A Differential Fault Attack Against Early Rounds of (Triple-)DES

Ludger Hemme

Giesecke & Devrient GmbH
Prinzregentenstr. 159, 81677 Munich, Germany
ludger.hemme@de.gi-de.com

Abstract. Previously proposed differential fault analysis (DFA) techniques against iterated block ciphers mostly exploit computational errors in the last few rounds of the cipher to extract the secret key. In this paper we describe a DFA attack that exploits computational errors in early rounds of a Feistel cipher. The principle of the attack is to force collisions by inducing faults in intermediate results of the cipher. We put this attack into practice against DES implemented on a smart card and extracted the full round key of the first round within a few hours by inducing one bit errors in the second and third round, respectively.

1 Introduction

In 1997 Biham and Shamir [4] proposed the so called Differential Fault Analysis (DFA) and applied it to secret key cryptosystems such as DES. Their attack exploits computational errors induced during the last few rounds of DES to extract the secret key of the last round. At least since the results of Anderson and Skorobogatov [2] the application of this attack to tamper resistant devices such as smart cards is a real threat: By exposing a chip to a laser beam or even the focused light from a flash lamp it is possible to induce the kinds of errors that are needed by the attack to succeed. Therefore in addition to possibly existing hardware countermeasures it is advisable to implement also adequate software countermeasures like verifying the correctness of an encryption by a subsequent encryption or decryption. To optimize performance, one might think of reducing these countermeasures to the critical last few rounds or, in case of Triple-DES, for example, to the last DES operation. This, however, can lead to a lack of security, as we will show in this paper. We will present a DFA attack against early rounds of a Feistel cipher and show that it is not sufficient to protect only the last few rounds against inducing computational errors. Since the attack targets at the first few rounds of the cipher (more exactly rounds 2,3,...) it is advisable to protect also these rounds.

The attack requires a chosen plaintext situation. The attacker must be able to choose various plaintexts and to encrypt them with the secret key that he wants to compromise. Associated with smart cards this might be a realistic scenario. By inducing a fault during the encryption of a plaintext P the attacker tries

M. Joye and J.-J. Quisquater (Eds.): CHES 2004, LNCS 3156, pp. 254–267, 2004.

to get a collision with another plaintext $\widetilde{P}$, meaning that the faulty ciphertext belonging to P equals the correct ciphertext belonging to $\widetilde{P}$. This is in some sense a reversion of the original DFA attack of Biham and Shamir [4]. The problem, however, is to find the pairs of colliding plaintexts in an efficient way. To solve this problem we make use of the concept of characteristics introduced by Biham and Shamir [3]. Once having found a pair of colliding plaintexts one can apply methods of differential cryptanalysis to gain some information about the first round key. Other pairs will provide further information until at last the full round key of the first round will be recovered.

In the following we will first provide some notations and definitions and then describe in detail the principle of the attack against a Feistel cipher. Finally we will describe the application of the attack on DES and Triple-DES, respectively.

2 Notations and Definitions

Definition 1. *A Feistel cipher of block length $2n$ with r rounds $(n, r \in \mathbb{N})$ is a function $F_K : \mathrm{GF}(2)^{2n} \longrightarrow \mathrm{GF}(2)^{2n}$ with a key $K = (K_1, \ldots, K_r)$ consisting of r round keys $K_i \in \mathrm{GF}(2)^m$ of length $m \in \mathbb{N}$, which maps a plaintext $P = (P^L, P^R) \in \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n$ to the corresponding ciphertext $C = (C^L, C^R) = F_K(P)$ in the following way:*

1. $L_0 := P^L, R_0 := P^R$
2. *For* $i = 1, \ldots, r$
   $(L_i, R_i) := (R_{i-1}, L_{i-1} \oplus f(R_{i-1}, K_i))$,
   where the round function $f : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^m \longrightarrow \mathrm{GF}(2)^n$ *is any mapping and* $\oplus$ *is the ordinary componentwise addition over* $\mathrm{GF}(2)$.
3. $C^L := R_r, C^R := L_r$

Traditionally, the round keys $(K_1, K_2, \ldots, K_r)$ are computed by a key schedule algorithm on input a master key, but in Definition 1 also the case of independent round keys is included. Figure 1 shows the Feistel scheme as a flowchart.

The attack described in Sect. 3 deals with inducing errors during the encryption of plaintexts. Hence we introduce a notation for the faulty encryption of a plaintext P. Let F_K be a Feistel cipher of block length $2n$ with r rounds and let $k \in \{1, \ldots, r\}$, $\varepsilon \in \mathrm{GF}(2)^n$. Then $F_K^{(k,\varepsilon)} : \mathrm{GF}(2)^{2n} \to \mathrm{GF}(2)^{2n}$ denotes the mapping which maps P to $F_K^{(k,\varepsilon)}(P)$ by applying the encryption algorithm F_K to P, whereby the output Y_k of the round function f in the k-th round is replaced with $Y_k \oplus \varepsilon$.

In the following we will have to deal with pairs of plaintexts, ciphertexts and intermediate results and with their differences with regard to $\oplus$, the so called XOR-differences. So for a pair of plaintexts $(P, \widetilde{P})$ and a Feistel cipher F_K we denote the XOR-differences occurring during the calculation of $F_K(P)$ and $F_K(\widetilde{P})$ in the following way:

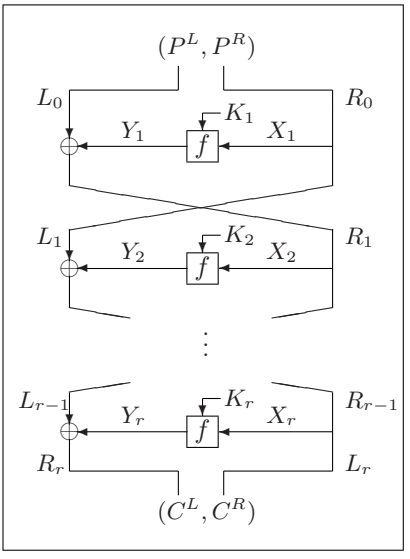

Fig. 1. Feistel scheme

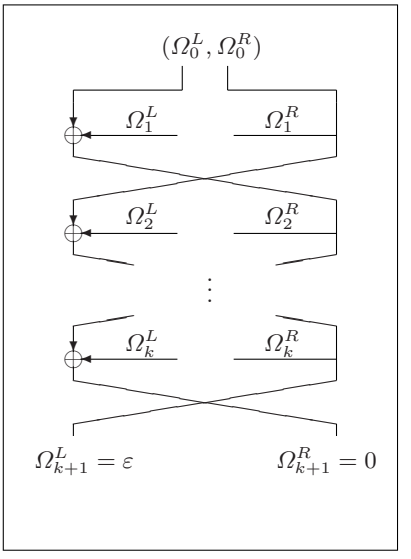

Fig. 2. A k-round ε-characteristic

$\Delta P := P \oplus \widetilde{P}$ plaintext difference
$\Delta C := C \oplus \widetilde{C}$ ciphertext difference
$\Delta L_i := L_i \oplus \widetilde{L}_i$,
$\Delta R_i := R_i \oplus \widetilde{R}_i$ differences of the intermediate results in the i-th round

Besides we denote the inputs of the round function f in the i-th round by $X_i, \widetilde{X}_i$, respectively, and its outputs by $Y_i, \widetilde{Y}_i$, respectively (see Fig. 1). The corresponding XOR-differences are denoted the same way as above:

$\Delta X_i := X_i \oplus \widetilde{X}_i$ input difference of the round function f in the i-th round
$\Delta Y_i := Y_i \oplus \widetilde{Y}_i$ output difference of the round function f in the i-th round

In this paper we will be mainly interested in differences between intermediate results occurring during the faulty encryption of P and the correct encryption of $\widetilde{P}$, more exactly during the calculation of $F_K^{(k,\varepsilon)}(P)$ and $F_K(\widetilde{P})$. For the sake of simplicity we denote also these differences with the above defined symbols. Though the exact meaning of the symbols should always be clear from the context.

Definition 2. *A k-round characteristic with respect to a Feistel cipher of block length $2n$ with r rounds $(k, n, r \in \mathbb{N}, k \leq r)$ is a tuple $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_{k+1})$, where the $\Omega_i = (\Omega_i^L, \Omega_i^R) \in \mathrm{GF}(2)^n \times \mathrm{GF}(2)^n$ satisfy the following conditions:*

i) $\Omega_1^R = \Omega_0^R$,
ii) $\Omega_2^R = \Omega_0^L \oplus \Omega_1^L$,

iii) $\Omega^L_{k+1} = \Omega^R_k$,
iv) $\Omega^R_{i+1} = \Omega^L_i \oplus \Omega^R_{i-1} \; \forall\, i \in \{2, 3, \ldots, k\}$.

For $\varepsilon \in \mathrm{GF}(2)^n$ a k-round characteristic Ω is called a k-round ε-characteristic if $(\Omega^L_{k+1}, \Omega^R_{k+1}) = (\varepsilon, 0)$.

Definition 3. *A right pair with respect to a k-round characteristic $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_{k+1})$ and with respect to a key K of the associated Feistel cipher F_K is a pair of plaintexts $(P, \widetilde{P})$ satisfying the following conditions:*

i) $\Delta P = \Omega_0$,
ii) $(\Delta Y_i, \Delta X_i) = (\Omega^L_i, \Omega^R_i) \; \forall\, i \in \{1, \ldots, k\}$,

where $\Delta Y_i, \Delta X_i$ are the above defined differences at the encryption by F_K.

Definition 4. *The probability $p_{\Omega,K}$ of a characteristic $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_{k+1})$ with respect to a key K of the associated Feistel cipher is the probability that a random pair of plaintexts $(P, \widetilde{P})$ satisfying $\Delta P = \Omega_0$ is a right pair with respect to Ω and K.*

3 Description of the Attack Against a Feistel Cipher

Let F_K be a Feistel cipher of block length $2n$ with r rounds, where K is the secret key that we would like to compromise. To carry out the attack the following preconditions must be fulfilled:

i) *Chosen plaintext scenario:* It is possible to encrypt arbitrarily chosen plaintexts with the secret key and to check the corresponding ciphertexts for pairwise equality. In particular, if the computed ciphertexts are returned to the attacker as result, this check is trivially possible.
ii) *Fault model:* It is possible to induce errors during computation of $F_K(P)$, more exactly to replace the output Y_k of the round function f in the k-th round ($k \geq 2$) with $Y_k \oplus \varepsilon$, where $\varepsilon \in E$ is a not necessarily known element of the a priori chosen subset $E \subseteq \mathrm{GF}(2)^n$. In the notation of Sect. 2 this means that it is possible to 'compute' $F_K^{(k,\varepsilon)}(P)$ for some $\varepsilon \in E$. Considering E as a probability space we denote by $\mathrm{prob}(\varepsilon)$ the probability that the induced error is ε.

By executing the following algorithm we will now try to get a pair of plaintexts $(P, \widetilde{P})$, where for the encryption by F_K the difference ΔY_1 at the output of the round function f in the first round is known. In the following we will call a triple $(P, \widetilde{P}, \Delta Y_1)$ consisting of a pair of plaintexts $(P, \widetilde{P})$ and the corresponding output difference ΔY_1 of the round function f 'a useful pair'.

Algorithm 1

INPUT
- *error round* $k \geq 2$
- *error set* $E \subseteq \mathrm{GF}(2)^n$
- *index set* $\hat{E} \subseteq E$
- *for each* $\varepsilon \in \hat{E}$ *a* $(k-1)$*-round* ε*-characteristic* $\Omega_\varepsilon = (\Omega_{\varepsilon,0}, \Omega_{\varepsilon,1}, \ldots, \Omega_{\varepsilon,k})$

1. *Choose a random plaintext* $P \in GF(2)^{2n}$ *and 'compute'* $F_K^{(k,\varepsilon)}(P)$ *for some random* $\varepsilon \in E$*;*
2. *For every* $\hat{\varepsilon} \in \hat{E}$
   a) *Set* $\widetilde{P} := P \oplus \Omega_{\hat{\varepsilon},0}$ *and compute* $F_K(\widetilde{P})$*;*
   b) *If* $F_K(\widetilde{P}) = F_K^{(k,\varepsilon)}(P)$ *then output the triple* $(P, \widetilde{P}, \Omega^L_{\hat{\varepsilon},1})$*;*

The following proposition shows why we may expect that Algorithm 1 will output useful pairs after a certain number of runs.

