A first-order chosen-plaintext DPA attack on the third round of DES

Oscar Reparaz[1,2] and Benedikt Gierlichs[1]
firstname.lastname@esat.kuleuven.be

[1] KU Leuven, imec-COSIC, Belgium
[2] Square, Inc.

Abstract. DPA attacks usually exhibit a "divide-and-conquer" property: the adversary needs to enumerate only a small space of the key (a key sub-space) when performing the DPA attack. This is achieved trivially in the outer rounds of a cryptographic implementation since intermediates depend on only few key bits. In the inner rounds, however, intermediates depend on too many key bits to make DPA practical or even to pose an advantage over cryptanalysis. For this reason, DPA countermeasures may be deployed only to outer rounds if performance or efficiency are critical. This paper shows a DPA attack exploiting leakage from the third round of a Feistel cipher, such as DES. We require the ability of fixing inputs, but we do not place any special restriction on the leakage model. The complexity of the attack is that of two to three DPA attacks on the first round of DES plus some minimal differential cryptanalysis.

1 Introduction

Cryptographic implementations on embedded devices are susceptible to side-channel attacks [Koc96]. Differential Power Analysis (DPA) attacks are a powerful strand of side-channel attacks [KJJ99]. DPA is based on the fact that in an unprotected embedded device, the instantaneous power consumption depends somehow on the intermediate data handled by the implementation.

The basic working principle of DPA is to compare power consumption measurements from the device when executing the cryptographic implementation with a key-dependent model of its behavior. When modeling the device behavior, the practitioner places hypotheses on subkey values (obviously the key is secret and hence unknown). By comparing a model with the actual device behavior, DPA allows to verify or reject hypotheses on subkeys, and hence learn the actual key values. DPA and countermeasures are nowadays topics of intense research with dozens of scientific papers published per year on conferences devoted to the field.

Basic DPA attacks target the outer rounds: either the first one (if the input is known) or the last one (for known output). In outer rounds, every sensitive intermediate variable depends on only few key bits. Thus, a side-channel adversary can easily model the device behavior when handling such intermediates by placing hypothesis on only few key bits. A critical property of DPA attacks is that they allow the adversary to "divide and conquer": the adversary just repeats the same methodology with different intermediates to learn different subkey bits until he learns enough key material to break the device.

DPA countermeasures aim to prevent DPA attacks, usually by lowering the SNR of the side-channel and by data randomization. However, countermeasures come with a considerable implementation overhead, e.g. increased execution time. Therefore, if performance is of importance, one may consider to protect only outer rounds until the cipher provides enough diffusion and intermediates depend on "many" key bits. This prevents basic DPA attacks on outer rounds and allows to use a more efficient, unprotected implementation for the inner rounds.

There are DPA attacks that target inner rounds. One way to circumvent the problem of an intermediate depending on too many key bits is to deactivate portions of input texts by fixing them to a constant value. As an example, suppose we target an intermediate V that is the xor of four S-box outputs $V = S(p_1 + k_1) + S(p_2 + k_2) + S(p_3 + k_3) + S(p_4 + k_4)$. If we set p_2, p_3 and p_4 to constant values, the intermediate V can be rewritten as $V = S(p_1 + k_1) + c$ for some constant c. Then, one can perform a DPA on V to jointly recover k_1 and the constant c. This is less effort (about 2^{2w} for w-bit variables) than jointly recovering (k_1, k_2, k_3, k_4) (about 2^{4w}). In many situations, when the practitioner carefully chooses the appropriate statistical distinguisher tools, it is even possible to first recover k_1 alone, and later, in a separate step, search for the constant c, further decreasing complexity to 2×2^w.

