The Davies-Murphy Power Attack

Sébastien Kunz-Jacques, Frédéric Muller, and Frédéric Valette

DCSSI Crypto Lab 51, Boulevard de Latour-Maubourg
75700 Paris 07 SP France
{Sebastien.Kunz-Jacques, Frederic.Muller,
Frederic.Valette}@sgdn.pm.gouv.fr

Abstract. In this paper, we introduce a new power analysis attack against DES. It is based on the well known Davies-Murphy attack. As for the original attack, we take advantage of non-uniform output distributions for two adjacent S-boxes. We show how to detect these biased distributions by power analysis on any DES inner round and thus obtain one bit of information about the key.
An advantage of this new attack is that no information about DES inputs or outputs is required. Therefore it is likely to defeat many actual countermeasures, in particular the popular masking techniques.

1 Introduction

Side-channel attacks have been developed in parallel to "classical" attack techniques since about 10 years. The initial publication by Kocher [13, 14] of Simple Power Analysis (SPA) and Differential Power Analysis (DPA) has been a major breakthrough in the domain. The general idea in this new family of attacks is to use "non-conventional" sources of information. Typically, the situation is we have a cryptographic device manipulating secret key or data which is protected against physical intrusion (we can think of this device as a smart-card, for instance). Then an attacker tries to obtain these secrets by measuring some external elements of information about the device. A leakage can result from the electric consumption of the device, its electromagnetic radiations, or simply by timing measurements. Some related attacks are also based on analyzing faults during the execution of the cryptographic computations [8].

Side channel attacks using the electric consumption are generally called "Power Attacks". It is widely believed that power consumption is always somehow correlated to the manipulated data. The question is thus to find appropriate countermeasures in order to thwart all known attacks. Power Attacks have been developed without distinction to secret and public key primitives. However in this paper, we mostly focus on the analysis of block ciphers. In this particular context, the most popular family of attacks are DPA [14] and its extended version, Higher-Order DPA [17, 21]. Advanced attacks usually revisit some techniques of "classical" cryptanalysis, like collision attacks [20] or differential attacks [15].

The goal of this paper is to propose a new power attack. We revisit the well-known Davies-Murphy cryptanalysis of DES [5, 11] and transform it into a power analysis attack. The "classical" attack uses non-uniform output distributions for each pair of adjacent S-boxes in DES. This property results from the duplication of some state bits by the expansion function. Non-uniform distributions result in detectable imbalance in electric consumption, and we propose several techniques to detect and exploit this imbalance. We call our new attack the Davies-Murphy Power Attack (DMPA).

First we discuss the model used to describe the correlation between intermediate data and power consumption. Then we recall the principles of DES, the Davies-Murphy attack and investigate some additional properties. In Section 5, the general principle of DMPA is exposed and we propose some tricks to apply it to various scenarios and different kinds of implementation. The final sections are dedicated to discussing the advantages and the extensions of DMPA.

2 The Power Consumption Model

In power analysis attacks, the basic assumption is that power consumption is somehow correlated with some data handled during the execution of an instruction. A classical assumption is the **Hamming weight model** [1, 9] where we suppose that the power consumption is proportional to the hamming weight of the manipulated data D. Let W be the power consumption and H the hamming weight function. We suppose that

$$W = \lambda\, H(D \oplus R) + \theta$$

where θ is a term of noise, λ a scalar and R a reference state from which we measure the number of bits flipped. For instance, R is often seen as a constant, unknown machine word (but R is not necessarily zero). The underlying assumption for electric consumption is that flipping a bit from 0 to 1 or flipping it from 1 to 0 costs almost the same thing, while keeping a bit unchanged costs almost nothing.

Many papers on side channel attacks [7, 10, 14, 18] observed empirically this correlation between the consumption of a smart card and the hamming weight of the operands. This model has also been verified more formally (see [9] for instance). Although a finer analysis has revealed that an extended linear model was sometimes more appropriate [1], it is still widely believed in practice that the Hamming weight model is a reasonable approximation. Actually it seems particularly well suited to model circuits based on the widely used CMOS technology, while it may be less appropriate for other technologies.

In the following, we suppose that the Hamming weight model is verified. We stress out that this model is not specifically helpful for our attack. We choose it because it is frequently used in the literature, and from our experience of cryptographic hardware, we believe it is very often appropriate. However our attack could probably be adapted to another model, as long as an actual correlation exists between W and D.

It is classically known that implementations can be subject to power analysis attacks when one of the following condition holds :

- the intermediate data D depends only on the plaintext and a small portion of key bits. This is the fundamental hypothesis for *Differential Power Analysis (DPA)*.
- a simple function of several intermediate data $D_1, \ldots, D_t$ depends only on the plaintext and a small portion of key bits. This is the fundamental hypothesis for *Higher-Order Differential Power Analysis (HO-DPA)* [1].

Then an attacker would use the correlation between intermediate data and power consumption to detect a correct guess of the key bits. Recent implementations take into account this threat by protecting all inner instructions. For instance a popular family of countermeasures consists in masking the manipulated data [2–4]. The underlying idea is that intermediate values should look random, even when the plaintext is known. However, most countermeasures do not take into account the fact that intermediate data may be biased **independently of the plaintext**. In the case of DES, this is actually the case because of Davies' observation about pairs of adjacent S-boxes [11]. In the next section, we focus on DES and recall the well-known Davies-Murphy attack.

