Differential Cryptanalysis of Lucifer

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

Abstract

Differential cryptanalysis was introduced as an approach to analyze the security of DES-like cryptosystems. The first example of a DES-like cryptosystem was Lucifer, the direct predecessor of DES, which is still believed by many people to be much more secure than DES, since it has 128 key bits, and since no attacks against (the full variant of) Lucifer were ever reported in the cryptographic literature. In this paper we introduce a new extension of differential cryptanalysis, devised to extend the class of vulnerable cryptosystems. This new extension suggests key-dependent characteristics, called *conditional characteristics*, selected to increase the characteristics' probabilities for keys in subsets of the key space. The application of conditional characteristics to Lucifer shows that more than half of the keys of Lucifer are insecure, and the attack requires about 2^{36} complexity and chosen plaintexts to find these keys. The same extension can also be used to attack a new variant of DES, called RDES, which was designed to be immune against differential cryptanalysis. These new attacks flash new light on the design of DES, and show that the transition of Lucifer to DES strengthened the later cryptosystem.

Key words: Differential Cryptanalysis, Lucifer, RDES.

1 Introduction

Differential cryptanalysis was introduced in [3,2] as an approach to analyze the security of DES-like cryptosystems. In a series of papers[3,4,5,6] this approach was used to attack the blockciphers DES[18], Feal[22,17], Khafre[15], REDOC-II[24], LOKI[8] and one variant of Lucifer[10], along with the hash functions N-Hash[16], and Snefru[14]. Lai et al[13] viewed a variant of this approach as a Markov chain and applied this approach to the PES[12] and the IDEA[13] ciphers. Other researchers studied how to make cryptosystems immune against differential cryptanalysis (some of which are [1,7,9,19,20,21]).

In this paper we extend differential cryptanalysis in several directions: The main extension of this paper enables differential cryptanalysis to analyze a wider set of cryptosystems. We define *conditional characteristics* as key-dependent characteristics selected to maximize the characteristic's probability (the fraction of right pairs) for only a specific subset of the key space. The required coverage of (almost) all the key space is done via selection of several conditional characteristics designed for different fractions of the key space.

In the attack on the full 16-round DES[6], structures which allow to gain one additional round for free with no additional cost are used. We extend this idea and show an implementation in which we gain two additional rounds for free, using the observations that the blocksize of Lucifer is larger than the one of DES and that the avalanche is slower. We also show two additional tools: a tool that gains a free additional round in Lucifer (described in the attack on the eight-round variant), and a tool that can increase the fraction of keys covered by differential cryptanalytic attacks when conditional characteristics are used. We suggest to use sets of characteristics whose Ω_P are the same, but which differ in their Ω_T. Since the same plaintexts can be shared for all these characteristics, the efficiency of the attacks is increased.

Many people still believe that Lucifer[23], the direct predecessor of DES, is stronger than DES, since it has 128 key bits rather than the 56 key bits of DES, and since they believe that the strength of DES was intentionally reduced by its designers. In this paper we study the strength of the variant of Lucifer described in [23] (the final variant of the Lucifer project, rather than the variant described in [10]). We apply our new techniques to this variant, and show an attack which can find the key with complexity about 2^{36}, if only the key resides within a particular subset of the key space containing about 55% of the keys. It is of interest to note that if the order of the two S boxes of Lucifer was reversed, a similar attack could cover more than 90% of the keys, but their replacement by S boxes satisfying the design rules of DES would invalidate the conditional characteristics used in this attack.

Several researchers studied how to make cryptosystems immune against differential cryptanalysis, but till now, this effort was not very successful. Many of them[1,9,19] suggested the use of S boxes whose difference distribution tables are uniform, and in particular they suggested the use of bent functions. However, the application of this suggestion to DES was studied in [2,7], and it was shown that the resultant cryptosystems become much weaker than DES.

Recently, Koyama and Terada[11] suggested to replace the deterministic swapping of the halves of the data between rounds in DES by a conditional swapping, which swap the halves only if a particular key bit (different for each round) has the value 1. They claim that the resultant cryptosystem, called RDES, is about 2^{15} times stronger than DES, although a small fraction of the keys, which do not swap the data even once, are bad. Our new extension developed in this paper can be applied to RDES, and shows that RDES is weaker than DES for almost all keys in the key space, leaving only a relatively small number of "good" keys, whose trial complexity is much smaller than exhaustive search of the whole key space.

2 Description of Lucifer

Lucifer[23] is the cryptosystem from which DES[18] was developed by IBM in the 1970's. Like DES, Lucifer has 16 rounds, but it has no initial and final permutations, and the sizes of its blocks and keys are 128 bits. The F function of Lucifer operates on the 64-bit right half of the data, 64-bit subkey and eight interchange control bits (ICBs). The F function uses only two four-bit to four-bit S boxes, called S_0 and S_1, applied in parallel to each byte of the input of the F function. The F function swaps the two nibbles (four bits) of each input byte whose corresponding interchange control bit (ICB) is zero. Then, the S box S_0 operates on the most significant (left-hand) nibble, and S_1 on the least significant (right-hand) nibble of every byte. The output of the S boxes is concatenated and the result is XORed with the subkey, in an operation called *key interruption*. The last stage of the F function permutes the output bits. Sorkin[23] describes the final bit permutation in two steps: each byte undergoes a fixed permutation (denoted P in [23]), and then the bits are mixed between the bytes – every bit enters a different byte in the same position in which it was in the original byte. This later step is called *diffusion*. We denote the product of these two permutations by P.

