Differential Cryptanalysis of the full 16-round DES

Eli Biham
Computer Science Department
Technion - Israel Institute of Technology
Haifa 32000, Israel

Adi Shamir
Department of Applied Mathematics and Computer Science
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract

In this paper we develop the first known attack which is capable of breaking the full 16 round DES in less than the 2^{55} complexity of exhaustive search. The data analysis phase computes the key by analyzing about 2^{36} ciphertexts in 2^{37} time. The 2^{36} usable ciphertexts are obtained during the data collection phase from a larger pool of 2^{47} chosen plaintexts by a simple bit repetition criteria which discards more than 99.9% of the ciphertexts as soon as they are generated. While earlier versions of differential attacks were based on huge counter arrays, the new attack requires negligible memory and can be carried out in parallel on up to 2^{33} disconnected processors with linear speedup. In addition, the new attack can be carried out even if the analyzed ciphertexts are derived from up to 2^{33} different keys due to frequent key changes during the data collection phase. The attack can be carried out incrementally with any number of available ciphertexts, and its probability of success grows linearly with this number (e.g., when 2^{29} usable ciphertexts are generated from a smaller pool of 2^{40} plaintexts, the analysis time decreases to 2^{30} and the probability of success is about 1%).

1 Introduction

The Data Encryption Standard (DES) is the best known and most widely used cryptosystem for civilian applications. It consists of 16 rounds of substitution and permutation operations, carried out under the control of a 56 bit key (see [6] for further details). It was adopted as a US national standard in the mid 70's, and had been extensively analyzed for over 15 years. However, no attack which is faster than exhaustive search (whose complexity is 2^{55} due to a simple complementation property that halves the number of searched keys) has ever been reported in the open literature.

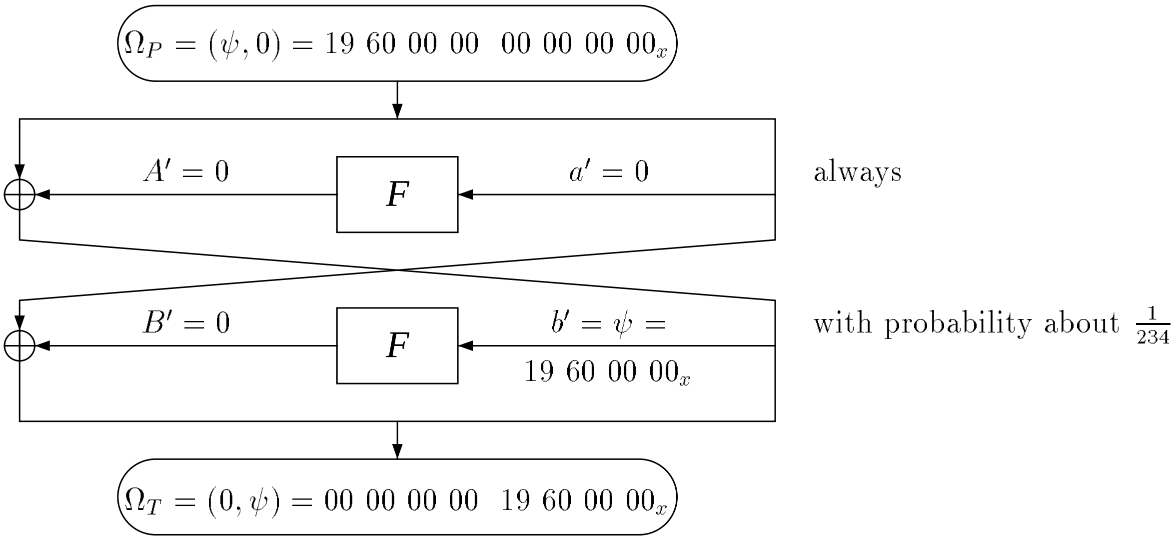
The lack of progress in the cryptanalysis of the full DES led many researchers to analyse simplified variants of DES, and in particular variants of DES with fewer than 16 rounds. Chaum and Evertse[4] described an attack on reduced variants of DES, whose complexity is 2^{54} for the six-round variant. They showed that their attack is not applicable to variants with eight or more rounds. Davies[5] devised a known plaintext attack whose application to DES reduced to eight rounds analyzes 2^{40} known plaintexts and has time complexity 2^{40}. This attack is not applicable to the full 16-round DES since it has to analyze more than the 2^{64} possible plaintexts. The most successful attack on reduced variants of DES was the method we called differential cryptanalysis [1], which could break variants of DES with up to 15 rounds faster than via exhaustive search. However, for the full 16-round DES the complexity of the attack was 2^{58}, which was slower than exhaustive search. Similar attacks were used to cryptanalyze a large number of DES-like cryptosystems and hash functions [2,3].