3 DES and the Davies-Murphy Attack

The Data Encryption Standard (DES) is one of the most popular block cipher. Since it was selected as a standard by the NBS in 1977 [19], it has been the target of many research on cryptanalysis. Among all the results against DES, three attacks have emerged :

- Differential Cryptanalysis (DC) [6] was proposed by Biham and Shamir in 1990. It has been a major breakthrough and many applications to other algorithms have been demonstrated thereafter. Since then, it was revealed that the principle of DC was already known by the designers of DES.
- Linear Cryptanalysis (LC) [16] was proposed by Matsui in 1993. Like DC, it became quickly very popular and was applied successfully to other algorithms. In addition, this attack was practically implemented by Matsui in the case of DES. This technique was presumably not known by the designers of DES.
- The Davies-Murphy Cryptanalysis [5, 11] is a dedicated attack against DES. It takes advantage of biased distributions for two adjacent S-boxes. Although less generic than the previous two, Davies-Murphy cryptanalysis is a concern for Feistel ciphers with a non-bijective round function.

First we remind the general structure of DES (see Figure 1). We call F the round function, iterated 16 times in this case.

F is represented in more details in Figure 2. The general idea of Davies-Murphy attack is to look at two adjacent S-boxes (say S_1 and S_2). Because of the expansion phase, two bits of the input have been duplicated and are shared

[1] Here it is t-th order DPA, since t intermediate data are considered

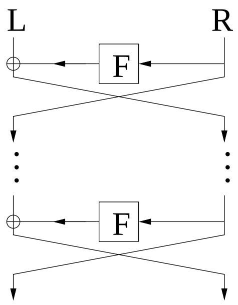

Fig. 1. The general structure of DES

by the inputs of S_1 and S_2. These two bits are the two rightmost bits of S_1 and the two leftmost bits of S_2. Consequently the output distributions for S_1 and for S_2 are not independent. A precise analysis shows that the joint distribution is not uniform. Moreover, depending on one key bit[2], two distributions (both non uniform) can be observed. Theoretically this allows an attacker to learn the sum of the 4 key bits corresponding to the shared positions in S_1 and S_2 (see [5, 11]).

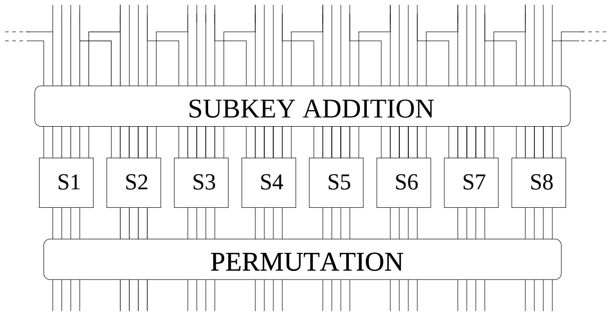

Fig. 2. The round function of DES

To give an illustration of the Davies-Murphy biased distributions, we focus on the S-boxes S_1 and S_2. We denote by (k_1, k_2, k_3, k_4) the 4 subkey bits corresponding to the "shared" positions of S_1 and S_2, and we call $k = k_1 \oplus k_2 \oplus k_3 \oplus k_4$ the sum of these 4 bits. In Table 1 we represent the output distributions for both cases $k = 0$ and $k = 1$. y_1 and y_2 represent respectively the outputs of S_1 and S_2. These distributions were simply obtained by looping on all possible inputs of S_1 and S_2.

[2] Actually, it is one linear combination of key bits

Case $k = 0$

y_2 \ y_1	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
01	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
02	2	2	4	6	4	4	6	4	6	4	0	4	4	2	6	6
03	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
05	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
06	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
08	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
09	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
11	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
12	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
13	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
14	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
15	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4

Case $k = 1$

y_2 \ y_1	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
01	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
02	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
04	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
05	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
07	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
08	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
10	2	2	4	6	4	4	6	4	6	4	0	4	4	2	6	6
11	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
12	3	3	4	5	4	4	5	4	5	4	2	4	4	3	5	5
13	5	5	4	3	4	4	3	4	3	4	6	4	4	5	3	3
14	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

Table 1. Biased Distributions for S_1 and S_2 (all elements in the table should be divided by 2^{10})

This kind of imbalance was initially observed by Davies [11]. At first, it was thought that the attack could not be extended to the full DES. Indeed the previous observation extends to 16 rounds by composing 8 times the distributions. The XOR of plaintext and ciphertext is therefore non uniform and it turns out things depend only on one combination of key bits. Unfortunately the resulting imbalance is too small to be detected. Later on, Biham and Biryukov demonstrated how to improve this attack to obtain an attack faster than exhaustive search for the full DES [5]. In this paper we focus on Davies-Murphy's biased distributions for just one round.

4 Extension of Davies-Murphy to the Hamming Weight

The key observation of Davies-Murphy attack is that, for any DES inner round, intermediate data are not distributed uniformly, for randomly-chosen inputs. However in a power attack we do not have access directly to the intermediate data but to the power consumption (which is hopefully correlated to the data). Since we assume the Hamming weight model, this correlation depends on the Hamming weight of the S-box output. Hence it is natural to consider how the Davies-Murphy property translates to the Hamming weight.

