A New Class of Collision Attacks and its Application to DES

Kai Schramm, Thomas Wollinger, and Christof Paar

Department of Electrical Engineering and Information Sciences
Communication Security Group (COSY)
Ruhr-Universität Bochum, Germany
Universitaetsstrasse 150
44780 Bochum, Germany
{schramm, wollinger, cpaar}@crypto.rub.de,
WWW homepage: www.crypto.rub.de

Abstract. Until now in cryptography the term collision was mainly associated with the surjective mapping of different inputs to an equal output of a hash function. Previous collision attacks were only able to detect collisions at the output of a particular function. In this publication we introduce a new class of attacks which originates from Hans Dobbertin and is based on the fact that side channel analysis can be used to detect internal collisions. We applied our attack against the widely used Data Encryption Standard (DES). We exploit the fact that internal collisions can be caused in three adjacent S-Boxes of DES [DDQ84] in order to gain information about the secret key-bits. As result, we were able to exploit an internal collision with a minimum of 140 encryptions[1] yielding 10.2 key-bits. Moreover, we successfully applied the attack to a smart card processor.

Keywords: DES, S-Boxes, collision attack, internal collisions, power analysis, side channel attacks

1 Introduction

Cryptanalysists have used collisions[2] to attack hash functions for years [Dob98, BGW98b]. Most of the previous attacks against hash functions only attacked a few rounds, e.g., three rounds of RIPEMD [Dob97,NIS95]. In [Dob98], Dobbertin revolutionized the field of collision attacks against hash functions by introducing an attack against the full round MD4 hash function [Riv92]. It was shown that MD4 is not collision free and that collisions in MD4 can be found in a few seconds on a PC. Another historic example of breaking an entire hash function is

[1] depending on the applied measurement hardware and sampling frequency a multiple of 140 plaintexts may have to be sent to the target device in order to average the corresponding power traces, which effectively decrease noise

[2] In the remainder of this publication we do *not* require an internal collision to be detectable at the output of the cryptographic algorithm.

the COMP128 algorithm [BGW98a]. COMP128 is widely used to authenticate mobile station to base stations in GSM (Global System for Mobile Communication) networks [GSM98]. COMP128's core building block is a hash function based on a butterfly structure with five stages. In [BGW98b], it was shown that it is possible to cause a collision in the second stage of the hash function, which fully propagates to the output of the algorithm. Hence, a collision can be easily detected revealing information about the secret key.

Cryptographers have traditionally designed new cipher systems assuming that the system would be realized in a closed, reliable computing environment, which does not leak any information about the internal state of the system. However, any physical implementation of a cryptographic system will generally provide a *side channel* leaking unwanted information. In [KJJ99], two practical attacks, Simple Power Analysis (SPA) and Differential Power Analysis (DPA), were introduced. The power consumption was analyzed in order to find the secret keys from a tamper resistant device. The main idea of DPA is to detect regions in the power consumption of a device which are correlated with the secret key. Moreover, little or no information about the target implementation is required. In recent years the were several publications dealing with side channel attacks: side channel analysis of several algorithms, improvements of the original attacks, e.g., higher order DPA and sliding window DPA and hard- and software countermeasures were published [CCD00b, CJR+99b, CJR+99a, Cor99, FR99, GP99, CCD00a, CC00, Sha00, Mes00, MS00]. Recently, attacks based on the analysis of electromagnetic emission have also been published [AK96, AARR02].

The main idea of this contribution is to combine 'traditional' collision attacks with side channel analysis. Traditional collision attacks implied that an internal collision fully propagates to the output of the function. Using side channel analysis it is possible to detect a collision at any state of the algorithm even if it does not propagate to the output.

Our Main Contributions

A New Class of Collision Attack: The work at hand presents a collision attack against cryptographic functions embedded in symmetric ciphers, e.g., the f-function in DES. The idea, which originally comes from Hans Dobbertin, is to detect collisions *within* the function by analysis of side channel information, e.g., power consumption. Contrary to previous collision attacks we exploit internal collisions, which are not necessarily detectable at the output. Modified versions of this attack can be potentially applied to any symmetric cipher, in which internal collisions are possible. Furthemore, we believe that our attack is resistant against certain side channel countermeasures, which we will show in future publications.

Collisions within the DES f-function: In [DDQ84], it was first shown that the f-function of DES is not one-to-one for a fixed round key, because collisions can be caused in three adjacent S-Boxes. We discovered that such internal collisions reveal information about the secret key. On average[3] 140 different encryptions are required to find the first collision, a significant lower number of encryptions is required to find further collisions. This result is a breakthrough for future attacks against DES and other cryptographic algorithms vulnerable to internal collisions.

Realization of the Attack: Smart cards play an increasingly important role for providing security functions. We applied our attack against an 8051 compatible smart card processor running DES in software. We focussed on the S-Box triple 2,3,4 and were able to gain 10.2 key-bits with 140 encryptions on average including key reduction.

We would like to mention that there exists another attack against DES based on internal collision which requires less measurements. This attack was developed by Andreas Wiemers and exploits collisions within the Feistel cipher [Wie03].
The remaining of this publication is organized as follows. Section 2 summarizes previous work on collision attacks, side channel attacks, and DES attacks. In Section 3 we explain the principle of our new attack. In Section 4 we apply our attack to the f-function of DES. In Section 5 further optimizations of our collision attack against DES are given. In Section 6 we compromise an 8051 compatible smartcard processor running DES. Finally, we end this contribution with a discussion of our results and some conclusions.

2 Previous Work

Collision Attacks

The hashing algorithm COMP128 was a suggested implementation of the algorithms A3 and A8 for GSM [GSM98]. Technical details of COMP128 were strictly confidential, however, in 1998 the algorithm was completely reverse engineered [BGW98a]. COMP128 consists of nine rounds and the core building block is a hash function. This hash function itself is based on the butterfly structure and consists of five stages. The output bytes contain a response used for the authentication of the mobile station with the base station and the session key used for the stream cipher A5.