In this paper we finally break through the 16-round barrier. We develop an improved version of differential cryptanalysis which can break the full 16-round DES in 2^{37} time and negligible space by analyzing 2^{36} ciphertexts obtained from a larger pool of 2^{47} chosen plaintexts. An interesting feature of the new attack is that it can be applied with the same complexity and success probability even if the key is frequently changed and thus the collected ciphertexts are derived from many different keys. The attack can be carried out incrementally, and one of the keys can be computed in real time while it is still valid. This is particularly important in attacks on bank authentication schemes, in which the opponent needs only one opportunity to forge a multi-million dollar wire transfer, but has to act quickly before the next key changeover invalidates his message.

2 The New Attack

The reader is assumed to be familiar with the general concept of differential cryptanalysis, and in particular with the definitions and notations introduced in [1]. As usual, we ignore the initial permutation IP and final permutation IP^{-1} of DES, since they have no effect on our analysis.

The old attack on the 15-round variant of DES was based on the following two-round iterative characteristic:



The 13-round characteristic results from iterating this characteristic six and a half times and it's probability is about $2^{-47.2}$. The attack used this characteristic in rounds 1 to 13, followed by a 2R-attack on rounds 14 to 15. Any pair of plaintexts which gives rise to the intermediate XORs specified by this characteristic is called a right pair. The attack tries many pairs of plaintexts, and eliminates any pair which is obviously wrong due to its known input and output values. However, since the cryptanalyst cannot actually determine the intermediate values, the elimination process is imperfect and leaves behind a mixture of right and wrong pairs.

In earlier versions of differential cryptanalysis, each surviving pair suggested several possible values for certain key bits. Right pairs always suggest the correct value for these key bits (along with several wrong values), while wrong pairs suggest random values. When sufficiently many right pairs are analyzed, the correct value (signal) overcomes the random values (noise) by becoming the most frequently suggested value. The actual algorithm is to keep a separate counter for the number of times each value is suggested, and to output the index of the counter with the maximal final value. This approach requires a huge memory (with up to 2^{42} counters in the attack on the 15-round variant of DES), and has a negligible probability of success when the number of analyzed pairs is reduced below the threshold implied by the signal to noise ratio.

In the new version of differential cryptanalysis, we work somewhat harder on each pair, and suggest a list of complete 56-bit keys rather than possible values for a subset of key bits. As a result, we can immediately test each suggested key via trial encryption, without using any counters. These tests can be carried out in parallel on disconnected processors with very small local memories, and the algorithm is guaranteed to discover the correct key as soon as the first right pair is encountered. Since the processing of different pairs are unrelated, they can be generated by different keys at different times due to frequent key changes, and the discovery of a key can be announced in real time while it is still valid (e.g., in order to forge authenticators for banking messages).

The key to success in such an attack is to use a high probability characteristic, which makes it possible to consider fewer wrong pairs before the first occurrence of a right pair. The probability of the characteristic used in the attack on the 15-round variant of DES is about $\left(\frac{1}{234}\right)^6 = 2^{-47.2}$. The obvious way to extend the attack to 16 rounds is to use the above iterative characteristic one more time, but this reduces the probability of the characteristic from $2^{-47.2}$ to $2^{-55.1}$, which makes the attack slower than exhaustive search. Our new attack adds the extra round without reducing the probability at all.

The assumed evolution of XORs of corresponding values during the encryption of a right pair of plaintexts in the new 16-round attack are summarized in Figure 1, which consists of the old 15-round attack on rounds 2 to 16, preceded by a new round 1.