As a first example, we consider the S-boxes S_1 and S_2 and look at the joint distribution of $(h_1, h_2) = (\mathrm{H}(S_1(x_1)), \mathrm{H}(S_2(x_2)))$ where x_1 and x_2 are uniformly chosen. The resulting distribution is given in Table 2.

Four values are biased in Table 2 (the corresponding positions are $(h_1, h_2) = (0, 2)$, $(4, 2)$, $(0, 3)$ and $(4, 3)$). Hence the imbalance exists but is not huge. Still, we hope to make it exploitable but we need to introduce appropriate statistical tools.

Random Distribution						Case $k = 0$						Case $k = 1$					
h_2 \ h_1	0	1	2	3	4	h_2 \ h_1	0	1	2	3	4	h_2 \ h_1	0	1	2	3	4
0	4	16	24	16	4	0	4	16	24	16	4	0	4	16	24	16	4
1	16	64	96	64	16	1	16	64	96	64	16	1	16	64	96	64	16
2	24	96	144	96	24	2	26	96	144	96	22	2	22	96	144	96	26
3	16	64	96	64	16	3	14	64	96	64	18	3	18	64	96	64	14
4	4	16	24	16	4	4	4	16	24	16	4	4	4	16	24	16	4

Table 2. Distributions of output hamming weight for S_1 and S_2 (all elements in the table should be divided by 2^{10})

Definition 1. *Let $\mathcal{D}_1$, $\mathcal{D}_2$ be two distributions over some finite domain X. The **statistical distance** between $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined as*

$$|\mathcal{D}_1 - \mathcal{D}_2| = \sum_{x \in X} |\mathcal{D}_1(x) - \mathcal{D}_2(x)|$$

Using this definition, we can compute the statistical distance between the previous distributions. Let $\mathcal{U}$ be the distribution of hamming weight for uniformly chosen inputs. $\mathcal{D}_i$ denotes the distribution in the case $k = i$. For S-boxes S_1 and S_2, we can easily compute :

$$|\mathcal{D}_0 - \mathcal{U}| = \frac{1}{128}$$
$$|\mathcal{D}_1 - \mathcal{U}| = \frac{1}{128}$$
$$|\mathcal{D}_1 - \mathcal{D}_0| = \frac{1}{64}$$

The imbalance for S-boxes S_1 and S_2 is not the best we can obtain. We repeated the same experience with different pairs of S-box and obtained better results. This is summarized in Table 3.

When using random inputs, all pairs of adjacent S-boxes present an imbalance regarding the output hamming weight. The best ones are obtained for (S_2,S_3), (S_7,S_8) and (S_8,S_1). One can also notice that

$$|\mathcal{D}_0 - \mathcal{U}| = |\mathcal{D}_1 - \mathcal{U}| = 0.5 \times |\mathcal{D}_1 - \mathcal{D}_0|$$

always holds due to the symmetry property :

$$\mathcal{D}_0(x) + \mathcal{D}_1(x) = 2 \cdot \mathcal{U}(x)$$

Therefore we have exhibited a Hamming weight version of the Davies-Murphy imbalance on DES, and we are confident that the electric consumption for adjacent S-boxes is biased, even **for randomly-chosen plaintexts**.

S-boxes	$\|\mathcal{D}_0 - \mathcal{U}\|$	$\|\mathcal{D}_1 - \mathcal{U}\|$	$\|\mathcal{D}_1 - \mathcal{D}_0\|$
S_1 and S_2	$\frac{1}{128}$	$\frac{1}{128}$	$\frac{1}{64}$
S_2 and S_3	$\frac{3}{64}$	$\frac{3}{64}$	$\frac{3}{32}$
S_3 and S_4	$\frac{1}{256}$	$\frac{1}{256}$	$\frac{1}{128}$
S_4 and S_5	$\frac{1}{128}$	$\frac{1}{128}$	$\frac{1}{64}$
S_5 and S_6	$\frac{1}{128}$	$\frac{1}{128}$	$\frac{1}{64}$
S_6 and S_7	$\frac{3}{128}$	$\frac{3}{128}$	$\frac{3}{64}$
S_7 and S_8	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$
S_8 and S_1	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{1}{16}$

Table 3. Statistical distance between distributions $\mathcal{U}$, $\mathcal{D}_0$ and $\mathcal{D}_1$

5 The Davies-Murphy Power Attack

In this section, we want to turn the imbalance of Hamming weight into a powerful side channel attack against DES. First, we need to specify which specific assumptions we make about the power consumption of the cryptographic device.

5.1 Assumptions

As mentioned previously, our general assumption is the Hamming weight model. However, before describing an attack, we need to precise more specifically this model.

A first and crucial question is to determine what the reference state R corresponds to in practice. In [9], some experiments were conducted on different hardwares to answer this question. Depending on the chips, different results were obtained. In many cases, R corresponded to the address of the input value or to the opcode of the current instruction. For other chips, R was always 0, presumably because these chips clear the bus between each instruction. Overall it is reasonable to consider that each instruction corresponds to a unique constant R.

More formally, we make the assumption that **there is a constant R_i, independent of the round, such that the electric consumption W_i of S-box** S_i is

$$W_i = \lambda \, \mathrm{H}(y_i \oplus R_i) + \theta$$

with the same notations than in Section 3.

