Structural Nonlinear Invariant Attacks on T-310: Attacking Arbitrary Boolean Functions

Nicolas T. Courtois

University College London, Gower Street, London, UK

Abstract. Recent papers [36, 12] show how to construct polynomial invariant attacks for block ciphers, however almost all such results are somewhat weak: invariants are simple and low degree and the Boolean functions tend by very simple if not degenerate. Is there a better more realistic attack, with invariants of higher degree and which is likely to work with stronger Boolean functions? In this paper we show that such attacks exist and can be constructed explicitly through on the one side, the study of Fundamental Equation of [12], and on the other side, a study of the space of Annihilators of any given Boolean function. Our approach is suitable for backdooring a block cipher in presence of an arbitrarily strong Boolean function not chosen by the attacker. The attack is constructed using excessively simple paper and pencil maths.

Key Words: block ciphers, Boolean functions, non-linearity, ANF, Feistel ciphers, weak keys, backdoors, history of cryptography, T-310, Generalized Linear Cryptanalysis, polynomial invariants, multivariate polynomials, annihilator space, algebraic cryptanalysis.

1 Introduction

In this paper we introduce a new **type** of attack on block ciphers. Not a new attack - as such attacks with non-linear[1] invariants were known for decades cf. [29, 30, 13]. The novelty is rather in the **features** of the attack and in the general philosophy of how the attack works and how it is used. The attack we present has unique interesting properties which have we never encountered before. We attempt to construct a higher degree polynomial invariant attack and we discover that such properties may work well beyond just one cipher setup we would like to break, but in a very large number of other cases. Imagine that we could break AES in a way such that the exact specification of the S-box does not really matter. Impossible? Furthermore, imagine that the S-box could be maliciously crafted by the choice of the affine transformation inside the S-box which makes the cipher breakable no matter what, and it is potentially hard to defend against this attack, because even if the choice of a suitable linear transformations is reduced, just a handful of good sparse and efficiently computable linear transformations exist, the power of the attack is such that it just works with high probability, for any choice of the internal non-linear component. Imagine also that our attack belongs to a larger space with the number of possible

---

[1] We consider that "nonlinear" and "non-linear" are synonyms the latter version being less formal and more suitable for a larger Computer Science audience.

attacks growing as 2^{2^n}. So that even if one attack accidentally does not work for one S-box, another similar invariant property may work out of the box for this exact cipher setup. Now of course we do not work on AES, which would be very ambitious. For the sake of scientific research we need to test our ideas on a simpler cipher. Numerous papers study attacks on toy ciphers [2, 27] however it is always better to work on a real-life cipher. We work on an older block cipher T-310 used in the 1980s to encrypt government communications.

1.1 Structural Non-Linear Attacks on Block Ciphers

Non-linear cryptanalysis was known for at least 20 years. What is new here is:

1. That such attacks may actually be found or constructed, with polynomials of higher degree than before, e.g. (rather than quadratic or cubic cf. [36, 12, 13]) and that such attacks may actually exist at all, for any block cipher.
2. It is an attack which works for an arbitrarily large number of rounds (which is rare but of course seen before cf. [28, 7, 4, 23]).
3. Finally it does something which is even more unique. Namely it is able to somewhat ignore or circumvent the non-linear component(s) of the cipher in a novel and powerful way: so that we obtain a purely **"structural"** attack on a block cipher which makes our block cipher insecure, more or less ignoring the non-linear components. An attack which is very likely to work also when they are chosen to be extremely strong and also when they are secret, unknown or key dependent [31]. This was seen before in stream cipher cryptanalysis [10, 11] but it is new and unprecedented for block ciphers.

In this paper we assume that attacker cannot choose the non-linear component (S-box or a Boolean function) but can make some "cipher wiring" choices of the type "long-term key" and his goal is to make the cipher extremely weak on purpose cf. [12, 1, 33, 3]. In other terms this paper is about whether block ciphers can be **backdoored** even though many components and a number of rules have already been imposed[2] by the designers and are assumed to be as strong as only possible. This is a realistic setting similar to what happens in real-life situations: for example the S-boxes in Serpent have been built in order to avoid any suspicion of backdooring. For the same reason some ciphers will have no S-boxes whatsoever but may use some natural transformations such as rotations, additions etc. (ARX philosophy). In AES potentially the affine transformation inside the S-box could be manipulated, and in LFSR-based ciphers the LFSR taps could be manipulated, yet the number of "good choices" is extremely small. Then the question is can we make the cipher weak nevertheless, and with a sufficiently large probability. As a proof of concept we will construct a round invariant property which with a number of appropriate connections inside the cipher works with probability up to 25 % over the choice of the non-linear component inside the cipher.

---

[2] The intention of the designers are on the contrary that there would be no way whatsoever to make the cipher weak, once some strong components are mandated like the S-boxes in DES which were designed using some classified knowledge and expertise which could not be made public at the time, cf. [19, 16].

1.2   What are we Looking for?

Block ciphers are in widespread use since the 1970s. Their iterated structure is prone to numerous round invariant attacks for example in Linear Cryptanalysis (LC). The next step is to look at non-linear polynomial invariants with Generalised Linear Cryptanalysis (GLC) cf. Eurocrypt'95. A major open problem in cryptanalysis is discovery of invariant properties of complex type. Researchers have until 2018 found extremely few such attacks with some impossibility results [5, 8]. Eventually recent papers [36, 12] show how to construct polynomial invariant attacks for block ciphers, however in almost all such results the Boolean function is extremely weak [12] and invariants are simple and of low degree [36, 12]. Can we do better? What could kind of better outcomes can we expect?

Most symmetric ciphers can be divided into two distinct parts: a set of relatively simple transformations which mix bits and a set of non-linear components (Boolean functions or S-boxes). We call a *"Structural Invariant Attack"* an attack which does not depend a lot on the non-linear components and depends strongly on the structure[3] (i.e. wiring) of the cipher. Most invariant attacks found to date can be seen as *"Weak Structural Invariant Attacks"*: there is a cipher wiring and some special Boolean function such that the cipher has an invariant property which propagates for an arbitrarily large number of rounds. We call these attacks **"weak"**, because typically, the choice of the Boolean function in these attacks is truly pathological: it is almost always of low degree, frequently it does NOT depend on some inputs. For most of the attacks in [12] very clearly, the probability that a cipher with a random Boolean function would be weak in a way as to enable one of the attacks will be negligible or close to zero. The recent paper [12] shows also the existence of polynomial invariant attacks which work for **any** Boolean function, based on the fact that all coefficients in a certain algebraic attack are equal to 0, cf. [12]. However this attack is not very realistic either. Previous attacks working for any Boolean function is an attack which would also work when the non-linear components are secret or unknown were not very convincing. Designing such attacks seems to impose too many strong constraints on other parts of the cipher so that it becomes too weak (e.g. Appendix of [12]) and that maybe no one would accept to use such a contrived cipher (e.g. weak ciphers in [14]). If a cipher has a backdoor, this fact should be well hidden and its components should not only look very strong, but even actually should be very strong in terms of non-linearity etc. [6].

