On Closed-Cycle Loops and Applicability of Nonlinear Product Attacks to DES

Nicolas T. Courtois[1], Matteo Abbondati[3], Hamy Ratoanina[1], and Marek Grajek[2]

[1] University College London, Gower Street, London, UK
[2] Independent cryptography and crypto history expert, Poland
[3] University of Trento, Italy

Abstract. In this article we look at the question of the security of Data Encryption Standard (DES) against non-linear polynomial invariant attacks. Is this sort of attack also possible for DES? We present a simple proof of concept attack on DES where a product of 5 polynomials is an invariant for 2 rounds of DES. Furthermore we present numerous additional examples of invariants with higher degrees. We analyse the success probability when the Boolean functions are chosen at random and compare to DES S-boxes. For more complex higher degree attacks the difficulties disappear progressively and up to 100 % of all Boolean functions in 6 variables are potentially vulnerable. A major limitation for all our attacks, is that they work only for a fraction of the key space. However in some cases, this fraction of the key space is very large for the full 16-round DES.

Key Words: block ciphers, Feistel ciphers, DES, weak keys, history of cryptography, algebraic cryptanalysis, generalized linear cryptanalysis, polynomial invariants, annihilator space, Boolean functions, k-normality.

1 Introduction

Block ciphers have occupied a dominating position in the applied cryptography space since the 1970s. Their iterated structure is (specifically) prone to round invariant attacks, for example in the form of Linear Cryptanalysis (LC) and Generalised Linear Cryptanalysis (GLC), cf. Eurocrypt'95 [29]. The space for possible attacks grows double-exponentially and researchers have found, until recently, extremely few such attacks [21, 34, 19, 4]. In many non-linear polynomial invariant attacks we obtain two polynomials which after substitution of the equations describing one round of the block cipher will become identical. Some such invariant attacks seem to happen by some coincidence and very few similar attacks are known or even expected to exist. Other, better attacks are such that they depend on events which happen with a larger probability, and they can be applied in a wide variety of cases or/and many similar attacks can be found. For example recent research suggests that attacks where we multiply several polynomials are potentially quite powerful [20, 14]. There are some specific reasons why such attacks work well. When we multiply several Boolean polynomials, it is possible to see that the attacker only needs to insure a number of bits to

be identical for a fraction of the input space. Another reason is that the ring of Boolean functions offers numerous possibilities of eliminating whole complex multivariate functions just by multiplying them with well-chosen polynomials.

Initially this "product" invariant attack was designed and studied for T-310 [32]. Eventually, though later and with some difficulties, it is claimed that this attack may also be applied to DES [20]. Previous non-linear attacks on DES used polynomials of degree 2 [21] and were not very successful. Also, many previous non-linear attack on other ciphers were of degree 2 or 3, see [34, 16]. With older results for DES, in the best case we obtained a probabilistic attack which was a very slight improvement over the best linear attack by Matsui [21]. In general in all known non-linear attack the invariants are existential over the key, no attack is known where the fraction of the key space for which such the attack works would be equal 100 %. This limitation will remain valid in the present article.

In our research we argue that the problem should be turned upside down and approached from the opposite end: essentially considering that the S-boxes are variable. Given any fixed polynomial $\mathcal{P}$, under what conditions the value of $\mathcal{P}$ is preserved for 1 or 2 rounds with probability 1, i.e. for every input. We need to understand what makes such attacks possible. We also need to push the attacks to their limits: for example, trying to research what is the lowest possible degree for which a non-linear attack for DES with the original P-box and modified S-boxes can be constructed. We need simple examples which are intelligible so that we can understand better what makes non-linear attacks possible (or not) and what are their limitations in terms of success probability are, when the DES key is chosen at random. With this approach we are able to produce a first proof of concept non-linear attack on DES with a polynomial of degree 5 and numerous further attacks at higher degrees.

1.1 Outline

This article is organised as follows. In Section 2 we look at the questions of research methodology. In Section 3 we introduce our notations used for DES and present a very simple invariant attack of degree 2. In Section 4 we introduce the question of "closed loop" configuration and on this basis in Section 5 we construct a simple attack of degree 5. In Section 5.3 we show that the same attack is actually impossible if we assumed that all our Boolean functions need to be balanced. In Section 6 we present an attack at degree 12 where S-boxes are more similar to real DES S-boxes. Then in Sections 6.4. to 7.3. we study how high quality Boolean functions with many zeros inside the Walsh spectrum and lack of k-normality may prevent our attacks. Nevertheless in Section 8 we will see that with cubic annihilators the attack becomes hard to avoid. In Section 9 we outline a more general attack in relation to a weaker security notion for Boolean functions.

2 Methodology, Assumptions and Related Research

In our attacks we allow the attacker to modify the DES S-boxes in arbitrary ways, which raises interesting questions about backdooring. Moreover we will

not[1] require that the attack works for 100% of the keys. In general, in the type of attacks we study, we expect that the "power" of the attack improves substantially[2] with the increase of the degree of the polynomial invariant. Given a specific polynomial attack with a fixed polynomial $\mathcal{P}$ of a certain degree, how do we measure how powerful the attack is? In general this question has two primary dimensions: when the S-box varies, and when the secret key varies. In DES the two questions are very closely related. Modifying the key for one S-box is equivalent to translating its input by a XOR with a 6-bit constant. We get two questions subject to probabilities. First, what is the probability, which we will call p_1 type result for simplicity, that a Boolean function chosen at random inside our cipher allows the attack to succeed? The second question is: if we modify the key at the input of the same S-box such that the attack works, what is the probability, which we could call p_2, that the attack still works? Finally there is also a global optimization question: designing an attack which involves several S-boxes, where several probabilities of type p_1 or p_2 are multiplied.

The examples of attacks in this paper have been chosen for their simplicity and elegance. Our main goal is to show the feasibility of this type of attacks and show that simple attacks are possible with degree as low as 5. This could be seen as taking the quadratic attacks in [21] to a new level. We will show that at degree 5 however, there extremely few Boolean functions s.t. the attack works, or in other terms, the p_1 type probabilities are very low. Then we will see that as the degree grows, the number of possible attacks and success probabilities will improve very substantially.

This paper is a proof of concept. We think it is too early to study how such attacks can be applied to decrypt communications encrypted with DES. This question is studied in Section 9 in [20] and in Section 6 in [16].

2.1 Discovery of Advanced Non-Linear Invariant Attacks

In recent research there exist two major types of invariant attacks: linear subspace invariants [30, 3, 7], and proper non-linear polynomial invariants [34, 19], which are somewhat more general. Several authors [3, 7, 19, 4] including inside the present work study both. Our product attack, is also a linear subspace attack when all the polynomials in the product are affine polynomials, which is the case here, and frequently also the case elsewhere e.g. in [15, 14, 20]. However the product attack is NOT yet the most powerful attack. In general we work in polynomial rings, where both addition and multiplication are allowed, and the general form of an invariant attack is a sum of several products.

The existence of some invariant properties does not imply that they can be found or computed. Finding such properties was so far considered as very hard. There are two major approaches to our problem: combinatorial and algebraic. A nice algebraic approach is through solving the so called Fundamental Equation (FE) cf. [19]. Solving such equation(s), or rather several equations simultaneously, **guarantees** that we obtain a Boolean function and the polynomial

[1] This seems inevitable already from the study of older attacks in [21].
[2] This phenomenon is sometimes called "phase transition", cf. [20, 13].

invariant $\mathcal{P}$, which propagates for any number of rounds. However this equation can be very complex and nothing guarantees that the FE has any solutions.

2.2 On the Bootstrapping Problem in Cryptanalysis

Research in block cipher cryptanalysis has suffered from a bootstrapping problem: we have hardly ever found any invariant attacks, except when the set of all possible attacks is not too large, e.g. in Linear Cryptanalysis (LC). An excessively rich space of attacks has been ignored, and we could not find many interesting attacks, because we failed to see or imagine how new attacks could look like. New examples of working attacks (to imitate in further attacks) are crucial. Further discussion of this question can be found in Section 1.8 of [19].

Essential insights about what makes non-linear invariant attacks actually possible can also be found in [14]. The whole idea that "product attacks' work well should makes us reflect on why and when two products of k complex polynomials can actually be equal. Here the lack of unique factorisation inside the ring of Boolean polynomials plays an important role. There exist also numerous opportunities where polynomials can be eliminated, for example through annihilation: a polynomial Z is not zero however a product fZ is 0 for every inputs. In other terms Z disappears after multiplying by another Boolean polynomial.

3 Polynomial Invariant Attacks on Block Ciphers

We call $\mathcal{P}$ a polynomial invariant, if the value of $\mathcal{P}$ is preserved after one round of encryption, i.e. if $\mathcal{P}(\text{Inputs}) = \mathcal{P}(\text{Outputs})$. An important point is then, that any block cipher round translates into relatively simple Boolean polynomials, if we look at just one round of encryption. In general we can denote this polynomial mapping by ϕ(Inputs. If we express round outputs as polynomials written in their standard Algebraic Normal Form (ANF) we get that

$$\mathcal{P}(\text{Inputs}) = \mathcal{P}(\text{Output ANF})$$

which is now a formal equality of two polynomials except that the polynomial on the right hand side is potentially ambiguous or it contains potentially additional inputs such as key bits. We further call this transformed polynomial $\mathcal{P}^\phi$ with:

$$\mathcal{P}^\phi \stackrel{def}{=} \mathcal{P} \circ \phi = \mathcal{P}(\text{Output ANF}) = \mathcal{P}(\phi(\text{Inputs}))$$

This ambiguity should be as small as possible: we simply want our polynomial $\mathcal{P}^\phi$ to depend on only few key bits, and preferably on none at all. The choice of $\mathcal{P}$ is therefore a crucial task. The attacker chooses this polynomial very carefully in order to avoid having to deal with too many key bits.

This concept can be applied to any block cipher, however finding a suitable $\mathcal{P}$ is notoriously a difficult task cf. [3]. If an attack if found it will propagate for any number of rounds, this if it is actually independent of the key and other bits which is frequently obtained for certain weaker ciphers such as T-310, cf. [19]. This is no longer the case in this article. DES turns out to be a substantially stronger cipher.

In general in our research we consider that the Boolean functions are unknown. We denote such functions by a special variables such as $Z5$ or $W1$ where $Z5$ will be the last output of S-box S5, cf. Fig. 1. Instead of considering that our cipher has a key, we will rather consider that the key is an operation translating one S-box into another, and if the attack works in one case, it might also work for another key.

3.1 Non-Linear Attacks on DES

In this article we show how a product attack can be applied to DES. This result has some historical significance, as DES is the most widely used cipher of all times. In Eastern Germany a modified DES was known under the name LAMBDA1 [23] and was implemented around 1990 inside a portable electronic cipher machine T-316. Triple DES is still widely used today in financial applications. In DES the number of inputs of each Boolean function is 6 and we concentrate our efforts on this case. This makes the analysis of our attacks similar as with T-310 block cipher cf. [19, 20]. In T-310 we have 4 identical Boolean functions in each encryption round and 9 new bits are produced out of 36 for the full cipher state. In DES we have 32 distinct Boolean functions in each encryption round and 32 new bits are produced out of 64 for the full cipher state. A considerable difficulty for the attacker in DES (see the conclusion in [19] and in [20]) is that DES has a large number of key bits to take into account (48 in each round). However our invariant attacks use very few bits from the cipher state and will depend on very few key bits. We consider the usual structure of DES with duplication of the bits at the boundaries of the S-boxes, cf. Fig. 1 below.