Moreover, we suppose that **all S-box computations are done separately**, hence we can observe any W_i separately by looking at an appropriate portion of the power consumption curves. This assumption is reasonable, but may be subject to discussions, depending on the implementation. Indeed some computations might be done in parallel (for instance, on a 8-bit architecture, it is likely that pairs of adjacent S-boxes are executed simultaneously, thus we could observe only $W_{2i} + W_{2i-1}$).

We further explore these different scenarios in Section 6. Here, we explore only the case where all S-boxes are computed sequentially. This is convenient to describe a basic attack.

5.2 The principle of the attack

We have already seen that $(H(y_i), H(y_{i+1}))$ is biased *a priori* for random plaintext, depending just on one key bit k. Actually what we observe also depends on the unknown constants R_i, R_{i+1} and on the noisy term θ. The general idea of the attack is decomposed in 3 steps :

- First, we observe that the distribution of $(H(y_i \oplus R_i), H(y_{i+1} \oplus R_{i+1}))$ is, in general, still biased for most constants R_i, R_{i+1}.
- Secondly, we build an empirical distribution of $(H(y_i \oplus R_i), H(y_{i+1} \oplus R_{i+1}))$ by encrypting a set of randomly chosen plaintexts. Hence we need to identify the portion of curves corresponding to W_i and W_{i+1}, then to counter the influence of the noisy term. The resulting empirical distribution is then matched with theoretical results.
- Finally, a good method to perform this matching is proposed. Our strategy is to compare distributions for two different inner rounds (not necessarily consecutive).

1 - Adding the constants in the distributions To analyze the influence of constants R_i, we simply explored all possible cases. Hence we have looked at distributions $(H(y_i \oplus R_i), H(y_{i+1} \oplus R_{i+1}))$ for various pairs of S-boxes, with all possible constants (R_i, R_{i+1}). These results are summarized in Table 4. As before, distribution $\mathcal{D}_i$ corresponds to the case $k = i$. Besides the column "constant = 0" corresponds to the previous results (see Table 3).

Clearly, we observe that the average distance is quite significant for all pairs (it ranges from $\frac{0.473}{32} \simeq \frac{1}{64}$ to $\frac{3.656}{32} \simeq \frac{1}{8}$). We also observe that there are "good" and "bad" constants, but in average an imbalance is expected.

2 - Getting rid of the noise In the second phase, our goal is to build empirical distributions. More precisely, we encrypt a set of M randomly chosen plaintexts and we monitor the electric consumption. We target the appropriate portion of the curves to observe (W_i, W_{i+1}). Our goal is, from these observations, to decide the underlying value of $(H(y_i \oplus R_i), H(y_{i+1} \oplus R_{i+1}))$ for each sample, despite the noise. Hence we obtain an empirical distribution over M samples. It is well known that when M grows, the empirical distribution converges to the theoretical distribution. More precisely, to get rid of the noise, two situations must be distinguished :

1. Suppose we can repeat each experiments. Typically, we can obtain twice from the cryptographic device the same encryption and the same execution.

S-boxes	Statistical Distance $\|\mathcal{D}_1 - \mathcal{D}_0\|$			
	constant = 0	worst constant	best constant	average value
(S_1,S_2)	$\frac{1}{64}$	0	$\frac{5}{32}$	$\frac{1.5}{32}$
(S_2,S_3)	$\frac{3}{32}$	$\frac{3}{32}$	$\frac{7}{32}$	$\frac{3.656}{32}$
(S_3,S_4)	$\frac{1}{128}$	0	$\frac{9}{128}$	$\frac{0.473}{32}$
(S_4,S_5)	$\frac{1}{64}$	0	$\frac{9}{64}$	$\frac{0.984}{32}$
(S_5,S_6)	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{3}{32}$	$\frac{1.195}{32}$
(S_6,S_7)	$\frac{3}{64}$	$\frac{1}{64}$	$\frac{9}{128}$	$\frac{1.262}{32}$
(S_7,S_8)	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{25}{128}$	$\frac{3.094}{32}$
(S_8,S_1)	$\frac{1}{16}$	$\frac{1}{128}$	$\frac{3}{32}$	$\frac{0.711}{32}$

Table 4. Statistical distances with constant R_i's

This assumption is commonly used in power analysis attacks. In this case the noise is eliminated by multiplying the samples for each trace and computing the average consumption.

2. Suppose we cannot repeat any experiments. Typically, any encryption corresponds to a random plaintext. This can result from masking countermeasures (with a fresh mask for each block !) or from a randomized mode of operation (CBC plus IV for instance).

In the first hypothesis, there is just an extra workload per message to make the noise arbitrarily small. Typically, if we have a Gaussian noise with expected value 0 and standard deviation σ, we expect to reduce the standard deviation by a factor $\sqrt{M}$ if we repeat M times each experiment. Therefore, since our model is :

$$W_i = \lambda \, \mathrm{H}(y_i \oplus R_i) + \theta$$

we consider the noise is sufficiently small when $\lambda \gg \frac{\sigma}{\sqrt{M}}$, *i.e.* the noise is negligible compared to the data-dependent term.