At the end, we consider a more realistic attack scenario than in previous works. We assume that the attacker knows the non-linear components [they are public and were chosen to be very strong] and that he can now **adapt** his attack to these components: for example look for weak keys and weak long-term keys for a given Boolean function. This brings the question of the existence of **"Strong Structural Invariant Attacks"**: which would work when the S-boxes or Boolean functions have not been chosen by the attacker, and are potentially

---

[3] For example the block cipher GOST has a strong structural property: it splits very neatly into two loosely connected parts, cf. Fig. 3 and Fig. 4 in [18] leading to advanced structural truncated differential attacks [17, 19].

random or extremely strong, or just have no[4] property which would be helpful for the attacker (except properties which cannot be avoided).

Related Research. Some basic attacks in cryptanalysis such as study of linear or differential properties propagating with probability 1 shows that such properties are quite "fragile" in the sense that a tiny modification of a Boolean function will ruin the attack and that cipher is simply secure in most cases. We work somewhat at the opposite end of the spectrum which is not so unusual. Several more advanced cryptanalytic attacks are less "fragile" and therefore more "robust" or more "versatile" than basic LC and DC: they work more frequently and it is hard to make they sure they do not work. For example in GOST block cipher, it is illusory to try to secure S-boxes against advanced forms of [Truncated] Differential Cryptanalysis, cf. [19]. In this paper we show that similarly the Generalized Linear Cryptanalysis of [29, 30] is very "robust" and more "versatile" and it works in a surprisingly large number of cases.

Expected Benefits. Frequently one attack hides another. Even if the probability that our *"Strong Structural Invariant Attack"* works is small say 5 %, this could mean that another similar or slightly more general property exists which covers maybe another 20 % of keys etc. At the end potentially we can hope to attack or break some real-life keys.

1.3 Long Term and Short Term Keys vs. Malicious Cipher Wiring

Our attack is very different than what is typically called a weak-key attack: in many weak key attacks the proportion of weak keys is negligible and therefore such attacks are not very practical with some exceptions [26]. We are interested in realistic attack scenarios where the proportion of weak keys is relatively large.

In our study of structural attacks on block ciphers we need to make a useful difference between a "short term key" which will be typically different for each transmission, and a "long term key" which will be used by all users for example for one year. Many commercial block ciphers such as DES or AES do not have such "long term key", however ciphers for more serious say government security purposes, tend to have a "long term key". A modified DES which is a common practice in the industry will have a long term key, an additional secret. A long-term key is a standard feature in T-310, one of the most important block ciphers of the Cold War [24, 35, 20, 34] and it takes a form of specifying the internal connections of the cipher implemented as a printed circuit board with a typical lifespan of one year. A related question is one of the feasibility of "backdooring" a block cipher, or whether it is possible to make the cipher weak on purpose, specifically through this type of "wiring" choice, such as for example the P-box in DES. This sort of studies were always done in block cipher research except that older literature would not use the word backdooring, but rather consider the "design" and "security analysis" of the cipher however. However the final objective yesterday and today is the same: we should observe that either "backdooring" is possible or the designers have done a good job to made

---

[4] A well-known folklore result is that well, very nearly 100 % of Boolean functions do not have any property we would like them to have, see for example [9].

it strong in all[5] cases. Overall for each cipher which has a "long term key" and for many other where we can still modify the cipher wiring without permission from the authors, the wiring can potentially be chosen maliciously. We will then say that we have a ***"Strong Structural Invariant Attack"*** if one can construct a "weak" wiring with a large broad applicability or/and large success probability which is does not require the other (non-linear) components of the cipher to be weak as well. In this paper we show how this can be done for T-310. Our attacks are more realistic[6] the previous *"Weak"*(er) structural invariant attacks described in [12]. When like in T-310 the modification of the long-term key is officially allowed and done on a regular basis, then a feasibility of a weak or malicious choice is an important and eminently practical question.

2 Non-Linear Cryptanalysis

The concept of cryptanalysis with non-linear polynomials a.k.a. Generalized Linear Cryptanalysis (GLC) was introduced by Harpes, Kramer, and Massey at Eurocrypt'95, cf. [29]. A key question is the existence of round-invariant I/O sums: when a value of a certain polynomial is preserved after 1 round. Many researchers have in the past failed to find any such properties, cf. for example Knudsen and Robshaw at Eurocrypt'96 cf. [32]. Bi-Linear and Multi-Linear attacks were introduced [13, 14] for Feistel ciphers branches specifically. The number of such attacks grows as 2^{2^n}, many authors stress that systematic exploration is not feasible [5]. For this reason, we there are extremely few positive results on this topic [36, 12] and any method to approximate the solution is valuable, as one working example as in this paper will typically allow the researchers to find more similar examples.

In this paper and unlike in [36] we focus on invariants which work for 100 % of the keys and we focus on stronger invariants which hold with probability 1. In addition we look at a strong us case: an expensive government cipher in which the block cipher is used for encryption in an extremely low-data rate mode, a lot more costly than 3DES, AES, cf. [24]. Here most cryptanalytic attacks simply do not work (!). However all this complexity is not that useful if we are able to construct powerful non-linear invariants which work for any number of rounds.

2.1 Mathematical Theory of Invariants.

In mathematics there exists an extensive theory of multivariate polynomial algebraic invariants going back to 1850s [25] which deals almost always with invariants w.r.t. linear transformations(!) and has very rarely considered invariants with more than 5 variables and in finite fields of small size. In our work we study

---

[5] Not quite true for T-310, for example in [22] we read that inside the so called class of KT1 keys mandated by the designers, as many as 3 % of are weak w.r.t non-linear cryptanalysis. This paper is about the remaining 97 % of KT1 keys.

[6] Attacks where the Boolean function is very weak are easier to find as it seems, or a plethora of such attacks exists [12]. Yet in practice they are rather irrelevant, as the attacker just cannot hope to change the main non-linear components of a cipher to become weaker.

invariants w.r.t **non-linear** transformations and 36 or more variables over $GF(2)$ and those which simultaneously hold in more than one case [required]. More details about how this compares to what is done elsewhere in mathematics can be found in Section 1.5 of [12].

A well-known polynomial invariant with applications in symmetric cryptography is the cross-ratio, [14] and the "whitening paradox" cf. [14, 15].

2.2 Discovery of Advanced Invariant Attacks

There are two major approaches to our problem: combinatorial and algebraic. A nice algebraic approach it through solving the so called Fundamental Equation cf. [12]. Solving such equation(s), or rather several such equations simultaneously, **guarantees** that we obtain a Boolean function and the polynomial invariant $\mathcal{P}$ which propagates for any number of rounds. However nothing guarantees that the FE equation has any solutions. In this paper we show how to construct just one higher degree invariant which has however the property that [with or without some extra wiring adjustments] it can work with a high probability for any non-linear component used.

2.3 On the Bootstrapping Problem in Cryptanalysis.

This paper is not only about a very specific attack on a very specific cipher but also about cryptanalysis of symmetric ciphers at large. Research in block cipher cryptanalysis has suffered from a bootstrapping problem: we have hardly found any invariant attacks ever except when the set of all possible attacks is not too large, e.g. in Linear Cryptanalysis (LC). An excessively rich space of attacks have been ignored, and we could not find new attacks because we failed to see or imagine how new attacks could even look like. New examples of working attacks (to adapt and imitate in further attacks) are crucial.