In [BGW98b], the COMP128 algorithm was cracked exploiting a weakness in the butterfly structure. Only the COMP128 input bits corresponding to the random number can be varied. A collision can occur in stage 2 of the hash function. It will fully propagate to the output of the algorithm and, as a result, it will be detectable at the output. To launch the attack, one has to vary bytes $i + 16$ and $i + 24$ of the COMP128 input and fix the remaining input bytes. The birthday paradox guarantees, that collision will occur rapidly and the colliding bytes are i, $i + 8$, $i + 16$, and $i + 24$. The attack requires $2^{17.5}$ queries to recover the whole 128-bit key.

Most of the presented attacks against hash functions only attacked a few rounds, e.g., three rounds of RIPEMD [Dob97, NIS95]. Also MD4 was first attacked partially. There were approaches to attack the two round MD4 [dBB94, Vau94] (also

[3] averaged over 10,000 random keys

an unpublished attack from Merkle). In [Dob98], Dobbertin introduced an attack against the whole MD4 hash function [Riv92]. It was shown, that an earlier attack against RIPEMD [Dob97] can be applied to MD4 very efficiently. An algorithm was developed that allows to compute a collision in a few seconds on a PC with a Pentium processor. Finally, it was demonstrated that a further development of the attack could find collisions for meaningful messages. The main result of that contribution was that MD4 is not collision-free and it requires the same computational effort as 2^{20} computations of the MD4-compression function to find a collision. The basic idea of the attack is that a difference of the input variables can be controlled in such a way that the differences occurring in the computation of the two associated hash values are compensated at the end.

Side Channel Attacks

A cryptographic system embedded into a microchip generally consists of many thousand logic gates and storage elements. The power consumption of the system can be analyzed with a shunt resistance put in series between the ground pad of the microchip and the external ground of the voltage source. A digital oscilloscope is used to digitize the voltage over the shunt resistance, which is proportional to the power consumption of the system.

Power analysis can be classified into Simple Power Analysis (SPA) and Differential Power Analysis (DPA) [KJJ99, KJJ98]. SPA directly interprets power consumption during cryptographic operations. Hence, an attacker must have detailed information about the target hardware and the implemented algorithm. Two types of information leakage have been observed in SPA: Hamming weight and transition count leakage of internal registers and accumulators [MDS99]. The Hamming weight is often directly proportional to the amount of current that is being discharged from the gate driving the data and address bus[4] [MDS99, Mui01]. Transition count information leaks during a gate transition from high to low or low to high when bits of internal registers flip [MDS99].

The main idea of the DPA is to detect regions in the power consumption of a cryptographic device correlated with particular bits of the secret key [KJJ99]. The adversary guesses a key (hypothesis) and encrypts random plaintexts. Depending on a particular observed bit within the algorithm, whose state can be computed based on the prior hypothesis, measured power traces are added or subtracted yielding a differential trace. A correct hypothesis will provide a high correlation of the differential trace with the observed bit, which will be indicated by distinct peaks. Contrary to SPA no information about the target implementation is required. In [KJJ99], it was shown that DES [NIS77] and RSA [RSA78] can be broken by DPA.

[4] if a precharged bus design is used

3 Principle of the Internal Collision Attack

An internal collision occurs if a function of a cryptographic algorithm computes two different input arguments, but returns an equal output argument. We propose the term 'internal' collision, because in general the collision will not propagate to the output of the algorithm. Since we are not able to detect it at the output we correlate side channel information of the cryptographic device, e.g., power traces, under the assumption that an internal collision will cause a high correlation of different encryptions (decryptions) at one point of time. Moreover, we assume that internal collisions which occur for particular plaintext (ciphertext) encryptions (decryptions) are somehow correlated with the secret key. A typical example of a function vulnerable to internal collisions is a surjective S-Box. However many other functions, e.g., based on finite field arithmetics, can cause collisions, too. In this publication, we exploit the fact that is possible to cause a collision in the non-linear f-function of DES in order to gain secret key-bits.

In Figure 1 the propagation path of a collision occurring in the f-function of round n is shown. The f-function in round $n+1$ processes the same input data, but any further rounds will not be affected by the collision.

An adversary encrypts (decrypts) particular plaintexts (ciphertexts) in order to cause an internal collision at one point of the algorithm. Detection of these collisions is possible by correlation of side channel information corresponding to different encryptions (decryptions), e.g., power traces of round $n+1$.

4 Collisions within the DES f-function

4.1 Collisions in single S-Boxes

In this section we briefly remind that it is possible to cause collisions in isolated S-Boxes. However, as stated in [DDQ84] overall collisions in the f-function can only be caused within three S-Boxes simultaneously. For a detailed description of DES the reader is referred to, e.g., [NIS77, MvOV97]. The eight S-Box mappings $2^6 \rightarrow 2^4$ are surjective. Moreover, the mappings are uniformly distributed, which means that for each input $z \in \{0, \ldots, 2^6-1\}$ of S-Box S_i, $i \in \{1, \ldots, 8\}$, there exist exactly three x-or differentials δ_1, δ_2 and $\delta_3 \in \{1, \ldots, 2^6-1\}$, which will cause a collision within a single S-Box

$$S_i(z) = S_i(z \oplus \delta_1) = S_i(z \oplus \delta_2) = S_i(z \oplus \delta_3), \qquad \delta_1 \neq \delta_2 \neq \delta_3 \neq 0, \qquad i \in \{1, \ldots, 8\}$$

If, for example, the first S-Box is examined and $z = 000000$, then there exist three differentials δ_1,δ_2 and δ_3 causing a collision:

$$\begin{aligned} S_1(000000) &= S_1(000000 \oplus 001001 = 001001) \\ &= S_1(000000 \oplus 100100 = 100100) \\ &= S_1(000000 \oplus 110111 = 110111) = 14 \end{aligned}$$

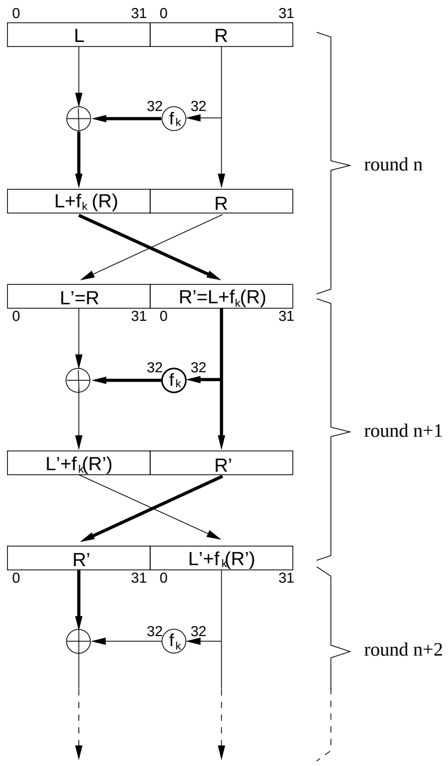

Fig. 1. Propagation path of an internal collision in DES.

