New Improvements of Davies-Murphy Cryptanalysis

Sébastien Kunz-Jacques and Frédéric Muller

DCSSI Crypto Lab
51, boulevard de La Tour-Maubourg
75700 PARIS-07 SP
{Sebastien.Kunz-Jacques, Frederic.Muller}@sgdn.pm.gouv.fr

Abstract. In this paper, we revisit the famous Davies-Murphy cryptanalysis of DES. First we improve its complexity down to the analysis of 2^{45} chosen plaintexts, by considering 6 distributions instead of 7. The previous improvement of the attack by Biham and Biryukov costed 2^{50} known plaintexts. This new result is better than differential cryptanalysis but slightly worse than linear cryptanalysis. Secondly, we explore the link between this attack and other cryptanalysis techniques, in particular linear cryptanalysis.

1 Introduction

DES (Data Encryption Standard) is a popular encryption algorithm published in the late 70's by the American National Bureau of Standards (NBS) for governmental use [12]. DES is a block cipher encrypting blocks of data of length 64 bits under a secret key of length 56 bits. DES quickly became a popular cipher and is still widely used today. Although it has been replaced by the more recent AES [13], DES is still an attracting topic for cryptographers. Indeed 64-bit block algorithms remain in use in many cryptographic devices and the migration to AES is quite slow.

Given the large amount of research on the topic, DES has surprisingly well resisted to cryptanalysis. In practice, the best way of attacking DES is by brute force on the 56 bits of the key. This is feasible with large resources and can be achieved using a dedicated hardware or a large cluster of standard machines [7]. Another topic of analysis has been the research of shortcut attacks (faster than exhaustive search). Several results have been published since the early 90's :

- **Differential Cryptanalysis** [4] has been the first published theoretical cryptanalysis of DES. This technique, proposed by Biham and Shamir, requires to encrypt (under the same key) 2^{47} chosen plaintexts.
- **Linear Cryptanalysis** [11] was published shortly after by Matsui. It is slightly more efficient than Differential Cryptanalysis, since it requires about 2^{43} known plaintexts. This attack was implemented by Matsui and the experience was repeated afterwards and even slightly improved [8, 9, 15].

- **Bi-Linear Cryptanalysis** [5] was published recently at Crypto 2004. It is an extension of Linear Cryptanalysis using some particular quadratic approximations instead of linear ones. Its complexity is roughly the same as Linear Cryptanalysis and the two techniques appear to be closely related.
- **Davies-Murphy Cryptanalysis** [6] is a dedicated attack against DES. The starting point was the observation by Davies that adjacent pairs (and triplets) of S-boxes in DES produced unbalanced output. At first, it was believed the attack was slower than exhaustive search. However, in 1995, Biham and Biryukov [3] demonstrated how to improve these results. Their resulting attack costs 2^{50} known plaintexts, which is worse than Linear or Differential cryptanalysis, but still represents a theoretical break of DES.
- There exists other attacks like differential-linear attack or partitioning attacks.

In this paper, we propose a further improvement of the Davies-Murphy cryptanalysis. Our new attack requires to encrypt and process 2^{45} chosen plaintexts, in order to recover the secret key. Therefore our results place the attack between linear cryptanalysis and differential cryptanalysis in terms of complexity (see Table 1).

Also, our improved attack is very closely related to linear cryptanalysis (we use a biased linear combination of intermediate bits). It is already well known (with Biham's work [2] in particular) that Matsui's attack and Davies-Murphy attack are closely related. In Section 4, we further explore this relation in the general case. We prove that linear distinguishers become almost optimal after several convolutions, which explains the convergence observed between the complexities of both attacks. It also shows that Davies-Murphy cryptanalysis cannot significantly outperform linear cryptanalysis.

Table 1. Summary of Cryptanalysis of DES

Cryptanalysis Technique	Time Complexity	Data Complexity
Exhaustive Search	2^{56}	1 known plaintext
Linear Cryptanalysis [11]	2^{43}	2^{43} known plaintexts
Bi-Linear Cryptanalysis [5]	$\simeq 2^{43}$	$\simeq 2^{43}$ known plaintexts
Differential Cryptanalysis [4]	2^{47}	2^{47} chosen plaintexts
Davies-Murphy Cryptanalysis [3, 6]	2^{50}	2^{50} known plaintexts
This paper	2^{45}	2^{45} chosen plaintexts

2 DES and Davies-Murphy cryptanalysis

2.1 DES

DES [12] was published in 1977. It is a Feistel cipher (see Figure 1) with 16 rounds. DES operates on a 64-bit block of data, which is split in two halves of equal length.

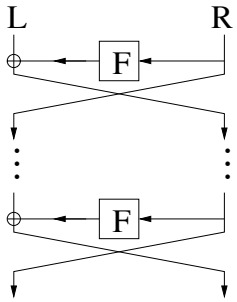

Fig. 1. General Structure of a Feistel cipher

The round function F of DES (also see Figure 2) first expands the state from 32 to 48 bits using a linear expansion E. Then a 48-bit subkey K is added bitwise to the state before a layer S of S-boxes is applied. This layer is built with 8 different S-boxes applied in parallel, each taking 6 input bits and producing 4 output bits. Therefore the layer S reduces the state size from 48 to 32 bits. Finally the state is permuted with a function P. Therefore

$$F(x) = P \circ S(K \oplus E(x))$$

Even though this round function is not bijective, the Feistel network remains invertible by construction. However a consequence of the non-invertibility is that for a given key, some outputs are produced more often than others by the round function F. This causes a natural imbalance in the cipher. The general idea of Davies-Murphy cryptanalysis is to take advantage of this property.