Figure 1 describes the F function of Lucifer. The pairs of adjacent S boxes are viewed as single combined boxes, to which we call *T boxes* (Transposition boxes). The T boxes are functions from nine bits to eight bits, whose one input bit is an ICB, and the eight others are data. The T boxes are defined by

$$\begin{aligned} T_0[XY] &= S_0[X]\ S_1[Y] \\ T_1[XY] &= S_0[Y]\ S_1[X]. \end{aligned}$$

They are described in Figure 2.

The key scheduling algorithm of Lucifer is much simpler than the one of DES. The key is assigned into a 128-bit shift register. Every round the subkey is chosen as the leftmost 64 bits of the register, the interchange control bits are chosen as the leftmost eight bits of the register, and after each round the shift register is rotated 56 bits to the left.

For the analysis it is convenient to use the following equivalent description: The key interruption is moved from after the S boxes to become the first operation in the F function (where a † is marked in Figure 1), and an initial XOR of the plaintext with a 128-bit subkey is added before the first round. The subkeys of this form are called *actual subkeys*, and are denoted by AK_i. The actual subkey of the last round (AK_{16}) is zero. AK_{15} is just the permuted value of the subkey of the last round ($AK_{15} = P(K_{16})$). The other actual subkeys $AK_1, \ldots, AK_{14}$ are $AK_i = AK_{i+2} \oplus P(K_{i+1})$, and

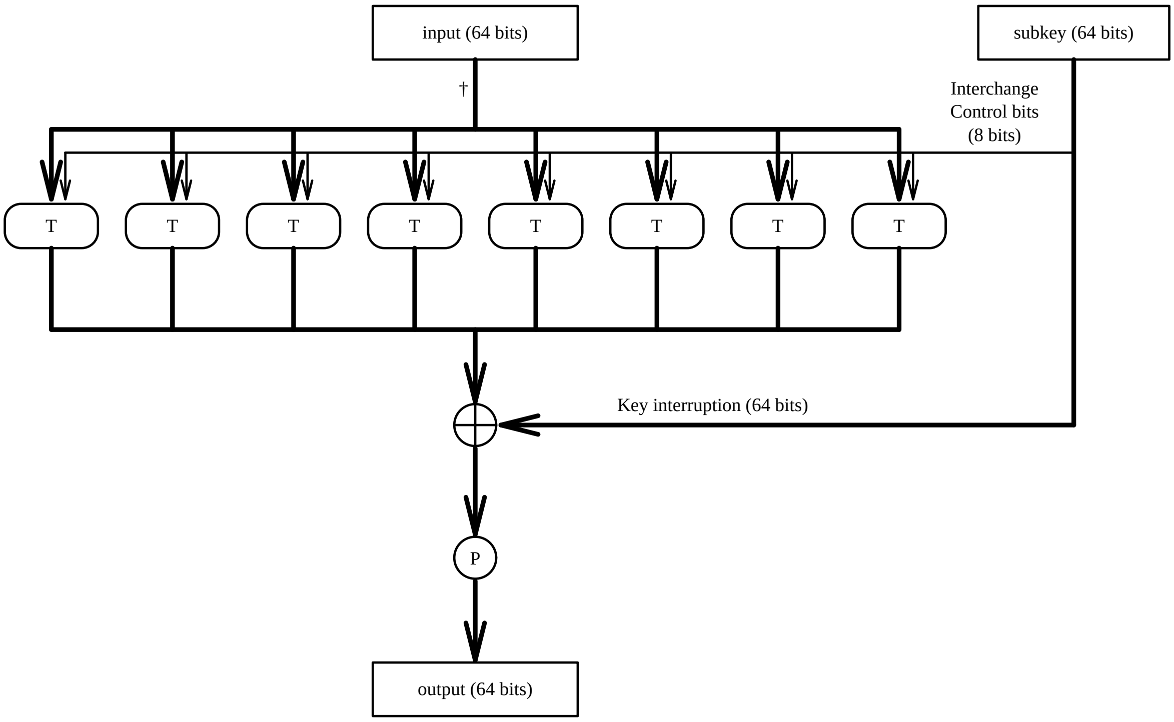


Figure 1. The F function of Lucifer.

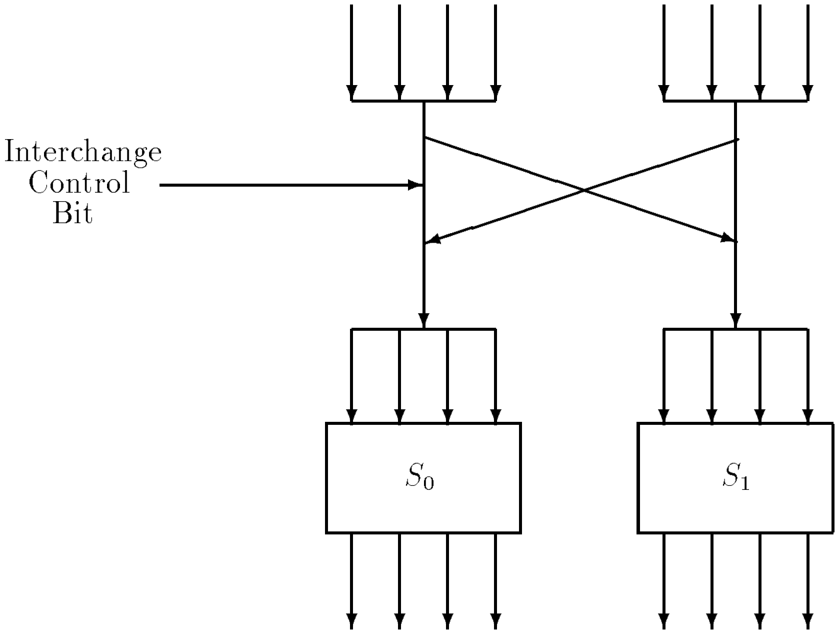


Figure 2. Lucifer T box structure.