In the second hypothesis, we cannot eliminate the noise by averaging methods. However we hope that it will only slightly perturb our empirical distributions. Hence we suppose $\lambda \gg \sigma$, so when making a decision for each hamming weight, we have a small probability p of making a mistake. Our practical experiments on a smart card confirmed this supposition (see Section 6). A justification is that the data-dependent terms represent the consumption of bus lines which is generally dominant in a chip. More precisely, our decisions are made using thresholds :

$$t - \frac{1}{2} < \frac{W_i}{\lambda} < t + \frac{1}{2} \Rightarrow \mathrm{H}(y_i \oplus R_i) = t$$

For example if $\frac{W_i}{\lambda}$ is in the range [2.5 - 3.5], we decide $\mathrm{H}(y_i \oplus R_i) = 3$. If the noise is indeed negligible, we are successful in predicting the hamming weight with overwhelming probability. This threshold strategy is summarized in Figure 3 and further analyzed in Appendix A. Of course, in practice, λ is not known but we can set up thresholds experimentally to fit to the observations. We stress out that this analysis requires a good knowledge of the electric behavior of the chip.

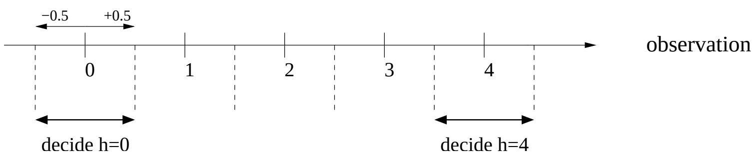

Fig. 3. Threshold rules of decision when observing $\frac{W_i}{\lambda}$

3 - Comparing two inner rounds After the step 2, we construct an empirical distribution of hamming weight from power consumption curves. We know this is biased depending on one round key bit k. However since the key is fixed, we have nothing to compare it with. Besides it is impossible to tell the value of k just by looking at the distribution, because it highly depends on the unknown R_i.

However an attack is still possible by **looking at two different inner rounds of DES** (not necessarily consecutive rounds). For instance, suppose we encrypt random plaintexts and compare the consumptions of round 1 and 2 for the adjacent S-boxes (S_2, S_3). At round 1 we observe (W_2, W_3), which is distributed differently depending on whether some first round key bit k is 0 or 1. These distributions are respectively called $\mathcal{D}_1$ and $\mathcal{D}_0$. A similar observation holds for round 2 with a second round key bit k'. Thus

- If $k = k' = i$, we have distributions $\mathcal{D}_i$ for both rounds.
- If $k \neq k'$, we have distributions $\mathcal{D}_0$ for one round and $\mathcal{D}_1$ for the other.

$\mathcal{D}_1$ and $\mathcal{D}_0$ depend on the constants R_i, but we have seen that, in average

$$|\mathcal{D}_1 - \mathcal{D}_0| = \frac{3.656}{32} \simeq 0.114$$

and even in the worst case, this value is $\frac{3}{32}$. So, in theory, if the number of samples M is sufficient (typically $M \geq \frac{1}{0.114^2} \simeq 100$), we should be able to tell if $k = k'$ or $k \neq k'$ and thus learn one bit of information about the key. In practice, we retrieve two empirical distributions $\mathcal{E}^0$ and $\mathcal{E}^1$. We must decide whether these distributions are the same ($k = k'$) or if they are different ($k \neq k'$). Because of the symmetry property exhibited in Section 4, we use the following indicator :

$$I = \sum_x (\mathcal{E}^0(x) - \mathcal{U}(x)) \times (\mathcal{E}^1(x) - \mathcal{U}(x))$$

Basically, we have normalized the empirical distributions by subtracting the $\mathcal{U}$ distribution, and then we compute a scalar product. If $k = k' = i$, then this indicator is positive :

$$I_{k=k'} = \sum_x (\mathcal{D}_i(x) - \mathcal{U}(x))^2$$

Otherwise, if $k \neq k'$, then the indicator should be negative

$$\begin{aligned} I_{k\neq k'} &= \sum_x (\mathcal{D}_0(x) - \mathcal{U}(x)) \times (\mathcal{D}_1(x) - \mathcal{U}(x)) \\ &= -\sum_x (\mathcal{D}_0(x) - \mathcal{U}(x))^2 \\ &= -\sum_x (\mathcal{D}_1(x) - \mathcal{U}(x))^2 \end{aligned}$$

because of the symmetry property described in Section 4. Therefore

$$I_{k=k'} = -I_{k\neq k'}$$

and these values are sufficiently large to be detected in practice.

5.3 Simulations

We ran some simulations of the previous distinguisher to evaluate our ability to predict correctly whether $k = k'$, and we obtained the results summarized in Table 5. Four intensity of noise where considered (see Appendix A for the role of the probability p) as well as several values for the number of samples M. We repeated the attack about 1000 times in each case, with a random choice of constants R_i.

It appears from Table 5 that the best pairs of S-boxes are (S_2, S_3) and (S_7, S_8), as predicted in Section 5.2. Hence, for our basic attack we will use any of these two pairs. For the variation attack with an 8-bit architecture, we can only use pairs with index of the form $(2i, 2i-1)$. Fortunately we can use the pair $(7, 8)$ here, which is strongly biased.