3.2 Notation

We denote by

$$\mathrm{L01}, \ldots, \mathrm{L32}; \mathrm{R01}, \ldots, \mathrm{R32}$$

the inputs of one rounds of DES. The same notations will be also used for the outputs and when it is needed to distinguish between different instances of the same variable we will use exponents, for example $\mathrm{L05}^i$ will be the 5-th input in one round and $\mathrm{L05}^o$ will be the 5-th output bit. Let I_{1-32} be the input of the DES round function, and let O_{1-32} be the output of the DES round function. In the first round we have $I_1 = \mathrm{R01}$ where $\mathrm{R01}^i$ denotes the first input on the right side, which will be sometimes denoted simply by R01 if there is no ambiguity. Similarly the notation $\mathrm{R02}^o$ denotes the second bit on the right hand side on the output side of the cipher. If our encryption is performed for 1 round only we have $\mathrm{R02}^o = O_1 + \mathrm{L02}^o$ where $+$ will always denote addition modulo 2 (when used for binary variables). Finally for one round of the DES Feistel scheme we have $\mathrm{L01}^o = \mathrm{R01}^i$ and the same applies for all 32 bits on the right side at the input.

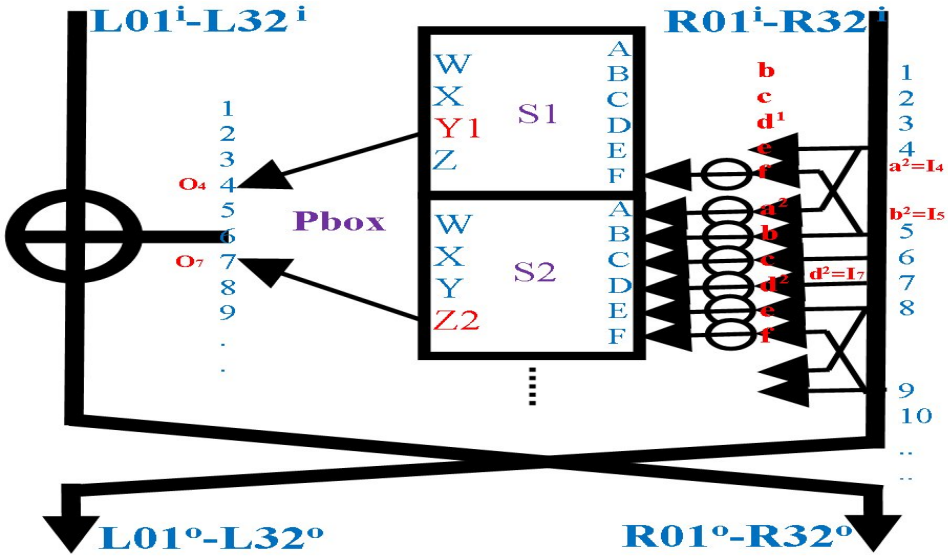

Fig. 1. One round of DES.

In what follows we are going to construct several examples of a "product attack" on DES.

We also introduce the following notation in order to simplify our polynomial expressions. We are going to denote by $\mathrm{OP}_i(.)$ the output polynomial which is connected to output O_i. This polynomial is always one of the 4 outputs $\mathrm{W}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}$ for exactly one of the S-boxes, and has 6 input variables $a - f$, which are also 6 consecutive variables of type I_{1-32} (with wrap-around). Our picture in Fig. 1 focuses on the pair of bits 4 and 7, showing that they could potentially be connected to any pair of outputs, depending on the P-box. For example output 4 could be connected to the third output Y^1 of the first S-box, which will be later called also simply Y1. Then output 7 could be connected to the last output of the second S-box, here denoted by Z^2 and sometimes also denoted by Z2. Later we will write that $\mathrm{OP}_4(\mathrm{R01}, \ldots, \mathrm{R32}) = \mathrm{Y2}(a, b, c, d, e, f)$ and also that $\mathrm{OP}_7(\mathrm{R01}, \ldots, \mathrm{R32}) = \mathrm{Z}(a, b, c, d, e, f)$, ignoring for the time being the key bits completely or considering that all key bits are zero. This is not quite accurate if we look at the actual P-box used in DES, see later Fig. 7, where output 4 and 7 are actually equal to Boolean functions W6 and Z7. However, we do not yet assume that we use the original DES P-box, and at this stage we assume that the P-box could have been modified by the attacker. For example in order to obtain a strong yet extremely simple attack, a sort of toy example, cf. Thm. 3.5 below.

We will consider arbitrary S-boxes, i.e. arbitrary sets of 32 Boolean functions which depend on the secret key of an arbitrary length in an arbitrary way. We denote inputs of each S-box by letters $\mathrm{A}, \ldots, \mathrm{F}$ in blue in our figure. The corresponding inputs before key whitening are denoted by more precise notations

$a^k, \ldots, f^k$ for S-box k, $k = 1 - 8$, which appear in red in our figure. When there is no ambiguity the same letters will be denoted simply by letters $a - f$.

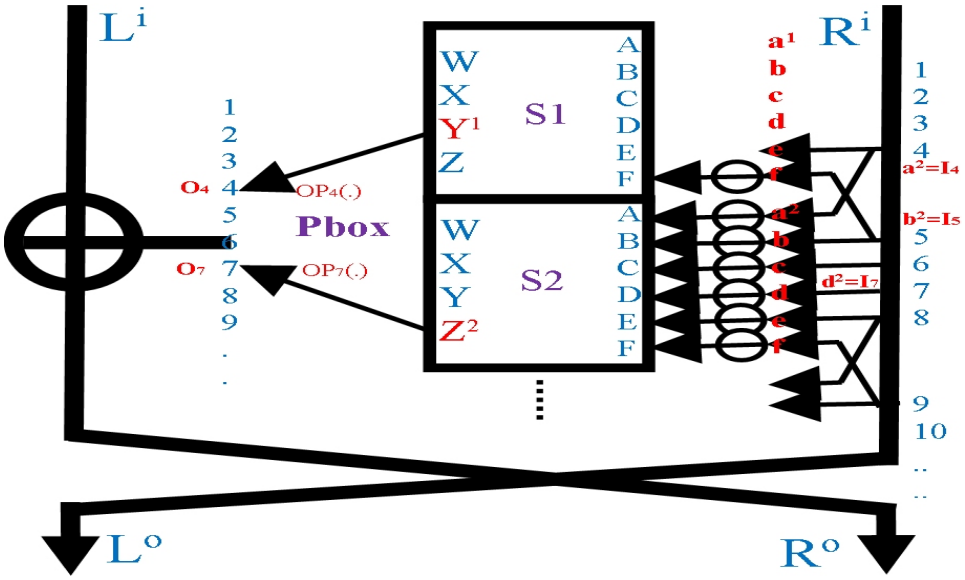

Fig. 2. One round of DES using OP_i notation.

3.3 A Basic Product Attack with 2+2 Active Bits

Our final goal will be at the end to obtain two polynomials which become equal, i.e. $\mathcal{P}(\text{Input}) = \mathcal{P}(\text{Output ANF})$, this if we make a number of assumptions. First we work on just one pair of bits for one S-box, for example S2, and consider inputs a, d of S2 which are also simply I_4 and I_7. We have:

$$\text{R04}^i = a^2$$
$$\text{R07}^i = d^2$$

Now we define the following 2 polynomials:

$$\begin{cases} \mathcal{A} \stackrel{def}{=} (\text{R04} + \text{R07}) & \text{which is right bits } 4, 7 \\ \mathcal{B} \stackrel{def}{=} (\text{L04} + \text{L07}) & \text{which is left bits } 4, 7 \end{cases}$$

We then consider how these polynomials compare at both input/output sides denoted by 'exponent' indices i and o. Since that example involves only one S-box, we omit the exponent on a, d.

$$\begin{cases} \mathcal{A}^i = (\text{R04}^i + \text{R07}^i) = a + d \\ \mathcal{B}^i = (\text{L04}^i + \text{L07}^i) \\ \mathcal{A}^o = (\text{L04}^o + \text{L07}^o) = (\text{L04}^i + \text{L07}^i + \text{OP}_4(.) + \text{OP}_7(.)) \\ \mathcal{B}^o = \mathcal{P}^i = a + d \end{cases}$$

From here we have:

$$\begin{cases} \mathcal{A}^i\mathcal{B}^i = (a+d)(\mathrm{L04}^i + \mathrm{L07}^i) \\ \mathcal{A}^o\mathcal{B}^o = (a+d)(\mathrm{L04}^i + \mathrm{L07}^i + \mathrm{OP}_4(.) + \mathrm{OP}_7(.)) \end{cases}$$

Could these two polynomials be identical? Yes, if the sum of polynomials $\mathrm{OP}_4(.) + \mathrm{OP}_7(.)$ can be annihilated by $(a+d)$. Interestingly this cancellation condition absolutely does NOT depend on inputs $\mathrm{L04}^i$ and $\mathrm{L07}^i$ and is only a property of the round function polynomials. Following [19], we can sum up the requirements for $\mathcal{AB}$ to be an invariant with the "Fundamental Equation" (FE) for our choice of two variables $(4, 7)$:

$$FE_{4,7} = \mathcal{A}^i\mathcal{B}^i + \mathcal{A}^o\mathcal{B}^o = (a+d)(\mathrm{OP}_4(.) + \mathrm{OP}_7(.))$$

When this equation collapses to a polynomial which is always zero (for any input), we obtain a working invariant attack.

3.4 How To Make this Attack Work

It is easy to see that if our DES P-box is very weak, an attack becomes possible. In order for $\mathcal{AB}$ to be an invariant polynomial after any number of rounds, we need to use a Boolean function which is annihilated by $a+d$ which requires that the two inputs a, d for BOTH $\mathrm{OP}_4(.)$ and $\mathrm{OP}_7(.)$ come from the same S-box. Let

$$\mathcal{P} = \mathrm{L04} * \mathrm{R07}$$

which is a non-zero polynomial of degree 2.

Theorem 3.5 (Simple Bi-Linear Attack for 1R). We assume that our S-box satisfies the following 2 conditions:

$$\begin{cases} (a+b) * Y = 0 \\ (a+b) * Z = 0 \end{cases}$$

where Y denotes the Boolean function connected to round output 4 in one round function, i.e. $\mathrm{OP}_4(\mathrm{R01}, \ldots, \mathrm{R32}) = Y(a, b, c, d, e, f)$ and also $\mathrm{OP}_7(\mathrm{R01}, \ldots, \mathrm{R32}) = Z(a, b, c, d, e, f)$ where the relevant 6 inputs are renamed $a - f$, and the P-box is such that $\mathrm{R04}, \mathrm{R07}$ are some of the $a - f$.
Then $\mathcal{P}$ is an invariant for one round of DES.

Proof of Thm. 3.5: The proof is trivial given that

$$FE_{4,7} = \mathcal{A}^i\mathcal{B}^i + \mathcal{A}^o\mathcal{B}^o = (a+d)(\mathrm{OP}_4(.) + \mathrm{OP}_7(.))$$

is zero for any input.

Clarification. This is a toy example which actually requires that 4 and 7 are simultaneously inputs and outputs of the same S-box S2, which does not happen

for the original DES P-box. This example cannot be considered as a valid attack on DES.

The next question is how to find or construct polynomials $\mathcal{P}$, such that they could be invariants for, say, 1 or 2 rounds with the original DES P-box but with modified S-boxes.