Proposition 1. *The probability that one pass of Algorithm 1 outputs at least one useful pair is at least* $\sum_{\varepsilon \in \hat{E}} \mathrm{prob}(\varepsilon) p_{\Omega_\varepsilon, K}$*, where the* $p_{\Omega_\varepsilon, K}$ *are the probabilities of the characteristics* Ω_ε *with respect to the secret key* K*.*

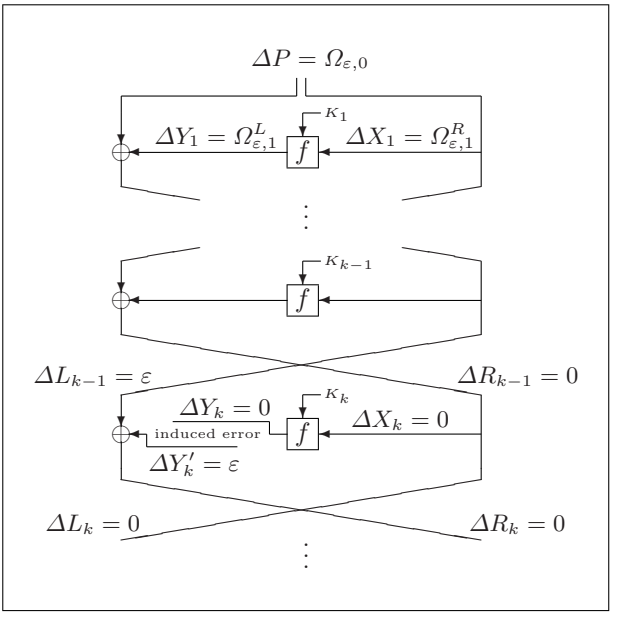

Fig. 3. $(P, \widetilde{P})$ is a right pair with respect to Ω_ε and K

Proof. Let $P \in \mathrm{GF}(2)^{2n}$ and $\varepsilon \in \mathrm{GF}(2)^n$ be the plaintext and the induced fault, respectively, from step 1 of Algorithm 1. Assume that $\varepsilon \in \hat{E}$. So during one pass of Algorithm 1, in one of the steps 2a the plaintext $\widetilde{P} \in \mathrm{GF}(2)^{2n}$ is defined, such that $\Delta P = \Omega_{\varepsilon,0}$. With probability $p_{\Omega_\varepsilon,K}$ the pair $(P, \widetilde{P})$ is a right pair with respect to Ω_ε and the secret key K. If this is the case (see Fig. 3), then in particular $\Delta Y_1 = \Omega^L_{\varepsilon,1}$, $\Delta L_{k-1} = \varepsilon$ and $\Delta R_{k-1} = 0$. In the k-th round of the encryption of P the output Y_k of the round function f is replaced by $Y'_k := Y_k \oplus \varepsilon$. At the end of the k-th round we have $\Delta L_k = \Delta R_{k-1} = 0$ and $\Delta R_k = \Delta L_{k-1} \oplus \Delta Y_k \oplus \varepsilon = \varepsilon \oplus \varepsilon = 0$, since $\Delta Y_k = f(R_{k-1}, K) \oplus f(\widetilde{R}_{k-1}, K) = 0$ due to $\Delta R_{k-1} = 0$. This implies that $F_K^{(k,\varepsilon)}(P) \oplus F_K(\widetilde{P}) = 0$ and the triple $(P, \widetilde{P}, \Omega^L_{\varepsilon,1})$, which is a useful pair due to $\Delta Y_1 = \Omega^L_{\varepsilon,1}$, is output by Algorithm 1. Since in step 1 the error $\varepsilon \in \hat{E}$ occurs with probability $\mathrm{prob}(\varepsilon)$, the probability that one pass of Algorithm 1 outputs at least one useful pair is at least $\sum_{\varepsilon \in \hat{E}} \mathrm{prob}(\varepsilon) p_{\Omega_\varepsilon,K}$. □

To get a high output rate of Algorithm 1 the probabilities of the used characteristics should be as high as possible. The existence of a $(k-1)$-round ε-characteristic $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_k)$ of probability $p_{\Omega,K} > 0$ for any $\varepsilon \in \mathrm{GF}(2)^n$ is ensured by the following consideration: Due to the invertibility of the Feistel structure there is a pair of plaintexts $(P, \widetilde{P})$ such that $L_{k-1} = \varepsilon$ and $R_{k-1} = \widetilde{L}_{k-1} = \widetilde{R}_{k-1} = 0$ (for the encryption by F_K). For the choice $\Omega_0 := \Delta P$, $\Omega_k := (\varepsilon, 0)$ and $\Omega_i := (\Delta Y_i, \Delta X_i)$, $i \in \{1, \ldots, k-1\}$, the tuple $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_k)$ is a $(k-1)$-round ε-characteristic and $(P, \widetilde{P})$ is a right pair with respect to Ω and K.

Although the existence of appropriate characteristics is ensured, it is a problem to find some without knowing the key K. However, depending on the round function of the considered Feistel cipher, it may be possible to define a probability of characteristics that is independent of the key K and still a good approximation for the probability of Definition 4. In case of DES, for example, we can use a definition of Biham and Shamir [3] that helps us to calculate characteristics of high probability.

Another problem we have to take into consideration is that Algorithm 1 might output pairs which are erroneously regarded as useful pairs. We denote by p_{err} the probability that a triple $(P, \widetilde{P}, \Omega^L_{\varepsilon,1})$ output by Algorithm 1 is not a useful pair. The following proposition says that we don't have to worry if the induced error occurs in the second round of the cipher.