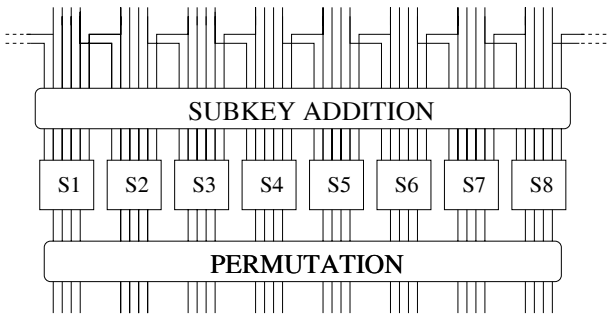

Fig. 2. The round function F of DES

2.2 Pairs of adjacent S-boxes

Any pair of adjacent S-box of DES "shares" two input bits (see Figure 2). To detail this phenomenon, we focus on the pair of S-boxes (S_1, S_2) and call (V_1, V_2) the corresponding outputs. We want to observe the distribution of (V_1, V_2) for a fixed key and a random round input.

The output of adjacent S-boxes is not balanced

Let $\{x_i\}_{i=1...32}$ be the round input bits and $\{k_i\}_{i=1...48}$ the bits of the subkey K. It directly follows from the specifications of DES that the input of S_1 - denoted $A = (a_1, ..., a_6)$ - is

$$A = (x_{32}, x_1, x_2, x_3, x_4, x_5) \oplus (k_1, k_2, k_3, k_4, k_5, k_6)$$

Similarly, the input of S_2 - denoted $B = (b_1, ..., b_6)$ - is

$$B = (x_4, x_5, x_6, x_7, x_8, x_9) \oplus (k_7, k_8, k_9, k_{10}, k_{11}, k_{12})$$

An important observation is that x_4 and x_5 are used twice : once in A and once in B. Suppose that the x_i's are random, then A and B are also random, except they have to verify the constraints :

$$a_5 \oplus b_1 = k_5 \oplus k_7 \tag{1}$$

$$a_6 \oplus b_2 = k_6 \oplus k_8 \tag{2}$$

Hence for a pair of adjacent S-boxes, like (S_1, S_2), the output distribution depends on two key bits $s = k_5 \oplus k_7$ and $t = k_6 \oplus k_8$.

The imbalance depends on 1 key bit only

DES S-boxes have a very particular form. Indeed, when the leftmost and rightmost input bits are fixed, each S_i performs a permutation of the remaining 4 input bits. A subtle consequence of this property is that the distribution of (V_1, V_2) does not depend on (s, t) but only on $s \oplus t$. In this section, we explain why this property is true.

Fix a target output called (z_1, z_2). Each z_i has exactly 4 preimages due to the row structure of the DES S-boxes. Hence there are 4 inputs of S_1 (one in each row of the S-box) such that $V_1 = z_1$. Similarly 4 inputs of S_2 yield $V_2 = z_2$. The total number of preimages of (z_1, z_2) is thus $4 \times 4 = 16$ where each solution is formed with an input of S_1 combined with an input of S_2. Let $N_{(s,t)}$ be the number among these 16 solutions that also satisfy the constraints (1) and (2) on s and t. Clearly,

$$N_{(0,0)} + N_{(0,1)} + N_{(1,0)} + N_{(1,1)} = 16 \tag{3}$$

For a fixed key, the probability $p(z_1, z_2)$ to obtain the output (z_1, z_2) is related to the quantity $N_{(s,t)}$ by the formula

$$p_{(s,t)}(z_1, z_2) = N_{(s,t)} \times 2^{-10}$$

Besides we can use symmetry arguments : since the bit a_6 is used to index the rows of the S-box S_1, it is well balanced among all preimages. So exactly half of the 4 S_1-preimages of z_1 satisfy $a_6 = 1$. Since all preimages of (z_1, z_2) are obtained by choosing independently a S_1-preimage of z_1 and a S_2-preimage of z_2, then $t = a_6 \oplus b_2$ is balanced among these 16 preimages and :

$$N_{(0,0)} + N_{(1,0)} = N_{(0,1)} + N_{(1,1)} = 8 \tag{4}$$

Using the same symmetry argument on the bit b_1 we see that

$$N_{(0,0)} + N_{(0,1)} = N_{(1,0)} + N_{(1,1)} = 8 \tag{5}$$

Putting together (4) and (5) we deduce :

$$N_{(0,0)} = N_{(1,1)}$$
$$N_{(0,1)} = N_{(1,0)}$$

Hence the output distribution of adjacent S-boxes depends only on the key-dependent bit k defined as

$$k = s \oplus t = k_5 \oplus k_6 \oplus k_7 \oplus k_8$$

An example

Two output distributions are therefore possible for (S_1, S_2) depending on the key-dependent bit k. Call $\mathcal{D}_0$ (resp. $\mathcal{D}_1$) the distribution corresponding to the case $k = 0$ (resp. $k = 1$). For instance $\mathcal{D}_1(z_1, z_2)$ is the probability that the output of (S_1, S_2) is (z_1, z_2) when $k = 1$.

The full distribution is represented in Table 2. It is interesting to notice that $\mathcal{D}_0$ and $\mathcal{D}_1$ are symmetric : they sum up to the uniform distributions. Denote by a single variable x the eight bits of (z_1, z_2). Then :

$$\frac{\mathcal{D}_0(x) + \mathcal{D}_1(x)}{2} = \frac{1}{256}$$

Hence, although the output is not balanced for a fixed key, it is globally balanced over all keys.

2.3 The resulting imbalance on 16 rounds

Since DES is a Feistel cipher, the XOR of plaintext and ciphertext is the XOR of 8 round outputs (see Figure 1). We focus on the output of adjacent S-boxes, like S_1 and S_2.[1] For these 8 bits, unbalanced distributions (like the one described in Table 2) are produced at each round. After XORing these outputs, the result is a **convolution** of several distributions of the form $\mathcal{D}_k$.