Previous work. Kunz-Jacques et al. describe a new DPA attack, called DMPA [KMV04], based on the Davies–Murphy attack on DES [DM95]. The basic idea is that the S-box output distribution of adjacent S-boxes is not independent, and the joint output distribution depends on (a linear function of) key bits. DMPA is a higher-order attack that does not need information on plaintexts but is rather expensive in terms of data and computational complexity. Handschuh and Preneel [HP06] present a differential attack on DES aided by collisions detected on power consumption traces. They hence require a device leakage behavior in the inner rounds that allows to reliably detect collisions on individual traces. Kim et al. showed that DES is vulnerable if not all rounds are masked [KLL10], relying also on collisions and subsequent cryptanalysis. Dodis and Pietrzak introduce highly theoretical attacks on generic Feistel networks in their CRYPTO 2010 publication [DP10]. Biryukov and Khovratovich present attacks that exploit leakage from inner rounds of AES in CHES 2007 [BK07].

Our contribution. We describe a simple DPA on the third round output of a Feistel cipher. The attack uses standard first-order DPA assumptions, and thus, it is very robust to noise and simple to mount. The attack is performed in two steps. In the first step, we perform a first-order DPA with chosen input texts to deactivate parts of the state and apply Jaffe's trick to push unknown constants into the key guess [Jaf07]. In the second step, we perform a minimal cryptanalytical differential attack. Contrary to other approaches, the number of required traces for our attack is not determined by any differential propagation probability, but only by the device SNR. We fully implemented and verified our attack on a software DES implementation.

2 A first-order chosen-plaintext DPA attack on the third round of DES

Notation. Figure 1 shows the relevant part of the first three rounds of a Feistel network and sets the notation for the remainder of this paper. In the case of DES, the initial permutation (IP) is applied to the 64-bit input, then the input is placed in two 32-bit words (L_0, R_0) and the iterated processing begins. The round function is applied to the right half R_i and the round key k_i and the result is xored to the left part L_i. Then, both parts are swapped. This is repeated for $r = 16$ rounds.

$$R_{i+1} = L_i \oplus F_{k_i}(R_i) \tag{1}$$

$$L_{i+1} = R_i \qquad\qquad 0 \leq i < r \tag{2}$$

In the last round, there is no swap and a final permutation is applied (IP^{-1}). The round function results from the composition of an Expansion stage E that maps 32 bits to 48 bits in a linear way, a key mixing stage that xors 48 subkey bits k_i, a non-linear substitution layer S and a linear permutation P as

$$F_{k_i}(R_i) = P(S(E(R_i) \oplus k_i)). \tag{3}$$

Decryption is identical to encryption up to a different key schedule. The observations of this paper can be applied either way. However, it is not possible to perform our attack to round 14 since we cannot choose the output.

Setting. In this paper, we assume the adversary acquires side-channel leakage corresponding to the third round, i.e., processing after (L_2, R_2). Normally, this would correspond to a device that deploys effective countermeasures only on the first two and last two rounds. We aim to recover the full DES key.

Our attack. Our attack consists of two steps. The first step is a DPA attack with chosen inputs. It recovers the second round key blinded by some unknown constant. The second step is a differential cryptanalysis that exploits differences in the unknown constant for different chosen inputs to reveal the first round key. Once the first round key is revealed, we can compute the blinding term of the second round key (this value was unknown after the first step) and thus derive the second round key. From two consecutive round keys, the full DES key is recovered.

2.1 Step 1

Step 1 consists of a DPA attack with chosen input targeting leakage of L_3. The input is chosen such that *after IP* we have varying L_0 and constant R_0. This

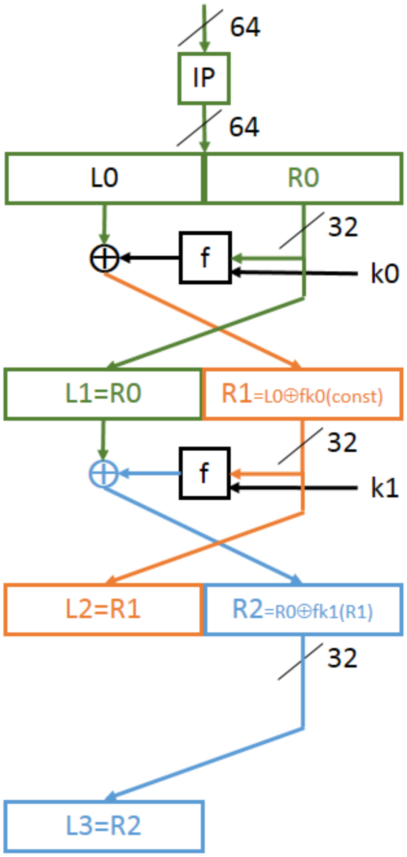

Fig. 1. First 2.5 rounds of a Feistel cipher.