However, it is not possible to directly set the six-bit input z of an S-Box. The input z corresponds to a particular six-bit input x entering the f-function. This input x is diffused[5] in the expansion permutation and x-ored with six key-bits k of the round key:

$$z = x \oplus k \Leftrightarrow k = x \oplus z \qquad k, x, z \in \{0, \ldots, 2^6 - 1\}$$

A table can be generated for each S-Box, which lists the three differentials δ_1, δ_2 and $\delta_3 \in \{1, \ldots, 2^6 - 1\}$ corresponding to all 64 S-Box inputs $z \in \{0, \ldots, 2^6 - 1\}$. These eight tables can be resorted in order to list the inputs $z \in \{0, \ldots, 2^6 - 1\}$ corresponding to all occurring differentials $\delta_i \in \{1, \ldots, 2^6 - 1\}$. In the remainder of this publication these latter tables will be referred to as the δ-tables (as an example we included the δ-table of S-Box 1 in the appendix).
In order to exploit the six key-bits k an adversary chooses a particular δ and varies the input x until he/she detects a collision $S(x \oplus k) = S(x \oplus k \oplus \delta)$. The two

[5] i.e. the two most and least significant bits of x will be x-ored with particular bits of the round key and then enter the adjacent S-Boxes

most and least significant bits of the inputs x and $x \oplus \delta$ will also enter the adjacent S-Boxes due to the bit spreading of the expansion box. As shown in Figure 2 the inputs of the adjacent S-Boxes only remain unchanged if the two most and least significant bits of differential δ are zero. However, such a differential δ does not exist, which is a known S-Box criterion [Cop94]. Therefore a collision attack targeting a single S-Box while preserving the inputs of the two adjacent S-Boxes is not possible.

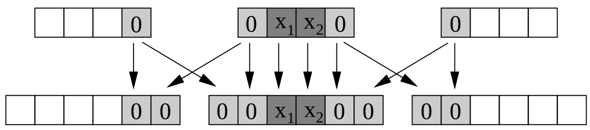

Fig. 2. Required Bit Mask of δ for a Single S-Box Collision.

4.2 Collisions in three S-Boxes

As stated in [DDQ84] it is possible to cause collisions within three adjacent S-Boxes simultaneously. In this case the inputs x and $x \oplus \Delta$ have a length of 18 bits[6]. The differential $\Delta = \delta_1|\delta_2|\delta_3$ denotes the concatenation of three S-Box differentials $\delta_1, \delta_2, \delta_3$ corresponding to each S-Box of the triple. In order not to alter the inputs of the two neighboring S-Boxes to the left and right of the S-Box triple, the two most and least significant bits of Δ must be zero:

$$\Delta[0] = \Delta[1] = \Delta[16] = \Delta[17] = 0$$

Moreover, in order to propagate through the expansion box, Δ must fulfil the condition:

$$\Delta[4] = \Delta[6], \Delta[5] = \Delta[7], \Delta[10] = \Delta[12], \Delta[11] = \Delta[13]$$

Thus $\Delta = \delta_1|\delta_2|\delta_3$ must comply with the bit mask $\Delta = 00x_1x_2vwvwx_3x_4yzyzx_5x_600$ with $x_i, v, w, y, z \in \{0, 1\}$, which is shown in Figure 3.
Analysis of the δ-tables reveals that there exist many differentials Δ, which comply with the properties stated above. As result, it is possible to cause collisions in an S-Box triple while preserving the inputs of the two neighboring S-Boxes. This means that there exist inputs x and $x \oplus \Delta$, which cause a collision $f(x) = f(x \oplus \Delta)$ in the f-function.

As an example we assume that an adversary randomly varies exactly those 14 input bits of function f in the first round, which enter the targeted S-Box triple.

[6] We refer to x and $x \oplus \Delta$ as the inputs of function f after having propagated through the expansion box, i.e., they have a length of 18 bits, but $x, x \oplus \Delta \in \{0, \dots, 2^{14} - 1\}$

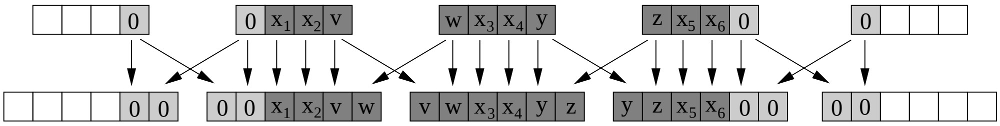

Fig. 3. Required S-Box triple Δ Bit Mask.

All 50 remaining bits of the plaintext are not changed. Within function f these bits are expanded to the 18 bit input x and x-ored with 18 corresponding key-bits k of the 48 bit round key. The result $z = x \oplus k$ enters the targeted S-Box triple. The adversary uses power analysis to record the power consumption of the cryptographic device during round two. Next, he sets the input to $x \oplus \Delta$ and again records the power consumption during round two. A high correlation of the two recorded power traces reveals that the same data was processed in function f in round two, i.e., a collision occurred. Once he detects a collision, analysis of the three corresponding δ-tables will reveal possible key candidates $k = z \oplus x$.