At first, one could expect the convolution of t output distributions to depend on t key-dependent bits, *i.e.* one bit per distribution. However it can easily be

[1] after the permutation P, the corresponding bits are $2, 9, 13, 17, 18, 23, 28$ and 31

z_2 \ z_1	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
01	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
02	2	2	4	6	4	4	6	4	6	4	0	4	4	2	6	6	6	6	4	2	4	4	2	4	2	4	8	4	4	6	2	2
03	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
04	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
05	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
06	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
07	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
08	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
09	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
10	6	6	4	2	4	4	2	4	2	4	8	4	4	6	2	2	2	2	4	6	4	4	6	4	6	4	0	4	4	2	6	6
11	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
12	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
13	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
14	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
15	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	Case $k = 0$																Case $k = 1$															

Table 2. Output distributions for (S_1, S_2). Values in the table should be divided by 1024

shown that only the parity of these t bits matters. For instance, consider the distribution $\mathcal{D}_1 \times \mathcal{D}_1$ obtained by the convolution of $\mathcal{D}_1$ with itself.

$$\begin{aligned}
\mathcal{D}_1 \times \mathcal{D}_1(x) &= \sum_a \mathcal{D}_1(a)\ \mathcal{D}_1(a \oplus x) \\
&= \sum_a \left(\frac{2}{256} - \mathcal{D}_0(a)\right)\left(\frac{2}{256} - \mathcal{D}_0(a \oplus x)\right) \\
&= \frac{4}{256} - \frac{2}{256} - \frac{2}{256} + \sum_a \mathcal{D}_0(a)\ \mathcal{D}_0(a \oplus x) \\
&= \sum_a \mathcal{D}_0(a)\ \mathcal{D}_0(a \oplus x) \\
&= \mathcal{D}_0 \times \mathcal{D}_0(x)
\end{aligned}$$

So it is equivalent to compose $\mathcal{D}_0$ with itself or $\mathcal{D}_1$ with itself. More generally only matters the parity of the t key-dependent bits involved. By extension, we simply denote $\mathcal{D}_0^t$ (resp. $\mathcal{D}_1^t$) the distribution after t convolutions when the parity bit is 0 (resp. 1). If an attacker can efficiently distinguish these two distributions, he learns one bit of information about the key. However, this analysis requires a large amount of pairs (plaintext, ciphertext) because distributions are almost uniform after a few convolutions.

2.4 Application to cryptanalysis

The problem of distinguishing two distributions is a classical topic in the literature, since it is related to many cryptanalysis problems (see [1] for example). In the particular case of DES, the problem is to distinguishing $\mathcal{D}_0^8$ from $\mathcal{D}_1^8$. One of these two distributions should be observed when XORing 8 appropriate bits from the plaintext and the ciphertext.

Davies and Murphy estimated in [6] the number of samples necessary to distinguish reliably theses 2 distributions. For several pairs of adjacent S-boxes,

these results are summarized in Table 3. The results depend highly on which pair is considered. In particular, (S_7, S_8) is the most favorable pair for the attack, although it falls short above the 2^{56} limit. Therefore it was first believed that Davies-Murphy cryptanalysis could not break DES.

Pair of S-boxes	$(1,2)$	$(2,3)$	$(3,4)$	$(4,5)$	$(5,6)$	$(6,7)$	$(7,8)$	$(8,1)$
Complexity	$2^{66.0}$	$2^{69.3}$	$2^{85.6}$	$2^{70.6}$	$2^{71.6}$	$2^{66.0}$	$2^{56.6}$	$2^{77.3}$

Table 3. Number of known plaintext needed for a 97% success rate

Later, further improvements of Davies-Murphy cryptanalysis have been proposed. Biham and Biryukov suggested to use 7 convolutioned distributions instead of 8. So their approximation no longer takes into account the full DES but only 15 rounds and accordingly an additional analysis is needed to handle the first (or last) round. The resulting attacks works by processing only 2^{50} known plaintexts, which is better than exhaustive search.

More recently, other extensions of Davies-Murphy Cryptanalysis were published. Pornin analyzed how to improve the resistance against the attack [14], and Kunz-Jacques *et al.* suggested to use the attack for side channel analysis [10].

3 Improving Davies-Murphy Cryptanalysis

In this section, we propose a new improvement of Davies-Murphy cryptanalysis. Our general idea is to use the convolution of only 6 distributions of round outputs (Davies and Murphy used 8 distributions [6], Biham and Biryukov only 7 distributions [3]). Therefore we approximate the behavior of only 13 rounds of DES. We take into account the 3 remaining rounds, but chosen plaintext is then needed, and several additional algorithmic tricks must be used.

3.1 General Framework

Like many statistical cryptanalysis, our attack is decomposed in three main phases.

- First we identify an internal object in the cipher that does not behave randomly. This statistical imbalance can be used to distinguish its behavior from a random one. Generally, such an object needs to be predictable from the plaintext, the ciphertext and eventually several key bits.
- Then we encrypt a large number of (chosen) messages and remember only a small part of information about each result. Typically, we store the number of occurrences of a small pattern of plaintext/ciphertext bits.

– Finally, we reconstruct the internal object from the collected data. This phase generally contains some partial exhaustive search and the statistical properties of the object are used as a stopping condition. Eventually we want to retrieve the secret key faster than exhaustive search.

3.2 The internal object

Davies-Murphy cryptanalysis targets the distribution of 8 bits from the round output, which are obtained from 2 adjacent S-boxes. After t convolutions, the resulting distribution is denoted $\mathcal{D}_0^t$ or $\mathcal{D}_1^t$ depending on the value of a key-dependent parity bit. Previous papers [3, 6] require to distinguish between these two distributions. Our attack has two important differences.

First we need to distinguish one of these two distributions (it does not matter whether the parity bit is 0 or 1 due to symmetry properties) from a uniform distribution. Secondly, to reduce the cost of the data collection, we propose to focus on the linear combination of these 8 bits with the strongest bias. Naturally, such a **linear distinguisher** cannot be more efficient than the **optimal distinguisher**, but it requires the storage of only 1 bit of information (instead of 8 bits) which turns out to be crucial for the data collection and data analysis phase.

Table 4 compares the samples needed by the optimal distinguisher and the best linear distinguisher for a fixed probability of success.