enables us to “skip” placing hypotheses on the first round subkey and instead we place hypotheses on second round keys. Note that $L_1 = R_0$ is known and that $R_1 = L_0 \oplus F_{k_0}(R_0)$ is only blinded by the unknown constant $C = F_{k_0}(R_0)$. We will for the moment assume that $R_1 = L_0$ and recover C later.

With a guess on k_1 we are able to compute the output of F in round 2, and since we know L_1 we can compute further until R_2. The Feistel construction gives us the next hop for free: $L_3 = R_2$. Therefore we can exploit the leakage of L_3 to recover k_1. More precisely, this attack recovers $k_1 \oplus E(C) = k_1 \oplus E[F_{k_0}(R_0)]$.

This approach works because one can view the S-box output in the second round $F_{k_1}(R_1)$ as

$$\begin{aligned} F_{k_1}(R_1) &= F_{k_1}(L_0 \oplus C) && (4)\\ &= F_{k_1}(L_0 \oplus F_{k_0}(R_0)) && (5)\\ &= F_{k_1 \oplus E[F_{k_0}(R_0)]}(L_0) && (6) \end{aligned}$$

where E is the expansion function inside the round function F. We are hence pushing the unknown constant $C = F_{k_0}(R_0)$ to the key hypothesis k_1. This can be thought of as a variant of Jaffe’s trick [Jaf07].

Before we proceed with step 2, we need to iterate step 1 some small number of times with different constant values R'_0 and R''_0, recovering $k_1 \oplus E(C') = k_1 \oplus E[F_{k_0}(R'_0)]$ and $k_1 \oplus E(C'') = k_1 \oplus E[F_{k_0}(R''_0)]$. The second step will untangle the two terms k_1 and $E(C)$ from the recovered, “blinded”, keys $k_1 \oplus E(C)$.

2.2 Step 2

Step 2 is a classic differential attack on 1-round Feistel to recover the first round subkey k_0 from the constants $E(C)$, $E(C')$ and $E(C)''$.

Consider the differences

$$\begin{aligned} \gamma &= (k_1 \oplus E(C)) \oplus (k_1 \oplus E(C')) && (7)\\ \gamma' &= (k_1 \oplus E(C')) \oplus (k_1 \oplus E(C'')) && (8)\\ \gamma'' &= (k_1 \oplus E(C'')) \oplus (k_1 \oplus E(C))\,. && (9) \end{aligned}$$

We have

$$\begin{aligned} \gamma &= E(C) \oplus E(C') && (10)\\ &= E(F_{k_0}(R_0)) \oplus E(F_{k_0}(R'_0))\,. && (11) \end{aligned}$$

The values γ, γ' and γ'' are thus the first round output differences after the expansion E, which is invertible. Note that the adversary knows the first round input differences $R_0 \oplus R'_0$. Therefore, given the first round input and output differences, we can launch a key-recovery differential attack to recover k_0. Since we are targeting only one round, this differential attack can be performed in a divide and conquer, S-box by S-box, fashion.

In more detail: for each S-box in round 1 we place a 6-bit hypothesis on the corresponding part of k_0 and compute the output difference corresponding to input R_0 and R_0'. If the obtained output difference (after applying the expansion) is the same as the corresponding part of γ for that S-box, the subkey is kept as a candidate. Otherwise it is discarded. We repeat the procedure for different output differences γ' and γ''. The intersection of candidates is expected to yield a unique and correct subkey.