Let Z_Δ denote the set of all possible 18 bit inputs z_i causing a collision in a particular S-Box triple for a particular differential Δ. For a fixed x, K is the set of all possible key candidates k_i:

$$K = \{x \oplus z_i\} = \{k_i\} \qquad z_i \in Z_\Delta$$

Therefore, the number of key candidates k_i is equal to the number of possible S-Box triple inputs z_i:

$$|K| = |Z_\Delta|$$

However, for a particular 18 bit key k only those values of z_i can cause collisions for which $x = z_i \oplus k$ can propagate through the expansion box. Hence, we have to check whether all possible keys $k \in \{0, \ldots, 2^{18} - 1\}$ can cause collisions for a particular $z \in Z_\Delta$. In particular, eight bits $k[4], k[5], k[6], k[7]$ and $k[10], k[11], k[12], k[13]$ of the key k determine whether $z_i \oplus k$ yields a valid value of x. In general, we only use those differentials Δ of an S-Box triple, for which there exist inputs z_i which will yield a valid $x = z_i \oplus k$ for any key $k \in \{0, \ldots, 2^{18} - 1\}$. Thus any 18 bit key k can be classified into one of 2^8 possible key sets $K_j, j \in \{0, \ldots, 2^8 - 1\}$. The set Z_{K_j} of valid S-Box triple inputs z_i causing a collision for a given key $k \in K_j$ is generally a subset of set Z_Δ:

$$Z_{\Delta, K_j} \subseteq Z_\Delta \qquad j \in \{0, \ldots, 2^8 - 1\}$$

For a fixed key $k \in K_j$ and a random $x \in \{0, \ldots, 2^{14} - 1\}$ the probability of a collision is

$$P(f(x) = f(x \oplus \Delta) | k \in K_j) = \frac{|Z_{\Delta, K_j}|}{2^{14}}$$

In general, two plaintexts x and $x \oplus \Delta$ have to be encrypted to check for a collision $f(x) = f(x \oplus \Delta)$. The average number of encryptions $\overline{\#M}$ until a collision occurs for a fixed key k is

$$\overline{\#M} = \frac{2}{P(f(x) = f(x \oplus \Delta) | k \in K_j)} = 2 \cdot \frac{2^{14}}{|Z_{\Delta,K_j}|} = \frac{2^{15}}{|Z_{\Delta,K_j}|}$$

The total probability of a collision for an arbitrary key $k \in K_j$ is

$$P(f(x) = f(x \oplus \Delta)) = \sum_{j=0}^{255} P(f(x) = f(x \oplus \Delta) | k \in K_j) \cdot P(k \in K_j) = 2^{-22} \cdot \sum_{j=0}^{255} |Z_{\Delta,K_j}|$$

The average number of encryptions $\overline{\#M}$ until a collision occurs for an arbitrary key $k \in K_j$ is

$$\overline{\#M} = 2 \cdot \frac{1}{256} \cdot \sum_{j=0}^{255} \frac{1}{P(f(x) = f(x \oplus \Delta) | k \in K_j)} = 2^7 \cdot \sum_{j=0}^{255} \frac{1}{|Z_{\Delta,K_j}|}$$

5 Optimization of the Collision Attack

5.1 Multiple Differentials

In order to decrease the number of encryptions until a collision occurs the attack can be extended to n differentials $\Delta_1, \ldots, \Delta_n$ yielding a set of 2^n possible encryptions $f(x)$, $f(x \oplus \Delta_1)$, $f(x \oplus \Delta_2)$, $f(x \oplus \Delta_2 \oplus \Delta_1)$,..., $f(x \oplus \Delta_n \oplus \ldots \oplus \Delta_1)$ for a fixed x. We are now looking for collisions between any two encryptions which has the potential to dramatically increase the likelihood of a collision due to the Birthday paradox. A collision $f(x') = f(x'')$ can only occur, if $x' \oplus x''$ equals a differential Δ_j, with $j \in \{1, \ldots, n\}$. In Table 1 the costs of the attacks using a single differential Δ and using n differentials $\Delta_1, \ldots, \Delta_n$ are compared.

	single Δ	multiple Δ's
$\#x$	m	m
$\#\Delta$	1	n
$\#M$	$2 \cdot m$	$m \cdot 2^n$
#collision tests	m	$m \cdot n \cdot 2^{n-1}$

Table 1. Comparison of the collision attacks using a single and multiple differentials

For example using a single Δ the random generation of $m = 64$ inputs x will result in $\#M = 128$ encryptions and will only yield $m = 64$ collision tests $f(x) = f(x \oplus \Delta)$. Using $n = 4$ differentials $\Delta_1, \ldots, \Delta_4$ the random generation of $m = 8$ inputs x will also result in $\#M = 8 \cdot 2^4 = 128$ encryptions, but will yield $8 \cdot 4 \cdot 2^3 = 256$ collision tests. In this example, with the same number of encryptions we are able to perform four times as many collision tests, which results in a higher probability of a collision.

As an example, Figure 4 shows a set of $2^n = 2^3 = 8$ encryptions for $n = 3$ differentials Δ_1, Δ_2 and Δ_3.

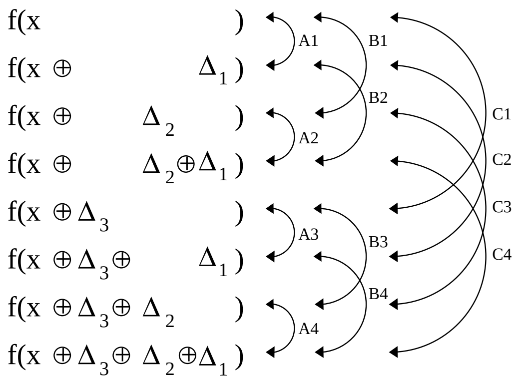

Fig. 4. Possible collision tests for $n = 3$ differentials

In this case $n \cdot 2^{n-1} = 3 \cdot 2^2 = 12$ possible collisions $A1, A2, \ldots, C4$ can occur with the following probabilities:

$$
\begin{aligned}
P_1 &= P(A1) = P(A2) = P(A3) = P(A4) = P(f(x) = f(x \oplus \Delta_1)) \\
P_2 &= P(B1) = P(B2) = P(B3) = P(B4) = P(f(x) = f(x \oplus \Delta_2)) \\
P_3 &= P(C1) = P(C2) = P(C3) = P(C4) = P(f(x) = f(x \oplus \Delta_3))
\end{aligned}
$$