Pair of S-boxes		$(1,2)$	$(2,3)$	$(3,4)$	$(4,5)$	$(5,6)$	$(6,7)$	$(7,8)$	$(8,1)$
Opt. Dist.	$t=1$	$2^{4.4}$	$2^{4.1}$	$2^{6.8}$	$2^{4.7}$	$2^{5.4}$	$2^{5.1}$	$2^{4.2}$	$2^{5.7}$
Best. Lin. Dist.	$t=1$	2^{8}	$2^{8.83}$	$2^{10.83}$	$2^{8.83}$	$2^{8.83}$	2^{8}	$2^{6.83}$	$2^{9.66}$
Opt. Dist.	$t=6$	$2^{47.9}$	$2^{49.6}$	2^{62}	$2^{50.9}$	$2^{51.9}$	$2^{47.9}$	$2^{40.8}$	$2^{55.9}$
Best. Lin. Dist.	$t=6$	2^{48}	2^{53}	2^{65}	2^{53}	2^{53}	2^{48}	2^{41}	2^{58}
Opt. Dist.	$t=8$	2^{64}	$2^{67.3}$	$2^{83.6}$	$2^{68.6}$	$2^{69.6}$	2^{64}	$2^{54.6}$	$2^{75.3}$
Best. Lin. Dist.	$t=8$	2^{64}	$2^{70.6}$	$2^{86.6}$	$2^{70.6}$	$2^{70.6}$	2^{64}	$2^{54.6}$	$2^{77.3}$

Table 4. Comparison of several distinguishers for Davies-Murphy cryptanalysis

The complexities obtained are very similar for both distinguishers. This comparison is further developed in Section 4. Here we are interested by $t = 6$ and target the most favorable pair of S-box, *i.e.* (S_7, S_8). We computed that the best linear combination λ is

$$\lambda(X) = x_5 \oplus x_7 \oplus x_{12} \oplus x_{21} \oplus x_{22} \oplus x_{27} \oplus x_{32}$$

where $X = (x_1, \ldots, x_{32})$ is the output of the round function F. We have

$$Pr[\lambda(X) = 1] = 0.5\ (1 \pm 2^{-3.4}) = 0.5 \pm 0.046875$$

depending on the key. After 6 convolutions, we have

$$Pr[\lambda(X) = 1] = 0.5\ (1 \pm (2^{-3.4})^6) = 0.5\ (1 \pm 2^{-20.5})$$

The amount of data needed for the corresponding distinguisher is about 2^{41} samples.

3.3 The data collection

In the following we do not take into account the initial and the final permutation of DES. Let $(p_i)_{i\in 1,\ldots,64}$ denote the plaintext bits. The left branch of the plaintext is called $p_L = (p_1, \ldots, p_{32})$ and the right branch $p_R = (p_{33}, \ldots, p_{64})$. Similar notations are used for the ciphertext bits c_i. In this data collection phase, we encrypt n messages that verify

- The left branch of the plaintext p_L is chosen at random
- 14 bits of the right branch are also random : $(p_{50}, \ldots, p_{63})$. These bits are involved only in S-boxes S_5, S_6, S_7 and S_8.
- The 18 remaining plaintext bits are set to an arbitrary but constant value.

Given the degrees of freedom, n cannot exceed 2^{46}. For each encryption, we store the following piece of information

- The bit $\lambda(p_R) \oplus \lambda(c_R)$
- The 14 bits $(p_{50}, \ldots, p_{63})$ from the plaintext, which are involved in the S-boxes S_5, S_6, S_7 and S_8 of the first round.
- The 10 bits $(p_1, p_{24}, \ldots, p_{32})$ from the plaintext, which are involved in the S-boxes S_7 and S_8 of the second round.
- The 10 bits $(c_1, c_{24}, \ldots, c_{32})$ from the ciphertext, which are involved in the S-boxes S_7 and S_8 of the last round.

Hence we have a pattern of $1 + 14 + 10 + 10 = 35$ bits to store. For sake of efficiency, we only store the number of occurrences of each pattern in a table. This requires a table of size 2^{35}, where each entry in the table is a counter[2].

This data collection phase is detailed in Figure 3. X and Y denote two intermediate states in the right branch of the Feistel. U is the output of the 1-st round, V the output of the 2-nd round and W the output of the 16-th round. $X \oplus Y$ is the XOR of 6 round outputs so $\lambda(X \oplus Y)$ is not uniformly distributed according to the results of Section 3.2. However this object is not directly accessible. The purpose of storing these pieces of information about each message is to later predict $\lambda(X \oplus Y)$ in the data analysis phase.

[2] two bytes should be sufficient to store the counter, since each pattern occurs in average $2^{45} \times 2^{-35} = 1024$ times

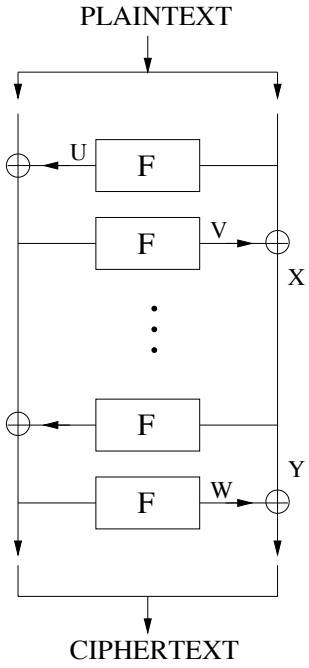

Fig. 3. Summary of the data collection phase

3.4 The data analysis

We want to predict $\lambda(X \oplus Y)$ from the data collected previously. For that purpose, we use the following relation :

$$\lambda(X \oplus Y) = \lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(V) \oplus \lambda(W) \tag{6}$$

Notation U_i, V_i and W_i is used to denote the bits from U, V and W.

The general idea of the attack is to perform an exhaustive search on a portion of the key bits. The pattern bits previously stored allow to determine the value of $\lambda(V)$ and $\lambda(W)$ in each case. Hence we determine all the terms involved in (6) and eventually predict how many times $\lambda(X \oplus Y)$ is equal to 1 among the samples. For the correct guess, this number should be significantly far from half of the samples.

Unfortunately, such a direct approach is way too expensive. Hence we need to decompose the attack in several steps. At each step, we only guess a few key bits, derive some intermediate information, and immediately get rid of what is no longer needed in the initial pattern.