6 Some Variations of the Attack

In the previous section, we considered a simple hypothesis where all S-boxes were computed separately. Therefore we could identify portions of the power consumption curves corresponding to each S-box. In practice, the implementations are often more complex and we need to investigate if our attack applies to other situations

	Probability of success in Deciding if $k = k'$								
$p =$	0			0.1			0.25		
$M =$	256	4000	40000	256	4000	40000	256	4000	100000
(S_1, S_2)	0.5	0.65	0.98	0.5	0.59	0.90	0.5	0.51	0.69
(S_2, S_3)	0.67	0.99	1	0.58	0.96	1	0.52	0.61	0.99
(S_3, S_4)	0.5	0.54	0.83	0.5	0.54	0.72	0.5	0.5	0.54
(S_4, S_5)	0.5	0.59	0.94	0.5	0.56	0.80	0.5	0.51	0.59
(S_5, S_6)	0.5	0.59	0.93	0.5	0.56	0.81	0.5	0.51	0.63
(S_6, S_7)	0.51	0.61	0.96	0.5	0.57	0.84	0.5	0.51	0.65
(S_7, S_8)	0.70	0.99	1	0.58	0.95	1	0.51	0.59	0.99
(S_8, S_1)	0.5	0.57	0.91	0.5	0.56	0.77	0.5	0.51	0.58

Table 5. Simulation results

6.1 A real-life situation : 8-bit architecture

As an example of our attack, we have considered a recent smart-card running a software DES implementation. The card also featured some usual hardware countermeasure (but no software countermeasure like masking). These countermeasures included a variable internal clock and some random peaks of power. Despite these protections, we managed to identify the power consumption corresponding to each portion of the DES execution. This analysis required first to understand well the behavior of the card. The trickiest part was to eliminate the random peaks of power but it turned out they were not "that" random and their presence was strongly correlated with the external clock.

In addition, we realized that two S-boxes are executed simultaneously by the card, *i.e.* each pair of adjacent S-boxes (S_1 and S_2, S_3 and S_4, etc ...). Therefore, the power consumption observed is $W_{2i} + W_{2i-1}$ for $i = 1 \ldots 4$. It is strongly correlated with the sum of the hamming weights:

$$\begin{aligned} &\mathrm{H}(y_{2i} \oplus R_{2i}) + \mathrm{H}(y_{2i-1} \oplus R_{2i-1}) \\ &= \mathrm{H}((y_{2i} \oplus R_{2i}) || ((y_{2i-1} \oplus R_{2i-1}))) \end{aligned}$$

Accordingly, we expect to observe 9 groups of curves locally, if we have many samples (corresponding to hamming weight ranging from 0 to 8). In fact, due to the noise influence, it is difficult to make the groups appear very distinctly, but if we display a few curves (see Figure 4), a clear distinction starts to appear depending on the hamming weight. Low hamming weight correspond to low consumption (few bits are flipped from the reference states), while high hamming weights curves are located at the top of this Figure. In addition, these experiments illustrate the fact that some noise θ is indeed present, but it is relatively small compared to the data-dependent term, since the Hamming weight distinction appears clearly.

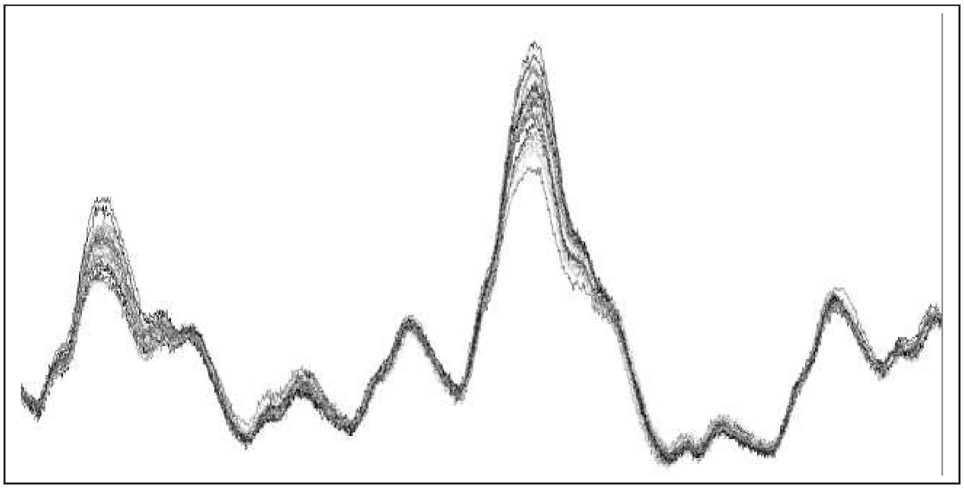

Fig. 4. Distinction of Curves According to the Hamming Weight

In this scenario, we can only observe the sum of power consumption for certain pairs of S-boxes. We computed theoretically the expected imbalance for these pairs (see Table 6). Here the distributions considered are over the sum of hamming weights $h_1 + h_2$ and not the joint distribution (h_1, h_2) as previously.

S-boxes	Statistical Distance $\lvert\mathcal{D}_1 - \mathcal{D}_0\rvert$			
	constant $= 0$	worst constant	best constant	average value
(S_1,S_2)	$\frac{1}{64}$	0	$\frac{1}{8}$	$\frac{1.097}{32}$
(S_3,S_4)	$\frac{1}{128}$	0	$\frac{3}{64}$	$\frac{0.406}{32}$
(S_5,S_6)	$\frac{1}{64}$	$\frac{1}{64}$	$\frac{21}{256}$	$\frac{1.076}{32}$
(S_7,S_8)	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{39}{256}$	$\frac{2.627}{32}$

Table 6. Statistical distances in the 8-bit scenario

Hence, the statistical distances are still relatively high for the four pairs of S-boxes. To perform the Davies-Murphy power attack here, we can use again the trick of comparing two different inner rounds, like in Section 5.2.