the initial subkey is $(AK_2 \oplus P(K_1), AK_1)$. In this description the last round becomes very simple, with a zero actual subkey. During encryption the actual subkey of the first round is cancelled by the initial subkey. Thus effectively both the first and the last rounds have no key interruption.

We also denote the ciphertext by T, and its left and right halves by T_L and T_R respectively.

3 Conditional Characteristics

Differential cryptanalysis requires the knowledge of good characteristics, i.e., to find pairs of messages, such that the difference of the output of the nth round during encryption of these messages is predictable with a relatively high probability. The key-dependent swaps make it quite difficult to find such characteristics, especially since characteristics which can predict the output for all the keys have a very low probability – thus making an attack infeasible. In order to solve this difficulty we define key-dependent characteristics which depend on the value of some ICBs. In [3,2] the characteristic's probability is defined as the probability that a random pair (whose plaintext difference is Ω_P) is a right pair with respect to a random key, and it is shown that the probability that a random pair is a right pair with respect to a fixed key may depend on the choice of the key. In this paper we are interested in characteristics for which the probability that a random pair is a right pair varies between different keys. We call these characteristics *conditional characteristics*.

Definition 1 The *probability of a characteristic Ω with respect to a fixed key K* is the probability that a random pair (whose plaintext difference is Ω_P) is a right pair with respect to the fixed key K.

Definition 2 The *probability of a characteristic Ω with respect to a set of keys U* is the minimal probability of the characteristic Ω with respect to a key K in U.

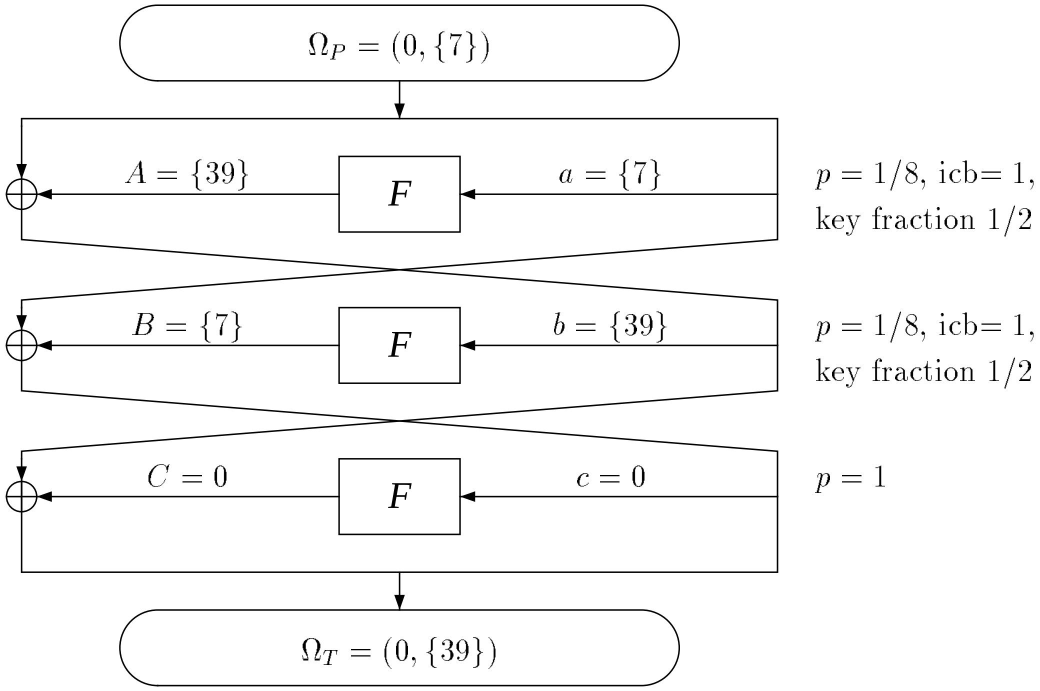
Definition 3 A *conditional characteristic* is a tuple (Ω, U, p_U^Ω) where Ω is a characteristic, U is a subset of the key space, and p_U^Ω is the probability of the characteristic Ω with respect to the subset U.

Definition 4 The *key fraction* of a conditional characteristic (Ω, U, p_U^Ω) is the ratio $|U|/|K|$ between the size of the subset U and the size of the key space.

These definitions suggest a tradeoff between the probability of a conditional characteristic and its key fraction. By reducing the size of U we can increase the probability of the conditional characteristic, but the key fraction is reduced. By increasing the size of U we increase the key fraction, but the probability may be reduced.