Let us detail the first step. The starting point is the table built in the data collection phase. We refer to it as T_0. Guess the following 6 bits from the secret key : $(K_7, K_{21}, K_{22}, K_{39}, K_{53}, K_{63})$. They are XORed to the bits $(c_{28}, c_{29}, c_{30}, c_{31}, c_{32}, c_1)$ before S-box S_8 at round 16. Hence we can determine S_8's output and in particular the combination $W_5 \oplus W_{21} \oplus W_{27}$, which is a portion of the term $\lambda(W)$. After this step, 4 bits from the ciphertext are no longer needed. Thus we replace T_0 by a new table T_1 of size only 2^{31} where the number of occurrences of the following 31-bit pattern is stored :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus W_5 \oplus W_{21} \oplus W_{27}$
- The 14 bits $(p_{50}, \ldots, p_{63})$ from the plaintext
- The 10 bits $(p_1, p_{24}, \ldots, p_{32})$ from the plaintext
- The 6 bits $(c_{24}, \ldots, c_{29})$ from the ciphertext, which are involved in the S-box S_7 of the last round.

In the second step, we guess 6 additional key bits which are involved in S_7 at round 16 : $K_4, K_6, K_{23}, K_{28}, K_{29}, K_{46}$. Up to this point, 12 key bits have been guessed. Then we use the remaining 6 ciphertext bits in T_1 to predict $W_7 \oplus W_{12} \oplus W_{22} \oplus W_{32}$. Now we know all of $\lambda(W)$ and can get rid of all ciphertext bits. Hence we replace T_1 by a new table T_2 where the number of occurrences of the following 25-bit pattern is stored :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W)$
- The 14 bits $(p_{50}, \ldots, p_{63})$ from the plaintext
- The 10 bits $(p_1, p_{24}, \ldots, p_{32})$ from the plaintext

Similarly, the next steps of the analysis allow us to predict the term $\lambda(V)$ in relation (6). To that purpose, we first need to predict some bits of U. These steps are detailed in Appendix A.

Table 5 summarizes the successive steps of this data analysis phase. At each step, the complexity corresponds to the number of bits guessed multiplied by

Step	Key bits guessed	Total bits guessed	Old table	New table	Time complexity
0	-	0	-	2^{35}	2^{35}
1	$7, 21, 22, 39, 53, 63$	6	2^{35}	2^{31}	2^{41}
2	$4, 6, 23, 28, 29, 46$	12	2^{31}	2^{25}	2^{43}
3	$37, 54$	14	2^{25}	2^{23}	2^{39}
4	$5, 30, 47$	17	2^{23}	2^{19}	2^{40}
5	$15, 20, 38, 61$	21	2^{19}	2^{15}	2^{40}
6	$13, 14, 31, 45, 55, 62$	27	2^{15}	2^{11}	2^{42}
7	3 internal bits	30	2^{11}	2^{7}	2^{41}
8	4 internal bits	34	2^{7}	2^{1}	2^{41}

Table 5. Successive steps of the data analysis phase

the size of the table to manipulate. The maximal complexity reached during the analysis is of 2^{43}.

After step 8, we have guessed a total of 34 bits, among which 27 are directly key bits. So we know how many times $\lambda(X \oplus Y)$ is equal to 1 using the relation (6) and the content of table T_8. Then we can apply our statistical distinguisher to determine the correct guess among the $2^{34} - 1$ wrong guesses.

3.5 Finishing the attack

How to finish the attack depends on the exact probability of success of the linear distinguisher, and thus on the number of samples n. Generally one assumes that both distributions occur with the same probability. Then, the probability P_{fa} of false alarm (*i.e.* the probability that a wrong guess is identified as correct) is the same as the probability P_{nd} of non-detection of the correct key (*i.e.* the probability that a correct guess is identified as wrong). But here we need to identify one correct guess among $2^{34} - 1$ wrong guesses, so the crucial point is to have a low probability of false alarms. Therefore we propose several trade-offs. First, we set P_{nd} to 50%. Then we have $P_{fa} = \phi(\sqrt{d})$ where d is a parameter computed from the number of samples n (see Section 4 for more details) and ϕ is defined as

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-t} e^{-\frac{1}{2}u^2} du$$

Secondly, we set $P_{nd} = 15.86\%$. This gives $P_{fa} = \phi(\sqrt{d} - 1)$. Table 6 presents various numeric applications. The number of samples n cannot exceed 2^{46} because we do not have enough degrees of freedom. It is not possible to completely eliminate false alarms as P_{fa} is always greater than 2^{-34}. But false alarms can be discarded by guessing the remaining key bits and testing each candidate with

n	d	Case $P_a = P_{nd}$	$P_{nd} = 50\%$	$P_{nd} = 15.86\%$
2^{41}	1	30.85%	15.86%	50%
2^{42}	2	23.98%	7.86%	22.94%
2^{43}	4	15.86%	2.28%	15.86%
2^{44}	8	7.86%	$2^{-8.74}$	3.37%
2^{45}	16	2.28%	$2^{-14.95}$	$2^{-9.53}$
2^{46}	32	$2^{-8.74}$	$2^{-26.95}$	$2^{-19.25}$

Table 6. Probability of false alarm depending on n and the scenario

a couple (plaintext, ciphertext). Since 34 key bits are guessed in the core of the attack[3], there are only $56 - 34 = 22$ bits left to guess.

Suppose we pick $n = 2^{45}$ samples and fix the probability of non-detection to 50%, then the number of false alarms is

$$P_{fa} \times 2^{34} = 2^{19.05}$$

Guessing the remaining 22 bits brings the complexity up to $2^{41.05}$ candidates. One couple (plaintext, ciphertext) is then enough to identify the full secret key.

3.6 Summary

- The memory complexity of the attack is always the size of T_0, *i.e* a table containing 2^{35} entries of 2 bytes each.
- The time complexity of the attack is at least the complexity of the data analysis, *i.e.* 2^{43} steps of computation.
- The data complexity of the attack can range between $n = 2^{41}$ and $n = 2^{46}$ chosen plaintexts. In all cases, the key recovery is faster than exhaustive search, but the exact complexity depends on n.
- For example, when $n = 2^{45}$, the full secret key can be recovered with probability of 50% after 2^{41} trial DES encryptions. This is the trade-off we suggest to use.