If collision tests $A1, A2, \ldots, C4$ are stochastically independent[7], the overall probability can also be expressed as:

$$
\begin{aligned}
&P((A1 \cup A2 \cup A3 \cup A4) \cup (B1 \cup B2 \cup B3 \cup B4) \cup (C1 \cup C2 \cup C3 \cup C4)) \\
&= 1 - [(1 - P(A1)) \cdot (1 - P(A2)) \cdot (1 - P(A3)) \cdot (1 - P(A4)) \cdot \\
&\quad (1 - P(B1)) \cdot (1 - P(B2)) \cdot (1 - P(B3)) \cdot (1 - P(B4)) \cdot \\
&\quad (1 - P(C1)) \cdot (1 - P(C2)) \cdot (1 - P(C3)) \cdot (1 - P(C4))] \\
&\approx P(A1) + P(A2) + \ldots + P(C4)
\end{aligned}
$$

In general, if n differentials are being used and there exist no stochastical dependencies among collision tests, the overall probability that at least one collision

[7] i.e. the occurrence of a collision does not depend on any other collision test within a set of 2^n encryptions

will occur within a set of 2^n encryptions is

$$P(collision) = 1-(\prod_{i=1}^{n}(1-P_i))^{2^{n-1}} \approx 2^{n-1}\cdot\sum_{i=1}^{n} P_i \qquad with \qquad P_i = P(f(x) = f(x \oplus \Delta_i))$$

So far we assumed that collision tests were stochastically independent, i.e. the occurrence of a particular collision does not condition any other collision within a set of encryptions. Surprisingly, analysis of the collision sets Z_Δ revealed that stochastical dependencies among collision tests do exist for certain differentials. In general, stochastical dependent collision tests are not desired, because they decrease the overall probability of a collision within a set of encryptions.

5.2 Linear Dependencies

By analysis we discovered that there exist many linear combinations among the differentials Δ of all eight S-Box triples. In an attack based on multiple differentials $\Delta_1, \ldots, \Delta_n$ linear combinations of these will eventually yield additionals differentials Δ_j. As result, further collision tests can be performed without increasing the number of encryptions. Thus the probability of a collision within a set of 2^n encryptions is increased:

$$\Delta_j = a_1 \cdot \Delta_1 \oplus \ldots \oplus a_n \cdot \Delta_n, a_i \in \{0,1\} \qquad \Delta_j \neq \Delta_1 \neq \ldots \neq \Delta_n$$

The improvement achieved by exploiting linear combinations among differentials is shown in the next example.

An adversary tries to cause a collision in S-Boxes 2,3,4 using $n = 5$ differentials $\Delta_3, \Delta_{13}, \Delta_{15}, \Delta_{16}$ and Δ_{21}. Analysis of the δ-tables of S-Boxes 2,3 and 4 reveals that there exist the following linear combinations:

$$\begin{aligned}
\Delta_1 &= \Delta_3 \oplus \Delta_{13} \oplus \Delta_{15} \\
\Delta_2 &= \Delta_3 \oplus \Delta_{13} \oplus \Delta_{16} \\
\Delta_4 &= \Delta_3 \oplus \Delta_{15} \oplus \Delta_{16} \\
\Delta_{14} &= \Delta_{13} \oplus \Delta_{15} \oplus \Delta_{16} \\
\Delta_{22} &= \Delta_{15} \oplus \Delta_{16} \oplus \Delta_{21} \\
\Delta_{23} &= \Delta_{13} \oplus \Delta_{15} \oplus \Delta_{21} \\
\Delta_{24} &= \Delta_{13} \oplus \Delta_{16} \oplus \Delta_{21}
\end{aligned}$$

These seven linear combinations will allow the adversary to check $7 \cdot 2^{n-1} = 112$ additional collision tests for each set of $2^n = 32$ encryptions. The total number of collision tests for a set of 32 encryptions is thus $(n+7) \cdot 2^{n-1} = 192$.

5.3 Key Candidate Reduction

Once a first collision occurred further collisions will provide additional key sets K_i. The intersection K_{int} of these sets delimits the number of key candidates:

$$K_{int} = K_1 \cap K_2 \cap \ldots \cap K_j$$

Additional collisions can be found efficiently by fixing the input of two S-Boxes and only varying the input of the third S-Box. Due to the bit spreading in the expansion box not all input bits of the third S-Box can be varied. Only bits 2-5 of the S-Box to the left, bits 2 and 3 of the middle S-Box and bits 0-3 of the S-Box to the right can be varied without altering the inputs of the other two S-Boxes.

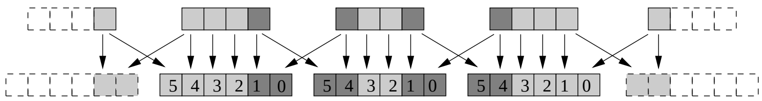

Fig. 5. Further collisions in single S-Boxes.

Analysis of the collision set Z_Δ provides all existing x-or differences $\epsilon = z' \oplus z''$ with $z', z'' \in Z_\Delta$. The theoretical[8] maximum of differentials ϵ, which only alter the input of a single S-Box is $15+3+15 = 33$. For any existing ϵ further collisions $f(x \oplus \epsilon) = f(x \oplus \epsilon \oplus \Delta)$ might be detected.

For example an adversary tries to cause collisions in S-Boxes 1,2,3 using differential Δ_3. A first collision $f(x) = f(x \oplus \Delta_3)$ yields $|Z_{\Delta_3}| = 1120$ possible key candidates. Analysis of the collision set Z_{Δ_3} reveals that there exist 18 out of 33 differentials ϵ_i, which comply with the conditions stated above. The adversary tries to find further collisions $f(x \oplus \epsilon_i) = f(x \oplus \epsilon_i \oplus \Delta_3)$ and is able to detect eight additional collisions delimiting the number of key candidates from 1120 down to 16.