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 $\mathcal{P}(\text{Inputs}) = \mathcal{P}(\text{Outputs})$. This concept applied to any block cipher except that such attacks are notoriously hard to find [5] in the last 20 years since [29]. In this paper we are going to work with one specific block cipher with 36-bit[7] blocks. The main point is that any block cipher round translates into relatively simple Boolean polynomials, if we look at just one round. We follow the methodology of [12] in order to specify the exact mathematical constraint, known as the Fundamental Equation or FE, so that we could have a polynomial invariant attack on our cipher. Such an attack will propagate for any number of rounds (if independent of key and other bits). In addition it makes sense following [12] to consider that the Boolean function is an unknown. We denote this function by a special variable Z. We then see that our attack works if an only if Z is a solution to a certain algebraic equation [with additional variables]. The main interest of making Z a variable is to see that even if Z is extremely strong, our attack is likely to nevertheless work.

---

[7] Block size could be increased and our attacks and methods would work all the same.

3.1 Notation and Methodology

In this paper the sign $+$ denotes addition modulo 2, and frequently we omit the sign $*$ in products. For the sake of compact notation we frequently use short or single letter variable names. For example let $x_1, \ldots, x_{36}$ be inputs of a block cipher each being $\in \{0, 1\}$. We will avoid this notation and name them with small letters $a - z$ and letters $M - V$ when we run out of lowercase letters. We follow the backwards numbering convention of [12] with $a = x_{36}$ till $z = x_{11}$ and then we use specific capital letters $M = x_{10}$ till $V = x_1$. This avoids some "special" capital letters following notations used since the 1970s [24, 35, 34]. We consider that each round of encryption is identical except that they can differ only in some "public" bits called F (and known to the attacker) and some "secret" bits called $S1$ or K and $S2 = L$. Even though these bits ARE different in different rounds we will omit to specify in which round we take them because our work is about constructing **one round** invariants (extending to any number of rounds). This framework covers most block ciphers ever made except that some ciphers would have more "secret" or "public" bits in one round. The capital letter Z is a placeholder for substitution of the following kind

$$Z(e_1, e_2, e_3, e_4, e_5, e_6)$$

where $e_1 \ldots e_6$ will be some 6 of the other variables. In practice, the e_i will represent a specific subset of variables of type a-z, or other such as L. At the end Z must be replaced by a formula like:

$$Z \leftarrow Z00 + Z01 * L + Z02 * c + Z03 * Lc + \ldots + Z62 * cklfh + Z63 * Lcklfh$$

Polynomial Invariants We are looking for arbitrary polynomial invariants. For example combinations we could have a symmetric homogenous polynomial like: $\mathcal{P}(a, b, \ldots) = P42 * (abc + abd + acd + bcd) + \ldots$ In this paper polynomials will be constructed primarily as simple products of super simple polynomials. However in general solutions are frequently complex irreducible polynomials, also far from being symmetric, cf. [12]. We simply study a tiny subcase of the general attack of [12]. One of the main points in discovery of innovative attacks on block ciphers is the most previous attacks polynomial invariants were of degree 2 [13, 12, 36]. In this paper the degree of invariants is very high, for example 8.

3.2 Constructive Approach Given the Cipher Wiring

Our attack methodology starts[8] from a given block cipher specified by its ANFs for one round. Specific examples will be show for T-310, and old Feistel cipher with 4 branches and undoubtedly the most important block ciphers of the Cold War with some 3,800 cipher machines in active service in 1989 [24, 35, 20, 34]. This cipher offers great **flexibility** in the choice of the internal wiring and entirely compatible with original historical hardware. The hardware encryption cost with T-310 is hundreds of times bigger than AES or 3DES, cf. [24]. Does it

---

[8] Our approach is to find invariant attack starting from arbitrary rounds ANFs is at the antipodes compared to [14, 15] where the ciphers are very special.

make this cipher very secure? Not quite, if we can construct algebraic invariants which work for any number of rounds. The block size is 36 bits and the key has 240 bits.

3.3   ANF Coding of One Full Round

We number the cipher state bits from 1 to 36 where bits $1, 5, 9 \ldots 33$ are those freshly created in one round, cf. Fig 1. Let $x_1, \ldots, x_{36}$ be the inputs and let $y_1, \ldots, y_{36}$ be the outputs. One round of our the cipher can be described as 36 Boolean polynomials out of which only 9 are non-trivial:

$$y_{33} = F + x_{D(9)}$$
$$Z1 \stackrel{def}{=} Z(S2, x_{P(1)}, \ldots, x_{P(5)})$$
$$y_{29} = F + Z1 + x_{D(8)}$$
$$y_{25} = F + Z1 + x_{P(6)} + x_{D(7)}$$
$$Z2 \stackrel{def}{=} Z(x_{P(7)}, \ldots, x_{P(12)})$$
$$y_{21} = F + Z1 + x_{P(6)} + Z2 + \quad x_{D(6)}$$
$$y_{17} = F + Z1 + x_{P(6)} + Z2 + \quad x_{P(13)} + x_{D(5)}$$
$$Z3 \stackrel{def}{=} Z(x_{P(14)}, \ldots, x_{P(19)})$$
$$y_{13} = F + Z1 + x_{P(6)} + Z2 + \quad x_{P(13)} + S2 + Z3 + x_{D(4)}$$
$$y_9 = F + Z1 + x_{P(6)} + Z2 + \quad x_{P(13)} + S2 + Z3 + x_{P(20)} + x_{D(3)}$$
$$Z4 \stackrel{def}{=} Z(x_{P(21)}, \ldots, x_{P(26)})$$
$$y_5 = F + Z1 + x_{P(6)} + Z2 + \quad x_{P(13)} + S2 + Z3 + x_{P(20)} + Z4 + x_{D(2)}$$
$$y_1 = F + Z1 + x_{P(6)} + Z2 + \quad x_{P(13)} + S2 + Z3 + x_{P(20)} + Z4 + x_{P(27)} + x_{D(1)}$$
$$x_0 \stackrel{def}{=} S1$$
$$y_{i+1} = x_i \text{ for all other } i \neq 4k \qquad (\text{ with } 1 \leq i \leq 36)$$

Two things remain unspecified: the P and D boxes or the internal wiring. In T-310 this specification is called an LZS or *Langzeitschlüssel* which means a long-term key setup. We simply need to specify two functions $D : \{1 \ldots 9\} \to \{0 \ldots 36\}$, $P : \{1 \ldots 27\} \to \{1 \ldots 36\}$. For example $D(5) = 36$ will mean that input bit 36 is connected to the wire which becomes $U5 = y_{17}$ after XOR of Fig. 1. Then $P(1) = 25$ will mean that input 25 is connected as v1 or the 2nd input of $Z1$. We also apply a special convention where the bit S1 is used instead of one of the $D(i)$ by specifying that $D(i) = 0$.