4 Link between Davies-Murphy Cryptanalysis and Linear Cryptanalysis

It is known since Biham's work [2] that there exists an underlying linear attack with similar complexity as Davies-Murphy's attack. In this paper, we also use a biased linear combination of bits, in order to improve the Davies-Murphy attack.

[3] 7 are only intermediate bits, but they give a condition on a few key bits. Hence their entropy is equivalent to 7 key bits in practice

Therefore a natural question is to explain the link between both techniques, in the general case.

An important parameter is the data complexity ratio between the optimal distinguisher (used in the Davies-Murphy original cryptanalysis) and the best *linear* distinguisher for outputs of pairs of adjacent S-boxes. As seen in Table 4, the more rounds are applied, the closer the complexities are. In this section, we explain this phenomenon and account for the exact values of the ratios observed in Table 4. We show that, due to the effects of the convolutions, the same phenomenon will always be observed, independently of the original distribution. To some extent, this shows that linear cryptanalysis is always optimal.

4.1 Optimal vs Best Linear Distinguishers

Suppose we have a random variable X that follows a distribution $\mathcal{D}$ or the uniform distribution $\mathcal{U}$. (in the Davies-Murphy case, $\mathcal{D} = \mathcal{D}_0^t$ or $\mathcal{D}_1^t$ for some t). Let $\mathcal{S} = \{0, \ldots, 2^n\}$ be the image set of X. Our goal is to distinguish between theses two distributions. Basically, there are two approaches : we can use the best (optimal) distinguisher, or we can restrict the analysis to linear distinguishers only.

Optimal Distinguisher It is well known (see [1] for instance) that the optimal distinguisher between $\mathcal{D}$ and $\mathcal{U}$ has probability of error

$$P_e = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{-\frac{\sqrt{d}}{2}} e^{-\frac{1}{2}u^2} du$$

when the number of samples n is related to the parameter d by

$$n = \frac{d}{\Delta(\mathcal{D})}$$

and $\Delta(\mathcal{D})$ is the Squared Euclidean Imbalance (SEI) of $\mathcal{D}$ from $\mathcal{U}$. If for any $x \in \mathcal{S}$, $\mathcal{D}(x)$ denotes the probability that $X = x$, the SEI is computed as

$$\Delta(\mathcal{D}) = |\mathcal{S}| \sum_x \left(\mathcal{D}(x) - \frac{1}{2^n} \right)^2$$

Linear Distinguisher Consider a linear combination $\lambda(X)$ of the bits of X. Suppose that, when X follows $\mathcal{D}$, it satisfies :

$$Pr_{\mathcal{D}}[\lambda(X) = 1] = \frac{1}{2}(1 + \varepsilon)$$

then it is well known that about $n = \varepsilon^2$ samples are needed to detect this bias. We introduce the usual notation

$$LP(\lambda) = (Pr_{\mathcal{D}}[\lambda(X) = 1] - Pr_{\mathcal{D}}[\lambda(X) = 0])^2 = \varepsilon^2$$

The question is to determine the $LP_{max} = \max_\lambda\{LP(\lambda)\}$ of the best linear distinguisher for a given distribution $\mathcal{D}$. By definition, it requires more data than the optimal distinguisher, but we are interested into the ratio between the two complexities.

Relation between $\Delta(\mathcal{D})$ and LP_{max} Using the Fourier transform (see Section 2.4 of [1]), one shows that

$$\Delta(\mathcal{D}) = \sum_{\lambda \neq 0} LP(\lambda) \tag{7}$$

Therefore we can derive the following bound for the ratio between the two data complexities :

$$LP_{max} \leq \Delta(\mathcal{D}) \leq (2^n - 1)LP_{max}$$

It can be shown that both bounds are actually tight, so the best linear distinguisher can be significantly worse (up to a factor of 2^n) than the optimal distinguisher. However, in Davies-Murphy cryptanalysis, we are dealing with particular distributions.

4.2 The case of Davies-Murphy cryptanalysis

The target distribution $\mathcal{D}_i^t$ in this case is obtained after t convolutions. In practice, when t grows, the ratio apparently gets small (see Table 4). In this Section, we explain the ratios observed. Since linear biases are just multiplied after each convolution, (7) can be re-expressed as :

$$\Delta(\mathcal{D}_i^t) = \sum_{\lambda \neq 0} LP(\lambda)^t \tag{8}$$

where $LP(\cdot)$ are computed with respect to the base distribution $\mathcal{D}_i$ (by symmetry it does matter whether the parity bit i is 0 or 1).

Suppose now that there are $m \leq 2^n - 1$ linear forms whose LP is equal to LP_{max}, and that all other λ are such that

$$LP(\lambda) \leq \alpha\, LP_{max}$$

for some $0 \leq \alpha < 1$. Then (8) yields

$$m\,(LP_{max})^t \leq \Delta(\mathcal{D}_i^t) \leq (m + \alpha^t(2^n - 1 - m))\,(LP_{max})^t$$

When t is big enough, then $\alpha^t \ll 1$ and

$$\Delta(\mathcal{D}_i^t) \simeq m\,(LP_{max})^t \tag{9}$$

We can compute LP_{max} and m in the case of DES. These results are summarized in Table 7.