4 Closed Loop Configurations or Key Property Which Makes a Cipher Vulnerable to Non-Linear Invariant Attacks

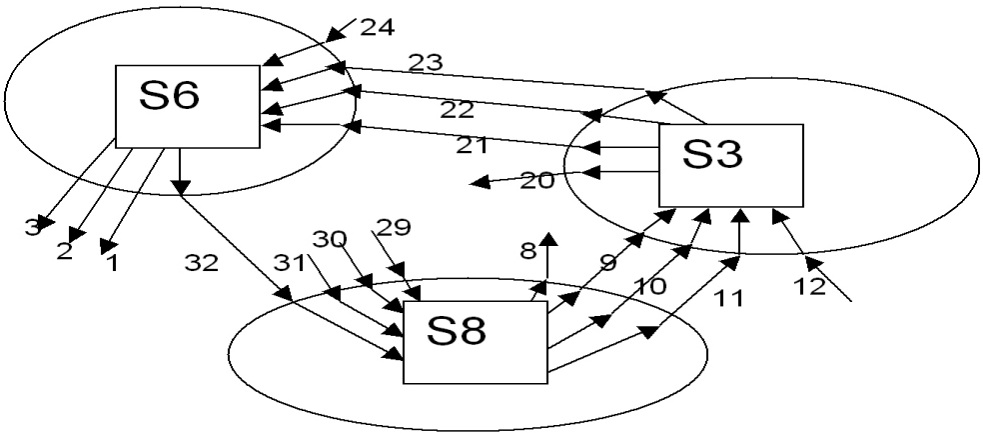

Fig. 3. Closed-loop connection between S-boxes S3,S6,S8 in GOST cf. Fig. 4 in [22].

Our experience shows that the primary problem in finding "interesting" non-linear invariant attacks is to find a configuration, where some set of bits and S-boxes are primarily connected to each other in closed loops, and the term "closed-loop invariants" is used in a very recent work [35]. This idea is not new: for T-310 it was also studied in [19] and for GOST it was already studied in [22], cf. Fig. 3.

The sets of S-boxes involved in all non-linear attacks on DES are also of this type: they have been constructed precisely and deliberately from such sets. Similar properties are expected to exist for other ciphers, for example PP-1 cf. [24], and to some extent for all block ciphers (unless the diffusion is very strong).

5 A Proof of Concept of the Applicability of Our Product Attack to DES - An Attack of Degree 5

Below we present a complete example of a polynomial invariant property for DES with a simple product of linear polynomials and the original P-box. Our proof of concept is based on a well-chosen invariant polynomial $\mathcal{P}$, which was found after extensive trial and error based on the idea of closed cycles involving 6 inputs/outputs of the DES round function, as illustrated in Fig. 4. However this

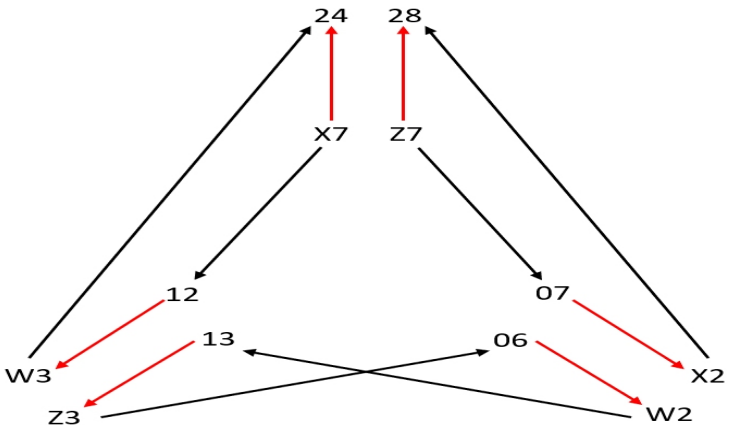

Fig. 4. Closed-loop connection between S-boxes S2,S3,S7 in DES.

leads in general to arbitrary polynomials of degrees up to 12, as each variable in Fig. 4 has two versions (left and right). In fact it is NOT obvious if any invariant properties of degree $5 \ll 12$ can be found or constructed[3] whatsoever. Our polynomial is as follows:

$$\mathcal{P} = (1 + \mathrm{L06} + \mathrm{L07}) * \mathrm{L12} \ * \ \mathrm{R13} * \mathrm{R24} * \mathrm{R28}$$

which is a non-zero polynomial of degree 5.

Related Research. A similar attack for 2 rounds with degree 5 and with factors which split into two groups relevant in the 1st or in the 2nd round was published recently in [26] with a similar polynomial $\mathrm{R05} * \mathrm{L07} * (\mathrm{R28} + 1) * (\mathrm{L27} + 1) * \mathrm{L32}$. The attack below is extremely similar and was found by our student Hamy Ratoanina few months earlier than the attack of [26]. We have also obtained generated a large number of attacks of this type with increasing degrees, some of which are listed in Section 7.4 and in Appendix A. The focus of the explorations have been so far primarily attacks where single variables appear as factors. In general we could also have factors which are sums of variables and a stronger and more general attack exists and is outlined in Section 9 following the methodology described in [26].

We go back to our main result with degree 5.

Notation: When we write

$$(1 + c + d) * \mathrm{W2} == 0$$

we mean that the polynomial $(1 + c + d)$ annihilates[4] the 1st output W of the second S-box S2.

[3] Moreover it is not obvious either if any invariant properties at all exist, as we do not have yet the equivalent of [18] for block ciphers, i.e. we are not able to prove that a polynomial attack always exists (in the worst case).

[4] The sign $==$ is used here to denote formal equality of polynomials in 6 variables, which at other places can also be denoted by $\equiv$ or just $=$ if there is no ambiguity.

Theorem 5.1 (A Simple Degree 5 Invariant Attack On 2 Rounds of DES). Let $\mathcal{P}$ be as above, and we assume that our three active[5] S-boxes (S2, S3 and S7), which include the key[6], and satisfy the following 6 conditions:

$$\begin{cases} (1+c+d) * \mathrm{W2} == 0 \\ (1+c+d) * \mathrm{X2} == 0 \\ e * \mathrm{W3} == 0 \\ f * \mathrm{Z3} == 0 \\ ae * \mathrm{X7} == 0 \\ ae * \mathrm{Z7} == 0 \end{cases}$$

Then $\mathcal{P}$ is an invariant for two rounds of DES.

Proof of Thm. 5.1:

We want to prove that for any input of the cipher

$$\mathcal{P}(\mathrm{Inputs}) = \mathcal{P}(\mathrm{Outputs})$$

where *Outputs* denotes the output of the cipher after 2 rounds. This is done ignoring the secret key bits completely. We either consider that they are all at zero, or they are considered to be a part of the S-box, and basically the attack should work for all S-boxes (comprising a possible input translation by the key bits) which satisfy our assumptions. In order to study this question we will introduce some auxiliary notations. For each variable such as L03 we have three instances of it: $\mathrm{L03}^i$ is the variable on the input side, $\mathrm{L03}^m$ will be the middle variable after 1 round of encryption, and finally $\mathrm{L03}^o$ will be the variable on the output side. We have 3 times 64 of these state variables. We will denote $\mathcal{P}^i = \mathcal{P}(\mathrm{Inputs})$ the value of the original polynomial applied at the input side, or formally this polynomial instantiated with the input side variables. Similarly $\mathcal{P}^o = \mathcal{P}(\mathit{Outputs})$ will be the same polynomial written at the output side, or formally the same polynomial instantiated with the output side variables. We assume that for any key one round of DES is a bijection denoted by ϕ, temporarily ignoring that in general ϕ depends on the secret key. Our reference set of variables will be the 64 variables in the middle, which are:

$$\mathcal{M} = (\mathrm{L01}^{\mathrm{m}}, \ldots \mathrm{L32}^{\mathrm{m}}; \mathrm{R01}^{\mathrm{m}}, \ldots \mathrm{R32}^{\mathrm{m}})$$

In order to prove that $\mathcal{P}^i = \mathcal{P}^o$ for any input, it is sufficient to prove that $\mathcal{P}^i = \mathcal{P}^o$ for any middle 64 variables $\mathrm{L01}^{\mathrm{m}} \ldots \mathrm{R32}^{\mathrm{m}}$. In order to show that we are going to express both values of our polynomial on both sides $\mathcal{P}^i$ and $\mathcal{P}^o$ as polynomials in these 64 middle variables and show that these two polynomials are simply exactly equal. This becomes a straightforward exercise, which amounts to substituting 64 native variables at the output inside $\mathcal{P}^o$ by their polynomial

[5] The content of the remaining five S-boxes can be arbitrary.

[6] We simply assume that the secret key in DES is implemented inside these S-boxes, transforming their input by a bitwise XOR.

ANF expressions in the 64 middle variables which comes from encryption with ϕ, i.e.

$$\mathcal{P}^o = \mathcal{P}(\phi(\mathcal{M}))$$

where $\phi(\mathcal{M})$ denotes a sequence of 64 polynomial expressions of $(\mathrm{L01^o}, \ldots, \mathrm{R32^o})$ expressed as polynomials in the 64 middle variables of $\mathcal{M}$. In the same way we express the 64 native variables of the input inside $\mathcal{P}^i$ by their polynomial ANF expressions in the middle 64 variables in $\mathcal{M}$ which come from decryption, i.e.

$$\mathcal{P}^i = \mathcal{P}(\phi^{-1}(\mathcal{M}))$$

where $\phi^{-1}(\mathcal{M})$ denotes a sequence of polynomial expressions of $(\mathrm{L01^i}, \ldots, \mathrm{R32^i})$ as a function of the 64 middle variables of $\mathcal{M}$. Half of these transformations are trivial. For example the variable L01 at the output is always replaced by the polynomial R01 at the input, and in general DES is a Feistel cipher and all right bits are preserved and become left bits in the next round. We recall that:

$$\mathcal{P}^i = (1 + \mathrm{L06} + \mathrm{L07}) * \mathrm{L12} \ * \ \mathrm{R13} * \mathrm{R24} * \mathrm{R28}$$

and we observe that:

$$\mathcal{P}^o = (1 + \mathrm{L06^o} + \mathrm{L07^o}) * \mathrm{L12^o} \ * \ \mathrm{R13^o} * \mathrm{R24^o} * \mathrm{R28^o}$$

Now we use the internal structure of DES cf. Fig. 4. This is based on the more detailed Fig. 7 which appears below, depicting the standard DES P-box. We can now to write down the exact Boolean functions needed in one round in order to compute the round outputs we use here. We obtain the following expression using middle variables only:

$$\mathcal{P}^o = (1 + \mathrm{R06^m} + \mathrm{R07^m}) * \mathrm{R12^m} \ * \ (\mathrm{L13^m} + \mathrm{W2}) * (\mathrm{L24^m} + \mathrm{W3}) * (\mathrm{L28^m} + \mathrm{X2})$$

In Fig. 5 we represent our whole proof graphically where the colour coding is the same as in Fig. 6.

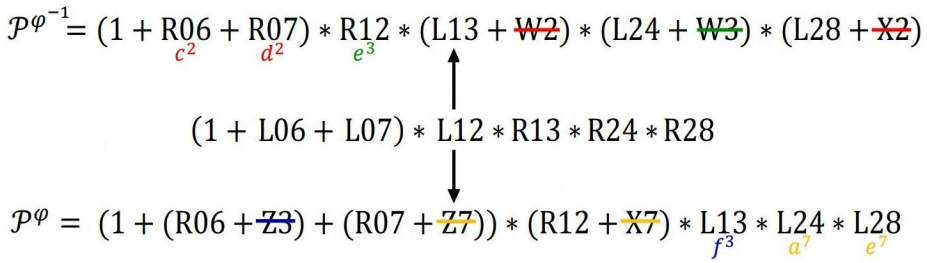

Fig. 5. A visual representation of our proof with colour coding, cf. Fig. 6.