Our goal is to generate without loss of probability pairs of plaintexts whose XORed outputs after the first round are the required XORed inputs $(\psi, 0)$ into the 13-round characteristic of rounds 2 to 14. Let P be an arbitrary 64-bit plaintext, and let $v_0, \ldots, v_{4095}$ be the 2^{12} 32-bit constants which consist of all the possible values at the 12 bit positions which are XORed with the 12 output bits of S1, S2 and S3 after the first round, and 0 elsewhere. We now define a structure which consists of 2^{13} plaintexts:

$$\begin{array}{lll} P_i = P \oplus (v_i, 0) & \bar{P}_i = (P \oplus (v_i, 0)) \oplus (0, \psi) & \text{for } 0 \leq i < 2^{12} \\ T_i = \mathrm{DES}(P_i, K) & \bar{T}_i = \mathrm{DES}(\bar{P}_i, K) & \end{array}$$

The plaintext pairs we are interested in are all the pairs P_i, $\bar{P}_j$ with $0 \leq i, j < 2^{12}$. There are 2^{24} such plaintext pairs, and their XOR is always of the form (v_k, ψ), where each v_k occurs exactly 2^{12} times. Since the actual processing of the left half of P and of the left half of P XORed with ψ in the first round under the actual key creates a XORed value after the first round which can be non-zero only at the outputs of S1, S2 and S3, this XORed value is one of the v_k. As a result, for exactly 2^{12} of the plaintext pairs, the output XOR of the first F-function is exactly cancelled by XORing it with the left half of the plaintext XOR, and thus the output XOR of the first round (after swapping the left and right halves) is the desired input XOR $(\psi, 0)$ into the iterative characteristic. Therefore, each structure has a probability of about $2^{12} \cdot 2^{-47.2} = 2^{-35.2}$ to contain a right pair.

The problem in this approach is that we do not know the actual value of v_k, which cancels the output XOR of the first F-function, and thus we do not know on which 2^{12} plaintext pairs to concentrate. Trying all the 2^{24} possible pairs takes too long, but we can use their cross-product structure to isolate the right pairs among them in just 2^{12} time. In any right pair, the output XOR after 16 rounds should be zero at the outputs of the five S-boxes S4, ..., S8 (i.e., , at 20 bit positions). We can thus sort