Whenever a conditional characteristic (Ω, U, p_U^Ω) improves the probability over the best probability of a non-conditional characteristic by a factor higher than the inverse of the key fraction $(|K|/|U|)$, the usage of the conditional characteristic is advisable. There are several additional cases in which the usage of conditional characteristics is advisable as well, especially if several such characteristics can efficiently share the same structure of chosen plaintexts.

We found four six-round iterative conditional characteristics of Lucifer. One of them is (only three rounds are described; the other three rounds are symmetric):



where $\{n\}$ denotes a 64-bit value whose nth bit ($n \in \{0, \ldots, 63\}$) is one and all the others are zero. The other three iterative conditional characteristics are similar with the replacement of the constants $\{7\}$ and $\{39\}$ by the constants (1) $\{15\}$ and $\{47\}$, (2) $\{23\}$ and $\{55\}$, and (3) $\{31\}$ and $\{63\}$. Each of these characteristics has six incarnations, starting from the six possible rounds.

4 The Attack on Lucifer

The differential cryptanalysis of Lucifer is slightly different than the cryptanalysis of DES. We describe the differential cryptanalysis of Lucifer in the following subsections. In the first subsection we describe the required structures and chosen plaintexts, then we describe the cryptanalysis, and finally we study modified variants and strength factors.

4.1 The Data

In order to pack all the required pairs into as few chosen plaintexts as possible, we use structures similar to the ones used in [6]. In [6] an additional "free" first round is gained and the characteristic starts only at the second round. Due to the larger blocksize of Lucifer, and to the slower avalanche, we can use two such "free" rounds in our attack on Lucifer. We use 3R-attacks, and thus, 11-round characteristics are required. The above conditional characteristics, iterated to 11 rounds, have probability 2^{-21} and a key fraction 2^{-7} in 16 of the incarnations, and probability 2^{-24} and a key fraction 2^{-8} in eight of the incarnations. In the rest of this section we ignore the details of the required data and the analysis of the eight incarnations, since (paradoxically) they require fewer chosen plaintexts and simpler structures than the other 16 incarnations.

The characteristics we use cause (in the 16 incarnations) a single bit difference in the input to the second round (the one preceding the characteristic). This bit enters a T box and affects one of its S boxes whose choice depend on an ICB. For each key it may affect up to four output bits, either the output bits of S_0 or the output bits of S_1. Given a fixed value of the input XOR of the third rounds (defined by the characteristic) we result with up to four affected bits in the input of the first round, which affect up to four S boxes in the first round, and up to 16 bits of its output (whose choice depend on up to four ICBs). The additional bit corresponding to the differing bit in the input of the second round and the (possible) bit which differ in the input of the third round are already counted in the $4 + 16 = 20$ bits. Thus we use structures of 2^{20} chosen plaintext with all the possible values of the 20 bits, and whose other 44 bits are fixed to some value. Each such structure is built to conform to some value of five ICBs of rounds 1 and 2. Thus, we have to create 32 such structures for all the 32 possible values of these ICBs[1]. Each structure contains 2^{19} pairs with the required difference before the third round. Since the characteristics' probability is about 2^{-21}, about four structures are required in average to have a right pair, if the ICBs of that characteristic have the required values. Therefore, a total of $2^{20} \cdot 4 \cdot 32 = 2^{27}$ chosen plaintexts are required for each incarnation of a characteristic to have one right pair.

The key fraction of the 16 incarnations is 2^{-7} and the key fraction of the other eight is 2^{-8}. The 24 incarnations cover a total fraction of about 15% of the key space. However, when we use some duplication techniques, which duplicate either the required data or the analysis for the two possible values of the extreme ICBs of the characteristics, we can enrich the set of covered keys and cover a fraction of about 25% of the key space. For this fraction, about $2^{27} \cdot 24 \cdot 16 \cdot 2 \approx 2^{36}$ chosen plaintexts are required (24 incarnations, 16 right pairs, 2 is the maximal duplication of the data).

We can increase the fraction of covered keys further using the observation that

[1] If the number of chosen plaintexts required was much larger, we could build huge more efficient structures for which such duplication is not required.

there are several conditional characteristics with the same Ω_P as of the characteristics we use, but with different Ω_T's and different key subsets U. Each Ω_P we use have about 9–10 such characteristics whose total key fraction is about three times the original key fraction, and their probabilities are about the same as of the original characteristics. In the appendix we show such additional characteristics (which we actually use in our attack). Due to the almost perfect identification of wrong pairs this attack has, we can analyze these characteristics with a negligible additional cost with the same data. Thus, this attack covers a fraction of about 55% of the keyspace[2]. We can still increase this fraction slightly using characteristics whose key fraction is slightly smaller than of the ones described, but whose Ω_P's have many additional characteristics with different Ω_T's.

4.2 The Analysis

For the analysis we use the notation h to be the input of the F function of the last round in the equivalent description of Lucifer, g and f are the inputs to the two preceding rounds, and H, G and F are the outputs of the F function in these rounds.

The first step of the analysis discards as many wrong pairs as possible. The value of f' contains at most one non-zero bit, thus, most bits of F' are zero, and at most four bits of F' are non-zero; the particular choice of the four bits is ICB dependent. The value of g' may contain at most five non-zero bits (these four bits plus one bit from e'), which may affect the output of at most five S boxes in G', and thus, $h' = T'_R$ may have at most $5 \cdot 4 + 1 = 21$ non-zero bits in positions depending on at most five ICBs. Thus the probability that a random T'_R is zero at all the 43 bits suggested by one of the 2^5 choices of the five ICBs is about $2^{-43} \cdot 2^5 = 2^{-38}$. Therefore, the identification of wrong pairs in a structure can be done efficiently by sorting (or hashing) by these bits, and choosing only pairs with common values. Each structure contains up to $(2^{20})^2/2 = 2^{39}$ potential pairs, and thus the average number of the remaining (wrong) pairs per structure is expected to be less than two for each characteristic.