3.4   The Substitutions.

Overall one round can be described as 36 Boolean polynomials of degree 6; out of which only 9 are non-trivial. One round of encryption is viewed as a sequence of substitutions where an output variable is replaced by a polynomial algebraic
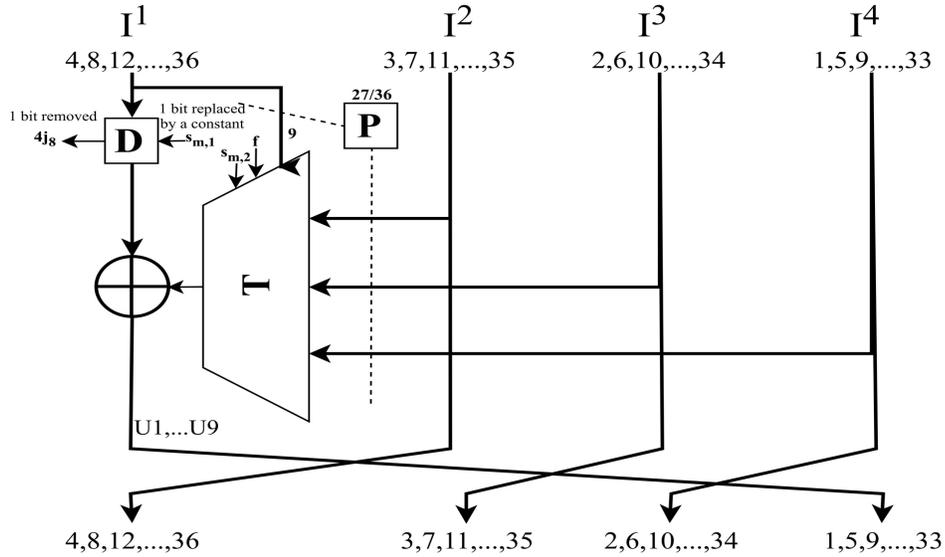
Fig. 1. T-310: a peculiar sort of Compressing Unbalanced Feistel scheme.

expression in the input variables. The full formulas in the general case can be found in [12]. Here is a concrete example following the cipher specification step-by-step for the long-term key 551 used in [12]:

$$a \leftarrow b$$
$$b \leftarrow c$$
$$c \leftarrow d$$
$$d \leftarrow F + i$$
$$[\ldots]$$
$$[\ldots]$$
$$V \leftarrow F + Z1 + O + Z2 + q + L + Z3 + i + Z4 + k + K$$

In order to have shorter expressions to manipulate we frequently replace $Z1 - Z4$ by shorter abbreviations Z, Y, X, W respectively. We also replace S2 by a single letter L (used at 2 places). The other key bits $S1 = K$ will only be used if some $D(i) = 0$.

4 The Fundamental Equation

In order to break our cipher we need to find a polynomial expression $\mathcal{P}$ say

$$\mathcal{P}(a, b, c, d, e, f, g, h, \ldots) = abcdijkl + efg + efh + egh + fgh$$

using any number between 1 and 36 variables such that if we substitute in $\mathcal{P}$ all the variables by the substitutions defined we would get exactly the same polynomial expression $\mathcal{P}$, i.e. $\mathcal{P}(Inputs) = \mathcal{P}(Outputs)$ are equal as multivariate polynomials. We obtain:

Definition 4.1 (Compact Uni/Quadri-variate FE). Our "Fundamental Equation (FE)" to solve is simply a substitution like:

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

or more precisely

$$\mathcal{P}(a, b, c, d, e, f, g, h, \ldots) = \mathcal{P}(b, c, d, F + i, f, g, h, F + Z1 + e, \ldots)$$

where again $Z1 - Z4$ are replaced by Z, Y, X, W. In the next step, Z will be replaced by an Algebraic Normal Form (ANF) with 64 binary variables which are the coefficients of the ANF of Z, and there will be several equations, and four **instances** Z, Y, X, W of the same Boolean function:

Definition 4.2 (A Multivariate FE). At this step we will rewrite FE as follows. We will replace Z1 by:

$$Z \leftarrow Z00 + Z01 * L + Z02 * j + Z03 * Lj + \ldots + Z62 * jhfpd + Z63 * Ljhfpd$$

Likewise we will also replace $Z2$:

$$Y \leftarrow Z00 + Z01 * k + Z02 * l + Z03 * kl + \ldots + Z62 * loent + Z63 * kloent$$

and likewise for $X = Z3$ and $W = Z4$ and the coefficients $Z00 \ldots Z63$ will be the same inside $Z1 - Z4$, however the subsets of 6 variables chosen out of 36 will be different in $Z1 - Z4$. Moreover, some coefficients of $\mathcal{P}$ may also be variable.

In all cases, all we need to do is to solve the equation above for Z, plus a variable amount of extra variables e.g. $Z63$. This formal algebraic approach, if it has a solution, still called Z for simplicity, or $(\mathcal{P}, Z)$ will **guarantee** that our invariant $\mathcal{P}$ holds for 1 round. This is, and in this paper we are quite lucky, IF this equation does not depend on three bits F, K, L. This is the discovery process of [12] which we do not use here. We rather work with basic paper and pencil maths and build our attack from scratch in stages.

5   Construction of An Attack

Our approach is to try to build some simple invariant attacks which initially will not work at all, but they will make us consider under which conditions they might work. The sort of result we are looking for is that a certain polynomial is an invariant if and only if a certain equation is true, the same as the FE approach before, except that we want to help to make it happen in several stages with some simplifications on the way. We recall that the Fundamental Equation FE is an equation such that an invariant attack $\mathcal{P} \to \mathcal{P}$ for 1 round holds if and only if FE is zero, cf. Def. 4.1 and Def. 4.2. We also consider more general transitions of type

$$\mathcal{P} \to \mathcal{Q}$$

and for any number of rounds, which should be interpreted as the value of $\mathcal{P}$ for the input variables should be equal to the value of $\mathcal{Q}$ for the output variables. In other terms the sum of these two values is zero, and informally we talk about "I/O Sums" which we already had in linear cryptanalysis [ignoring key bits] and here we do non-linear cryptanalysis (and we also try to eliminate all the key bits). It is important to see that some transitions are obvious and unconditional. For example in T-310, cf. Fig. 1, if bc denotes the product of variables 35 and 34 in our standard (backwards) numbering scheme, and cd is the product of variables 34 and 33, we have:

$$cd \to bc$$

which is un unconditional transition true in all cases. Here the polynomial $\mathcal{P} = cd$ taken and the input of the cipher is always equal to the value of the polynomial $\mathcal{Q} = bc$ at the output of one round of encryption. Now some other transitions are less obvious and will work if a certain I/O sum of two polynomials will be zero [which will maybe never happen with certainty]. For example in T-310 we always have the trivial transitions $d \to c \to b \to a$ and at degree 2

$$cd \to bc \to ab$$

and we are likely to be trouble with the next step. The problem in T-310 is actually that a is a multiple of 4, cf. Fig. 1. In general a is lost (erased) in the next round and for example $y_{33} = F + x_{D(9)}$ is created cf. general formulas in Section 3.3 where y_{33} is denoted by the letter d. Now imagine that $D(9) = 36$ and $F = 0$ which means that $a \to d$ and also $ab \to cd$ which happens when $F = 0$, i.e. not always. In this case, in the same way as with FE, we will call a **Transition Equation** or TE the sum of two polynomials for a transition of type $\mathcal{P} \to \mathcal{Q}$ for any number of rounds (typically just 1 round) such that the property works if and only if our equation is zero. Here our pre-condition is $D(9) = 36$ and our **Transition Equation** or TE is simply $TE = F$. In other terms if we insure that TE is 0 our two 4-round property works. Both our FE and TE are just I/O sums or sums of two polynomials, the input polynomial and the same polynomial after the substitutions defined on page 9: such as $a \leftarrow b$ and $d \leftarrow F + a$ knowing that the pre-condition $D(9) = 36$ is what makes that $d \leftarrow F + a$ here.