Proposition 2. *Let $(P, \widetilde{P}, \Omega^L_{\hat{\varepsilon},1})$ be a triple output by Algorithm 1 with error round $k = 2$. Then $(P, \widetilde{P}, \Omega^L_{\hat{\varepsilon},1})$ is a useful pair.*

Proof. Let ε be the error occurred in step 1 of Algorithm 1. Since $(P, \widetilde{P}, \Omega^L_{\hat{\varepsilon},1})$ was output by Algorithm 1, we have $\Delta C = F_K^{(2,\varepsilon)}(P) \oplus F_K(\widetilde{P}) = 0$. Due to the structure of the Feistel cipher this implies $\Delta R_1 = 0$. By the choice of $\widetilde{P}$ we have $\Delta P^L = P^L \oplus \widetilde{P}^L = \Omega^L_{\hat{\varepsilon},0} = \Omega^R_{\hat{\varepsilon},2} \oplus \Omega^L_{\hat{\varepsilon},1} = \Omega^L_{\hat{\varepsilon},1}$, where $\Omega_{\hat{\varepsilon}} = (\Omega_{\hat{\varepsilon},0}, \Omega_{\hat{\varepsilon},1}, \Omega_{\hat{\varepsilon},2})$ is the characteristic used by Algorithm 1 to define $\widetilde{P}$. Thus the difference at the output

of the round function f in the first round is $\Delta Y_1 = \Delta R_1 \oplus \Delta P^L = 0 \oplus \Omega^L_{\hat{\varepsilon},1} = \Omega^L_{\hat{\varepsilon},1}$ and $(P, \widetilde{P}, \Omega^L_{\hat{\varepsilon},1})$ is a useful pair. □

Once having found useful pairs we can exploit them by means of differential cryptanalysis to get some information about the round key K_1 of the first round. So let $(P, \widetilde{P}, \Delta Y_1)$ be a useful pair. Then we test for each candidate $\hat{K}_1 \in \mathrm{GF}(2)^m$ of the round key K_1 if it satisfies

$$f(P^R, \hat{K}_1) \oplus f(\widetilde{P}^R, \hat{K}_1) = \Delta Y_1. \tag{1}$$

If this is the case we increment a counter for this candidate. After having processed several useful pairs, the candidate with the highest counter is taken to be the value of K_1. The success of this method depends on the signal to noise ratio S/N which is the expected number of times the right key is counted over the expected number of times a randomly picked wrong key is counted. For $S/N > 1$ the method succeeds and the number of needed pairs decreases with increasing S/N. If $S/N = 1$ the method does not succeed.

In our case the pairs to be analysed are output by Algorithm 1. For every useful pair the right key is counted and in addition several wrong keys, which we suppose to be uniformly distributed. With probability p_{err} an output pair is not a useful pair. In this case there are counted several keys, which we suppose again to be uniformly distributed. This time it is not guaranteed that the right key is counted but it can happen. Let γ be the average number of counted keys per pair and Q be the number of key candidates. Then the signal to noise ratio is

$$S/N = \frac{(1 - p_{\mathrm{err}}) + p_{\mathrm{err}} \cdot \gamma/Q}{\gamma/Q} = \frac{(1 - p_{\mathrm{err}}) \cdot Q}{\gamma} + p_{\mathrm{err}}. \tag{2}$$

In the worst case we have $p_{\mathrm{err}} = 1$ and no output pair is a useful pair. The signal to noise ratio is then $S/N = 1$ and the attack fails. If however p_{err} is small, then $S/N \approx Q/\gamma$.

As a consequence of these considerations a characteristic $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_k)$ used in Algorithm 1 should satisfy the condition $\Omega^R_0 \neq 0$. Assume this is not the case. Then every useful pair output by Algorithm 1 using Ω has the form $(P, \widetilde{P}, 0)$, where $\Delta P^R = \Omega^R_0 = 0$. Thus (1) is obviously satisfied for all key candidates $\hat{K}_1 \in \mathrm{GF}(2)^m$ and Ω does not contribute to compromise the key.

Before discussing the application of the attack on DES we give a slightly generalised version of Algorithm 1 that provides higher flexibility and thus better possibilities of optimizing the attack. In Algorithm 1 for each possible error $\varepsilon \in E$ there is used at most one characteristic. According to Proposition 1 the probability of this characteristic should be as high as possible to ensure a high output rate of useful pairs. In general, however, for some errors $\varepsilon \in E$ there might be only ε-characteristics of relatively low probability, whereas for some other errors there are even several ε-characteristics of relatively high probability. Algorithm 2 takes this situation into account. At least in the case of DES this generalised algorithm yields slightly better results than Algorithm 1 as we will see in Sect. 4.

Algorithm 2

INPUT · *error round* $k \geq 2$
· *error set* $E \subseteq \mathrm{GF}(2)^n$
· *for each* $\varepsilon \in E$ *a set* $\mathcal{C}_\varepsilon$ *of* $(k-1)$*-round* ε*-characteristics*

1. *Choose a random plaintext* $P \in GF(2)^{2n}$ *and 'compute'* $F_K^{(k,\varepsilon)}(P)$ *for some random* $\varepsilon \in E$*;*
2. *For every* $\hat{\varepsilon} \in E$*, for every* $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_k) \in \mathcal{C}_{\hat{\varepsilon}}$
   a) *Set* $\widetilde{P} := P \oplus \Omega_0$ *and compute* $F_K(\widetilde{P})$*;*
   b) *If* $F_K(\widetilde{P}) = F_K^{(k,\varepsilon)}(P)$ *then output the triple* $(P, \widetilde{P}, \Omega_1^L)$*;*

A lower bound for the output probability of Algorithm 2 is given by Proposition 3. The proof is in principle the same as for Proposition 1.

Proposition 3. *The probability that one pass of Algorithm 2 outputs at least one useful pair is at least* $\sum_{\varepsilon \in E} \sum_{\Omega \in \mathcal{C}_\varepsilon} \mathrm{prob}(\varepsilon) p_{\Omega,K}$*, where the* $p_{\Omega,K}$ *are the probabilities of the characteristics* Ω *with respect to the secret key* K*.*

Again we can be sure that an output triple is a useful pair, if the error is induced in the second round.

Proposition 4. *Let* $(P, \widetilde{P}, \Omega_1^L)$ *be a triple output by Algorithm 2 with error round* $k = 2$*. Then* $(P, \widetilde{P}, \Omega_1^L)$ *is a useful pair.*

The proof is exactly the same as for Proposition 2.