Since effectively there is no key interruption in the last round, and since $h = T_R$, we can calculate for any ciphertext the 256 possible outputs H of the F function of the last round using the 256 possible choices of the interchange control bits, and get 256 possible values for H' for any pair. Independently, we can calculate 56 bits of H' for any pair, using the facts that $H' = T'_L \oplus F' \oplus e'$ and that 56 particular bits of F' are zero. This value should match one of the 256 possible values calculated directly. If it does not match, the pair is clearly a wrong pair, and should be discarded. The probability of a random pair to pass this test is about $2^8 \cdot 2^{-56} = 2^{-48}$. Thus, the average number of wrong pairs in a structure which pass both the previous test and this test is $2 \cdot 2^{-48} = 2^{-47}$ for each characteristic. In practice only right pairs are expected to pass both tests. From these right pairs we can easily derive the values

[2]It can be verified easily (but inaccurately) by $1 - (1 - 0.25)^3 = 0.58$. The exact calculation results with a value slightly higher than 0.55.

of seven ICBs of the last round, the seven (or eight) ICBs controlling the conditional characteristic[3], the five ICBs affecting rounds 14 and 15 during the analysis, and the five ICBs affecting the choice of the chosen plaintexts in the first two rounds. All these ICBs are different (since each key bit is used only once as an ICB) and thus we get a total of $7+7+5+5=24$ bits of the key.

Now we can calculate the output of the F function of the last round for any given ciphertext, and find the value of g, effectively reducing the cryptosystem to 15 rounds. The value of G' can be calculated from the characteristic and the ciphertexts by $G' = T'_R \oplus f'$, where f' is the value suggested from the characteristic. Thus, we can mount a simple counting scheme to find many additional bits of the actual subkey AK_{15}, and then use other standard differential cryptanalytic techniques to complete the rest of the key.

4.3 Modified Variants and Weaknesses

As in DES, the order of the S boxes is important. If we only replace the S boxes S_0 and S_1 by each other, the number of (iterative) conditional characteristics grows to 20 (rather than four) and the fraction of the keys vulnerable to these attacks grows to more than 60% using about 2^{38} chosen plaintexts (rather than 25%). When using several characteristics with the same Ω_P's, the fraction of keys vulnerable to the attack grows to more than 90%.

On the other hand, replacement of the S boxes by single lines of the S boxes of DES (or by S boxes satisfying the design rules of DES) would invalidate the kind of characteristics used in the above attacks, in which a difference of one input bit of an S box may cause a difference of only one output bit. However, in order to strengthen the cryptosystem, we should make sure that no other kinds of high probability characteristics exist.

In order to disable conditional characteristics, we may choose the interchange control bits as combinations of key bits and data bits, rather than of key bits alone. It was actually done in DES.

The key interruption in the F function is done in Lucifer after the S boxes. This order effectively eliminates the key interruption in the first round and in the last round and allows the analyst to analyze an equivalent description with one or two fewer rounds. The replacement of the order of the key interruption and the S boxes, as was done in DES, solves this weakness (but enables complementation properties).

The F function of Lucifer has a rotational symmetry, in which rotations by multiples of eight bits of the input half, the subkey, and concurrent rotation by the same multiple of one bit of the interchange control bits cause rotation of the same multiple

[3] Whenever there are two different characteristics with the same Ω_P and Ω_T, we get two possible sets of seven or eight ICBs. This fact affects the remainder of the analysis only slightly.

of eight bits in the output. Therefore, characteristics can be rotated by multiples of eight bits as well, causing each characteristic to have seven rotated counterparts (when a characteristic is a rotation of itself we get fewer counterparts; the four characteristics used to attack Lucifer are such examples). In order to disable this property we have to use different S boxes in different entries, as was done in DES.

The eight-round reduced variant of Lucifer is very weak. Using the same conditional characteristics, with a new first round technique, we result with an attack requiring 256 chosen plaintexts, which cover about 90% of the keys. This attack places four-round characteristics built from the iterative characteristics described for the full variant, such that the $0 \rightarrow 0$ rounds are set in rounds 2 and 5, and such that rounds 3 and 4 have probabilities 1/8 and key fraction 1/2. There are eight such possible four-round characteristics, which cover together 90% of the key space. In order to get the first round for free, we can simply choose the right half of the plaintexts in any way (with the required input difference) and to calculate the output of the F function of the first round in the equivalent description (which can be done since AK_1 equals the right half of the initial subkey). Then we have only to choose the left halves in such a way that cancels the difference received from the output of the first round. Two structures of all the eight characteristics are used: one assumes that the affecting ICBs in the first round are zero, and the other assumes they are one. These structures contain 128 pairs for each characteristic. Since the characteristics' probability is 1/64, we get in average two right pairs which can be used to find directly many key bits. Additional standard techniques using the same structures can complete the key.

The fact that the fraction of right pairs may depend on the choice of the key was already noted in [3]. It was shown that the conditional characteristics of DES can enrich the fraction of right pairs by a medium factor, but the key fraction of these characteristics is too small to make an attack feasible. It was concluded that the use of these characteristics does not help to attack DES.