Pair of S-boxes		$(1,2)$	$(2,3)$	$(3,4)$	$(4,5)$	$(5,6)$	$(6,7)$	$(7,8)$	$(8,1)$
m		1	10	8	4	2	1	1	4
$\log_2(m)$		0	3.3	3	2	1	0	0	2
LP_{max}		2^{8}	$2^{8.83}$	$2^{10.83}$	$2^{8.83}$	$2^{8.83}$	2^{8}	$2^{6.83}$	$2^{9.66}$
Opt. Dist. $\Delta(\mathcal{D}^6)$	$t=6$	$2^{47.9}$	$2^{49.6}$	2^{62}	$2^{50.9}$	$2^{51.9}$	$2^{47.9}$	$2^{40.8}$	$2^{55.9}$
Best. Lin. Dist. LP^6_{max}	$t=6$	2^{48}	2^{53}	2^{65}	2^{53}	2^{53}	2^{48}	2^{41}	2^{58}
Expected value from (9)	$t=6$	$2^{47.9}$	$2^{52.9}$	2^{65}	$2^{52.9}$	$2^{52.9}$	$2^{47.9}$	$2^{40.8}$	$2^{57.9}$
Opt. Dist. $\Delta(\mathcal{D}^8)$	$t=8$	2^{64}	$2^{67.3}$	$2^{83.6}$	$2^{68.6}$	$2^{69.6}$	2^{64}	$2^{54.6}$	$2^{75.3}$
Best. Lin. Dist. LP^8_{max}	$t=8$	2^{64}	$2^{70.6}$	$2^{86.6}$	$2^{70.6}$	$2^{70.6}$	2^{64}	$2^{54.6}$	$2^{77.3}$
Expected value from (9)	$t=8$	2^{64}	$2^{70.6}$	$2^{86.6}$	$2^{70.6}$	$2^{70.6}$	2^{64}	$2^{54.6}$	$2^{77.3}$

Table 7. Difference Between Optimal and Linear Distinguisher Explained

In practice, the approximation of equation (9) accurately predicts the maximum linear bias and the loss between optimal and linear distinguishers. The weakest couples of DES S-boxes w.r.t. linear distinguishers are the ones that have a small number of linear forms reaching the maximum bias LP_{max}. For the best pair of S-boxes (S_7, S_8), LP_{max} is only reached once, so the ratio between both distinguishers is almost 1 after 6 convolutions. Hence, replacing the optimal distinguisher with the best linear one does not result in a significant deterioration.

4.3 Summary

A consequence of the convolutions involved in Davies-Murphy cryptanalysis is that distributions become very quickly "smooth". Therefore the complexity of the optimal distinguisher can increase very quickly after several rounds, while the complexity of a linear distinguisher increases more regularly.

Hence, using a linear distinguisher becomes almost optimal after several convolutions. This explains the phenomenon that Biham observed in [2] and it also explains why we obtained good results in this paper, while restricting our analysis to linear distinguishers. This observation is independent of the initial distribution, so it would make no difference if used other S-boxes for instance. However, the linear characteristic used in our attack has some nice properties :

- it is iterative
- it uses only output bits of the round function
- the same linear form is used at every round

These properties allow us to concentrate on one half of the Feistel network, reducing the effective number of rounds to consider down from 16 to 8 (algorithmic

tricks further reduce this number to 6). Therefore, although this linear characteristic is not the best one known for DES, its a particular form may be helpful to optimize the data analysis phase.

5 Conclusion

In this paper, we improve the famous Davies-Murphy cryptanalysis of DES, by using 6 round output distributions (instead of 7 or 8 like in previous papers on the topic [3, 6]). Several trade-offs are possible, but we describe a key-recovery attack with complexity of 2^{45} chosen plaintexts. This positions the attack at the second rank of cryptanalysis of DES : slightly better than Biham and Shamir's differential cryptanalysis but slightly worse than Matsui's linear cryptanalysis.

In addition, we have shown that using linear distinguishers for the Davies-Murphy cryptanalysis was almost an optimal choice, because of the particular structure of the attack. Therefore Davies-Murphy cryptanalysis is closely related to a particular family of linear attacks, where the linear mask involves only the round output. This allows for efficient optimizations of the data collection and data analysis. At the same time, it shows that it is unlikely to (significantly) outperform Matsui's attack with further algorithmic improvements.

References

1. T. Baignères, P. Junod, and S. Vaudenay. How Far Can We Go Beyond Linear Cryptanalysis? In P-J. Lee, editor, *Advances in Cryptology – Asiacrypt'04*, volume 3329 of *Lectures Notes in Computer Science*, pages 432–450. Springer, 2004.
2. E. Biham. On Matsui's Linear Cryptanalysis. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt'94*, volume 950 of *Lectures Notes in Computer Science*, pages 341–355. Springer, 1995.
3. E. Biham and A. Biryukov. An Improvement of Davies' Attack on DES. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt'94*, volume 950 of *Lectures Notes in Computer Science*, pages 461–467. Springer, 1995.
4. E. Biham and A. Shamir. Differential Cryptanalysis of the Full 16-round DES. In E.F. Brickell, editor, *Advances in Cryptology – Crypto'92*, volume 740 of *Lectures Notes in Computer Science*, pages 487–496. Springer, 1992.
5. N. Courtois. Feistel Schemes and Bi-linear Cryptanalysis. In M. Franklin, editor, *Advances in Cryptology – CRYPTO'04*, volume 3152 of *Lectures Notes in Computer Science*, pages 23–40. Springer, 2004.
6. D. Davies and S. Murphy. Pairs and Triplets of DES S-Boxes. *Journal of Cryptology*, 8(1):1–25, 1995.
7. DES Cracker Project. See `http://www.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/`.
8. P. Junod. On the Complexity of Matsui's Attack. In S. Vaudenay and A. Youssef, editors, *Selected Areas in Cryptography – 2001*, volume 2259 of *Lectures Notes in Computer Science*, pages 199–211. Springer, 2001.
9. L. Knudsen and J-E. Mathiassen. A Chosen-Plaintext Linear Attack on DES. In B. Schneier, editor, *Fast Software Encryption – 2000*, volume 1978 of *Lectures Notes in Computer Science*, pages 262–272. Springer, 2001.