Now we are going to rewrite our 6 assumptions knowing that inputs $abcdef$ of S1 are, in order, R32 up to R05. This is for 1 round of DES and assuming that the key is included as a part of the S-box and therefore we don't need (yet) to worry about what happens when the key changes.

$$\mathcal{P} = (1 + \mathrm{L06} + \mathrm{L07}) * \mathrm{L12} * \mathrm{R13} * \mathrm{R24} * \mathrm{R28}$$

$$\begin{cases} 1 + \mathrm{c} + \mathrm{d} & \text{is an annihilator for W2 and X2} \quad (1) \\ \mathrm{e} & \text{is an annihilator for W3} \quad (2) \\ \mathrm{f} & \text{is an annihilator for Z3} \quad (3) \\ \mathrm{ae} & \text{is an annihilator for X7 and Z7} \quad (4) \end{cases}$$

Fig. 6. Assumptions in our attack.

Here we work on 2 rounds, so it is important to remember that W2 in the second round encryption (studied here) is different than W2 in the first round of encryption (studied later).

We obtain for the inputs of the second encryption round:

$$\begin{cases} (1 + c + d) * \mathrm{W2} = 0 \text{ becomes } (1 + \mathrm{R06} + \mathrm{R07}) * \mathrm{W2} = 0 \textit{ in the 2nd round} \\ (1 + c + d) * \mathrm{X2} = 0 \text{ becomes } (1 + \mathrm{R06} + \mathrm{R07}) * \mathrm{X2} = 0 \textit{ in the 2nd round} \\ e * \mathrm{W3} = 0 \text{ becomes } \mathrm{R12} * \mathrm{W3} = 0 \textit{ in the 2nd round} \end{cases}$$

We see that W3 can be simply erased from our last product because $\mathrm{R12}^m$ is a factor in the whole product. Therefore the difference is a multiple of $\mathrm{R12}^{\mathrm{m}} * \mathrm{W3}$ which polynomial is zero (for any input). In the same way we can simply erase Z3. Then we can also just erase W2 and X2, because $(1 + \mathrm{R6}^{\mathrm{m}} + \mathrm{R7}^{\mathrm{m}})$ is a factor of our product.

We get the following equality:

$$\mathcal{P}^o = (1 + \mathrm{R06}^{\mathrm{m}} + \mathrm{R07}^{\mathrm{m}}) * \mathrm{R12}^{\mathrm{m}} \ * \ \mathrm{L13}^{\mathrm{m}} * \mathrm{L24}^{\mathrm{m}} * \mathrm{L28}^{\mathrm{m}}$$

Now we are going to work on the first round of encryption and backwards, starting from the middle state $\mathcal{M}$. We have

$$\mathcal{P}^i = (1 + \mathrm{L06}^{\mathrm{i}} + \mathrm{L07}^{\mathrm{i}}) * \mathrm{L12}^{\mathrm{i}} \ * \ \mathrm{R13}^{\mathrm{i}} * \mathrm{R24}^{\mathrm{i}} * \mathrm{R28}^{\mathrm{i}}$$

We obtain for the outputs of the first encryption round which is applied in the backwards direction:

$$\begin{cases} f * \mathrm{Z3} = 0 \text{ becomes } \mathrm{R13} * \mathrm{Z3} = 0 \textit{ in the 1st round} \\ ae * \mathrm{X7} = 0 \text{ becomes } \mathrm{R24} * \mathrm{R28} * \mathrm{X7} = 0 \textit{ in the 1st round} \\ ae * \mathrm{Z7} = 0 \text{ becomes } \mathrm{R24} * \mathrm{R28} * \mathrm{Z7} = 0 \textit{ in the 1st round} \end{cases}$$

which is equal to

$$(1 + \mathrm{R06}^{\mathrm{m}} + \mathrm{Z3} + \mathrm{R07}^{\mathrm{m}} + \mathrm{Z7}) * (\mathrm{R12} + \mathrm{X7}) \ * \ \mathrm{L13}^{\mathrm{m}} * \mathrm{L24}^{\mathrm{m}} * \mathrm{L28}^{\mathrm{m}}$$

Here we can erase X7 and Z7, because $\mathrm{L24}^{\mathrm{m}} * \mathrm{L28}^{\mathrm{m}}$ is a factor of the whole product, which is equal to $\mathrm{R24}^{\mathrm{i}} * \mathrm{R28}^{\mathrm{i}}$, which are inputs of the first round. We can also erase Z3, because it is annihilated by $\mathrm{R13}^{\mathrm{i}}$, which is present here disguised as $\mathrm{L13}^{\mathrm{m}}$. Thus finally we obtain:

$$\mathcal{P}^i = (1 + \mathrm{R06}^{\mathrm{m}} + \mathrm{R07}^{\mathrm{m}}) * \mathrm{R12}^{\mathrm{m}} \ * \ \mathrm{L13}^{\mathrm{m}} * \mathrm{L24}^{\mathrm{m}} * \mathrm{L28}^{\mathrm{m}}$$

which completes the proof that both polynomials are equal for 2 rounds of DES (only if the S-boxes with the key satisfy our 6 assumptions in both rounds).

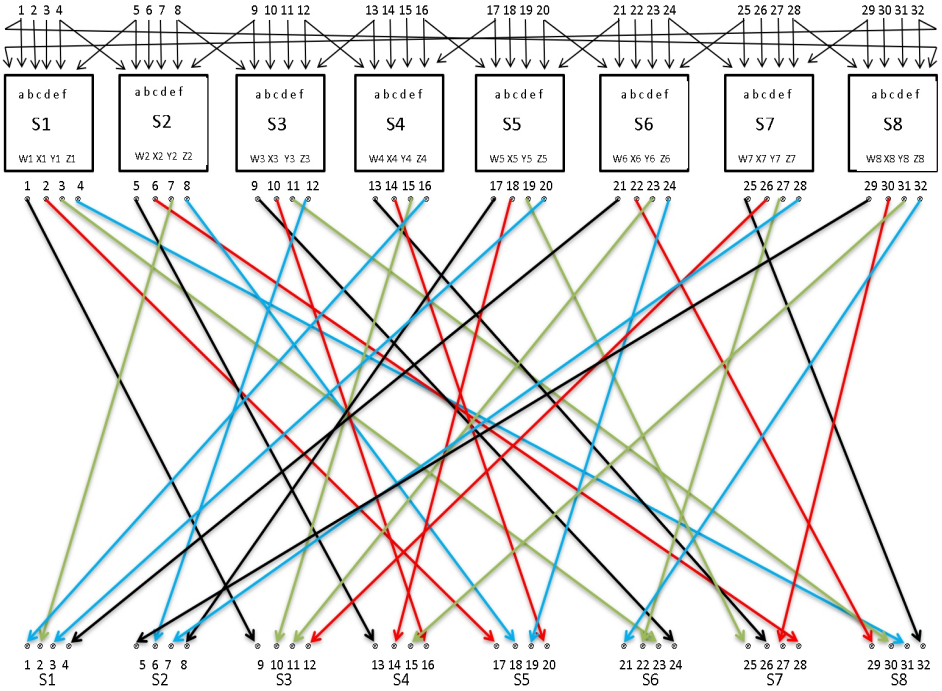

Fig. 7. Full round function of DES showing connections with S-boxes in the next round.

5.2 Important Remark

It may seem that our proof also proves that $\mathcal{P}$ is an invariant for 1 round. However a quick examination shows that the two sides R and L get swapped[7] after 1 round. If we denote by $\mathcal{P}'$ the symmetric version of $\mathcal{P}$ above, then it is easy to see that $\mathcal{P} + \mathcal{P}'$ is an invariant of degree 5 for 1 round.

5.3 How Powerful is Current Attack?

It may seem that we are able to backdoor DES by modifying certain S-boxes (and nothing else). Unfortunately the current result is extremely weak. We recall the following result, which is adapted from Theorem 6.4 in [26].

Theorem 5.4 (Impossibility result for Balancedness). It is not possible to generate a set of S-boxes are required by Thm. 5.1 in such a way, that these Boolean functions are simultaneously balanced and non-linear. This impossibility result holds for any key (there exists no key for which our conditions could be satisfied).

Proof of Thm. 5.4: This is because at least one of the pre-conditions required was of the form $f(a, b, c, d, e, f) * Z(a, b, c, d, e, f) = 0$ where both f and Z are required to be balanced. For example we had:

[7] This is closely related to the question of reflection attacks in GOST, cf. [25].

$$(1 + \mathrm{c} + \mathrm{d}) * \mathrm{W2} = 0$$

with $f(a, b, c, d, e, f) = 1 + c + d$ which is balanced and affine. This is not possible and a simple counting argument shows that if both f and Z are balanced, then $f * Z = 0$ implies that $f = Z + 1$ for any input. To see this we observe that for some 2^{6-1} inputs we have $f = 1$ and for all those we must have $Z = 0$. Now since Z is balanced, it must be 1 on all the remaining 2^{6-1} inputs and it is completely determined in this way, and we have $f = Z + 1$ for any input. Finally since f is affine, Z also must be affine. This contradiction ends our proof. □

Remark. We see that our attack will not work in general, for all non-linear balanced S-boxes. Then obviously it will not work either when we transform their inputs by a secret key. Our methodology is such that attacks are in general existential over the key, i.e. work for a fraction of the key space. However any impossibility result such as above will hold for any key.

Next Steps. An interesting open problem is the minimum degree d, such that a product of polynomials of degree d is an invariant for X rounds of DES for some S-boxes, which would be both balanced and non-linear. In the next section we show that this can be achieved with degree being at most 12.

Future Research. This question extends to all known S-box quality measures, such as non-linearity, algebraic degree, algebraic immunity, correlation immunity, multiplicative complexity etc, cf. [2] and adding more criteria will probably further increase the degree d required.

6 A Non-Linear Attack on DES with Balanced Boolean Functions

We are now going to show that if we increase the degree from 5 to about[8] 12, the Boolean functions can be balanced and non-linear. This example was found after extensive trial and error based on the idea of closed cycles and starting from using exactly the same set of 6 bits and 3 S-boxes as before, cf. Fig. 4. Our polynomial is as follows:

$$\mathcal{P} = \mathrm{L06} * \mathrm{R06} * \mathrm{L07} * \mathrm{R07} \ * \ \mathrm{L12} * \mathrm{R12} * \mathrm{L13} * \mathrm{R13} \ * \ \mathrm{L24} * \mathrm{R24} * \mathrm{L28} * \mathrm{R28}$$

which is a non-zero polynomial of degree 12.

Theorem 6.1 (A Balanced Degree 12 Invariant Attack On 1 Round of DES). Let $\mathcal{P}$ be as above, and we assume that our three active S-boxes, which include the key[9], satisfy the following 6 conditions:

$$\begin{cases} cd * \mathrm{W2} = 0 \\ cd * \mathrm{X2} = 0 \\ ef * \mathrm{W3} = 0 \\ ef * \mathrm{Z3} = 0 \\ ae * \mathrm{X7} = 0 \\ ae * \mathrm{Z7} = 0 \end{cases}$$

Then $\mathcal{P}$ is an invariant for 1 round of DES.