6 Practical Attack

In order to verify the DES collision attack, we simulated it on a PC. In addition, an 8051 compatible microcontroller running a software implementation of DES was successfully compromised using the proposed collision attack. The measurement setup used in this practical attack is shown in Figure 6.
In this setup a PC sends chosen plaintexts to the microcontroller and triggers new encryptions. In order to measure the power consumption of the microcontroller a small shunt resistance (here $R_s = 10\Omega$) is put in series between the ground pad and ground of the power supply. We also replaced the original voltage source of the microcontroller with a low-noise voltage source to minimize noise superimposed by the source. The digital oscilloscope HP1662AS was used to sample the voltage over the shunt resistance at 1 GHz. Collisions were caused in the first round of DES. Power traces of round two were transferred to the PC using the GPIB interface. The PC was used to correlate power traces of different

[8] disregarding the S-Box disign criteria

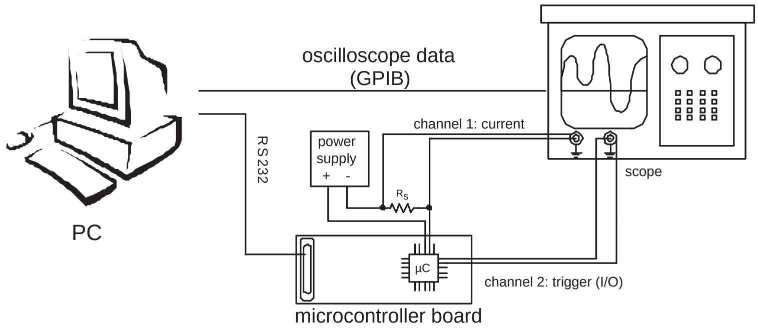

Fig. 6. Measurement setup for power analysis of a microcontroller.

encryptions in order to detect collisions. In our experiments we discovered that a correlation coefficient greater than 95% generally indicated a collision. If no collision occurred, the correlation coefficient was always well below 95%, typically ranging from 50% to 80%. In general, uncorrelated noise such as voltage source noise, quantization noise of the oscilloscope or intrinsic noise within the microcontroller can be decreased by averaging power traces of equal encryptions[9]. In our experiments we found out that averaging of $N = 10$ power traces was clearly sufficient to achieve the significant correlation results stated above. Averaging may not even be necessary at all if additional measurement circuitry is used in order to decouple the external voltage source from the target hardware or data is acquired at higher sampling rates. For example, in Figures 7 and 8 the averaged power traces of two different plaintext encryptions x and $x \oplus \Delta$ during the S-Box look-up in round two is shown. The power traces 7 and 8 clearly differ in peaks. This indicates a low correlation, i.e., no collision occured.

7 Results and Conclusions

We proposed a new kind of attack, which uses side channel analysis to detect internal collisions. In this paper the well known block cipher DES is attacked. However, the attack can be applied to any cryptographic function in which internal collisions are possible. We showed that internal collisions can be caused within three adjacent S-Boxes of DES yielding secret key information. Furthermore, we presented different methods in order to minimize the cost of finding such collisions. In our computer simulations we heuristically searched for the optimum combination of differentials Δ_i for all eight S-Box triples in order to minimize the number of required encryptions until a collision occured. The results of this exhaustive search are listed in Table 2, where $\overline{\#M}$ denotes the

[9] we assume that no countermeasures such as random dummy cycles are present

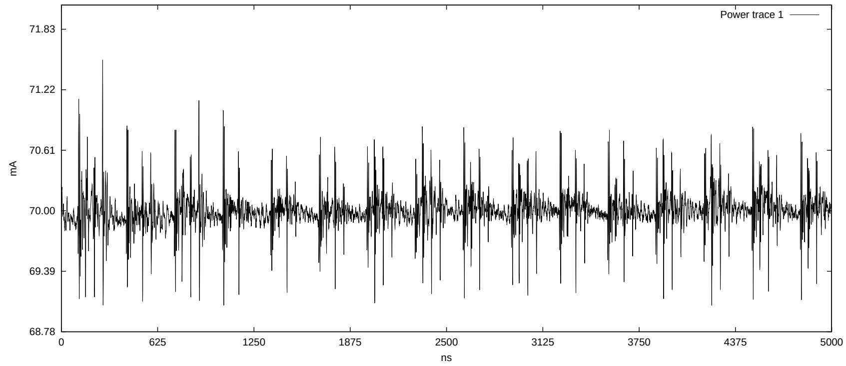

Fig. 7. Power consumption of the microcontroller encrypting x.

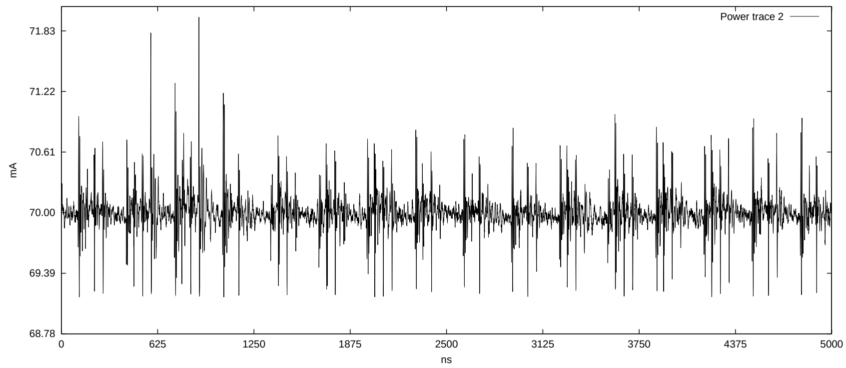

Fig. 8. Power consumption of the microcontroller encrypting $x \oplus \Delta$.

average[10] number of encryptions until a collision occurs. $\overline{\#K}$ denotes the average number of key candidates corresponding to 18 key-bits found after applying the key reduction method. As result, we were able to cause a collision in S-Box triple 2,3,4 with a minimum average of 140 encryptions. Using the key reduction method we were able to delimit 18 key-bits to an average of 220 key candidates which is equivalent to $\log_2(220) \approx 7.8$ key-bits, i.e., 10.2 key bits were broken. Moreover, we were able to cause collisions in S-Box triple 7,8,1 with an average of 165 encryptions yielding on average 19 key candidates, thus breaking $18 - \log_2(19) \approx 13.8$ key-bits. Finally, we successfully validated our attack by compromising an 8051 compatible microcontroller running DES in software.