4 Application of the Attack on (Triple-)DES

Now we will show how the attack described above can be applied to the Data Encryption Standard (DES) [7]. Here we refer to 'DES' as a Feistel cipher of block length 64 with 16 rounds that has additional bit permutations at the beginning and at the end. Throughout this section we ignore the existence of these permutations. We can do this by the following convention. Whenever we use the word 'plaintext' ('ciphertext') we imagine that this is the already permuted actual plaintext (ciphertext). The round function f consists of the E-expansion, the addition of the 48 bit round key, the S-box transformations and the P-permutation. For details refer to [7].

From the principle of the attack it is clear that Triple-DES can be attacked in exactly the same way as DES, meaning that the determination of the first round key is equal for both ciphers. Thus, in the following, whenever we write 'DES' one may also read 'Triple-DES' instead.

To carry out Algorithm 1 and Algorithm 2, respectively, we first have to choose an appropriate error set $E \subseteq \mathrm{GF}(2)^n$. Generally the choice depends on the implementation of DES, on the hardware platform and on the equipment used for inducing the faults. The more information an attacker has about the kind of

faults he is able to induce, the more selective he can choose the set E. For the beginning we choose E to be the set $E_{\text{onebit}} := \{\varepsilon \in \text{GF}(2)^{32}; \text{hammingweight}(\varepsilon) = 1\}$ of all one bit faults. So our goal is to induce a one bit error in the output of the round function f. This could be done, for example, by inducing a bit flip in the register containing this intermediate result [2]. But this is not the only way to reach the goal. Another possibility is to disturb the program flow during the calculation of the P-permutation, the S-box transformations or even the E-expansion or the addition of the round key [1]. Assume, for example, that at one point of the DES calculation we are able to prevent the correct reading of an S-box table entry, so that a random value instead of the correct 4 bit result is returned. Then with probability 1/4 there will be a one bit error in the output of the S-box transformations and thus in the output of the round function f. Such a 'random' S-box output can also be forced indirectly, for example by inducing an appropriate error during the E-expansion or the addition of the round key.

Once having chosen the set E, we have to choose the error round number $k \geq 2$ and to calculate $(k-1)$-round ε-characteristics Ω of high probability $p_{\Omega,K}$. Unfortunately we cannot calculate $p_{\Omega,K}$ of a given characteristic Ω because we do not know the secret key K. Though for the case of DES, Biham and Shamir [3] defined a probability p_Ω of a characteristic Ω that is independent of the secret key K and a good approximation for the probability $p_{\Omega,K}$ of Definition 4.

Definition 5. *Let $\Omega = (\Omega_0, \Omega_1, \ldots, \Omega_{k+1})$ be a k-round characteristic with respect to DES.*
The probability $p_\Omega^{(i)}$ of round i of Ω is the fraction

$$p_\Omega^{(i)} := 2^{-32 \cdot 48} \cdot |\{(x, y) \in \text{GF}(2)^{32} \times \text{GF}(2)^{48}\,;\, f(x, y) \oplus f(x \oplus \Omega_i^R, y) = \Omega_i^L\}|$$

of all input pairs $x, x \oplus \Omega_i^R$ of f, 'encrypted' by all round keys y, for which the output difference equals Ω_i^L.
The probability p_Ω of the characteristic Ω is the product

$$p_\Omega := \prod_{i=1}^{k} p_\Omega^{(i)}$$

of the probabilities of all rounds.

With this definition it is possible to calculate the best $(k-1)$-round ε-characteristics $\Omega_\varepsilon = (\Omega_{\varepsilon,0}, \Omega_{\varepsilon,1}, \ldots, \Omega_{\varepsilon,k})$ for all $\varepsilon \in E$. Here 'the best' means those with the highest probability p_{Ω_ε}. For the calculation we implemented the search algorithm of Matsui [6], slightly modified due to the side condition for Ω_ε. The results of the calculation for the one bit fault model $E = E_{\text{onebit}}$ are listed in the appendix, where tables 4 to 6 show the best $(k-1)$-round ε-characteristics for all $\varepsilon \in E_{\text{onebit}}$ and for $k = 2, 3, 4$. Note that all these ε-characteristics satisfy $\Omega_{\varepsilon,1}^R \neq 0$ and thus can be used to recover the first round key. A brief overview of the corresponding probabilities is given in Table 1.

We implemented Algorithm 1 and Algorithm 2 on a PC and simulated the attack against DES for the following fault model. *It is possible to flip a single*

Table 1. Probabilities of the best $(k-1)$-round ε-characteristics for $\varepsilon \in E_{\text{onebit}}$

$k-1$	$\frac{1}{32} \sum_{\varepsilon \in E_{\text{onebit}}} p_{\Omega_\varepsilon}$	$\max_{\varepsilon \in E_{\text{onebit}}} p_{\Omega_\varepsilon}$	$\min_{\varepsilon \in E_{\text{onebit}}} p_{\Omega_\varepsilon}$
1	0.111	0.250	0.016
2	$1.78 \cdot 10^{-3}$	$7.32 \cdot 10^{-3}$	$3.13 \cdot 10^{-6}$
3	$1.38 \cdot 10^{-6}$	$4.89 \cdot 10^{-6}$	$1.43 \cdot 10^{-9}$

random bit (uniformly distributed) of the output of the round function f in the k-th round. That means that the error set E is the set E_{onebit} of the 32 possible one bit faults and $\text{prob}(\varepsilon) = 1/32$ for all $\varepsilon \in E$. For the analysis of the found pairs we use a counting scheme that counts over 6 bit subkeys (corresponding to the 6 bit S-box inputs) of the first round key. Let us assume that the probability p_{err} of an output triple being not a useful pair is negligible. For error round $k = 2$ this is guaranteed by Proposition 2 and Proposition 4, respectively. According to (2) for the signal to noise ratio of this counting scheme we have

$$S/N \approx \frac{Q}{\gamma} \geq \frac{2^6}{16} = 4,$$

where the upper bound 16 for the counted keys per pair is given by the difference distribution tables of DES [3]. The ratio S/N is high enough for the attack to succeed with a reasonable amount of useful pairs. Table 2 shows some results of the simulation using Algorithm 1 for the error rounds $k = 2, 3, 4$. For various numbers of runs of Algorithm 1 and various numbers $|\hat{E}|$ of used characteristics it is stated how many faulty and how many correct encryptions were calculated, how many useful pairs were found and how many bits of information about the key were extracted. The numbers of found pairs and key bits are averaged over the number of performed simulations, which can be found in the last column. Note that for each single simulation the secret key to be compromised was randomly chosen. Finally Table 2 shows for each case the expected minimum number of useful pairs, calculated using Proposition 1. The characteristics used for the simulation were chosen in the following way. Assume the number of used characteristics given by Table 2 being N. Then the used characteristics are the N most probable characteristics of the appropriate table in the appendix.