Once k_0 is recovered we can resolve $C = F_{k_0}(R_0)$, plug it in $k_1 \oplus E(C)$ to solve for k_1 and we are done. We recovered two round keys, thus, we can invert the key schedule and recover the DES key.

3 Implementation

We have fully implemented and verified our attack on an unprotected software implementation of DES in an 8-bit microcontroller. Figure 2, top, shows a power trace.

Step 1 is a classical DPA attack exploiting leakage from L_3. We made sure that this DPA attack does not exploit any leakage of rounds one and two. The target intermediate L_3 also appears as output of round 2, but we are assuming that the implementation starts leaking after round 2. In Figure 2, bottom, we plot the result for the attack on one S-box after 200 traces. The correct value for a 6-bit chunk of $k_1 \oplus E(C)$ is distinguished with a comfortable margin, as Figure 3, left, shows.

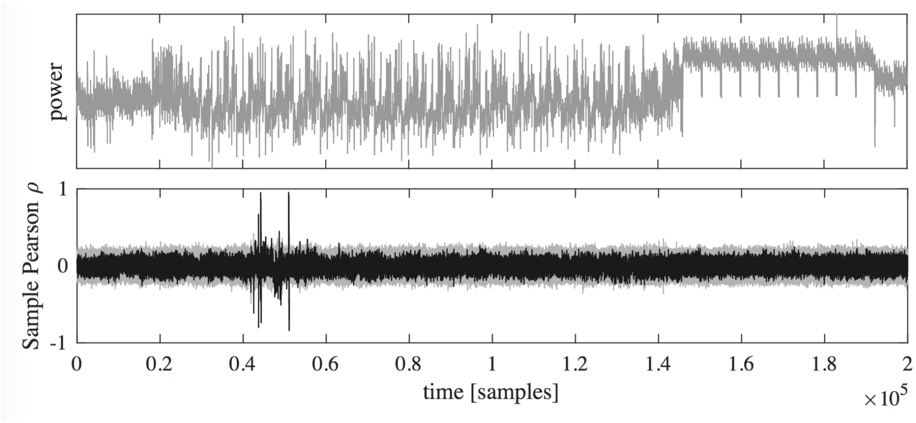

Fig. 2. Top: power consumption trace, heavily low-pass filtered to make SPA features more visible. Bottom: correlation traces. Incorrect key guesses in gray, correct key hypothesis in black. Peaks appear at the end of round 2 and at the end of round 3.

3.1 Step 1

We repeat this step three more times with different fix value R_0. The results of this step are:

- $R_0 =$ 88 00 17 FD, recovered key $k_1 \oplus E[F_{k_0}(R_0)] =$ 25 0D 02 24 15 00 06 1F.
- $R_0' =$ A9 60 1B 9F, recovered key $k_1 \oplus E[F_{k_0}(R_0')] =$ 2A 34 11 1A 31 08 05 23.
- $R_0'' =$ 3E 57 8B 11, recovered key $k_1 \oplus E[F_{k_0}(R_0'')] =$ 0B 2B 2D 11 0B 27 37 09.
- $R_0''' =$ 3E 3E 3E 3E, recovered key $k_1 \oplus E[F_{k_0}(R_0''')] =$ 0B 2E 39 18 1F 2F 32 19.

(R is given as 4 8-bit values in hexadecimal; $k_1 \oplus E[F_{k_0}(R_0)]$ is given as 8 6-bit values, one per S-box.)