5 RDES

RDES[11] (Randomized DES) is an attempt to strengthen DES against differential cryptanalysis. In order to reduce the probability of characteristics, the designers suggested to replace the deterministic swaps of the halves of the data between rounds by key dependent swaps. They claim that since the 15 key dependent swaps occur with 2^{15} possible instances, the probability of the characteristics used against DES is reduced by a similar factor. As a result, they claim that RDES is much stronger than DES, and that the differential cryptanalytic technique of the full 16-round DES[6] is not applicable to RDES.

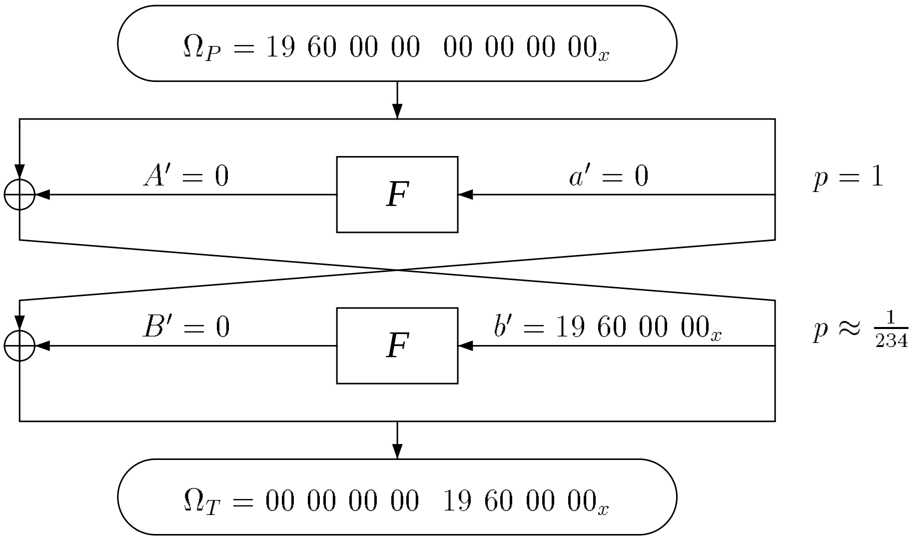
The new conditional technique suggested in this paper cancels the cryptanalytic effect of the key dependent swaps, and shows that RDES is weaker than DES.

The simplest weakness of RDES (already noted by the designers) is that one of every 2^{15} keys does not swap the data even once. Thus, half of the ciphertext bits (corresponding to the right half of the data during the various rounds) are the same in both the plaintext and the ciphertext. If this property is found under an attack, the attacker can immediately conclude the value of the 15 key bits affecting the swaps, and thus, an exhaustive search for the remaining key bits would require only about 2^{41} steps. Such property should be avoided in cryptosystems, and thus keys leading to this property are weak, and should not be used.

The next simplest weakness of RDES is that one of every 2^{15} keys swaps the data only once, just before the last round. In this case the attacker can easily derive the output of the F function of the last round, along with its input, and can find all the 48 bits of the subkey K_{16}, resulting with at most 256 possibilities for the key.

These two examples show that many keys are quite weak, thus it is interesting to ask whether elimination of these weak keys would make RDES more secure. Using the conditional differential cryptanalytic technique we can show that almost any key of RDES is weaker than the corresponding key of DES, and thus that RDES should not be used.

In DES the following two-round iterative characteristic is used:



along with a similar characteristic with $\Omega_P = 1B\ 60\ 00\ 00\ \ 00\ 00\ 00\ 00_x$. This characteristic can be iterated any number of times, since there is a deterministic swap between any two consecutive rounds. In RDES many swaps are cancelled due to the key dependent swapping policy. Thus, this characteristic cannot be iterated, and cannot be used (as is) against RDES.

However, when we look carefully, we see that whatever is the choice of the swaps, these two one-round characteristics (the two rounds of this two-round characteristic) can be combined to longer characteristics in two ways: In the first, choose the first one-round characteristic ($0 \to 0$) to appear in the first round, and the second to appear in the round after the first swap. The rest of the rounds can be completed uniquely using these two one-round characteristics. In the second way we replace all the occurrences of the one-round characteristics by each other. These two combined characteristics are duals: when one one-round characteristic occurs in a round of one combination, then the other one-round characteristic occurs in the same round of the other combination. As a results, such two r-round combined characteristics have probabilities $(\frac{1}{234})^q$ and $(\frac{1}{234})^{r-q}$, when q is the number of occurrences of $19\ 60\ 00\ 00_x \to 0$ (or of $1B\ 60\ 00\ 00_x \to 0$) in the first combined characteristic, and $r-q$ is the number of occurrences of $19\ 60\ 00\ 00_x \to 0$ in the second combined characteristic. Thus, for any choice of the key dependent swapping, we can easily find at least two r-round characteristics with probability $p \geq (\frac{1}{234})^{\lfloor r/2 \rfloor}$. Note that the right hand side of the inequality is the probability of the r-round iterative characteristics of DES, and thus RDES always has characteristics with probabilities higher or equal to the ones of DES.