6.2 Case when more S-boxes are computed simultaneously

When considering software implementations of DES, we believe the most common situation are those where 1 or 2 S-boxes at most are computed simultaneously. This was developed in Section 5 and Section 6.1. To our knowledge, no software implementation presents a higher degree of parallelization than that.

However, when turning to DES hardware implementations, more than two S-boxes are often computed at the same time. Things are thus more complex because we observe many biased distributions simultaneously, and this depends on many key bits. We are currently investigating some refined version of our attack in this case. We believe an attack can be achieved since the imbalance is detectable in theory, but it will probably not be very efficient.

7 Impact on DES Implementations

Modern countermeasures against Side Channel Attacks are often focused against DPA. Accordingly they try to make intermediate data handled during the block cipher computation as random and unpredictable as possible. Two main techniques have receive a huge interest in recent years

- *Masking Techniques* [2–4] where the idea is to ensure that critical intermediate data are equal to the "true" data XOR some random mask. Masking the round input has clearly no effect, since we process randomly-chosen data. However masking the output is problematic since the Davies imbalance now depends on the mask. But, for consistency, masking countermeasures generally require some unmasked round output (see [4]). This was not believed to be critical because the goal was to thwart DPA and HO-DPA. However DMPA will still work here, except we need to choose two inner rounds with unmasked output.
  If all round outputs were masked with the same value, an higher-order version of DMPA could be envisaged [3]. So the best protection consists in masking all round outputs with a distinct value. But this is probably too expensive in practice (actual countermeasures use one or two masking values at most).
- *Duplication Techniques* [12] where all intermediate data are split into two parts, using the secret sharing principle. However, by analyzing simultaneously the behavior of both parts, the Davies imbalance should still be observed. Since everything is duplicated, the analysis is probably more complicated because 4 S-boxes need to be considered instead of just 2.

Therefore the Davies-Murphy Power Attack (DMPA) is likely to defeat most "software" countermeasures. In fact, this new attack is not fundamentally different from classical DPA : both gather power traces and sort them according to some intermediate data, the goal being to verify a guess on a few key bits. However, while DPA focuses on predicting some data from the plaintext and a few key bits, DMPA does not require the knowledge of the plaintext. The analysis is based only on the internal structure of DES and we can predict intermediate

[3] Actually the trick from Section 5.2 of using power traces of any two inner rounds is already, by definition, a second-order attack

data (actually a bias on intermediate data), only from a few key bits. The advantage is that we can focus our analysis on any inner round, while DPA usually focuses on the first (or last) rounds of DES.

An other advantage is that countermeasures designed specifically to thwart the family of DPA attack (like masking, duplication, or others ...) are unlikely to be very efficient as a protection against DMPA. The main drawback of the attack is that it is rather expensive in terms of messages encrypted. Moreover it requires a fine analysis of the electric behavior of the target cryptographic hardware, in order to find an appropriate power consumption model and to identify each portion of the DES execution. So there is a lot of preliminary analysis to do before applying the attack.

Finally DMPA proves that even slight weakness or small "non-random" behavior of a cipher can be exploited to mount a side channel attack. Software countermeasures are helpful to complicate the task of the attacker, but a better protection against power attacks will be obtained if

- the cipher behaves as randomly as possible.
- efficient hardware countermeasures are implemented, to limit the information leaked in the electric consumption.

8 Extensions

All our analysis has focused on the case of DES. Indeed the principle of Davies-Murphy attack was initially developed specifically against DES. However, more generally, for any Feistel cipher with a non-bijective round function, some imbalance in the round output necessarily exists. In this case, the requirements for DMPA are

- Express the output imbalance with a small number of key bits.
- Find a correlation between the non-randomly distributed data and the electric consumption

The first requirement depends on the cipher, while the second depends on the cryptographic hardware considered. We did not explore further to find applications on other algorithms but we believe it is an interesting topic for further research.

9 Conclusion

We have proposed a new side channel attack against DES, the Davies-Murphy Power Attack. It is based on the well known Davies-Murphy attack. Like its predecessor, our attack uses non-uniform output distributions of adjacent S-boxes. Then we detect this imbalance using electric consumption curves.

DMPA is very powerful, because it requires no information about the plaintext and can be performed on any inner rounds of DES. Therefore we believe it can defeat software countermeasures, which do not take into account this type of threat. However DMPA is rather expensive : good knowledge of the device behavior regarding power consumption is required, and the data processing complexity is rather high. For a non-protected implementation of DES, simpler side-channel attacks (like DPA) should be preferred.