Proof of Thm. 6.1: This proof is substantially simpler than the previous proof, because we have only 1 round. For each variable, such as L03, we have only two instances of it: $\mathrm{L03^i}$ is the variable on the input side, and $\mathrm{L03^o}$ will be the same variable on the output side. Again we will denote $\mathcal{P}^i = \mathcal{P}(\mathrm{Inputs})$ our polynomial instantiated with the input side variables. Similarly $\mathcal{P}^o = \mathcal{P}(Outputs)$ is the same polynomial written with the output side variables. We assume that for any key one round of DES is a bijection, and we will denote this bijection by ϕ, temporarily ignoring that in general ϕ depends on the secret key. Our reference set of variables will be the 64 input variables:

$$\mathcal{I} = (\mathrm{L01^i}, \ldots \mathrm{L32^i}; \mathrm{R01^i}, \ldots \mathrm{R32^i})$$

In order to prove that $\mathcal{P}^i = \mathcal{P}^o$ for any input, it is sufficient to prove that the values $\mathcal{P}^i = \mathcal{P}^o$ for any input 64 variables $\mathrm{L01^i} \ldots \mathrm{R32^i}$. We are going to express both values as a polynomial in the input 64 variables and show, that these two polynomials are formally equal. We will substitute 64 native variables at the

[8] Later we show that the same can also be done at degree 10, cf. Section 7.4.

[9] We assume that the secret key in DES as implemented inside these S-boxes transforming their input by a bitwise XOR.

output side $\mathcal{P}^o$ by their polynomial ANF expressions in the 64 input variables, which is due to the application of one round of encryption with ϕ, i.e.

$$\mathcal{P}^o = \mathcal{P}(\phi(\mathcal{I}))$$

where $\phi(\mathcal{I})$ denotes a sequence of 64 polynomial expressions of $(\mathrm{L01^o}, \ldots, \mathrm{R32^o})$ as a function of the 64 inputs-side variables in $\mathcal{I}$ which are the $\mathrm{L01^i}, \ldots, \mathrm{R32^i}$.

We observe that:

$$\mathcal{P}^o = \mathrm{L06^o} * \mathrm{R06^o} * \mathrm{L07^o} * \mathrm{R07^o} \ * \ \mathrm{L12^o} * \mathrm{R12^o} * \mathrm{L13^o} * \mathrm{R13^o}$$

$$* \ \mathrm{L24^o} * \mathrm{R24^o} * \mathrm{L28^o} * \mathrm{R28^o}$$

Again half of our transformations are trivial and for example the variable $\mathrm{L01^o}$ is always replaced by the polynomial $\mathrm{R01^i}$. We need to follow the connection of DES of Fig. 4 and we are going to write down which exact Boolean functions are added in our round. We obtain the following expression using input variables only and therefore we omit the notation ^{i} as no confusion is possible anymore:

$$\mathcal{P}^o = \mathrm{R06} * (\mathrm{L06} + \mathrm{Z3}) * \mathrm{R07} * (\mathrm{L07} + \mathrm{Z7}) * \mathrm{R12} * (\mathrm{L12} + \mathrm{X7}) \ *$$

$$\mathrm{R13} * (\mathrm{L13} + \mathrm{W2}) * \mathrm{R24} * (\mathrm{L24} + \mathrm{W3}) * \mathrm{R28} * (\mathrm{L28} + \mathrm{X2})$$

Now, just like in the previous proof, we are going to re-write our 6 assumptions:

$$\begin{cases} cd * \mathrm{W2} = 0 \text{ becomes } \mathrm{R06} * \mathrm{R07} * \mathrm{W2} = 0 \\ cd * \mathrm{X2} = 0 \text{ becomes } \ \mathrm{R06} * \mathrm{R07} * \mathrm{X2} = 0 \\ ef * \mathrm{W3} = 0 \text{ becomes } \mathrm{R12} * \mathrm{R13} * \mathrm{W3} = 0 \\ ef * \mathrm{Z3} = 0 \text{ becomes } \ \mathrm{R12} * \mathrm{R13} * \mathrm{Z3} = 0 \\ ae * \mathrm{X7} = 0 \text{ becomes } \ \mathrm{R24} * \mathrm{R28} * \mathrm{X7} = 0 \\ ae * \mathrm{Z7} = 0 \text{ becomes } \ \mathrm{R24} * \mathrm{R28} * \mathrm{Z7} = 0 \end{cases}$$

We have 6 consecutive double products and we observe and check that:

1. W2 can be erased from our 4th double product because $\mathrm{R06} * \mathrm{R07}$ is a factor in the whole product.
2. The same holds for X2 in our 6th double product.
3. W3 can be erased from our 5th double product because $\mathrm{R12} * \mathrm{R13}$ is a factor.
4. The same holds for Z3 in our 1st double product.
5. X7 can be erased from our 3rd double product because $\mathrm{R24} * \mathrm{R28}$ is a factor.
6. The same holds for Z7 in our 2nd double product.

After removing these terms we obtain the exact result we wanted to prove:

$$\mathcal{P}^o = \mathrm{R06} * \mathrm{L06} * \mathrm{R07} * \mathrm{L07} * \mathrm{R12} * \mathrm{L12} * \mathrm{R13} * \mathrm{L13} * \mathrm{R24} * \mathrm{L24} * \mathrm{R28} * \mathrm{L28} = \mathcal{P}^i$$

6.2 Can Our Attack Work with Balanced Boolean Functions?

It is easy to see that S-boxes can now be balanced, and in order to show this it is sufficient to exhibit one working example. This example was constructed using the fact, that our annihilation conditions only imply that our Boolean functions must be zero at some 1/4 of the input space. All the other 3/4 coefficients can be arbitrary and the overall number of 1s can be easily adjusted in order to obtain balanced Boolean functions.

```
S2:12,15,9,15,14,5,0,0,7,15,7,15,11,11,0,0,7,6,12,7,9,4,0,0,14,11,14,
15,15,13,0,0,10,2,10,13,15,9,0,0,15,2,5,2,13,2,0,0,11,5,8,14,13,5,0,0,
11,14,15,11,5,7,0,0
S3:12,15,11,15,15,5,7,15,7,12,11,11,7,7,14,15,8,0,5,0,15,0,11,0,12,0,
14,0,15,0,13,0,10,2,10,13,13,11,13,2,5,2,13,2,9,7,8,14,7,0,4,0,11,0,15,
0,15,0,11,0,5,0,6,0
S7:12,15,9,11,14,5,5,15,7,12,11,11,7,7,14,15,8,4,15,11,13,12,13,15,10,
2,10,15,15,9,13,3,11,0,2,0,7,0,2,0,11,0,7,0,13,0,14,0,5,0,4,0,15,0,14,0,
15,0,11,0,5,0,6,0
```

We also need to show that our Boolean functions are non-linear and display their Walsh spectrum. We limit our display to the 6 Boolean functions $W2, X2, W3, Z3, X7, Z7$ actually used in our attack. It is easy to see that the remaining 6 outputs are not used in our proof and therefore could be replaced by arbitrary (strong) Boolean functions. We can also see that all our 6 functions are of degree 5. The Walsh spectra are displayed in the standard format used by the popular SAGE software and need no further explanation.

```
S2: X2   {0: 13, 4: 29, 8: 16, 12: 1, 16: 3, 20: 1, 36: 1}
    W2   {0: 15, 4: 23, 8: 15, 12: 8, 16: 1, 24: 1, 28: 1}
S3: Z3   {0: 13, 4: 25, 8: 16, 12: 5, 16: 3, 20: 1, 28: 1}
    W3   {0: 14, 4: 24, 8: 15, 12: 6, 16: 2, 20: 2, 24: 1}
S7: Z7   {0: 13, 4: 26, 8: 16, 12: 5, 16: 2, 20: 1, 32: 1}
    X7   {0: 14, 4: 23, 8: 16, 12: 7, 16: 2, 20: 1, 28: 1}
```

6.3 Comparison to DES S-boxes

These spectra do not seem substantially different or of lower quality[10] than in DES itself, where the maximum values are also typically 24 or 28, and sometimes also as high as 36. In comparison here are the Walsh spectra of S-box S7 in DES:

```
S7: Z7   {0: 21, 4: 22, 8: 6, 12: 10, 16: 3, 24: 2}
    Y7   {0: 16, 4: 29, 8: 12, 12: 1, 16: 4, 20: 1, 36: 1}
    X7   {0: 18, 4: 24, 8: 8, 12: 6, 16: 6, 20: 2}
    W7   {0: 20, 4: 23, 8: 10, 12: 8, 24: 2, 28: 1}
```

[10] This is explained by the fact that in order to satisfy our annihilator conditions of type $Z * f$ where f is a product of 2 linear factors, we only need to impose that $Z = 0$ in 1/4 of the cases, all the other values can be arbitrary, so our Boolean functions are in some sense random for 3/4 of the inputs.

We have obtained an attack where the actual DES S-boxes do not seem any stronger than vulnerable S-boxes.

6.4 On Number of Zeros Inside Walsh Spectra

However it seems that there is a substantial difference if we look at the number of zeros in the Walsh spectra. DES S-boxes have larger numbers on average.

This is related to the widely studied question of correlation immunity[11] in block ciphers. However it is naive to believe that a Boolean function with high correlation immunity would always be also immune to annihilation by a product of 2 linear factors. In order to show this we have found a counter-example:

$$Z(a,b,c,d,e,f) = a(b+1) + (c+d+e+f)$$

is 3-resilient and it has two annihilators being a product of 2 linear factors:

$$Zb(c+d+e+f+1) = 0 \quad \text{and} \quad Z(a+1)(c+d+e+f+1) = 0.$$

Instead we need to look at this question from the point of view of probability.

7 DES S-boxes and Normality

In 1990s Dobbertin has proposed the notion of normality of Boolean functions. This notion was later extended to k-normality by Charpin in [12]. This is maybe not immediately obvious, but this notion is exactly mathematically equivalent to what we need to study here and we just need to reformulate it in terms of annihilators.

Let $\mathcal{B}_n$ be the ring of all Boolean functions in n variables and let $Z \in \mathcal{B}_n$. In order to study these notions it is useful to see that we have the following one-to-one correspondence. Any affine sub-space U of dimension k, which is sometimes called a flat [12], can be also seen as a set of points, where a certain product $\Pi = \prod_i L_i$ of k affine and linearly independent polynomials is non-zero and equal to 1. Then we observe that we have $Z \cdot \Pi = 0$ if and only if $Z = 0$ when restricted to our affine space U. Then we have $(Z+1) \cdot \Pi = 0$ if and only if $Z = 1$ when restricted to our affine space U. Therefore we have two equivalent ways to define our notion:

Definition 7.1 (k-normality). A Boolean function $E \in \mathcal{B}_n$ is said to be k-normal if either of the following equivalent conditions holds:
i) There exists a $k-$dimensional flat U where Z is constant.
ii) Either Z or $Z+1$ are annihilated by at least one product $\Pi = \prod_i L_i$ of $n-k$ linearly independent affine polynomials with either:

$$Z \prod_{i=1}^{k} L_i = 0 \qquad \text{or} \qquad (Z+1) \prod_{i=1}^{k} L_i = 0$$

[11] Correlation immune Boolean functions cf. [9] have numerous zeros in the Walsh spectrum.

7.2 How Many Boolean Functions are Vulnerable?

Theorem 7.3 (Frequency of Annihilation with Two Affine Terms). Given a Boolean function Z in 6 variables chosen uniformly at random and which is such, that it has at least 14 zeros in Walsh spectrum, the probability that it is 2-normal i.e. it has an annihilation of type

$$Z \cdot f \cdot g = 0 \text{ or } (Z+1) \cdot f \cdot g = 0$$

with two arbitrary affine factors f, g is equal to $2^{-3.42}$.