Many further steps are needed in constructing the attack but they key point is that we are going to try to make this polynomials do something interesting and eventually construct a working invariant polynomial in several steps, while trying to make as few assumptions as possible [e.g. $D(9) = 36$], so that our attack will apply to a large class of keys. We will no start from scratch and we do longer assume that $D(9) = 36$.

5.1 Eliminating FKL

In this process there are some heuristics, for example we will try to eliminate the secret key variables K, L and the IV or public round constant variable F to be eliminated early on, an approach suitable for a human, rather than having very complex polynomials where some variables would maybe disappear at the end, which approach would be potentially more complex and suitable only if we used powerful formal algebra or constraint satisfaction tools to find a solution. Instead we work by paper and pencil. One of key problems with T-310 is that some sums are very long, for example $y_9 = F + Z1 + x_{P(6)} + Z2 + x_{P(13)} + L + Z3 + x_{P(20)} + x_{D(3)}$ and the most such expressions contain F and many of the $Z1 - Z4$. A potential solution is that we can eliminate many (but maybe not all) variables quite easily by XORing together consecutive freshly created outputs for example 9 and 13.

We have for example:

$$y_9 + y_5 = x_{D(3)} + W(x_{P(21)}, \ldots, x_{P(26)}) + x_{D(2)}.$$

and similarly

$$y_{25} + y_{21} = x_{D(7)} + Y(x_{P(7)}, \ldots, x_{P(12)}) + x_{D(6)}.$$

where W is a shorter name for the function $Z4$ and Y is another name for $Z2$ which functions are identical except that the inputs are typically connected to two disjoint sets of variables.
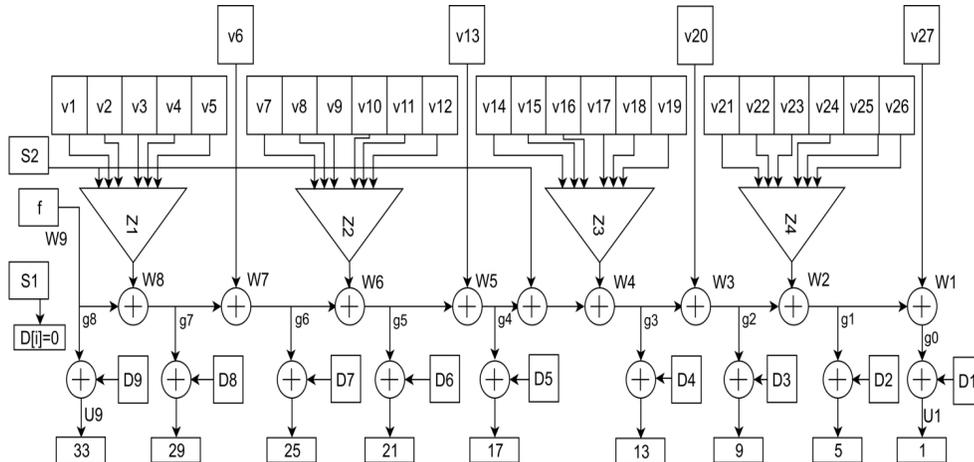

Fig. 2. The internal structure of one round of T-310 block cipher.

5.2 Connecting $Z2$ and $Z4$

A direct inspiration for our attack is the attack with key 799 described in Section 8.1 of [12]. This attack achieves something quite incredible which was previously seen in stream cipher attacks cf. [10, 11] but seems very hard to achieve in block cipher cryptanalysis. It manages to connect two remote parts on both sides of the cipher, into an attack which eliminates everything else. This even though variables uses in our attack can and do depend countless extra variables. Yet all variables we do not want to see get eliminated. This attack is a model for us except that we are aiming at an attack would work for a Boolean function which would be infinitely more complex than the one obtained in Section 8.1 of [12]. Moreover will work on the couple of $Z2$ and $Z4$ and avoid Z1 because one of the inputs of $Z1$ is the key bit $L = S2$ which we want to avoid, (likewise we avoid $Z3$ as the same key bit L is XORed to its output).

Looking at the two equations above, we will now connect Y and W in the simplest possible way, we will assume that (order inside these two pairs does not matter):

$$\begin{cases} \{D(2), D(3)\} = \{6 \cdot 4, 7 \cdot 4\} \\ \{D(6), D(7)\} = \{2 \cdot 4, 3 \cdot 4\} \end{cases}$$

where the numbers $\{6 \cdot 4, 7 \cdot 4\}$ are simply the bits numbers obtained from the second equation after 3 steps with trivial transitions like $25 \to 26 \to 27 \to 28$ later. Our goal is basically in some way connect these two equations to form a closed cycle with 8 steps as follows, hypothetically:

$$5, 9 \to 6, 10 \to 7, 11 \to 8, 12 \to ?21, 25 \to 22, 26 \to 23, 27 \to 24, 28 \to ?5, 9$$

where the question marks relate to transitions which are not true and will actually never become true. Now we define the following 8 polynomials:

$$\begin{cases} A \stackrel{def}{=} (i + m) & \text{which is bits } 24, 28 \\ B \stackrel{def}{=} (j + n) & \text{which is bits } 23, 27 \\ C \stackrel{def}{=} (k + o) & \text{which is bits } 22, 26 \\ D \stackrel{def}{=} (l + p) & \text{which is bits } 21, 25 \\ E \stackrel{def}{=} (y + O) & \text{which is bits } 8, 12 \\ F \stackrel{def}{=} (z + P) & \text{which is bits } 7, 11 \\ G \stackrel{def}{=} (M + Q) & \text{which is bits } 6, 10 \\ H \stackrel{def}{=} (N + R) & \text{which is bits } 5, 9 \end{cases}$$

and then our cycle becomes

$$H \to G \to F \to E \to ?D \to C \to B \to A \to ?H$$

Now we are going to work on polynomials of type say BD and it easy to see that $BD \to AC$ after one round, and again we are stuck because the polynomial AC contains two multiples of 4 which are $8, 12$ which are erased inside this round so that we do not see an easy transition into something very simple.

Now we are ready for a big jump. As we cannot apparently achieve a lot with low degree invariants, we will directly consider an invariant of degree 8.

5.3 An Interesting Question

The question is assuming our 4 assumptions on $D(2), D(3), D(6), D(7)$, under what conditions we could have $P = ABCDEFGH$ to be an invariant for 1 round, in other terms what is the FE? The answer is remarkably simple.

Theorem 5.4 (A Simple Degree 8 Attack). Let

$$\mathcal{P} = ABCDEFGH$$

then P is a non-zero polynomial[9] and it is a one-round invariant if and only if the FE is equal zero for any input of the cipher and any F, K, L and this FE is actually equal to

$$FE = BCDFGH \cdot ((Y + E)W(.) + AY(.))$$

Note. Here the bits which are used as inputs to $W()$ and $Y()$ are exactly those which would be used inside the cipher and we study them in more detail later.