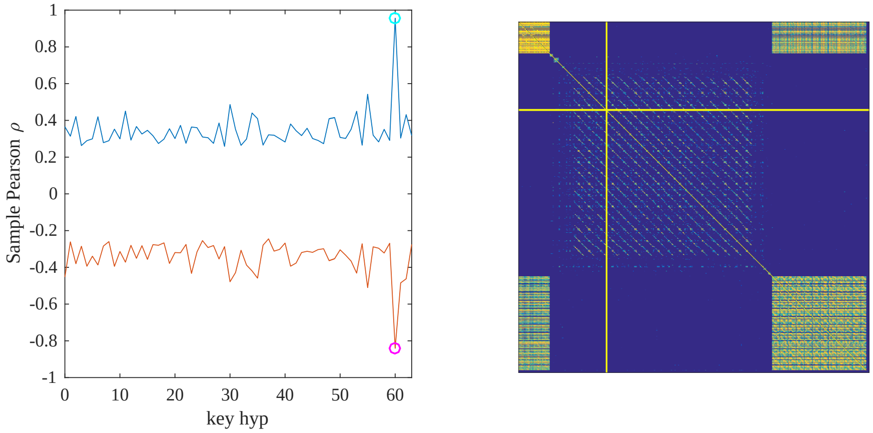

Fig. 3. Left: min/max correlation coefficient for each key, over all timesamples from round 3. The correct key hypothesis is marked with a circle. Right: cross-correlation matrix of a single trace, spanning the same time window as Figure 2. The time sample for which the Pearson correlation is maximal is marked in the picture.

3.2 Step 2

The differential attack from step 2 applied to the previous results yields the following four candidates for the k_0 round key. From each candidate for k_0 we can derive one candidate for the k_1 second round key by resolving C.

- $k_0 =$ 17 00 21 0C 15 18 3D 0F $\implies k_1 =$ 14 12 37 30 19 09 1F 0C
- $k_0 =$ 17 00 21 1F 15 18 3D 0F $\implies k_1 =$ 14 12 37 30 19 09 1F 0C
- $k_0 =$ 17 00 21 23 15 18 3D 0F $\implies k_1 =$ 14 12 3F 30 1B 29 1F 0C
- $k_0 =$ 17 00 21 30 15 18 3D 0F $\implies k_1 =$ 14 12 3F 30 1B 29 1F 0C

From every candidate for (k_0, k_1) we can invert the key schedule. (We could detect already incorrect candidates if the candidate (k_0, k_1) do not correspond to the DES key schedule, but since the number of candidates is so low in our case, we did not implement this option.) The correct round keys are found to be $k_0 =$ 17 00 21 0C 15 18 3D 0F and $k_1 =$ 14 12 37 30 19 09 1F 0C; this corresponds to the DES key 3B 38 98 37 15 20 F7 5E. We verified the correctness of the entire procedure with plaintext/ciphertext pairs.

4 Discussion

Distance leakage. In a typical hardware implementation, the attacker measures leakage roughly corresponding to $\mathrm{HW}(L_3 \oplus L_2)$. Exactly the same attack can be mounted in this case, *mutatis mutandis*, adjusting predictions in the DPA of step 1. The practitioner knows that $L_2 = R_1 \oplus C$. This C is unknown at this stage, but constant, so that he can revert to single-bit DPA (which would ignore the effect of C) or perform DPA recovering C as well.

If L_2 is masked, e.g. because the first two rounds are masked, the Hamming distance from L_2 to L_3 is also masked and the attack does not work immediately. However, a device that exhibits Hamming distance leakage typically exhibits also Hamming weight leakage (albeit possibly weaker).

Jaffe's trick. Jaffe [Jaf07] used a similar trick in a different context. He gave a surprisingly elegant attack on the CTR mode of operation, even when the starting counter value is unknown. (The amusing part here is that his is effectively a *blind* DPA attack with unknown inputs and outputs.) The basic idea is to push the unknown counter value to the subkey hypothesis, so that the DPA attack recovers at the same time the subkey and the initial counter value.

Optimizations. It may be possible to choose clever values for input differences $R_0 \oplus R_0'$ to minimize the number of candidate keys output in the second step, and thus to accelerate the whole attack. However, the gain is very thin. One condition that the input difference should satisfy is that all first-round S-boxes should be active (otherwise, the differential attack of step 2 cannot eliminate any incorrect key guess for the inactive S-box). This can be achieved, for example, with the easy-to-memorize difference $R_0 \oplus R_0' =$ `FF FF FF FF`.