Proof. This result is obtained by checking all the 150357 classes of Boolean functions based on a database of Boolean functions published together with [8, 31]. Important relevant work on this topic is [28, 12].

Table 1. Classes of Boolean Functions with 6 Variables w.r.t. k-normality

total $\downarrow$ (any k)	k-normal Boolean functions				31079 with ≥ 14 Walsh at 0			
k value $\rightarrow$	6	≥ 5	≥ 4	≥ 3	6	≥ 5	≥ 4	≤ 3
150357	1	205	47446	150357	1	100	13969	31079
100%	$2^{-17.2}$	$2^{-9.52}$	$2^{-1.66}$	$2^{-0.0}$	$2^{-14.9}$	$2^{-10.55}$	$2^{-3.42}$	$2^{-0.0}$

This probability is not very small, however the attack in Thm. 6.1 requires as many as six such events for specific single variable linear factors. It appears that having numerous zeros in the Walsh spectrum in DES S-boxes is a plausible explanation why our attack with products of 2 linear factors such as Thm. 6.1 does not work with the original S-boxes. We need a better attack.

7.4 Further Attacks with Two Affine Factors

It is possible to see that this can be further improved at the cost of considering invariants of higher degree. In Table 2 below we explore a larger number of possible attacks. For example we found that a better result can be achieved with

$$\mathcal{P} = \text{L02} * \text{L05} * \text{L09} * \text{L28} * \text{L31} * \text{R02} * \text{R05} * \text{R09} * \text{R28} * \text{R31}$$

which is of degree 10. Here the active S-boxes are S1,S2,S8 with annihilators being respectively cf, bf, ad. All attacks we present here are based on the concept of closed loop invariants. More such attacks exist if we allow 4 active S-boxes.

Table 2. List of possible attacks with DES P-box and with 3 active S-boxes

```
L02*L03*L08*L09*L17*L18*R02*R03*R08*R09*R17*R18 1:cd 2:ef 5:bc
L02*L05*L09*L28*L31*R02*R05*R09*R28*R31 1:cf 2:bf 8:ad
L01*L03*L14*L17*L20*R01*R03*R14*R17*R20 1:bd 4:cf 5:be
L01*L05*L15*L17*L31*R01*R05*R15*R17*R31 1:bf 4:df 8:df
L03*L04*L17*L19*L23*L25*R03*R04*R17*R19*R23*R25 1:de 5:bd 6:df
L04*L05*L21*L23*L29*L31*R04*R05*R21*R23*R29*R31 1:ef 6:bd 8:bd
L06*L08*L13*L16*L18*R06*R08*R13*R16*R18 2:ce 3:af 5:ac
L06*L07*L12*L13*L24*L28*R06*R07*R12*R13*R24*R28 2:cd 3:ef 7:ae
L05*L07*L27*L28*L32*R05*R07*R27*R28*R32 2:bd 7:de 8:ae
L08*L10*L14*L16*L20*R08*R10*R14*R16*R20 3:ac 4:ce 5:ae
L10*L12*L16*L24*L26*R10*R12*R16*R24*R26 3:ce 4:ae 7:ac
L08*L11*L16*L19*L24*L25*R08*R11*R16*R19*R24*R25 3:ad 5:ad 6:ef
L11*L12*L22*L24*L29*R11*R12*R22*R24*R29 3:de 6:ce 7:af
L01*L12*L15*L26*L27*L32*R01*R12*R15*R26*R27*R32 4:ad 7:cd 8:ef
L21*L22*L27*L29*L32*R21*R22*R27*R29*R32 6:bc 7:df 8:be
```

8 Attacks with Three Affine Factors

Furthermore it is possible to construct an attack where every single annihilation polynomial is of degree 3, so that only 1/8th of the entries in the truth table for certain Boolean functions inside the whole cipher need to be as required. This comes at the price of increasing the degree of $\mathcal{P}$ from 10 to 20. One example of such polynomial we found which we expect to be the best possible is $\mathrm{R01} * \mathrm{R02} * \mathrm{R03} * \mathrm{R06} * \mathrm{R08} * \mathrm{R09} * \mathrm{R13} * \mathrm{R16} * \mathrm{R17} * \mathrm{R20}$ multiplied by the same polynomial for the left side variables. This attack works with five S-boxes $1, 2, 3, 4, 5$ with annihilations respectively by bcd, cef, abf, bef and abe. A longer list of attacks with products of three or more factors is given in Appendix A.

8.1 On Boolean Function Vulnerability with 3 Factors

Here the situation improves dramatically w.r.t. Boolean functions: no Boolean function is such that annihilations with 3 linear factors are impossible. We have:

Theorem 8.2 (All Boolean functions in 6 variables are 3-normal). Given a Boolean function Z in 6 variables chosen uniformly at random the probability that it is 3-normal i.e. it has an annihilation of type

$$Z \cdot f \cdot g \cdot h = 0$$

with three arbitrary affine factors f, g, h is equal to exactly 100%.

Proof: This is a known result. It can be easily shown by using the complete classification of Boolean functions in 150,357 classes, see [31, 8]. It is also a special case of a theorem proven by Dubuc [28] and then studied by Charpin in [12].

8.3 Attacks with 4 Factors

Then if further increase the degree of $\mathcal{P}$ we have constructed invariant attacks, such that all annihilators are products of ≤ 4 terms and only 1/16 out of 64 entries in the truth table need to be modified. Currently known attacks of this type are extremely poor: they involve a very large number of bits and $\mathcal{P}$ is of very high degrees such as 40 and are very weak in terms of potential bias which such invariants for the cipher state. We need to consider another type of attack.

9 More General Attacks Based on Imperfect Cycles

In fact we have not studied the most general attack yet. There exists a more general attack framework, where Boolean functions can be annihilated in a wider variety of ways, and where annihilators are not single variables, but arbitrary products of polynomials (which do not actually have to be linear or affine). One basic idea is that instead of considering sets of bits as in Section 3.4, we consider some affine polynomials $\mathcal{Q}_i$ and see if they could depend in some sense "essentially" on themselves. Then we need to say what exactly we mean by "essentially". This generalized attack is outlined in Section 12 of [20] and described in full in Section 4 and Section 7 of [26]. It is based on multiplying the polynomials over several cycles which correspond to conditional transitions between polynomials denoted by $\mathcal{Q}_i$. These cycles are imperfect in the sense that some additional polynomials $\mathcal{Z}_i$ may be added on the way. These polynomials $\mathcal{Z}_i$ are actually an extension of the concept of the Fundamental Equation (FE) in [19] to the case of transitions of type $\mathcal{Q}_i \rightarrow \mathcal{Q}_{i+1}$, i.e. which are no longer invariants. The polynomials $\mathcal{Z}_i$ are then present each time the transitions are not actually true, and they correspond to the notion of *Transition Equation* or (TE) in[12] Section 5 of [19].

The main interest with this more recent approach for constructing invariant attacks, is that it allows one to rediscover in an ordered and structured way many previous attacks, and understand better why such attacks can be made to work, cf. for example Fig. 4 in Section 5.6. in [20]. In this type of attack we also allow arbitrary additions of original Boolean functions with affine polynomials with the same set of inputs of variables, such as for example $\mathrm{W2} + \mathrm{R06} + \mathrm{R07}$. This corresponds to another well-known notion in the theory of Boolean functions, one of k-weakly-normal functions also studied in [12].

9.1 Attacks with Sums of Outputs

The next stage is to search for attacks, which is also permitted by our general attack framework, of Section 12 and Appendix B in [20] and [26]. In such more general attacks sums of outputs of Boolean functions such as $(X8 + W8)$, or even better combinations, such as say $(X8 + W8 + R25 + R28)$, are permitted

[12] One interpretation of $\mathcal{Z}_i$ is that it is as a polynomial in the input variables only, such that the transition $\mathcal{Q}_{i+1} \rightarrow \mathcal{Q}_{i+1}$ would be true for one round ϕ for inputs such that our polynomial $\mathcal{Z}_i$ is equal to zero.

and can be annihilated. Here the degree of annihilators which actually exist fort the real-life DES S-boxes is eventually lower and **better**. For example in DES S-box S5 we have:

$$R17 * (R16 + R20) * (W5 + X5 + Y5 + Z5) = 0$$

which is a direct consequence[13] of strong biases inside presented by Shamir as early as 1985, cf. [33]. Similar properties exist for other DES S-boxes, for example for S4 and S1 and will be studied inside another article. It is too early to say if these properties can lead to better[14] non-linear invariant attacks on DES than those currently known.

10 Attacks Exploiting the DES Key Schedule

In this section we finally look at questions which is a realistic with respect to the full 16-round DES as it is standardized and used by billions of users. The main idea is that DES key schedule is weak, however only a tiny fraction of properties we study are compatible with DES key schedule. The weakness of DES key schedule is widely known. An essential reference on key scheduling in DES Brown and Seberry 1990 paper [6], and many observations were already published in an earlier paper [5] from 1988. In Section 3.4.2. in page 6 of [5] we read that

1. "There is no interaction across the two halves" in DES key scheduling
2. we have "two distinct 28 to 24-bit selections".

An obvious consequence of these observations is that half of the key on 28 bits is used at inputs of S-boxes S1-S4, and the other half is used at inputs of S-boxes S5-S8. In our research we have examined 177 different closed sets such as in Table 2 and found only three one-side configurations which use only S-boxes from either S1-S4 or S5-S8 range, but do not mix them. We call these configurations a14, a58 and b58.

Table 3. List of simple product attacks compatible with DES key schedule

```
a14: L01*L02*L06*L09*L10*L13*L16*L17*R01*R02*R06*R09*R10*R13*R16*R17 1:bc 2:cf 3:bcf 4:bef
a58: L21*L22*L27*L29*L32*R21*R22*R27*R29*R32 6:bc 7:df 8:be
b58: L19*L21*L22*L25*L27*L29*L32*R19*R21*R22*R25*R27*R29*R32 5:df 6:bcf 7:bdf 8:be
```

The first property uses 8+8 bits, the two other properties uses 5+5 and 7+7 bits. These are the properties which are most likely, among those studied in this

[13] It is easy to see that the more a Boolean function is biased the more annihilators or absorbers it will have, cf. Thm. C.2. in Appendix C of [20].

[14] The contrary could also be imagined: that these particular annihilations will force all "interesting" attacks to use all the 4 outputs for several S-boxes, which could overall increase the number of S-boxes which must be active inside the attack.

article, to be useful in an attack on DES which takes into account the DES key schedule. We see that there are extremely few such configurations (3 out of 177).