Table 3 shows some results of the simulation using Algorithm 2. The used characteristics were chosen in the following way. For errors $\varepsilon \in E$, for which there exist ε-characteristics Ω_ε of relatively high probability, we chose various ε-characteristics, whereas for other errors $\varepsilon \in E$ we did not choose any ε-characteristic due to the low probabilities. Furthermore the choice was made taking care that the number of different values of $\Omega^R_{\varepsilon,1}$ is as high as possible. The reason for that is a better distribution of the wrong subkey values counted during the differential analysis, resulting in a higher signal to noise ratio of the counting scheme. The results in Table 3 show that the number of induced errors required

Table 2. Simulation on PC (Algorithm 1, fault model $E = E_{\text{onebit}}$)

k	faulty + correct DES-calculations	characteristics	found pairs	expected pairs	extracted key bits	simulations
2	$100 + 1600$	16	9.16	9.18	17.46	10000
	$500 + 8000$	16	45.95	45.90	41.03	10000
	$500 + 16000$	32	55.43	55.53	46.74	10000
	$1000 + 32000$	32	110.92	111.05	47.94	10000
	$1500 + 48000$	32	166.46	166.58	48.00	10000
3	$5 \cdot 10^3 + 4.0 \cdot 10^4$	8	6.20	6.26	17.65	1000
	$1 \cdot 10^4 + 1.6 \cdot 10^5$	16	17.67	17.64	36.36	1000
	$5 \cdot 10^4 + 8.0 \cdot 10^5$	16	88.88	88.19	45.59	1000
	$5 \cdot 10^5 + 1.6 \cdot 10^7$	32	888.11	889.24	47.86	1000
	$1 \cdot 10^6 + 3.2 \cdot 10^7$	32	1779.18	1778.48	48.00	1000
4	$5 \cdot 10^6 + 7.0 \cdot 10^7$	14	6.25	6.57	34.52	100
	$1 \cdot 10^7 + 1.4 \cdot 10^8$	14	13.40	13.15	42.42	100
	$5 \cdot 10^7 + 1.0 \cdot 10^9$	20	67.25	68.55	47.30	20

Table 3. Simulation on PC (Algorithm 2, fault model $E = E_{\text{onebit}}$)

k	faulty + correct DES-calculations	characteristics	found pairs	expected pairs	extracted key bits	simulations
2	$10 + 990$	99	3.96	3.96	10.55	10000
	$200 + 12800$	64	58.62	58.59	45.84	10000
	$400 + 40000$	100	94.21	94.17	48.00	10000
3	$100 + 28100$	281	2.09	2.17	10.23	1000
	$500 + 140500$	281	11.07	10.87	31.25	1000
	$1000 + 160000$	160	12.67	12.62	36.18	1000
	$2000 + 278000$	139	15.67	15.70	41.34	1000
	$5000 + 555000$	111	26.42	26.57	46.83	1000
	$10000 + 1110000$	111	53.42	53.14	47.95	1000
4	$1 \cdot 10^5 + 1.62 \cdot 10^7$	162	1.56	1.35	13.82	100
	$5 \cdot 10^5 + 8.10 \cdot 10^7$	162	6.52	6.72	36.21	100
	$1 \cdot 10^6 + 1.62 \cdot 10^8$	162	13.96	13.44	45.78	100
	$5 \cdot 10^6 + 8.10 \cdot 10^8$	162	66.68	67.20	48.00	100

for extracting a certain amount of key bits is lower than for Algorithm 1. This gain, however, is diminished by the higher amount of correct DES-calculations resulting from the higher number of used characteristics. Apart from this one can see that for error rounds $k = 3, 4$ the total amount of DES-calculations required for extracting many key bits is slightly less than for using Algorithm 1.

Now let us consider another fault model. Assume that during the DES calculation we are able to disturb the access to the S-box tables in a way that instead of the correct S-box entry a random 4 bit value is read. Hence we consider the error sets $E_i := \{\pi(x) \in \mathrm{GF}(2)^{32}; x = (x_1, \ldots, x_8) \in \mathrm{GF}(2)^{4 \cdot 8} \wedge x_j = 0 \text{ for } j \neq i\}$ of all

the errors that arise from a random fault in the output of S-box S_i ($i = 1, \ldots, 8$), where π denotes the P-Permutation of the DES round function. The calculation of the best ε-characteristics for all $\varepsilon \in \bigcup_{i=1}^{8} E_i$ showed that the probabilities for $\varepsilon \in \bigcup_{i=1}^{8} E_i \setminus E_{\text{onebit}}$ are much smaller than for $\varepsilon \in E_{\text{onebit}}$. Hence it is reasonable also for this fault model to use ε-characteristics for $\varepsilon \in E_{\text{onebit}}$ only.