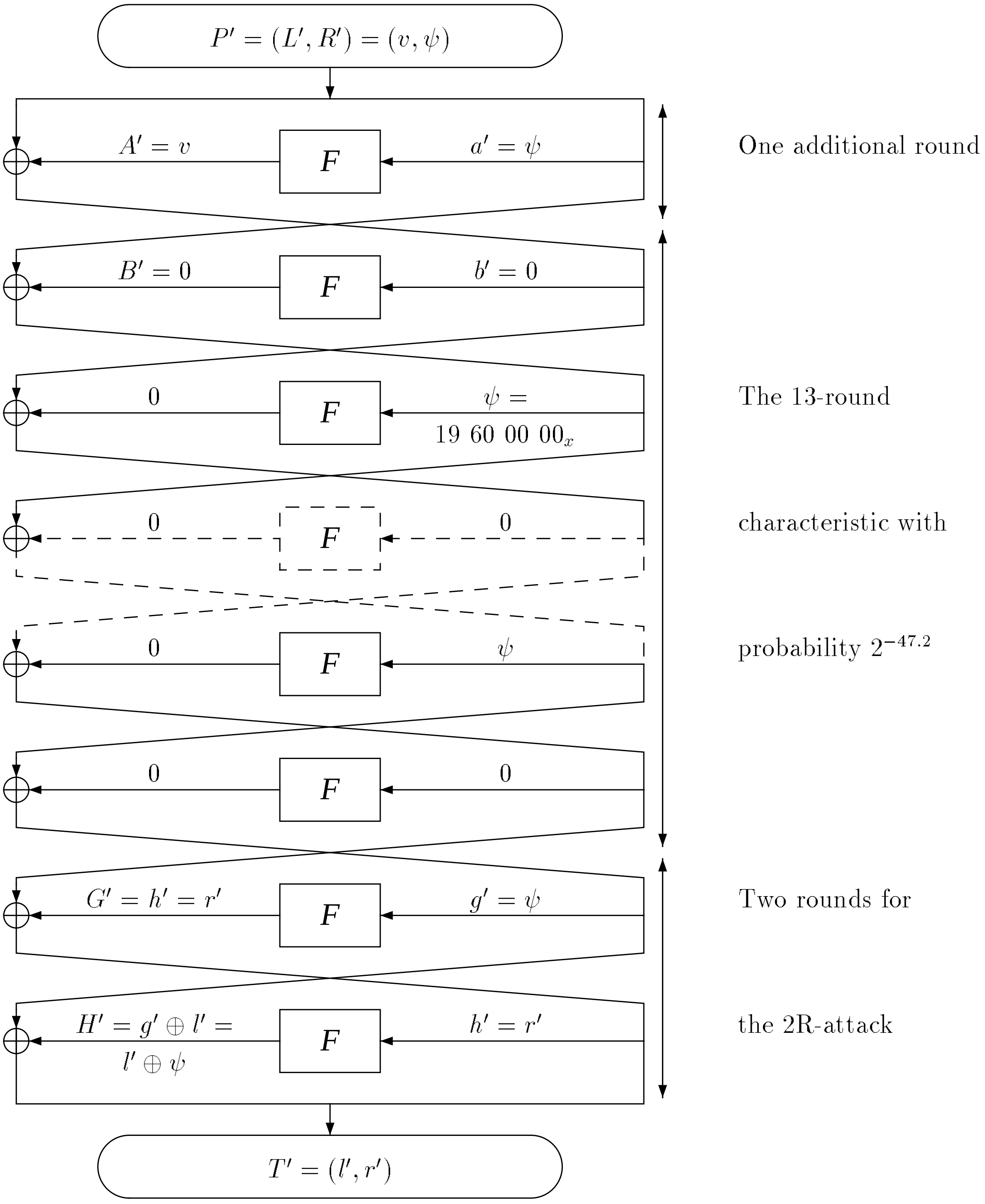


Figure 1. The extension of the attack to 16 rounds.

(or hash) the two groups of 2^{12} ciphertexts T_i, $\bar{T}_j$ by these 20 bit positions, and detect all the repeated occurrences of values among the 2^{24} ciphertext pairs in about 2^{12} time. Any pair of plaintexts which fails this test has a non-zero ciphertext XOR at those 20 bit positions, and thus cannot be a right pair by definition. Since each one of the 2^{24} possible pairs passes this test with probability 2^{-20}, we expect about $2^4 = 16$ pairs to survive. By testing additional S boxes in the first, fifteenth, and sixteenth rounds and eliminating all the pairs whose XOR values are indicated as impossible in the pairs XOR distribution tables of the various S boxes, we can discard about 92.55% of these surviving pairs[1] leaving only $16 \cdot 0.0745 = 1.19$ pairs per structure as the expected output of the data collection phase. All these additional tests can be implemented by a few table lookup operations into small precomputed tables, and their time complexity is much smaller than the time required to perform one trial encryption during an exhaustive search. Note that this filtering process removes only wrong pairs but not all of them and thus the input of the data analysis phase is still a mixture of right and wrong pairs.

The data analysis phase of previous differential cryptanalytic attacks used huge arrays of up to 2^{42} counters to find the most popular values of certain key bits. The new variant of differential attack described in this paper uses only negligible space. We want to count on all the key bits simultaneously but cannot afford the huge array of 2^{56} counters. Instead, we immediately try each suggested value of the key. A key value is suggested when it can create the output XOR values of the last round as well as the expected output XOR of the first round and the fifteenth round for the particular plaintext pairs and ciphertext pairs. In the first round and in the fifteenth round the input XORs of S4 and S5, ..., S8 are always zero. Due to the key scheduling algorithm, all the 28 bits of the left key register are used as inputs to the S boxes S1, S2 and S3 in the first and the fifteenth rounds and S1, ..., S4 in the sixteenth round. Only 24 bits of the right key register are used in the sixteenth round. Thus, $28 + 24 = 52$ key bits enter these S boxes. $\frac{2^{-32}}{0.8^8}$ of the choices of the 52-bit values remain by comparing the output XOR of the last round to its expected value and discarding the ones whose values are not possible and $\frac{2^{-12}}{\frac{14}{16}\cdot\frac{13}{16}\cdot\frac{15}{16}}$ of the remaining ones remain by comparing the output XOR of the three S boxes in the first round to its expected value. A similar fraction of the remaining 52-bit values remain by analyzing the three S boxes in the fifteenth round. Each analyzed pair suggests about $2^{52} \cdot \frac{2^{-32}}{0.8^8} \cdot \frac{2^{-12}}{\frac{14}{16}\cdot\frac{13}{16}\cdot\frac{15}{16}} \cdot \frac{2^{-12}}{\frac{14}{16}\cdot\frac{13}{16}\cdot\frac{15}{16}} = 0.84$ values for these 52 bits of the key, and each one of them corresponds to 16 possible values of the full 56-bit key. Therefore, each structure suggests about $1.19 \cdot 0.84 \cdot 16 = 16$ choices for the whole key. By peeling up two additional rounds we can verify each such key by performing about one quarter of a DES encryption (i.e., executing two rounds for each one of the two members of the pair), leaving only about 2^{-12} of the choices of the key. This filtering costs about