It only remains to prepare characteristics for all the 2^{15} possible swap choices and choose sufficient number of plaintexts for all these choices. Fortunately, all these characteristics have only two possible values for Ω_P, and the same two possible values for Ω_T: $19\ 60\ 00\ 00\ \ 00\ 00\ 00\ 00_x$ and $00\ 00\ 00\ 00\ \ 19\ 60\ 00\ 00_x$. Therefore, the number of chosen plaintexts required for this attack is only up to twice the number required for the attack on DES, if characteristics with the same probability are used. However, for most keys these characteristics have probabilities much higher than $(\frac{1}{234})^{\lfloor r/2 \rfloor}$. The swap choice of many keys have q much smaller than $r/2$. Even when $q \approx r/2$ and the characteristics have two (or more) consecutive rounds of $19\ 60\ 00\ 00_x \to 0$, the probability is larger than $(\frac{1}{234})^{\lfloor r/2 \rfloor}$ since the probability that the exclusive-or of two (or more) output XORs (in which only in three particular S boxes the output XOR can be non-zero) is at least 2^{-12}, and not $(\frac{1}{234})^2 \approx 2^{-16}$, 2^{-24}, etc. We can conclude that the probability of one of the two dual characteristics must satisfy

$$p \geq 2^{(-8(s+1)-4(r-s-1))/2} = 2^{-2s-2r-2},$$

where s is the number of swaps during the r rounds (we approximate $\frac{1}{234}$ by 2^{-8}). The application of this formula to the attack on the full 16-round DES, which require a 13-round characteristic, shows that any choice of up to nine swaps during these 13 rounds would result with characteristic probabilities greater than $2^{-2\cdot 9-2\cdot 13-2} = 2^{-46}$. Therefore the attacks on these cases are faster than the attacks on DES and require fewer chosen plaintexts. Note that these attacks usually find the subkey of the last round, but if there is no single swap in the final few rounds, they identify this fact (which identifies several key bits) along with the number s of swaps (estimated from the probability). Using auxiliary techniques the full key can later be completed in both cases.

The fraction of keys which cause up to nine swaps during the 13 rounds is

$$\frac{\sum_{s=0}^{9} \binom{12}{s}}{2^{12}} \approx 0.98$$

so, at most one of every 50 keys may be strong as the corresponding key of DES against this attack. Even if only such "strong" keys are used, differential cryptanalysis requires about 2^{47} chosen plaintexts, and an exhaustive search of all the possibilities of these keys takes only about 2^{50} steps. Therefore, RDES is not more secure than DES, and for almost all keys it is even much weaker.

Note that unlike in Lucifer, even if we replace the swap control bits by combinations of key bits and data bits, the cryptosystem does not become more secure, since then for any key, a fraction of 2^{-15} of the plaintexts would be encrypted to ciphertexts whose right halves are just the same as those of the plaintexts (in RDES, a fraction of 2^{-15} of the keys are weak due to this property).

References

[1] Carlisle M. Adams, *On Immunity against Biham and Shamir's "Differential Cryptanalysis"*, Information Processing Letters, Vol. 41, No. 2, pp. 77–80, 1992.

[2] Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[3] Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Journal of Cryptology, Vol. 4, No. 1, pp. 3–72, 1991.

[4] Eli Biham, Adi Shamir, *Differential Cryptanalysis of FEAL and N-Hash (extended abstract)*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'91, pp. 1–16, 1991.

[5] Eli Biham, Adi Shamir, *Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer (extended abstract)*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'91, pp. 156–171, 1991.

[6] Eli Biham, Adi Shamir, *Differential Cryptanalysis of the full 16-round DES*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'92, pp. 487–496, 1992.

[7] Lawrence Brown, Matthew Kwan, Josef Pieprzyk, Jennifer Seberry, *Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of ASIACRYPT'91, pp. 36–50, 1991.

[8] Lawrence Brown, Josef Pieprzyk, Jennifer Seberry, *LOKI - A Cryptographic Primitive for Authentication and Secrecy Applications*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of AUSCRYPT'90, pp. 229–236, 1990.

[9] M. H. Dawson, S. E. Tavares, *An Expanded Set of S-box Design Criteria Based On Information Theory and its Relation to Differential-Like Attacks*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'91, pp. 352–367, 1991.

[10] H. Feistel, *Cryptography and Data Security*, Scientific American, Vol. 228, No. 5, pp. 15–23, May 1973.

[11] Kenji Koyama, Routo Terada, *How to Strengthen DES-like Cryptosystems against Differential Cryptanalysis*, IEICE Transactions on Fundumentals of Electronics, Communications and Computer Science, Vol. E76-A, No. 1, pp. 63–69, January 1993.

[12] Xuejia Lai, James L. Massey, *A Proposal for a New Block Encryption Standard*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'90, pp. 389–404, 1990.

[13] Xuejia Lai, James L. Massey, Sean Murphy, *Markov Ciphers and Differential Cryptanalysis*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'91, pp. 17–38, 1991.

[14] Ralph C. Merkle, *A Fast Software One-Way Hash Function*, Journal of Cryptology, Vol. 3, No. 1, pp. 43–58, 1990.

[15] Ralph C. Merkle, *Fast Software Encryption Functions*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'90, pp. 476–501, 1990.

[16] S. Miyaguchi, K. Ohta, M. Iwata, *128-bit hash function (N-Hash)*, proceedings of SECURICOM'90, pp. 123–137, March 1990.