[10] averaged over 10,000 random keys

S-Boxes	$\#\Delta$	$\Delta_1, \Delta_2, \ldots$	$\#M$	$\#K$
1,2,3	3	$\Delta_3, \Delta_{15}, \Delta_{18}$	227	20
2,3,4	5	$\Delta_3, \Delta_{13}, \Delta_{15}, \Delta_{16}, \Delta_{21}$	140	220
3,4,5	3	$\Delta_3, \Delta_{10}, \Delta_{12}$	190	110
4,5,6	3	$\Delta_2, \Delta_{10}, \Delta_{11}$	690	71
5,6,7	5	$\Delta_2, \Delta_5, \Delta_8, \Delta_{23}, \Delta_{29}$	290	24
6,7,8	5	$\Delta_7, \Delta_{10}, \Delta_{19}, \Delta_{20}, \Delta_{32}$	186	52
7,8,1	5	$\Delta_1, \Delta_2, \Delta_7, \Delta_{17}, \Delta_{19}$	165	19
8,1,2	4	$\Delta_1, \Delta_2, \Delta_8, \Delta_{38}$	208	158

Table 2. Results of the exhaustive search for the S-Box triple/Δ set optimum.

References

[AARR02] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi. The EM Side – Channel(s). In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2002*. Springer-Verlag, 2002.

[AK96] R. Anderson and M. Kuhn. Tamper Resistance - a Cautionary Note. In *Second Usenix Workshop on Electronic Commerce*, pages 1–11, November 1996.

[BGW98a] M. Briceno, I. Goldberg, and D. Wagner. An Implementation of the GSM A3A8 algorithm, 1998. http://www.scard.org/gsm/a3a8.txt.

[BGW98b] M. Briceno, I. Goldberg, and D. Wagner. GSM cloning, 1998. http://www.isaac.cs.berkely.edu/isaac/gsm–faq.html.

[CC00] C. Clavier and J.-S. Coron. On Boolean and Arithmetic Masking against Differential Power Analysis. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 231 – 237. Springer-Verlag, 2000.

[CCD00a] C. Clavier, J.-S. Coron, and N. Dabbour. Differential Power Anajlysis in the Presence of Hardware Countermeasures. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 252 – 263. Springer-Verlag, 2000.

[CCD00b] C. Clavier, J.S. Coron, and N. Dabbous. Differential Power Analysis in the Presence of Hardware Countermeasures. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 252–263. Springer-Verlag, 2000.

[CJR+99a] S. Chari, C. S. Jutla, J. R. Rao, , and P. Rohatgi. A Cauttionary Note Regarding the Evaluation of AES Condidates on Smart Cards. In *Proceedings: Second AES Candidate Conference (AES2)*, Rome, Italy, March 1999.

[CJR+99b] S. Chari, C. S. Jutla, J. R. Rao, , and P. Rohatgi. Towards Sound Approaches to Counteract Power-Analysis Attacks. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 398 – 412. Springer-Verlag, August 1999.

[Cop94] D. Coppersmith. The Data Encryption Standard (DES) and its Strength Against Attacks. Technical report rc 186131994, IBM Thomas J. Watson Research Center, December 1994.

[Cor99] J.-S. Coron. Resistance against Differentail Power Analysis for Elliptic Curve Cryptosystems. In Ç. K. Koç and C. Paar, editors, *Cryptographic*

Hardware and Embedded Systems — CHES 1999, volume LNCS 1717, pages 292 – 302. Springer-Verlag, 1999.

[dBB94] B. den Boer and A. Bosselaers. Collisions for the Compression Function of MD5. In T. Hellenseth, editor, *Advances in Cryptology — EUROCRYPT '93*, volume LNCS 0765, pages 293 – 304, Berlin, Germany, 1994. Springer-Verlag.

[DDQ84] M. Davio, Y. Desmedt, and J.-J. Quisquater. Propagation Characteristics of the DES. In *Advances in Cryptology — CRYPTO '84*, pages 62–74. Springer-Verlag, 1984.

[Dob97] H. Dobbertin. RIPEMD with two-round compress function is not collision-free. *Journal of Cryptology*, 10:51–68, 1997.

[Dob98] H. Dobbertin. Cryptanalysis of md4. *Journal of Cryptology*, 11:253–271, 1998.

[FR99] P. N. Fahn and P.K. Rearson. IPA: A New Class of Power Attacks. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 1999*, volume LNCS 1717, pages 173 – 186. Springer-Verlag, 1999.

[GP99] L. Goubin and J. Patarin. DES and Differential Power Analysis. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 1999*, volume LNCS 1717, pages 158 – 172. Springer-Verlag, 1999.

[GSM98] Technical Information - GSM System Security Study, 1998. http://jya.com/gsm061088.htm.

[KJJ98] P. Kocher, J. Jaffe, and B. Jun. Introduction to Differential Power Analysis and Related Attacks. http://www.cryptography.com/dpa/technical, 1998. Manuscript, Cryptography Research, Inc.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In *Advances in Cryptology — CRYPTO '99*, volume LNCS 1666, pages 388–397. Springer-Verlag, 1999.

[MDS99] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Investigations of Power Analysis Attacks on Smartcards. In *USENIX Workshop on Smartcard Technology*, pages 151–162, 1999.

[Mes00] T. S. Messerges. Using Second-Order Power Analysis to Attack DPA Resistant Software. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 238 – 251. Springer-Verlag, 2000.

[MS00] R. Mayer-Sommer. Smartly Analyzing the Simplicity and the Power of Simple Power Analysis on Smart Cards. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 78 – 92. Springer-Verlag, 2000.

[Mui01] J.A. Muir. Techniques of Side Channel Cryptanalysis. Master thesis, 2001. University of Waterloo, Canada.

[MvOV97] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, Boca Raton, Florida, USA, 1997.

[NIS77] NIST FIPS PUB 46-3. *Data Encryption Standard.* Federal Information Processing Standards, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., 1977.

[NIS95] NIST FIPS PUB 180-1. *Secure Hash Standard.* Federal Information Processing Standards, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., April 1995.