[1] A fraction of about $\left(\frac{14}{16}\cdot\frac{13}{16}\cdot\frac{15}{16}\right)^2 \cdot 0.8^8 = 0.0745$ of these pairs remain and thus a fraction of about 0.9255 of them are discarded. The input XOR values of the S boxes in the first and the fifteenth rounds of right pairs are known and fixed, and thus we use the fraction of non-zero entries of the corresponding lines in the pairs XOR distribution tables whose values are $\frac{14}{16}$, $\frac{13}{16}$ and $\frac{15}{16}$, rather than the fraction of the non-zero entries in the whole tables, which is approximated by 0.8.

		K16									
		Left Key Register					Right Key Register				
		S1	S2	S3	S4	X	S5	S6	S7	S8	X
K1	S1		2	1	1	2					
	S2	2		1	2	1					
	S3	2			3	1					
	S4	2	3	1							
	X		1	3							
	S5							1	2	2	1
	S6						3		2	1	
	S7							2		2	2
	S8						2	3			1
	X						1		2	1	

Table 1. The number of common bits entering the S boxes in the first round (K1) and in the sixteenth round (K16).

$16 \cdot \frac{1}{4} = 4$ equivalent DES operations. Each remaining choice of the 56-bit key is verified via trial encryption of one of the plaintexts and comparing the result to the corresponding ciphertext. If the test succeeds, there is a very high probability that this key is the right key. Note that the signal to noise ratio of this counting scheme is $S/N = \frac{2^{52} \cdot 2^{-47.2}}{1.19/2^{12} \cdot 0.84} = 2^{16.8}$.

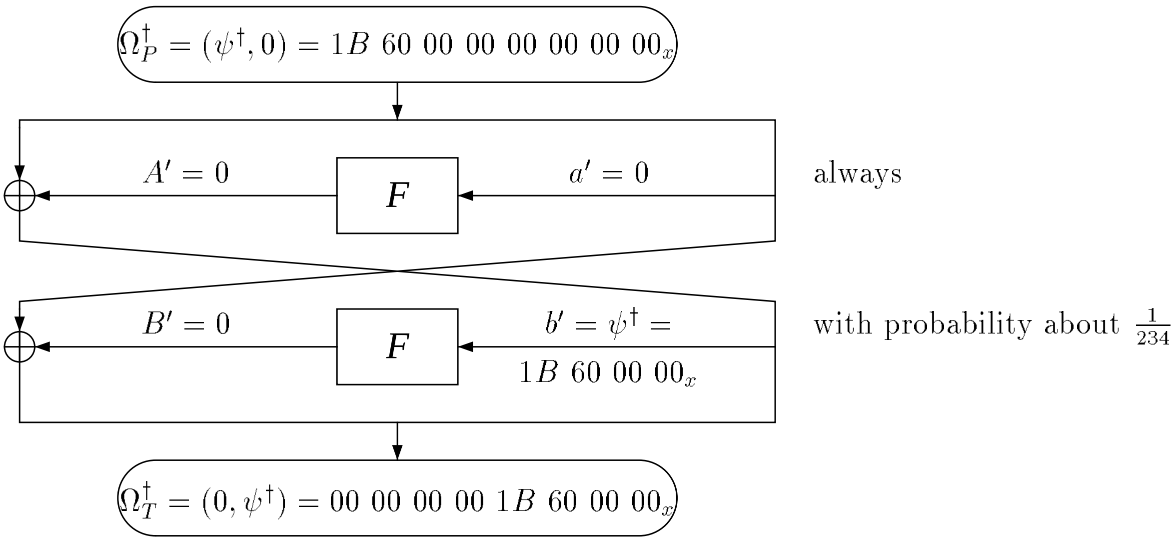
This data analysis can be carried out efficiently by carefully choosing the order in which we test the various key bits. We first enumerate all the possible values of the six key bits of $S4_{Kh}$, and eliminate any value which does not give rise to the expected XOR of the four output bits from this S box. This leaves four out of the 64 possibilities in average. Table 1 shows the number of common bits entering the S boxes in the first round and in the sixteenth round. The notation X denotes the bits which are not used in the specific subkey. We see that three of the bits of $S4_{Kh}$ are shared with $S3_{Ka}$. We complete the three missing bits of $S3_{Ka}$ in all possible ways, and reduce the average number of possibilities to two. Two bits of $S1_{Kh}$ are shared with $S3_{Ka}$. By completing the four missing bits of $S1_{Kh}$ and then the two missing bits of $S2_{Ka}$ we can reduce the average number of possibilities to about half. After completing the 13 remaining bits of the left key register in a similar way, the average number of values suggested for this half of the key is one.