Proof: When necessary we will distinguish input and output-side variables and polynomials by an index in the exponent such as A^o vs. A^i. In fact in most cases this is not even needed as there is no ambiguity: at the end the FE is expected to be written using input side and internal secret/public variables FKL ONLY and all the output-side variables must be eliminated, thus removing any ambiguity (at later stages). We recall our assumption:

$$\begin{cases} \{D(2), D(3)\} = \{6 \cdot 4, 7 \cdot 4\} \\ \{D(6), D(7)\} = \{2 \cdot 4, 3 \cdot 4\} \end{cases}$$

and in Section 5.1 we have already established that

$$H^o = y_9 + y_5 = x_{D(3)} + W(.) + x_{D(2)} = W(.) + A^i$$

and

$$D^o = y_{25} + y_{21} = x_{D(7)} + Y(.) + x_{D(6)} = Y(.) + E^i$$

which expressions can be reinterpreted as showing how the polynomials D and H on the output side can be rewritten as expressions using only input-side variables and the combination of bits used here were already chosen in Section 5.1 in such a way that all bits FKL are already eliminated. Following Def. 4.1 all we have to do is to add the polynomial $ABCDEFGH$ to itself on the output side after substitutions with input-side variables. This latter polynomial is equal to:

$$A^o B^o C^o D^o E^o F^o G^o H^o = B^i C^i D^i (Y(.) + E^i) F^i G^i H^i (W(.) + A^i) =$$

$$BCDFGH(Y(.) + E^i)(W(.) + A^i) = BCDFGH(YW + YA + EW + EA)$$

---

[9] Important to check each time we multiply many terms.

Finally we add this expression to the input and obtain that the FE is equal to:

$$ABCDEFGH + BCDFGH(YW + YA + EW + EA) = BCDFGH((Y+E)W + AY)$$

which is the exact result we need:

$$FE = BCDFGH \cdot ((Y(.) + E)W(.) + AY(.))$$

5.5 Solving the FE

In [12] constructing some invariants seems an extremely complex process. In this paper everything seems just incredibly simple. We just need to make sure that FE is equal to zero. We want to solve:

$$0 = BCDFGH \cdot \big((O + y + Y)W(x_{P(21)}, \ldots, x_{P(26)}) + (i + m)Y(x_{P(7)}, \ldots, x_{P(12)})\big)$$

Interestingly in this expression we do not enumerate the inputs of the first Y which are also $Y(x_{P(7)}, \ldots, x_{P(12)})$, which is because we aim at eliminating the term $(O + y + Y)$ completely later on by making sure that the product of other terms is always zero. We also have a polynomial where F, K, L are already eliminated. However there are many other variables. At the first sight it may seem that our equation which is polynomial of degree 18 in as many as 12 input variables plus 63 variables specifying the Boolean function is unlikely to be systematically equal to 0. What we are going to show now is simply mind-blowing: this polynomial can be made to be exactly zero in all 2^{12} cases, even though **no effort** will be made whatsoever to use a particularly simple Boolean function or even to influence it. In fact we have chosen our example really well and something incredible will happen. This equation will become solvable after a certain choice of which 12 bits will be connected to the 12 inputs of $Y()$ and $W()$ and then we will see that an exponentially large space of Boolean functions will just disappear, they can be annihilated and our equation will be zero which happens later in Section 6. Before we get there, the first step is to convert a single equation which is a sum into insuring that both components are zero. For example we will aim at insuring simultaneously that:

$$\begin{cases} CFH \cdot W(x_{P(21)}, \ldots, x_{P(26)}) = 0 \\ BDG \cdot Y(x_{P(7)}, \ldots, x_{P(12)}) = 0 \end{cases}$$

which will imply that our FE is equal to 0 in all cases, i.e. our one round invariant attack works for any key and any F. The choice of the two subsets of $BCDFGH$ is arbitrary. Moreover there are many solutions which work. At this stage it may be hard to believe but we will see that it is sufficient to put as inputs to W the 6 inputs of CFH in a well-chosen order, not every order works but many do, and the inputs of Y needs to be the 6 inputs of BDG in another order, and that both orders must be compatible (both functions W and Y are just two instances of the same Boolean function). Moreover this actually works

for any possible way to split the product $BCDFGH$ in two products of degree 3 for example $BCF \cdot DGH$ provided that later the inputs of W and Y are suitably aligned. Here is for example one solution which works well as we will see later:

```
265: P=1,20,33,34,15,13,27,6,10,23,21,25,16,14,2,4,3,
19,35,29,26,9,5,22,7,11,17 D=36,28,24,32,20,8,12,16,4
```

below we explain is full detail how such a key can be created.

5.6   The Satisfiability Problem

Initially we want to do something quite which may seem extremely difficult. Make sure that a polynomial equation of degree 18 in as many as 12 input variables and 64 extra variables $Z00 - Z63$ which represent the coefficients inside the ANF of our Boolean function, cf. Def. 4.2 is systematically equal to 0 in all 2^{12} cases, and that we will do it for an absolutely general Boolean function in 6 variables which can be random and is not chosen by the attacker, i.e. the solution should work for as many choices of $Z00 - Z63$ as only possible. The we made it harder be requesting that the two terms of the equation are simultaneously zero. Overall we recall that we decided to insure simultaneously that for example:

$$\begin{cases} CFH \cdot W(x_{P(21)}, \ldots, x_{P(26)}) = 0 \\ BDG \cdot Y(x_{P(7)}, \ldots, x_{P(12)}) = 0 \end{cases}$$

and that the 6 inputs of CFH will become inputs of W and vice versa for Y. For example with our example 265 above we have $P(24) = 22$, which corresponds to the 4-th input d of W. In fact we have already assigned the right 12 variables at some "right" 6+6 places inside LZS 265.

Now we need to see that this alone it not yet sufficient to make the attack work. We need to make sure that the polynomial CFH is an annihilator of the Boolean function W, and that simultaneously BDG is an annihilator for the second instance of the same Boolean function [two conditions]. But is this at all possible that a Boolean function **chosen at random** or not chosen by the attacker would have such peculiar annihilators such as CFH? The answer will be provided in the next section. At this moment we have reduced [module checking that all the bits are at the right positions] the problem of backdooring a block cipher to a problem of annihilating a Boolean function **twice** with two annihilators of the form $(a + b)(c + d)(e + f)$.

This is going to be a lot easier than one might expect.

6 Some Basic Results On Boolean Functions

Let B_n be the ring of all Boolean functions in n variables with $n = 6$ with the usual addition and multiplication of polynomials modulo 2. We recall that an annihilator of a Boolean function $f \in B_n$ is a function $g \in B_n$ such that the product of these functions $f \cdot g$ is zero: i.e. $\forall_{x \in \{0,1\}^n}$ $f(x) \cdot g(x) = 0$. There are countless divisors of zero in the ring B_n and annihilators exist in vast [10] quantities for any Boolean function. What is more surprising is that annihilators with special properties which will be exactly those which we need in our attack also exist with a large probability. Fir of all, is it normal that a Boolean function of degree 6 would have an annihilator of degree 3? Yes, it is and we have:

Theorem 6.1 (Annihilator Degree Bound). Let Z be a Boolean function with $n = 6$ variables $abcdef$. Then either Z of $Z + 1$ has at least one annihilator of degree 3.