Below we show a complete list of all 177 closed-loop sets we have studied:

0x19323,0x8a011943,0x391a6,0x88801962,0x5155001c,0x68001132,0xc0152a,0x10a00c00,0x38901438,0xb3187,0x49110b,0x48015113,0x78d9d73b,0x147018e,0x70d00518,
0x890301c6,0x48130196,0x18ca680,0x9860014a,0xd4700018,0x5850011a,0xcc000152,0x9a385,0x2800d231,0x2808a00,0xe4800910,0xc98709,0x3098c601,0x82818b01,
0xb0a38c0,0x6001c311,0x1c5858c,0x81818984,0xcf1b79d7,0x60118194,0x92690809,0x889628,0x90e00d08,0xc6014811,0x14d200d,0xae80da71,0x83092805,0xab2a0,
0x40196015,0x50594019,0xa680ca01,0x19a69ce8,0x8c000050,0x9165000c,0xc5110014,0xa809a60,0xb8abae0,0x31948480,0x18694a8,0xfef9df7b,0x281290b0,0x98298e0,
0x8a819b63,0x18a01c68,0x2018e281,0x6019e395,0xac801870,0x10e3088,0x18185019,0x86004801,0x81a7091,0x48000112,0x1a281848,0x79d795be,0xb4b00c00,0x8e005851,
0x192600c8,0x19160098,0x79dff7bf,0x8d1200d0,0x9c300058,0x388aa80,0x12a88e00,0x11a48c80,0xa5908880,0xe5918994,0x132c2800,0x111c6001,0x87186801,0x8801860,
0x96384801,0x95340000,0xbfbefef9,0x9befbfef,0xef9bfbf7,0xfdf79dfe,0x91e58d8c,0xdf7f79df,0x6801d333,0xf7fdef9d,0x1cfb7af,0x8b8bbbe7,0x681bf3b7,0x9ae99f6b,
0xee81db73,0x99e79dee,0xed9399f6,0xfcf01d7a,0x9b6f39cf,0x595f719f,0x681391b6,0xde79595b,0xdd7701de,0x93edaf8d,0x71dde79d,0xa798ea81,0xe799eb95,0xf6f9cf19,
0xf5f58d9c,0xd77d681d,0x1baebee8,0x399ef6b9,0xaf9afaf1,0xbeb8de79,0xbdb69cf8,0x9f3e78d9,0xb7bcee81,0xbb3a7,0xc9972b,0x1c795ae,0x898399e6,0x98e01d6a,
0x78d0153a,0xec801972,0x14f318f,0x8b0b39c7,0x481b7197,0x9a69194b,0x5859511b,0xce015953,0x996701ce,0x5957019e,0x936d280d,0xcd1301d6,0xdc70015a,0x1cda78d,
0x8389ab85,0x92e98f09,0x70d9c719,0xe681cb11,0x71d5859c,0xf4f00d18,0x515d601d,0xc7196815,0xd6794819,0xd575001c,0x18eb6a8,0x281af2b1,0x8a280,0x1aa89e68,
0x3898d639,0x399694b8,0xad9298f0,0xbcb01c78,0x1b2e38c8,0x191e7099,0x8f1a78d1,0x9e385859,0x9d3600d8,0x13acae80,0x319ce681,0xb6b8ce01,0xb5b48c80,0x973c6801,
0x30186,0x92005,0x40014011,0x145000c,0x50500018,0x290a0,0x1848480,0x94300000

11 Affine to Affine Mappings

A central question in cryptanalysis of block ciphers is the question of affine to affine mappings. By definition affine spaces are cosets of linear spaces (which some authors also call flats). Sometimes, a non-linear mapping maps an affine space to an affine space. There are zillions of special cases like this for every cipher ever made. We can first study this question at the level of individual S-boxes with 6 inputs and 4 outputs and only for spaces of dimension 2.

Table 4. Number of affine spaces U of dimension 2 which can be mapped to another affine space W of dimension 2 also, for all 4-bit permutations defined by DES S-boxes with 2 outer bits a, f fixed

		DES S-box								s^5DES S-box								S*DES S-box								DESL
a	f	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	any
0	0	14	10	12	13	12	8	14	10	36	28	36	28	24	28	24	24	12	13	13	12	12	12	10	10	12
0	1	14	12	14	13	12	11	12	16	16	24	24	14	14	24	24	24	10	11	13	12	12	13	8	13	12
1	0	14	12	12	13	9	13	13	12	26	24	32	20	48	36	24	24	12	12	13	12	10	13	8	8	24
1	1	12	13	16	13	12	10	16	14	16	20	20	16	18	20	20	24	8	13	12	12	9	13	8	13	24

It seems that the Korean version of DES, known as s^5DES, is substantially weaker than other versions of DES. Below we show a more detailed picture of essentially the same events where we classify them by the input difference vectors $U1$ and $U2$ actually used. Again it seems that s^5DES is a particularly weak version of DES.

Table 5. Counting mappings of affine spaces U of dimension 2 which can be mapped to an affine space W of dimension 2, classified by **input** linear spaces ignoring the offset and when a and f are fixed.

		DES S-box								s^5DES S-box								S*DES S-box								DESL
$U1$	$U2$	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	any
1	2	1	–	2	–	4	1	–	–	2	4	**8**	**6**	2	4	2	2	–	–	–	–	3	2	1	2	–
1	4	1	3	–	–	1	–	**7**	–	4	2	4	4	**6**	**6**	2	4	1	3	–	2	–	–	1	–	–
1	6	–	2	1	–	–	3	2	2	2	2	**6**	–	2	**6**	–	–	1	1	2	3	2	1	–	2	4
1	8	1	2	5	4	3	5	5	2	5	4	5	3	4	3	**6**	2	3	3	3	3	4	4	–	3	–
1	*A*	3	–	–	–	2	1	–	1	2	–	5	2	1	2	3	2	1	–	1	2	2	2	–	–	–
1	*C*	1	1	–	–	2	–	5	3	2	4	1	2	3	2	2	1	1	2	3	–	–	–	1	–	1
1	*E*	5	1	1	4	1	1	–	–	1	2	1	1	2	1	1	1	1	2	1	2	1	–	1	–	–
2	4	4	**7**	3	4	–	4	4	**6**	**6**	4	**6**	**6**	**8**	4	4	4	**6**	5	5	3	3	4	**8**	**6**	**8**
2	5	–	1	–	–	1	1	–	–	4	**8**	**8**	**10**	2	**10**	4	**8**	–	1	1	–	1	2	1	3	–
2	8	–	2	3	4	–	1	–	–	5	2	2	1	5	3	2	3	–	–	2	2	–	–	1	1	**6**
2	9	1	–	2	–	1	–	–	3	2	–	2	1	3	1	2	2	–	–	1	–	–	1	–	–	–
2	*C*	–	–	–	4	–	1	–	–	3	–	–	–	4	–	1	4	2	–	2	3	1	2	1	2	4
2	*D*	3	–	–	–	2	–	1	3	2	**6**	–	2	2	2	1	1	–	2	1	–	1	1	–	–	–
3	4	1	3	2	4	1	1	2	–	4	4	**6**	2	4	4	**6**	2	1	2	–	–	1	2	1	–	–
3	5	–	–	4	–	1	2	2	1	2	2	**6**	2	**6**	**8**	2	**6**	1	2	2	1	2	1	–	3	4
3	8	1	1	1	4	1	1	1	1	4	3	**6**	3	3	1	4	4	1	3	3	1	–	–	2	–	–
3	9	–	–	2	–	–	–	1	2	2	1	2	–	2	–	1	1	–	–	–	–	–	1	–	–	2
3	*C*	3	–	2	4	1	2	2	–	2	1	–	1	1	2	3	–	2	3	3	3	2	–	1	1	3
3	*D*	–	2	5	4	1	3	1	3	2	1	–	–	4	3	2	3	1	2	3	4	2	2	3	1	–
4	8	2	–	2	–	2	1	1	–	4	4	1	3	5	3	2	3	3	1	–	–	3	1	–	2	2
4	9	2	–	–	–	1	1	3	2	1	2	2	2	4	2	2	2	1	–	1	–	–	–	–	2	–
4	*A*	4	3	1	–	2	1	1	1	–	1	3	1	4	1	–	3	3	2	–	1	1	3	–	–	**6**
4	*B*	1	–	–	–	1	1	1	1	1	1	2	2	3	2	–	–	1	–	3	4	2	–	–	2	–
5	8	–	–	3	–	1	–	–	2	5	3	2	4	3	4	4	**6**	1	–	–	–	1	1	1	2	–
5	9	–	–	–	4	3	1	–	–	3	1	3	2	1	2	–	4	1	1	1	–	–	1	–	2	5
5	*A*	3	2	–	–	2	–	2	1	2	3	4	5	1	2	3	**6**	1	1	1	3	4	3	2	3	2
5	*B*	2	4	3	4	2	4	2	4	2	1	3	3	3	4	3	4	1	2	4	4	4	2	3	2	–
6	8	1	2	3	–	1	–	–	2	5	5	–	3	–	3	2	4	2	1	–	1	–	2	1	1	4
6	9	1	1	2	–	1	2	1	1	2	–	1	–	3	–	4	–	–	1	–	1	1	1	1	–	4
6	*A*	2	2	2	–	–	1	2	3	3	2	2	–	–	–	1	1	2	2	1	2	1	2	1	2	2
6	*B*	3	1	1	4	3	–	1	2	2	1	1	1	3	1	5	3	–	1	–	1	–	2	1	–	4
7	8	3	1	1	–	1	–	–	1	4	**7**	4	3	4	**7**	**8**	2	1	1	–	2	1	2	2	1	–
7	9	1	2	3	–	–	–	2	–	1	4	5	–	3	4	3	3	–	1	2	–	–	1	–	–	–
7	*A*	1	2	–	–	1	1	3	3	–	**7**	**6**	3	1	**6**	2	4	–	2	1	–	–	3	–	–	4
7	*B*	3	2	–	4	2	2	3	2	3	4	5	–	2	5	5	1	3	2	4	–	–	2	–	1	**7**
total		54	44	54	33	45	41	54	50	75	77	88	54	76	84	78	76	40	47	46	36	41	48	33	37	58
all		401								**780**								362								72

12 Conclusion

Nonlinear polynomial invariant attacks are very popular in the recent years cf. for example [34, 19]. For DES they typically work only for a fraction of the key space, which was already the case in early invariant attacks on DES at degree 2 cf. [21]. In this article we show that various non-linear invariant attack with degrees ranging between 5 and 20 can be constructed for DES. Our attacks are constructed from "closed loop" configurations [35], cf. for example Fig. 3 page 9. Our methodology is to study the question of the existence of the attacks independently of any considerations which would involve any secret key bits, similarly to the suggestion in Sec 4.2. in [4]. We have a pure question of existence of polynomial invariants $\mathcal{P}$ for any specific P-box and arbitrary S-boxes with the key bits. Then we have two separate questions of how many S-boxes are vulnerable and if these properties are preserved or not when we translate the S-boxes by a key on the input side. Our first impossibility result is Thm. 5.4. We show that balanced non-linear Boolean functions cannot work with our specific attack of degree 5. The impossibility holds for any key (worst case). Possibly there is little hope to break DES with invariants of degree 5.

Then we show that substantially more powerful attacks will be obtained when the degree of the polynomial invariant increases. In Thm. 6.1 we show that with invariants of degree 12, Boolean functions can be balanced and highly non-linear. We obtained a first proof of concept of how to backdoor DES by modifying a small number of entries inside certain S-boxes and nothing else – everything else is like in the original FIPS Data Encryption Standard.

An important question how to avoid this type of attacks which is closely related to the question of k-normality for Boolean functions, cf. Section 7.2. Furthermore in Section 8 we discover that with cubic annihilators the attack becomes very hard to avoid, cf. Thm. 8.2. In Section 9 we outline a larger family of attacks based on cycles involving no longer just bits but complex polynomials. In Section 10 we explain that some of these are highly compatible with the DES key scheduling, while most are not. Finally in Section 11 we provide a glimpse of how a larger picture looks like, when we consider full S-boxes instead of individual Boolean functions.