[Riv92] R. Rivest. *RFC 1320: The MD4 Message-Digest Algorithm.* Corporation for National Research Initiatives, Internet Engineering Task Force, Network Working Group, Reston, Virginia, USA, April 1992.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[Sha00] Adi Shamir. Protecting Smart Cards form Power Analysis with Detached Power Supplies. In Ç. K. Koç and C. Paar, editors, *Cryptographic Hardware and Embedded Systems — CHES 2000*, volume LNCS 1965, pages 71 – 77. Springer-Verlag, 2000.

[Vau94] S. Vaudenay. On the need of Multipermutations: Cryptanalysis of MD4 and SAFER. In *Fast Software Encryption — FSE '94*, volume LNCS 1008, pages 286 – 297, Berlin, Germany, 1994. Springer-Verlag.

[Wie03] A. Wiemers. Partial Collision Search by Side Channel Analysis. Presentation at the Workshop: Smartcards and Side Channel Attacks, January 2003. Horst Goertz Institute, Bochum, Germany.

A S-Box 1 δ-table

As an example the δ-table of S-Box 1 lists all inputs z corresponding to occurring differentials δ, which fulfil the condition $S_1(z) = S_1(z \oplus \delta)$. The inputs z are listed in pairs of $(z, z \oplus \delta)$, because both values will fulfil the condition $S_i(z) = S_i(z \oplus \delta) \Leftrightarrow S_i((z \oplus \delta)) = S_i((z \oplus \delta) \oplus \delta)$. For convenience, the column and row position of inputs z within the S-Box matrix is also given in parentheses.

δ	#z	$(z_1, z_1 \oplus \delta)$, $(z_2, z_2 \oplus \delta)$, ...
000011	14	((001000(04,0),001011(05,1)), ((010001(08,1),010010(09,0)), ((010101(10,1),010110(11,0)), ((011000(12,0),011011(13,1)), ((011001(12,1),011010(13,0)), ((100101(02,3),100110(03,2)), ((111001(12,3),111010(13,2))
000101	4	((000010(01,0),000111(03,1)), ((111011(13,3),111110(15,2))
000111	2	((010011(09,1),010100(10,0))
001001	10	((000000(00,0),001001(04,1)), ((000011(01,1),001010(05,0)), ((000100(02,0),001101(06,1)), ((000110(03,0),001111(07,1)), ((100000(00,2),101001(04,3))
001011	2	((100111(03,3),101100(06,2))
001101	6	((010000(08,0),011101(14,1)), ((110001(08,3),111100(14,2)), ((110101(10,3),111000(12,2))
001111	2	((100010(01,2),101101(06,3))
010001	6	((001110(07,0),011111(15,1)), ((100001(00,3),110000(08,2)), ((100011(01,3),110010(09,2))
010011	2	((100100(02,2),110111(11,3))
010111	4	((101000(04,2),111111(15,3)), ((101010(05,2),111101(14,3))
011001	2	((101111(07,3),110110(11,2))
011011	4	((000101(02,1),011110(15,0)), ((001100(06,0),010111(11,1))
011101	4	((000001(00,1),011100(14,0)), ((101110(07,2),110011(09,3))
011111	2	((101011(05,3),110100(10,2))
100010	10	((000010(01,0),100000(00,2)), ((000011(01,1),100001(00,3)), ((001100(06,0),101110(07,2)), ((001111(07,1),101101(06,3)), ((011100(14,0),111110(15,2))
100100	12	((000000(00,0),100100(02,2)), ((000110(03,0),100010(01,2)), ((001000(04,0),101100(06,2)), ((010110(11,0),110010(09,2)), ((010111(11,1),110011(9,3)), ((011000(12,0),111100(14,2))
100101	6	((001101(06,1),101000(04,2)), ((010000(08,0),110101(10,3)), ((011101(14,1),111000(12,2))
100111	10	((000111(03,1),100000(00,2)), ((001011(05,1),101100(06,2)), ((010101(10,1),110010(09,2)), ((011011(13,1),111100(14,2)), ((011100(14,0),111011(13,3))
101000	12	((001110(07,0),100110(03,2)), ((010000(08,0),111000(12,2)), ((010001(08,1),111001(12,3)), ((010010(09,0),111010(13,2)), ((011101(14,1),110101(10,3)), ((011110(15,0),110110(11,2))
101001	4	((010100(10,0),111101(14,3)), ((011000(12,0),110001(08,3))
101010	4	((000101(02,1),101111(07,3)), ((011011(13,1),110001(08,3))
101011	12	((000010(01,0),101001(04,3)), ((000110(03,0),101101(06,3)), ((001010(05,0),100001(00,3)), ((001110(07,0),100101(02,3)), ((010001(8,1),111010(13,2)), ((010010(09,0),111001(12,3))
101100	4	((000100(02,0),101000(04,2)), ((001011(05,1),100111(03,3))
101101	6	((001001(04,1),100100(02,2)), ((001111(07,1),100010(01,2)), ((011001(12,1),110100(10,2))
101110	6	((000111(03,1),101001(04,3)), ((010011(09,1),111101(14,3)), ((011010(13,0),110100(10,2))
101111	2	((001000(04,0),100111(03,3))
110001	4	((011010(13,0),101011(05,3)), ((011110(15,0),101111(07,3))
110010	4	((001101(06,1),111111(15,3)), ((011001(12,1),101011(05,3))
110011	4	((000011(01,1),110000(08,2)), ((000101(02,1),110110(11,2))
110101	2	((010110(11,0),100011(01,3))
110110	2	((010101(10,1),100011(01,3))
110111	2	((000000(00,0),110111(11,3))
111001	6	((010011(09,1),101010(05,2)), ((010111(11,1),101110(07,2)), ((011111(15,1),100110(03,2))
111010	6	((000001(00,1),111011(13,3)), ((001010(05,0),110000(08,2)), ((011111(15,1),100101(02,3))
111011	2	((000100(02,0),111111(15,3))
111110	4	((001001(04,1),110111(11,3)), ((010100(10,0),101010(05,2))
111111	4	((000001(00,1),111110(15,2)), ((001100(06,0),110011(09,3))

Table 3. S-Box 1: $S_1(z) = S_1(z \oplus \delta)$