Proof: Following Thm 6.0.1 in [10] there exists a function g of degree at most 3 such that

$$gZ = h$$

where h has degree of at most 3. In addition following Theorem C.0.1. in the Appendix, of [10], there exists a solutions g such that either $h = 0$ or $h = g$. In the first case $gZ = 0$ and we found an annihilator g of degree 3. In the second case $gZ = g$ and so $g(Z + 1) = g + g = 0$ and we found an annihilator of degree 3 for Z+1.

Remark. It is NOT true however that every function Z has an annihilator of degree 3, a nice counter-example is $abcdef + 1$ which has no annihilators at any degree less than 6. Moreover such counterexamples are quite rare. In general for most Boolean functions both Z and $Z + 1$ will have numerous annihilators of degree 3.

6.2 Annihilators of a Special Form

In addition to the existence of annihilators of degree 3, and in addition to the fact that h is likely to very special (or that there exists annihilators g s.t. h is very special) it is interesting that with large probability there exist annihilators such that g is also very special. In what follows we are going to establish some simple yet significant results on this question showing that even though Z is an arbitrary Boolean function, the space of annihilators is very likely to contain some very interesting and highly symmetric polynomials. These facts are essentially consequences of the fact that the space of annihilators of Z is an ideal, i.e. if $fZ = 0$ then $fgZ = 0$, and it is also a highly structured finite partially ordered set (for divisibility) which being finite is also a lattice, i.e. every two elements have LUB/Sup and in GLB/Inf and where the top element which dominates all the other is 0. This lattice contains many some special quite symmetric elements which dominate many other in the partial order. We start by an auxiliary result.

---

[10] For example for every function $f \in B_n$ any multiple of $f + 1$ is an annihilator for f.

Theorem 6.3 (Special Type of Annihilators). Let f be a Boolean function with $n = 6$ variables $abcdef$ chosen uniformly at random. We consider the following multiple of f: $g = f(a + b)(c + d)(e + f)$. We do it many times and by linear algebra [or by a careful study of how different monomials imply also the presence of other monomials] we observe that this polynomial lives in a linear space of dimension only 8. Consequently if f is chosen at random then g is equal to 0 with high probability equal to 2^{-8}.

Proof: This result comes from numerous symmetries imposed on g by the linear factors. The linear space of all possible polynomials of type $g = f(a + b)(c + d)(e + f)$ is finite and a quick computation with SAGE maths software shows that its dimension is only 8. This means that every polynomial in this space can be seen as being of the form

$$a_0 G_0 + \ldots + a_7 G_7$$

where $a_i \in \{0, 1\}$ and G_i are some polynomials forming a basis. Then we need to that the probability distribution of the 8-tuples a_i must be uniform, this is because it is an image of the input space G_n which is an Abelian group for addition, through a linear application $f \mapsto f(a + b)(c + d)(e + f)$ and all preimages for any 8-tuple a_i are cosets w.r.t. H defined as sub-group of annihilators of $(a + b)(c + d)(e + f)$, which must be of equal sizes as they define a partition of B_6 and in fact each polynomial $f \in B_n$ can be viewed as bijection which maps one coset into another coset. Given the uniform distribution, the probability that our polynomial is zero is exactly 2^{-8}.

Finally here is the main result which we need in block cipher cryptanalysis:

Theorem 6.4 (Existence of Structured 2x2x2 Annihilators). Let Z be a Boolean function with $n = 6$ variables $abcdef$ chosen uniformly at random. The probability that $Z(a + b)(c + d)(e + f) = 0$ or similar for at least one permutation of variables $abcdef$ is approximately equal to 0.254.

Justification: For any 6 variables there are $\binom{6}{2} \cdot \binom{4}{2}/3! = 15$ ways to select three pairs of variables if we consider as identical all the 3! permutations of the three sets of 2. In the first approximation, each of these choices has a probability of 2^{-8} of being zero, cf. Thm. 6.3 above. If all these 15 choices of a polynomial of type say $Q = (a + f)(c + b)(d + e)$ were independent we would obtain that the probability that none of the 15 choices works is about $(1 - 2^{-8})^{15} \approx 0.94$. Therefore we would obtain a result $Pr \approx 0.06$. In reality this probability is substantially higher. This comes from the fact that the space of annihilators of Z contains multiple elements of degree 3 typically, for example for the original Boolean function of T-310, the dimension of this space A is 10 while Thm. 6.1 would guarantee at least 1. The question whether A which is a subgroup of B_6 contains at least one of the 15 elements [which do not form a subgroup] is complicated and it is possible to see that A contains always either 0, 3 or exactly 15 such elements which means that this is not at all a set of random and independent events with complex dependencies. Overall, a computer simulation shows that this $Pr \approx 0.254$.

7 Putting It All Together

We recall that in order for our invariant $\mathcal{P}$ to be preserved after one encryption round, we need to solve [in terms of the wiring of the cipher and the choice of the Boolean function] the following two equations:

$$\begin{cases} (a+b)(c+d)(e+f) \cdot W(a,b,c,d,e,f) = 0 \\ (a'+b')(c'+d')(e'+f') \cdot T(a',b',c',d',e',f') = 0 \end{cases}$$

In our solution we are going use the same annihilator for both W and Y, which is not even required (therefore there exist way more ways of backdooring this cipher than the simple method we present here). Following our Thm. 6.4, this works for 1/4 of all Boolean functions, and all we need to do is to assign in arbitrary way the 2 time 3 sets of 2 inputs inside each Boolean function. For example, in order to identify $C \overset{def}{=} (k+o) = x_{22} + x_{26}$ to the sum $(a+b)$ for W we must insure that $P(21) = 22$ and $P(22) = 26$ or vice versa. This must be done for all 12 variables a, b, c, d, e, assigned at $P(7)$ to $P(12)$ and the variables a', b', c', d', e', f' are assigned at $P(21)$ to $P(26)$. All this is actually already done in key 265 above and there are numerous ways to do it.

7.1 The Bijectivity Problem

A slight technical difficulty is also that we want our long-term key to be a bijection on 36 bits. The reason for this requirement is that if it is not bijective, the cipher is already broken by a ciphertext-only attack, see [21]. This in practice it not so difficult and we have used a version of a free open source tool described in Appendix I.11 of [20] which allows to create many random bijective LZS keys keys with arbitrary specific constraints on $D()$ and $P()$. It appears that this process works with probability 1, i.e. once we have specified our constraints as above (four values for the $D(i)$ and 12 values for the $P(i)$) there are still 5+15 coefficients being numbers between 0 and 36 which can be adjusted to make a key which is bijective (and looks like a secure key w.r.t all previously known attacks cf. [21]). The space of long-term keys is simply enormous and our attack does not seem to contradict the bijectivity requirement in any way. We obtained for example the following solution:

```
265: P=1,20,33,34,15,13,27,6,10,23,21,25,16,14,2,4,3,
19,35,29,26,9,5,22,7,11,17 D=36,28,24,32,20,8,12,16,4
```

and there are plenty of other possibilities: we can also assign inputs a, b of W to be D, or F etc and we are always as it seems able to adjust the missing 15+5 coefficients to form a bijective LZS.