To compute bits from the right key register, we first extract actual S box bits from their assumed XORed values. In the fifteenth round we know the input XORs and the output XORs of S1, S2 and S3. We can thus generate about 4–5 candidate inputs for each one of these S boxes, and deduce the corresponding bits in g by XORing with the known bits of the left key register. In a similar way, we can calculate the outputs of the S boxes S1, S2, S3 and S4 in the sixteenth round, XOR these bits of H with the known bits of the left half of the ciphertext l and get 16 bits of g, from which two bits enter S1, two bits enter S2 and three bits enter S3 in the fifteenth round.

By comparing these bit values to the candidate inputs of the S boxes we end up with about one candidate input for S1, one for S2, and only about half of the trials would result with a candidate input for S3. We can now deduce all the bits of g which enter these three S boxes and deduce the corresponding bits of H by $H = g \oplus l$. Two of these bits are outputs of S5, two bits are outputs of S6, three are outputs of S7 and one is output of S8. For each of these four S boxes we know the input XOR and the output XOR, and can deduce about 4–5 possible inputs. Since we also know actual output bits, the number of possible inputs is reduced to about one for S5 and S6, two for S8, but only half of the trials would result with a candidate for S7. We can deduce 24 out of the 28 bits of the right key register by XORing the 24 computed bits at the inputs of these four S boxes with the expanded value of the known right half of the ciphertext.

We can now summarize the performance of the new attack in the following way. Each structure contains a right pair with probability $2^{-35.2}$. The data collection phase encrypts a pool of about 2^{35} structures, which contain about $2^{35} \cdot 2^{13} = 2^{48}$ chosen plaintexts, from which about $2^{35} \cdot 1.19 = 2^{35.25}$ pairs ($2^{36.25}$ ciphertexts) remain as candidate inputs to the data analysis phase. The probability that at least one of them is a right pair is about 58%, and the analysis of any right pair is guaranteed to lead to the correct key. The time complexity of this data analysis phase is about $2^{35} \cdot 4 = 2^{37}$ equivalent DES operations.

In order to further reduce the number of chosen plaintexts, we can use the quartet method of [1]. Since the basic collection of plaintexts in the new attack is a structure rather than a pair, we create metastructures which contain 2^{14} chosen plaintexts, built from two structures which correspond to the standard iterative characteristic and from two structures which correspond to the following iterative characteristic:



This characteristic has the same probability as the previous one. With these metastructures, we can obtain four times as many pairs from twice as many plaintexts, and thus reduce the number of chosen plaintexts encrypted in the data collection phase

Rounds	Chosen Plaintexts	Analyzed Plaintexts	Complexity of Analysis	Best Previous Time	Best Previous Space
8	2^{14}	4	2^{9}	2^{16}	2^{24}
9	2^{24}	2	2^{32}	2^{26}	2^{30}
10	2^{24}	2^{14}	2^{15}	2^{35}	–
11	2^{31}	2	2^{32}	2^{36}	–
12	2^{31}	2^{21}	2^{21}	2^{43}	–
13	2^{39}	2	2^{32}	2^{44}	2^{30}
14	2^{39}	2^{29}	2^{29}	2^{51}	–
15	2^{47}	2^{7}	2^{37}	2^{52}	2^{42}
16	2^{47}	2^{36}	2^{37}	2^{58}	–

Table 2. Summary of the new memoryless results on DES.

from 2^{48} to 2^{47}.