[17] Shoji Miyaguchi, Akira Shiraishi, Akihiro Shimizu, *Fast Data Encryption Algorithm FEAL-8*, Review of electrical communications laboratories, Vol. 36, No. 4, pp. 433–437, 1988.

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

[19] Kaisa Nyberg, *Perfect nonlinear S-boxes*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'91, pp. 378–386, 1991.

[20] Luke O'Connor, *On the Distribution of Characteristics in Bijective Mappings*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'93, pp. 360–370, 1993.

[21] Luke O'Connor, *On the Distribution of Characteristics in Composite Permutations*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of CRYPTO'93, pp. 403–412, 1993.

[22] Akihiro Shimizu, Shoji Miyaguchi, *Fast Data Encryption Algorithm FEAL*, Lecture Notes in Computer Science, Advances in Cryptology, proceedings of EUROCRYPT'87, pp. 267–278, 1987.

[23] Arthur Sorkin, *Lucifer, a Cryptographic Algorithm*, Cryptologia, Vol. 8, No. 1, pp. 22–41, January 1984.

[24] Michael C. Wood, technical report, Cryptech Inc., Jamestown, NY, July 1990.

6 Appendix

In this appendix we show conditional characteristics with the same Ω_P but with different Ω_T's, which cover different fractions of the key space. Many conditional characteristics of Lucifer have this property of their Ω_P. These characteristics are actually used by our attacks.

The following conditional characteristic is an iterated version of the characteristic described in section 3.

Round	output difference	input difference	p	icb	key fraction
	$\Omega_P = (\{7\}, 0)$				
1	$A = 0$	$a = 0$	1		1
2	$B = \{39\}$	$b = \{7\}$	$p = 1/8$	1	1/2
3	$C = \{7\}$	$c = \{39\}$	$p = 1/8$	1	1/2
4	0	0	1		1
5	$\{7\}$	$\{39\}$	$p = 1/8$	1	1/2
6	$\{39\}$	$\{7\}$	$p = 1/8$	1	1/2
7	0	0	1		1
8	$\{39\}$	$\{7\}$	$p = 1/8$	1	1/2
9	$\{7\}$	$\{39\}$	$p = 1/8$	1	1/2
10	0	0	1		1
11	$\{7\}$	$\{39\}$	$p = 1/8$	1	1/2
	$\Omega_T = (\{7\}, \{39\})$				

The following characteristics have the same Ω_P, but different Ω_T and cover different fractions of the key space.

Round	output difference	input difference	p	icb	key fraction
	$\Omega_P = (\{7\}, 0)$				
1	$A = 0$	$a = 0$	1		1
2	$B = \{45\}$	$b = \{7\}$	$p = 1/8$	0	1/2
3	$C = \{6\}$	$c = \{45\}$	$p = 1/8$	1	1/2
4	$\{45\}$	$\{6, 7\}$	$p = 1/8$	0	1/2
5	0	0	1		1
6	$\{45\}$	$\{6, 7\}$	$p = 1/8$	0	1/2
7	$\{6\}$	$\{45\}$	$p = 1/8$	1	1/2
8	$\{45\}$	$\{7\}$	$p = 1/8$	0	1/2
9	0	0	1		1
10	$\{45\}$	$\{7\}$	$p = 1/8$	0	1/2
11	$\{6\}$	$\{45\}$	$p = 1/8$	1	1/2
	$\Omega_T = (\{6, 7\}, \{45\})$				

where $\{m, n\}$ denotes a 64-bit value whose mth and nth bits have the value one and all the others have value zero.

Round	output difference	input difference	p	icb	key fraction
	$\Omega_P = (\{7\}, 0)$				
1	$A = 0$	$a = 0$	1		1
2	$B = \{39\}$	$b = \{7\}$	$p = 1/8$	1	1/2
3	$C = \{7\}$	$c = \{39\}$	$p = 1/8$	1	1/2
4	0	0	1		1
5	$\{13\}$	$\{39\}$	$p = 1/8$	0	1/2
6	$\{38\}$	$\{13\}$	$p = 1/8$	1	1/2
7	$\{13\}$	$\{38, 39\}$	$p = 1/8$	0	1/2
8	0	0	1		1
9	$\{13\}$	$\{38, 39\}$	$p = 1/8$	0	1/2
10	$\{38\}$	$\{13\}$	$p = 1/8$	1	1/2
11	$\{13\}$	$\{39\}$	$p = 1/8$	0	1/2
	$\Omega_T = (0, \{39\})$				

The following conditional characteristic even has the same Ω_T as the previous one:

Round	output difference	input difference	p	icb	key fraction
	$\Omega_P = (\{7\}, 0)$				
1	$A = 0$	$a = 0$	1		1
2	$B = \{45\}$	$b = \{7\}$	$p = 1/8$	0	1/2
3	$C = \{6\}$	$c = \{45\}$	$p = 1/8$	1	1/2
4	$\{45\}$	$\{6, 7\}$	$p = 1/8$	0	1/2
5	0	0	1		1
6	$\{45\}$	$\{6, 7\}$	$p = 1/8$	0	1/2
7	$\{6\}$	$\{45\}$	$p = 1/8$	1	1/2
8	$\{45\}$	$\{7\}$	$p = 1/8$	0	1/2
9	0	0	1		1
8	$\{39\}$	$\{7\}$	$p = 1/8$	1	1/2
9	$\{7\}$	$\{39\}$	$p = 1/8$	1	1/2
	$\Omega_T = (0, \{39\})$				