References

1. M.-L. Akkar, R. Bevan, P. Dischamp, and D. Moyart. Power Analysis, What Is Now Possible ... In T. Okamoto, editor, *Advances in Cryptology – Asiacrypt'00*, volume 1976 of *Lectures Notes in Computer Science*, pages 489–502. Springer, 2000.
2. M.-L. Akkar, R. Bevan, and L. Goubin. Two Power Analysis Attacks against One-Mask Methods. In B. Roy and W. Meier, editors, *Fast Software Encryption – 2004*, pages 308–325, 2004. Pre-proceedings Version.
3. M.-L. Akkar and C. Giraud. An Implementation of DES and AES Secure against Some Attacks. In Ç. Koç, D. Naccache, and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 2162 of *Lectures Notes in Computer Science*, pages 309–318. Springer, 2001.
4. M.-L. Akkar and L. Goubin. A Generic Protection against High-Order Differential Power Analysis. In T. Johansson, editor, *Fast Software Encryption – 2003*, volume 2887 of *Lectures Notes in Computer Science*, pages 192–205. Springer, 2003.
5. E. Biham and A. Biryukov. An Improvement of Davies' Attack on DES. In A. De Santis, editor, *Advances in Cryptology – Eurocrypt'95*, volume 950 of *Lectures Notes in Computer Science*, pages 461–467. Springer, 1995.
6. E. Biham and A. Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In A. Menezes and S. Vanstone, editors, *Advances in Cryptology – Crypto'90*, volume 537 of *Lectures Notes in Computer Science*, pages 2–21. Springer, 1990.
7. E. Biham and A. Shamir. Power Analysis of the Key Scheduling of the AES Candidates. In *Second AES Candidate Conference*, 1999.
8. D. Boneh, R. DeMillo, and R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract). In W. Fumy, editor, *Advances in Cryptology – Eurocrypt'97*, volume 1233 of *Lectures Notes in Computer Science*, pages 37–51. Springer, 1997.
9. E. Brier, C. Clavier, and F. Olivier. Optimal Statistical Power Analysis, 2003. Available on Eprint : `http://eprint.iacr.org/2003/152/`.
10. C. Clavier, J.-S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lectures Notes in Computer Science*, pages 252–263. Springer, 2000.
11. D. Davies and S. Murphy. Pairs and Triplets of DES S-Boxes. *Journal of Cryptology*, 8(1):1–25, 1995.
12. L. Goubin and J. Patarin. DES and Differential Power Analysis, The "Duplication" Method. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 1717 of *Lectures Notes in Computer Science*, pages 158–172. Springer, 1999.
13. P. Kocher. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Others Systems. In N. Koblitz, editor, *Advances in Cryptology – Crypto'96*, volume 1109 of *Lectures Notes in Computer Science*, pages 104–113. Springer, 1996.
14. P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In M. Wiener, editor, *Advances in Cryptology – Crypto'99*, volume 1666 of *Lectures Notes in Computer Science*, pages 388–397. Springer, 1999.

15. H. Ledig, F. Muller, and F. Valette. Enhancing Collision Attacks. In *Cryptographic Hardware and Embedded Systems (CHES)*, 2004. to appear.
16. 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.
17. T. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant software. In Ç. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems (CHES)*, volume 1965 of *Lectures Notes in Computer Science*, pages 238–251. Springer, 2000.
18. T. Messerges, E. Dabbish, and R. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology 1999*, 1999. Available at `http://www.usenix.org/`.
19. NIST FIPS PUB 46-3. *Data Encryption Standard*, 1977.
20. K. Schramm, T. Wollinger, and C. Paar. A New Class of Collision Attacks and its Application to DES. In T. Johansson, editor, *Fast Software Encryption – 2003*, volume 2887 of *Lectures Notes in Computer Science*. Springer, 2003.
21. J. Waddle and D. Wagner. Towards Efficient Second Order Power Analysis. In *Cryptographic Hardware and Embedded Systems (CHES)*, 2004. to appear.

A Statistical influence of the noise

In Section 5, we are interested in distinguishing two distributions, $\mathcal{D}_0$ and $\mathcal{D}_1$. Our analysis of DES revealed that the statistical distance $d = |\mathcal{D}_0 - \mathcal{D}_1|$ was sufficiently large to distinguish between these two distributions. But in practice we need to build an empirical observation of these distributions in the presence of noise. As we argued, this noise is generally small, but still it may result in errors of prediction in the threshold technique of Figure 3. There is a small probability p that the noise is larger than $0.5\ \lambda$ and thus we predict $h+1$ or $h-1$ instead of the "true" h. We call $\mathcal{D}'_i$ the new distribution obtained when the noise is taken into account. We have :

$$\mathcal{D}'_i(h) = p\ \mathcal{D}_i(h-1) + p\ \mathcal{D}_i(h+1) + (1-2p)\ \mathcal{D}_i(h)$$

Thus

$$\Delta(h) = \mathcal{D}'_i(h) - \mathcal{D}_i(h) = p \cdot [\mathcal{D}_i(h-1) + \mathcal{D}_i(h+1) - 2\ \mathcal{D}_i(h)]$$

$\Delta(h)$ is the difference of probability of deciding h, resulting from the noise influence. We see that if $p \ll 1$, then $\Delta(h) \ll 1$ for all $h \in \{0, \ldots, 4\}$. Therefore

$$|\mathcal{D}'_i - \mathcal{D}_i| = \sum_h |\Delta(h)| = \varepsilon \ll 1$$

for $i = 0, 1$. Hence, it is still possible to make the difference between the two distributions since

$$\begin{aligned} |\mathcal{D}'_1 - \mathcal{D}'_0| &\le |\mathcal{D}'_1 - \mathcal{D}_1| + |\mathcal{D}_1 - \mathcal{D}_0| + |\mathcal{D}'_0 - \mathcal{D}_0| \\ &\le |\mathcal{D}_1 - \mathcal{D}_0| + 2\ \varepsilon \end{aligned}$$

Therefore as long as the noise results in small probabilities of incorrect decisions, we can still apply the same methods.