To test the attack in a real life situation, we implemented DES on a smart card and induced computational errors during calculation of the S-boxes by exposing the chip to a laser beam. For the determination of the correct timing we exploited information obtained by measuring the power consumption of the chip. The distribution of the induced 1-, 2-, 3- and 4-bit faults showed that we managed it to approximately realise the just considered fault model. Of course not every shot induced an error in the desired S-box output and finally we reached average probabilities between 13% and 17% for generating a 1-bit error in the output of a certain S-box by one shot. After these preliminary examinations we carried out the attack against the smart card by applying Algorithm 1 for the inputs $k = 2$, $E = E_i$ and $\hat{E} = E_i \cap E_{\text{onebit}}$ ($i = 1, 5, 6, 7$), using the characteristics from Table 4. In other words we induced errors in the outputs of the S-boxes 1,5,6 and 7 in the second round and looked for useful pairs using four characteristics in each of the four cases. In total we carried out 13000 passes of Algorithm 1, i.e. 13000 faulty and 52000 correct DES calculations, and found 187 useful pairs that revealed the full round key of the first round. As one DES calculation took about 0.1 seconds, including the time for communication between smart card and terminal, the attack took about two hours.

Next we attacked the third DES round on the same smart card by disturbing the S-boxes 1,4 and 8 in the same manner as described above. This time we found in total 263 pairs by $6.9 \cdot 10^5$ runs of Algorithm 1, meaning an effort of $6.9 \cdot 10^5$ faulty and $2.76 \cdot 10^6$ correct DES calculations or 96 hours runtime, respectively. Again the found pairs compromised the full round key of the first round.

5 Conclusion

In this paper we introduced a DFA attack on early rounds of a Feistel cipher showing that it is not sufficient to protect only the last few rounds of the cipher against inducing computational errors. By carrying out the attack against DES implemented on a smart card we proved that the attack is not only of theoretical nature but a real threat in practice. An evident question is if the principle of the attack can also be applied to a non-Feistel cipher, for example to the AES. The answer is yes. In case of AES, for example, it is possible to combine the attack of Dusart et al. [5] with our principle to force collisions by inducing computational errors in early rounds of the cipher. The problem is that the probabilities of characteristics or differentials, respectively, for AES are much smaller than for DES. So even by using counting schemes over four key bytes, as Piret and Quisquater [8] did to optimise the attack in [5], the amount of AES calculations required to extract the secret key is quite large at the moment, but our investigations are still in progress.

References

1. R. Anderson and M. Kuhn. Low Cost Attacks on Tamper Resistant Devices. In *IWSP: 5th International Workshop of Security Protocols*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136. Springer, 1997.
2. R. Anderson and S. Skorobogatov. Optical Fault Induction Attacks. In *Cryptographic Hardware and Embedded Systmes - CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*. Springer, 2002.
3. E. Biham and A. Shamir. Differential cryptanalysis of DES-like cryptosystems. *Journal of Cryptology*, 4(1):3–72, 1991.
4. E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystems. *Lecture Notes in Computer Science*, 1294:513–525, 1997.
5. P. Dusart, G. Letourneux, and O. Vivolo. Differential Fault Analysis on A.E.S. Available at http://eprint.iacr.org/, 2003/010, 2003.
6. M. Matsui. On Correlation Between the Order of S-boxes and the Strength of DES. *Lecture Notes in Computer Science*, 950:366–375, 1995.
7. NIST FIPS PUB 46-3. *Data Encryption Standard (DES)*. U.S. Department of Commerce, 1999.
8. G. Piret and J.J. Quisquater. A Differential Fault Attack Technique against SPN Structures, with Application to the AES and KHAZAD. In *Cryptographic Hardware and Embedded Systmes - CHES 2003*, volume 2779 of *Lecture Notes in Computer Science*. Springer, 2003.

A Tables of Characteristics

The following tables show the best 1-, 2- and 3-round ε-characteristics Ω_ε of DES for all $\varepsilon \in E_{\text{onebit}}$. The components of the characteristics are listed in hexadecimal notation. Components not given in the tables can be calculated according to Definition 2.

Table 4. Best 1-round ε-characteristics of DES for $\varepsilon \in E_{\text{onebit}}$

ε	$\Omega^L_{\varepsilon,0}$	p_{Ω_ε}
80 00 00 00	00 00 8A 22	0.0546875
40 00 00 00	00 00 02 02	0.1875
20 00 00 00	00 00 80 02	0.15625
10 00 00 00	00 88 80 10	0.041015625
08 00 00 00	40 80 02 10	0.029296875
04 00 00 00	40 08 00 00	0.25
02 00 00 00	40 00 40 10	0.21875
01 00 00 00	44 09 01 10	0.046875
00 80 00 00	04 09 41 00	0.0478515625
00 40 00 00	04 00 00 04	0.15625
00 20 00 00	04 00 01 00	0.1875
00 10 00 00	80 40 11 04	0.03515625
00 08 00 00	04 40 11 00	0.041015625
00 04 00 00	80 00 10 00	0.125
00 02 00 00	00 40 10 00	0.1875
00 01 00 00	A0 40 00 80	0.015625

ε	$\Omega^L_{\varepsilon,0}$	p_{Ω_ε}
00 00 80 00	A0 04 10 80	0.046875
00 00 40 00	00 04 00 80	0.15625
00 00 20 00	00 04 00 80	0.15625
00 00 10 00	30 24 00 08	0.0244140625
00 00 08 00	01 24 20 08	0.03515625
00 00 04 00	00 20 00 08	0.25
00 00 02 00	10 20 20 00	0.15625
00 00 01 00	02 00 24 08	0.0341796875
00 00 00 80	12 00 24 09	0.0390625
00 00 00 40	00 00 04 01	0.1875
00 00 00 20	02 00 04 01	0.1875
00 00 00 10	02 12 0C 01	0.0546875
00 00 00 08	02 12 0C 20	0.041015625
00 00 00 04	08 02 08 20	0.1875
00 00 00 02	00 02 08 20	0.1875
00 00 00 01	00 00 88 22	0.029296875

Table 5. Best 2-round ε-characteristics of DES for $\varepsilon \in E_{\text{onebit}}$