How many different inputs do we need? It is possible to mount the attack with just one input difference, i.e., one known plaintext and one chosen plaintext. We have empirically determined that, if the input difference is `FF FF FF FF`, step 2 will (in the worst case) return 8, 14, 10, 16, 8, 8, 14 and 10 sub-key candidates for S-box number 1, ..., 8 respectively. This means that the step 2 yields $8 \times 14 \times \ldots \times 10 < 2^{28}$ keys, which can be easily bruteforced in a matter of seconds in a workstation. (This is a very rough upper bound, one can cut this number by first applying a consistency check if k_0 and k_1 fit the DES key schedule.)

Influence of the key schedule. Note that in the process of deriving k_1 from k_0 by resolving C in Section 2.2, we did not exploit the fact that in DES the round keys k_0 and k_1 are heavily correlated (since the DES key schedule is so simple). This method can thus be used even for other Feistel ciphers with an arbitrary key schedule algorithm, even when k_1 is completely independent of k_0. Our method, as described, recovers the first two round keys k_0, k_1. If two round keys are not enough to invert the key schedule, once the adversary learns k_0 and k_1 he can iterate the attack peeling off the first two rounds to recover k_2 and k_3 until he gets the desired amount of round keys.

5 Conclusion

In this paper, we have described a first-order chosen-plaintext DPA attack on the DES exploiting leakage stemming from the third round. This stresses, once again, the necessity of protecting implementations of outer and inner rounds in Feistel ciphers. Our attack is very easy to carry out, is resilient to noise (we only make use of first-order statistics), can be carried out with negligible computational power and recovers the full DES key.

Acknowledgments. The authors wish to thank the anonymous reviewers for their insightful comments. This work was supported in part by the Research Council KU Leuven: C16/15/058. In addition, this work was supported by the Flemish Government, FWO G.00130.13N, FWO G.0876.14N and Thresholds G0842.13; by the Hercules Foundation AKUL/11/19; and through the Cathedral ERC Advanced Grant 695305. Oscar Reparaz is Postdoctoral Fellow of the Fund for Scientific Research - Flanders (FWO). Benedikt Gierlichs is Postdoctoral Fellow of the Fund for Scientific Research - Flanders (FWO).

References

BK07. Alex Biryukov and Dmitry Khovratovich. Two new techniques of side-channel cryptanalysis. In Paillier and Verbauwhede [PV07], pages 195–208.

DM95. Donald W. Davies and Sean Murphy. Pairs and triplets of DES s-boxes. *J. Cryptology*, 8(1):1–25, 1995.

DP10. Yevgeniy Dodis and Krzysztof Pietrzak. Leakage-resilient pseudorandom functions and side-channel attacks on feistel networks. In Tal Rabin, editor, *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*, volume 6223 of *Lecture Notes in Computer Science*, pages 21–40. Springer, 2010.

HP06. Helena Handschuh and Bart Preneel. Blind differential cryptanalysis for enhanced power attacks. In Eli Biham and Amr M. Youssef, editors, *Selected Areas in Cryptography, 13th International Workshop, SAC 2006, Montreal, Canada, August 17-18, 2006 Revised Selected Papers*, volume 4356 of *Lecture Notes in Computer Science*, pages 163–173. Springer, 2006.

Jaf07. Joshua Jaffe. A first-order DPA attack against AES in counter mode with unknown initial counter. In Paillier and Verbauwhede [PV07], pages 1–13.

KJJ99. Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

KLL10. Jongsung Kim, Yuseop Lee, and Sangjin Lee. DES with any reduced masked rounds is not secure against side-channel attacks. *Computers & Mathematics with Applications*, 60(2):347–354, 2010.

KMV04. Sébastien Kunz-Jacques, Frédéric Muller, and Frédéric Valette. The davies-murphy power attack. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December*

5-9, 2004, Proceedings, volume 3329 of *Lecture Notes in Computer Science*, pages 451–467. Springer, 2004.

Koc96. Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

PV07. Pascal Paillier and Ingrid Verbauwhede, editors. *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*. Springer, 2007.