1972.
2. "Data Encryption Standard," *Federal Information Processing Standards Publication No. 46*, National Bureau of Standards, January 15, 1977.
3. C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Computer Data Security*, John Wiley & Sons, Inc., New York, 1982.
4. E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *J. Cryptol.* **4**, 3–72 (1991).
5. E. Biham and A. Shamir, "Differential Cryptanalysis of the Full 16-round DES," *Lecture Notes in Computer Science: Advances in Cryptology—Proceedings of CRYPTO '92*, Springer-Verlag, pp. 487–496. See also E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
6. B. Den Boer, "Cryptanalysis of F.E.A.L.," *Lecture Notes in Computer Science: Advances in Cryptology—Proceedings of EUROCRYPT '88*, Springer-Verlag, pp. 293–299.
7. S. Murphy, "The Cryptanalysis of FEAL-4 with 20 Chosen Plaintexts," *J. Cryptol.* **2**, 145–154 (1990).
8. E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *Lecture Notes in Computer Science: Advances in Cryptology—Proceedings of CRYPTO '90*, Springer-Verlag, 1990, pp. 2–21.
9. E. Biham and A. Shamir, "Differential Cryptanalysis of FEAL and N-Hash," *Lecture Notes in Computer Science: Advances in Cryptology—Proceedings of EUROCRYPT '91*, Springer-Verlag, 1991, pp. 1–16.
10. M. Hellman, R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Pohlig, and P. Schweitzer, "Results of an Initial Attempt to Cryptanalyze the NBS Data Encryption Standard," *Information Systems Laboratory Report*, Stanford University (September 9, 1976) (revised November 10, 1976).
11. P. C. van Oorschot and M. J. Wiener, "A Known-Plaintext Attack on Two-Key Triple Encryption," *Lecture Notes in Computer Science: Advances in Cryptology—Proceedings of EUROCRYPT '90*, Springer-Verlag, 1990, pp. 318–325.

Received March 11, 1993; accepted for publication February 14, 1994

Don Coppersmith *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (COPPER at YKTVMV, copper@watson.ibm.com).* Dr. Coppersmith received his B.S. in mathematics from the Massachusetts Institute of Technology in 1972 and his M.S. and Ph.D. in mathematics from Harvard University in 1975 and 1977. Since that time he has been a Research Staff Member at the IBM Thomas J. Watson Research Center. His current research interests include cryptography and computational complexity. Dr. Coppersmith is a Fellow of the Institute of Electrical and Electronics Engineers.