ε	$\Omega^L_{\varepsilon,0}$	$\Omega^R_{\varepsilon,0}$	p_{Ω_ε}
80 00 00 00	10 02 38 A0	00 02 8A 02	1.760199666e − 5
40 00 00 00	50 22 28 20	00 00 02 02	5.493164063e − 3
20 00 00 00	30 22 28 20	00 00 02 02	2.746582031e − 3
10 00 00 00	02 36 2C 81	00 00 42 12	1.564621925e − 5
08 00 00 00	1A 32 2E 03	40 00 02 12	1.173466444e − 5
04 00 00 00	00 40 13 02	40 08 00 00	1.922607422e − 3
02 00 00 00	06 40 13 02	40 08 00 00	1.201629639e − 3
01 00 00 00	A2 54 0C A1	00 01 40 14	3.129243851e − 6
00 80 00 00	E5 CC 01 80	04 09 40 00	4.380941391e − 5
00 40 00 00	48 4A 08 20	04 00 00 04	7.32421875e − 3
00 20 00 00	48 2A 08 20	04 00 00 04	2.9296875e − 3
00 10 00 00	28 16 08 20	00 00 11 44	3.755092621e − 5
00 08 00 00	44 00 20 0C	04 40 01 40	3.650784492e − 5
00 04 00 00	04 04 04 05	00 40 00 40	3.662109375e − 3
00 02 00 00	00 02 8E 23	80 00 00 40	1.922607422e − 3
00 01 00 00	12 01 A8 28	A0 00 00 C0	1.907348633e − 5
00 00 80 00	52 08 A0 1C	01 40 00 C0	5.722045898e − 5
00 00 40 00	80 00 D0 02	20 04 00 00	2.44140625e − 3
00 00 20 00	80 00 B0 02	20 04 00 00	1.220703125e − 3
00 00 10 00	04 80 D2 06	31 20 00 00	7.724761963e − 5
00 00 08 00	C4 0C 18 94	01 24 20 00	1.609325409e − 5
00 00 04 00	04 04 05 80	00 20 20 00	2.746582031e − 3
00 00 02 00	04 04 03 80	00 20 20 00	3.662109375e − 3
00 00 01 00	04 A0 80 08	12 20 04 00	1.87754631e − 5
00 00 00 80	84 65 10 0C	00 30 24 00	1.609325409e − 5
00 00 00 40	40 20 40 58	02 00 04 00	6.8359375e − 3
00 00 00 20	40 20 40 38	02 00 04 00	6.8359375e − 3
00 00 00 10	C0 80 43 04	0A 12 00 00	1.086294651e − 5
00 00 00 08	40 40 DA 38	0A 02 00 01	2.011656761e − 5
00 00 00 04	02 40 14 05	00 02 00 20	2.197265625e − 3
00 00 00 02	02 40 14 03	00 02 00 20	3.295898438e − 3
00 00 00 01	90 00 30 81	00 02 8A 00	7.152557373e − 5

Table 6. Best 3-round ε-characteristics of DES for $\varepsilon \in E_{\text{onebit}}$

ε	$\Omega^L_{\varepsilon,0}$	$\Omega^R_{\varepsilon,0}$	$\Omega^R_{\varepsilon,2}$	p_{Ω_ε}
80 00 00 00	04 03 8A 00	00 60 00 00	00 02 8A 00	6.034970284e − 7
40 00 00 00	46 88 C5 01	58 20 00 20	00 00 02 02	4.526227713e − 6
20 00 00 00	46 08 45 03	38 20 00 20	00 00 02 02	1.885928214e − 6
10 00 00 00	00 00 03 16	03 34 00 00	00 00 42 12	4.400499165e − 8
08 00 00 00	40 80 83 06	13 34 00 00	00 00 42 12	4.813045962e − 9
04 00 00 00	50 08 8A 02	80 00 03 42	40 08 00 00	2.695305739e − 6
02 00 00 00	50 08 CA 12	86 00 03 42	40 08 00 00	3.684988314e − 7
01 00 00 00	04 88 C2 06	30 20 00 00	00 00 41 14	1.432454155e − 9
00 80 00 00	42 14 0D 94	00 00 20 0A	40 00 01 14	8.952838471e − 9
00 40 00 00	C4 C0 92 56	48 4A 00 00	04 00 00 04	4.190951586e − 6
00 20 00 00	C0 C1 92 56	48 2A 00 00	04 00 00 04	1.676380634e − 6
00 10 00 00	84 C0 83 04	60 1C 00 00	04 00 11 40	1.426087692e − 7
00 08 00 00	66 58 0D C0	04 00 60 0C	04 40 01 40	3.655742375e − 8
00 04 00 00	88 62 58 78	06 04 04 04	00 40 00 40	3.129243851e − 6
00 02 00 00	C6 29 25 51	02 80 06 20	80 00 00 40	4.125467967e − 7
00 01 00 00	B2 64 04 00	00 00 1B 60	A0 44 00 00	8.381903172e − 9
00 00 80 00	23 72 0C 69	40 00 04 17	21 40 00 40	6.446043699e − 9
00 00 40 00	A0 04 14 89	00 00 D2 40	20 04 00 00	2.682209015e − 6
00 00 20 00	A0 00 14 09	00 00 B2 40	20 04 00 00	1.173466444e − 6
00 00 10 00	A1 00 10 88	00 00 D0 00	31 20 00 00	1.676380634e − 7
00 00 08 00	B0 04 30 08	00 00 C8 00	31 20 00 00	2.514570951e − 8
00 00 04 00	40 08 A0 0B	24 00 05 80	00 20 20 00	2.514570951e − 6
00 00 02 00	40 28 80 0B	24 00 03 80	00 20 20 00	2.793967724e − 6
00 00 01 00	10 30 09 00	04 A0 00 0A	12 20 04 00	1.403805072e − 8
00 00 00 80	90 00 34 00	00 06 00 20	12 00 20 01	3.274180926e − 8
00 00 00 40	04 51 10 21	00 28 00 58	02 00 04 00	4.889443517e − 6
00 00 00 20	04 41 14 21	00 28 00 38	02 00 04 00	4.889443517e − 6
00 00 00 10	1A 12 08 20	00 00 03 46	0A 12 00 00	2.900719664e − 8
00 00 00 08	1A 10 08 20	00 00 03 5E	0A 12 00 00	1.178705133e − 8
00 00 00 04	08 22 8C 29	80 00 04 45	00 02 00 20	2.011656761e − 6
00 00 00 02	00 22 8C 09	80 00 04 43	00 02 00 20	3.017485142e − 6
00 00 00 01	04 01 80 00	A0 60 00 01	00 02 8A 00	1.005828381e − 7