10. S. Kunz-Jacques, F. Muller, and F. Valette. The Davies-Murphy Power Attack. In P.-J. Lee, editor, *Advances in Cryptology – Asiacrypt'04*, volume 3329 of *Lectures Notes in Computer Science*, pages 451–467. Springer, 2004.
11. M. Matsui. Linear Cryptanalysis Method for DES Cipher. In T. Helleseth, editor, *Advances in Cryptology – Eurocrypt'93*, volume 765 of *Lectures Notes in Computer Science*, pages 386–397. Springer, 1993.
12. National Bureau of Standards (NBS), U.S. *FIPS 46, "Data Encryption Standard", Federal Information Processing Standards Publication 46*, 1977.
13. National Institute of Standards and Technology (NIST). Advanded Encryption Standard (AES) FIPS Publication 197, November 2001. Available at `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.
14. T. Pornin. Optimal Resistance Against the Davies and Murphy Attack. In K. Ohta and D. Pei, editors, *Advances in Cryptology – Asiacrypt'98*, volume 1514 of *Lectures Notes in Computer Science*, pages 148–159. Springer, 1998.
15. T. Shimoyama and T. Kaneko. Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES. In H. Krawczyk, editor, *Advances in Cryptology – Crypto'98*, volume 1462 of *Lectures Notes in Computer Science*, pages 200–211. Springer, 1998.

A Detailed steps of the data analysis phase

A.1 Step 3

In the step number 3, we guess the key bits involved in S_5 at the first round. Luckily, 4 of these bits $(K_4, K_{22}, K_{28}, K_{39})$ are already known. Thanks to the key scheduling properties, only K_{37} and K_{54} need to be guessed. We know the plaintext bits involved in S_5 (part of it are arbitrary constants, the rest is contained in the pattern of table T_2). So we can predict S_5's output and in particular the bit U_{24}. 2 plaintext bits are no longer needed and the new table T_3 contains the number of occurrences of the 23-bit pattern formed by :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W)$
- The 12 bits $(p_{52}, \ldots, p_{63})$ from the plaintext
- The 9 bits $(p_1, p_{25}, \ldots, p_{32})$ from the plaintext
- The intermediate bit $p_{24} \oplus U_{24}$

A.2 Step 4

In the step number 4, we guess the key bits involved in S_6 at the first round. Luckily, 3 of these bits (K_{23}, K_{29}, K_{53}) are already known. Thanks to the key scheduling properties, only K_5, K_{30} and K_{47} need to be guessed. We predict S_6's output and in particular the bits U_{27} and U_{32}. 4 plaintext bits are no longer needed and the new table T_4 contains the number of occurrences of the 19-bit pattern formed by :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W)$
- The 8 bits $(p_{56}, \ldots, p_{63})$ from the plaintext
- The 7 bits $(p_1, p_{25}, p_{26}, p_{28}, p_{29}, p_{30}, p_{31})$
- The 3 intermediate bits $(p_{24} \oplus U_{24}, p_{27} \oplus U_{27}, p_{32} \oplus U_{32})$

A.3 Step 5

In the step number 5, we guess the key bits involved in S_7 at the first round. Luckily, 2 of these bits (K_{21}, K_{63}) are already known. Thanks to the key scheduling properties, only K_{15}, K_{20}, K_{38} and K_{61} need to be guessed. We predict S_7's output and in particular the bit U_{30}. 4 plaintext bits are no longer needed and the new table T_5 contains the number of occurrences of the 15-bit pattern formed by :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W)$
- The 4 bits $(p_{60}, \ldots, p_{63})$ from the plaintext
- The 6 bits $(p_1, p_{25}, p_{26}, p_{28}, p_{29}, p_{31})$
- The 4 intermediate bits $(p_{24} \oplus U_{24}, p_{27} \oplus U_{27}, p_{30} \oplus U_{30}, p_{32} \oplus U_{32})$

A.4 Step 6

In the step number 6, we guess the key bits involved in S_8 at the first round. Hence we need to guess $K_{13}, K_{14}, K_{31}, K_{45}, K_{55}$ and K_{62}. Then we predict S_8's output and in particular the bit U_{25}. 4 plaintext bits are no longer needed and the new table T_6 contains the number of occurrences of the 11-bit pattern formed by :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W)$
- The 5 bits $(p_1, p_{26}, p_{28}, p_{29}, p_{31})$
- The 5 intermediate bits $(p_{24} \oplus U_{24}, p_{25} \oplus U_{25}, p_{27} \oplus U_{27}, p_{30} \oplus U_{30}, p_{32} \oplus U_{32})$

A.5 Step 7

In the step number 7, we guess the missing input bits of S-box S_7 at the second round. The actual input is

$$(p_{24} \oplus U_{24}, \ldots, p_{29} \oplus U_{29}) \oplus (K_{53}, K_{13}, K_{30}, K_{55}, K_6, K_{11})$$

Thanks to the key scheduling properties, we already know 4 of these key bits. Besides we already know 3 intermediate bits of the form $p_i \oplus U_i$. The missing U_i's are not known but they depend only on the key and the fixed plaintext bits, so their value is the same for all samples. So we can guess the 3 bits $(U_{26}, U_{28} \oplus K_6, U_{29} \oplus K_{11})$ and predict S_7's output. Then, we determine $V_7 \oplus V_{12} \oplus V_{22} \oplus V_{32}$. The new table T_7 contains the number of occurrences of the 7-bit pattern formed by :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W) \oplus V_7 \oplus V_{12} \oplus V_{22} \oplus V_{32}$
- The 4 bits $(p_1, p_{28}, p_{29}, p_{31})$
- The 2 intermediate bits $(p_{30} \oplus U_{30}, p_{32} \oplus U_{32})$

A.6 Step 8

In the step number 8, we guess the missing input bits of S-box S_8 at the second round. Thanks to the key scheduling properties, all key bits involved ($K_5, K_6, K_{23}, K_{37}, K_{47}$ and K_{54}) are already known. Hence we just need to guess the 4 missing input bits : $U_1, U_{28}, U_{29}, U_{31}$. in order to predict S_8's output and in particular $V_5 \oplus V_{21} \oplus V_{27}$. Hence we know the value of $\lambda(V)$. The new table T_8 contains the number of occurrences of the bit :

- The bit $\lambda(p_R) \oplus \lambda(c_R) \oplus \lambda(W) \oplus \lambda(V)$