The general form of the new attack can be summarized in the following way: Given a characteristic with probability p and signal to noise ratio S/N for a cryptosystem with k key bits, we can apply a memoryless attack which encrypts $\frac{2}{p}$ chosen plaintexts in the data collection phase and has complexity of $\frac{2^k}{S/N}$ trial encryptions during the data analysis phase. The number of chosen plaintexts can be reduced to $\frac{1}{p}$ by using appropriate metastructures, and the effective time complexity can be reduced by a factor of $f \leq 1$ if a tested key can be discarded by carrying out only a fraction f of the rounds. Therefore, memoryless attacks can be mounted whenever $p > 2^{1-k}$ and $S/N > 1$. The memoryless attacks require fewer chosen plaintexts compared to the corresponding counting schemes, but if the signal to noise ratio is too low or if the number of the key bits on which we count is small, the time complexity of the data analysis phase may be higher than the corresponding complexity of the counting scheme.

In the attack described in this paper, $p = 2^{-47.2}$, $k = 56$, $f = \frac{1}{4}$ and $S/N = 2^{16.8}$. Therefore, the number of chosen plaintexts is $\frac{2}{p} = 2^{48.2}$ which can be reduced to $\frac{1}{p} = 2^{47.2}$ by using metastructures, and the complexity of the data analysis phase is $2^{37.2}$ equivalent DES operations.

The performance of the new attack for various numbers of rounds is summarized in Table 2. Variants with an even number of rounds n have a characteristic with probability $p = \left(\frac{1}{234}\right)^{(n-4)/2}$, require p^{-1} chosen plaintexts, and analyze $p^{-1} \cdot 2^{-10.75}$ plaintexts in time complexity $p^{-1} \cdot 2^{-10}$. The known plaintext variant of the new attack needs about $2^{31.5} \cdot p^{-0.5}$ known plaintexts (using the symmetry of the cryptosystem which makes it possible to double the number of known encryptions by reversing the roles of the plaintexts and the ciphertexts). Variants with an odd number of rounds n have a characteristic with probability $p = \left(\frac{1}{234}\right)^{(n-3)/2}$, require p^{-1} chosen plaintexts, and analyze $p^{-1} \cdot 2^{-40.2}$ plaintexts in time complexity $p^{-1} \cdot 2^{-10}$. For such odd values of n, if $p > 2^{-40.2}$ then the number of analyzed plaintexts is two and the complexity of the data analysis phase is 2^{32}. However, using about four times as many chosen plaintexts, we can use the clique algorithm (described in [1]) and reduce the time complexity of the data analysis phase to less than a second on a personal computer. The known plaintext attacks need about $2^{32} \cdot p^{-0.5}$ known plaintexts (in this case the symmetry does not help). The application of the known plaintext attack to eight rounds needs a pool of $2^{38.5}$ known plaintexts. The application to 12 rounds needs a pool of $2^{47.2}$ known plaintexts. The application to 15 rounds needs a pool of $2^{55.6}$ known plaintexts and the application to the full 16-round DES needs a pool of $2^{55.1}$ known plaintexts. This is slightly worse than the 2^{55} complexity of exhaustive search (which in the case of a known plaintext attack requires about 2^{33} plaintexts in order to generate a complementary pair via the birthday paradox).

This specific attack is not directly applicable to plaintexts consisting solely of ASCII characters since such plaintexts cannot give rise to the desired XOR differences. By using several other iterative characteristics we can attack the full 16-round DES with a pool of about 2^{49} chosen ASCII plaintexts (out of the 2^{56} possible ASCII plaintexts).

References

[1] Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Journal of Cryptology, Vol. 4, No. 1, pp. 3–72, 1991. The extended abstract appears in Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'90, pp. 2–21, 1990.

[2] Eli Biham, Adi Shamir, *Differential Cryptanalysis of FEAL and N-Hash*, technical report CS91-17, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, 1991. The extended abstract appears in Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'91, pp. 1–16, 1991.

[3] Eli Biham, Adi Shamir, *Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer*, technical report CS91-18, Department of Applied Mathematics and Computer Science, The Weizmann Institute of Science, 1991. The extended abstract appears in Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'91, pp. 156–171, 1991.

[4] David Chaum, Jan-Hendrik Evertse, *Cryptanalysis of DES with a reduced number of rounds, Sequences of linear factors in block ciphers*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'85, pp. 192–211, 1985.

[5] D. W. Davies, *Investigation of a Potential Weakness in the DES Algorithm*, 1987, private communication.

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