References

1. Arnaud Bannier, Nicolas Bodin, and Eric Filiol: *Partition-Based Trapdoor Ciphers,* `https://ia.cr/2016/493`.
2. Joan Boyar, Magnus Find, René Peralta: *Four Measures of Nonlinearity,* In Algorithms and Complexity, CIAC 2013, LNCS 7878, pp. 61-72, Springer.
3. C. Beierle, A. Canteaut, G. Leander, Y. Rotella: *Proving resistance against invariant attacks: how to choose the round constants,* in Crypto 2017, Part II. LNCS, 10402, pp. 647–678, Springer 2017.
4. Tim Beyne: *Block Cipher Invariants as Eigenvectors of Correlation Matrices,* in Asiacrypt 2018, pp. 3-31. One version is also avail. at `https://eprint.iacr.org/2018/763.pdf`

5. Lawrence Brown: *A Proposed Design for an Extended DES,* In IFIP/Sec'88, May 1988.
6. Lawrence Brown, Jennifer Seberry, *Key scheduling in DES type cryptosystems* In AUSCRYPT '90, LNCS 453, pp. 221-228.
7. Marco Calderini: *A note on some algebraic trapdoors for block ciphers,* last revised 17 May 2018, `https://arxiv.org/abs/1705.08151`
8. Cagdas Calik and Meltem Sonmez Turan and Rene Peralta: *The Multiplicative Complexity of 6-variable Boolean Functions,* `https://ia.cr/2018/002.pdf`
9. P. Camion, C. Carlet, P. Charpin, and N. Sendrier, *On correlation immune functions,* In Crypto'91, LNCS 576, pp 86-100.
10. Claude Carlet: *On the degree, nonlinearity, algebraic thickness and non-normality of Boolean functions, with developments on symmetric functions. IEEE Trans. Inf. Theory 50, 2178–2185 (2004).*
11. *Claude Carlet, Sihem Mesnager; Four decades of research on bent functions,* In Designs Codes and Cryptography vol. 78, pp: 5–50, 2006.
12. Pascale Charpin: *Normal Boolean functions,* Journal of Complexity, vol. 20, Issues 2–3, pp 245–265, 2004.
13. Nicolas Courtois: *Two Philosophies For Solving Non-Linear Equations in Algebraic Cryptanalysis,* avail. at `http://www.nicolascourtois.com/papers/Igamma-Mycrypt2016.pdf`, in Paradigms in Cryptology, Mycrypt 2016. Malicious and Exploratory Cryptology, pp. 506-520, LNCS 10311, Springer 2017.
14. Nicolas T. Courtois, Aidan Patrick: *Lack of Unique Factorization as a Tool in Block Cipher Cryptanalysis,* Preprint, `https://arxiv.org/abs/1905.04684` 12 May 2019.
15. Nicolas T. Courtois: *Invariant Hopping Attacks on Block Ciphers,* presented at WCC'2019, Abbaye de Saint-Jacut de la Mer, France, 31 March - 5 April 2019. Extended version available at `https://arxiv.org/pdf/2002.03212.pdf`, 8 February 2020.
16. Nicolas T. Courtois, Marios Georgiou: *Variable elimination strategies and construction of nonlinear polynomial invariant attacks on T-310,* In Cryptologia, vol. 44, Iss. 1, pp. 20-38. At `https://doi.org/10.1080/01611194.2019.1650845`
17. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback,* Eurocrypt 2003, LNCS 2656, pp. 345–359, Springer. Extended version: `www.nicolascourtois.com/toyolili.pdf`.
18. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs,* ICISC 2004, LNCS 3506, pp. 3–20, Springer 2005. Extended version available on `https://ia.cr/2003/125/`.
19. Nicolas T. Courtois: *On the Existence of Non-Linear Invariants and Algebraic Polynomial Constructive Approach to Backdoors in Block Ciphers,* `https://ia.cr/2018/807`, revised 3 Dec 2018.
20. Nicolas T. Courtois: *Structural Nonlinear Invariant Attacks on T-310: Attacking Arbitrary Boolean Functions,* `https://ia.cr/2018/1242`, revised 12 Sep 2019.
21. Nicolas Courtois: *Feistel Schemes and Bi-Linear Cryptanalysis,* in Crypto 2004, LNCS 3152, pp. 23–40, Springer, 2004.
22. Nicolas Courtois: *An Improved Differential Attack on Full GOST,* In Cryptology ePrint Archive, Report 2012/138. 15 March 2012, updated December 2015, `https://ia.cr/2012/138`.
23. Nicolas Courtois, Jörg Drobick and Klaus Schmeh: *Feistel ciphers in East Germany in the communist era,* In Cryptologia, vol. 42, Iss. 6, 2018, pp. 427-444.

24. Nicolas Courtois, Michał Misztal: *Aggregated Differentials and Cryptanalysis of PP-1 and GOST,* In CECC 2011, 11th Central European Conference on Cryptology. In Periodica Mathematica Hungarica Vol. 65 (2), 2012, pp. 11-26, Springer.
25. Nicolas Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST,* Monograph study on GOST cipher, 2010-2014, 224 pages, available at `https://ia.cr/2011/626`.
26. Nicolas T. Courtois, Matteo Abbondati, Hamy Ratoanina, and Marek Grajek: *Systematic Construction of Nonlinear Product Attacks on Block Ciphers,* In ICISC 2019, LNCS 11975, pp 20-51, Springer, 2020.
27. Hans Dobbertin: *Construction of bent functions and balanced Boolean functions with high nonlinearity*, in: FSE'94, LNCS 1008, Springer, Berlin, pp. 61–74, 1994.
28. S. Dubuc: *Etude des propriétés de dégénérescence et de normalité des fonctions booléennes et construction de fonctions q-aires parfaitement non-linéaires,* Ph.D. Thesis, Université de Caen, 2001.
29. C. Harpes, G. Kramer, and J. Massey: *A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma,* Eurocrypt'95, LNCS 921, Springer, pp. 24–38.
30. G. Leander, M.A. Abdelraheem, H. AlKhzaimi, E. Zenner: *A cryptanalysis of PRINTcipher: The invariant subspace attack,* In Crypto 2011, LNCS 6841, pp. 206–221, 2011.
31. James A. Maiorana: *A classification of the cosets of the Reed-Muller code R(1,6),* In Mathematics of Computation, 57(195):403-414, 1991.
32. Klaus Schmeh: *The East German Encryption Machine T-310 and the Algorithm It Used,* In Cryptologia, vol. 30, iss. 3, pp. 251–257, 2006.
33. Adi Shamir: *On the security of DES,* Crypto'85, LNCS 218, Springer, pages 280-281.
34. Yosuke Todo, Gregor Leander, and Yu Sasaki: *Nonlinear invariant attack: Practical attack on full SCREAM, iSCREAM and Midori64,* In Journal of Cryptology, pp. 1–40, April 2018.
35. Yongzhuang Wei, Tao Ye, Wenling Wu, Enes Pasalic: *Generalized Nonlinear Invariant Attack and a New Design Criterion for Round Constants,* In IACR Tr. on Symm. Crypt. Vol. 2018, No. 4, pp. 62-79.

A Enumeration of Product Attacks with Cubic Annihilators and Single Variables

We have computed a list of potential non-linear invariant attacks on DES with modified S-boxes. These configurations were constructed based on random subsets of 32 bits considering that annihilators can be any sets of bits which are connected primarily to themselves cf. Section 4. Annihilations by a product of two variables or less are avoided, as too few S-boxes have such properties, cf. Table 1. This is not the most general polynomial invariant attack on DES, cf. Section 9 and Section 12 in [20]. However these attack are quite interesting because many inputs of DES round functions are duplicated, and many terms with single variables are such that they can used TWICE (annihilating outputs for two different S-boxes). We obtain a small finite set of attacks where all annihilations are of degree 3 or higher and where the degree of $\mathcal{P}$ is ≥ 20. Our list of attacks is shown in Table 6 below, each polynomial is a product of two identical polynomials with R variables and L variables. For the sake of compactness we omit the part with L variables. In addition, due to the lack of space we present here only results the "stronger" half of such examples we have generated, where three S-boxes are annihilated by a products of degree 4, which condition is easier to satisfy. Moreover we do not specify which outputs need to be annihilated, which will be simply all those actually used inside our set, cf. Fig. 7.

Enumerating all possible attacks with these characteristics is our view quite useful because we have a sufficient variety of attacks in order to work on questions such as what is the probability that one attack out of many works with specific S-boxes and how to optimize it. Or is DES P-box adequate and would a random P-box lead to a larger set of attacks? Finally when we instantiate the attack with concrete S-boxes, we obtain a ring of invariants which sometimes contains elements of substantially lower degree than initially planned[15]. Therefore we can use these polynomials as a starting point to discover a substantially larger set of invariant attacks on DES with the original P-box. We leave these questions for future research.

[15] For example the exact attack of degree 12 in Section 6.1 hides the existence of another attack of degree 5 in Section 5 operating on exactly the same set of 6+6 bits.

s **Table 6.** List of attacks with 5 active boxes and two cubic annihilators each.

```
R01*R02*R03*R06*R08*R09*R10*R13*R14*R16*R17*R18*R20 1:bcd 2:cef 3:abcf
4:bcef 5:abce
R02*R03*R04*R06*R08*R09*R11*R13*R16*R17*R18*R19*R23*R24*R25 1:cde 2:acef
3:abdf 5:abcd 6:def
R01*R02*R03*R05*R08*R09*R13*R14*R15*R17*R18*R20*R21*R28*R31 1:bcdf 2:bef
4:bcdf 5:bcef 8:adf
R01*R02*R05*R07*R09*R12*R13*R15*R17*R26*R27*R28*R31*R32 1:abcf 2:bdf
4:abdf 7:cde 8:adef
R02*R03*R04*R05*R08*R09*R17*R18*R19*R21*R23*R25*R28*R29*R31 1:cdef 2:abef
5:bcdf 6:bdf 8:abd
R02*R04*R05*R07*R09*R21*R22*R23*R27*R28*R29*R31*R32 1:acef 2:abdf 6:bcd
7:def 8:abde
R01*R03*R04*R08*R09*R10*R11*R14*R16*R17*R19*R20*R23*R24*R25 1:bde 3:abcd
4:cef 5:abde 6:adef
R01*R04*R05*R09*R10*R11*R15*R16*R17*R20*R21*R23*R24*R29*R30*R31 1:bef
3:bcd 4:def 6:abde 8:bcdf
R01*R03*R04*R05*R14*R15*R17*R19*R20*R21*R23*R25*R29*R31 1:bdef 4:cdf
5:bdef 6:abdf 8:bdf
R01*R03*R05*R12*R14*R15*R17*R20*R21*R25*R26*R27*R31*R32 1:abdf 4:acdf
5:bef 7:bcd 8:def
R01*R04*R05*R12*R15*R17*R20*R21*R22*R23*R26*R27*R29*R31*R32 1:abef 4:adf
6:abcd 7:cdf 8:bdef
R06*R07*R08*R10*R12*R13*R14*R16*R18*R20*R24*R25*R26*R28 2:cde 3:acef
4:abce 5:ace 7:abce
R04*R05*R06*R07*R11*R12*R13*R21*R22*R24*R27*R28*R29*R30*R32 2:abcd 3:def
6:bce 7:adef 8:abce
R08*R10*R11*R12*R14*R16*R19*R20*R22*R24*R25*R26*R29 3:acde 4:ace 5:ade
6:acef 7:abcf
R01*R10*R11*R12*R15*R16*R20*R21*R22*R24*R26*R27*R29*R30*R32 3:cde 4:ade
6:abce 7:acdf 8:bcef
```