7.2   How to Backdoor T-310

As any Boolean function works in our attack with large probability, cf. Thm. 6.4 all we have to do now is to try some super secure Boolean functions and expect that the cipher will be broken sort of accidentally, well, the accident will happen 25 % of th time. For example with key 265 (and with any other variant of our attack) we have for example the following solution:

$$ad + bc + be + de + ef + acd + acf + ade + bcf + bdf + cef+$$

$$abcd + abce + abef + bcdf + abcde + acdef + 1 + a + e + c$$

which differs from the original Boolean function by only one linear term, the last letter was changed from f to c. We apologize that it does not work for the original Boolean function but we consider that this is not because this Boolean function is well chosen. It is rather simply a mater of good versus bad luck, and the chances that the original Boolean function would work without any modification were quite high, about 1/4.

7.3   An Attack Can Hide Another

We do not only find the exact invariant $\mathcal{P}$ which was the intended property. With this Boolean function our cipher has a linear space of dimension as high as 19 composed of various invariants being polynomials in $A - D$ including some complex invariants of degree 7 and probably many more which we are unable[11] to detect. The dimension of 19 is not constant and varies for different LZS and different Boolean functions. This means that in addition to the attack we have constructed we ALWAYS obtain countless other invariant attacks the existence of which we have not anticipated, more complex, more sporadic, and this happens systematically without any exceptions as it seems. This is another testimonial to how powerful high degree polynomial invariant attacks are in practice.

8   Cryptanalytic Relevance of Round Invariant Attacks

Many cryptanalytic attacks do not matter at all for the security of encryption systems in real-life attacks scenarios. Or so it may seem initially. The T-310 cipher actually operates in a very secure and super-paranoid low-data rate encryption mode which makes that many attacks even if they seem very strong, still do not seem to break it in sense of decrypting some communications or recovering the secret key, cf. [22]. However if we dig a big deeper we will find that we can actually break T-310 and recover the key and decrypt communications.

------

[11] It is infeasible to search the space of all possible invariants of degree 8.

8.1 Decryption Oracle Attacks

We refer to [23] to see one major method in which round invariant attacks such as studied and constructed in this paper affect the security of T-310 in real-life attack scenarios. Such attacks exist, but they require a lot more work. The actual attack is a certain type of long-distance[12] slide attack with a decryption oracle, which is far from being obvious and has additional requirements[13]. One example of a relevant non-linear attack with LZS 771 is given in [23].

8.2 Higher-Order Correlation Attacks

A second major way to exploit these properties is to see that many such attack create partitions of the space of 2^{36} elements into two sets of **unequal** sizes which inevitably leads to powerful ciphertext-only higher-order correlation attacks. Basically we get partitions which will bias the cipher internally in a pervasive way which works for any number of rounds. This modifies very deeply the joint probability distributions for numerous sets of internal variables and is likely to introduce some biases in such distributions. Given the fact that they key is sued in a regular periodic way in this cipher, and that the plaintext in a natural language will also be strongly biased cf. [21], such biases lead directly to ciphertext-only key recovery attacks. The actual exploitation requires many additional technical steps. Several examples of such biases appear in a separate paper. We given here one example which comes from the Section 10 of [12]. Let

$$\mathcal{P} = eg + fh + eo + fp + gm + hn + mo + np$$

and we use LZS 551^{14} from [12]:

```
551: P=17,4,33,12,10,8,5,11,9,30,22,24,20,2,21,34,1,25,
13,28,14,16,36,29,32,23,27 D=0,12,4,36,16,32,20,8,24
```

This gives a remarkably simple FE being simply

$$Yg + Yo + gm + mo$$

and one solution is $Z = 1 + d + e + f + de + cde + def$. This attack is very different[15] compared to what we see in this paper: $\mathcal{P}$ is irreducible and not at all a product of linear terms like ABC. The invariant $\mathcal{P}$ as above has very interesting properties: it does indeed divide the space of possible cipher states into two unequal size sets and introduces strong biased on the various tuples of internal variables (for subsets of 8 variables $e - h$ and $m - p$ actually used in $\mathcal{P}$). We have checked that already with $N = 3$ variables we obtain some strong biases on N-tuples of variables which can be used in higher-order correlation attacks. For example when $\mathcal{P} = 0$ there are 40 events with $efg = 1$ but only

---

[12] As opposed to the shorter-distance 1-bit correlation attacks of [23].

[13] Such as looking for invariants actually using some key bits in a very specific way rather then totally eliminating them as we do here and in [12].

[14] A strong example fully compliant with all the requirements of the KT1 class of keys approved to protect government communications cf. [21, 35, 34].

[15] Actually Section 10 in [12] shows further keys such as 558 or 550 with more sophisticated quartic invariants closer to what we see in present paper.

16 events with $ef(g + 1) = 1$. Here the main idea is that correlation attacks involving $N = 1$ or $N = 2$ variables will not work, but starting from $N = 3$ we have plenty of interesting correlations which may be exploited by the attacker (which is not obvious in a well-designed cipher and leads to further technical questions).

9 Further Developments and Open Problems

We have just started to explore an incredibly rich space of new attacks and the number of possibilities is overwhelming. We list here a few questions which we would like to find answers to in the future:

Question 1. It it possible to find a modified version of the current attack which also works[16] with the exact original Boolean[17] function? If so it can be used directly to backdoor the original historical encryption hardware by inserting a printed board with a malicious LZS.

Question 2. Is it possible to find strong keys against our attacks or guarantee in any way that the attacks will not work, cf. [5, 8]? For example:

Question 2a. Are there some specific requirements in the style of KT1 or KT2 classes cf. [21] which avoid polynomial invariant attacks?

Question 2b. Is it possible to find some super-strong Boolean functions which can make T-310 secure against invariants attacks in the style of

Question 3. What about other block ciphers? DES? Ciphers not being Feistel ciphers such as AES? In theory, as in [12] our attack is applicable directly to many other ciphers. In practice, given the fact that most modern ciphers use way more key material in one round than T-310 cf. [24], at some moment they could become secure or maybe even provably secure against polynomial invariant attacks, see Section 5.6 in [12] and [5, 8].

---

[16] The probability it would is expected to be high very e.g. 25 % in our current attack.

[17] It is important to see that a Boolean function has typically more than one special or extremely simple annihilator, see Appendix I.19 in [20].

10   Conclusion.

This paper proposes a new way of attacking block ciphers. A non-linear invariant attack which has some unique features. Given a non-linear component not chosen by the attacker, or rather largely ignoring the exact component, we show step by step using simple by paper and pencil maths with multivariate polynomials, how one can construct a polynomial invariant property of degree 8 which has the ability to work in a large number of cases. It becomes than easy to make our invariant work for an arbitrarily large number of rounds and for any key and any IV. We impose very few constraints on internal cipher wiring (the LZS) and on the Boolean function so that this or similar attack may also work accidentally.

In the most recent paper on this topic [12] the Boolean functions were extremely weak and pathological in order to make the attack work. In contrast our new attack is able to adapt to (or work directly) with arbitrary Boolean functions. The success probability of our attack is surprisingly high and close to 25 % modulo some adaptations in the cipher wiring (the LZS). This is due to the fact that our invariant polynomial properties operate at a higher degree than previously [12, 36], and also due to some surprising results on the existence of symmetric annihilators of degree 3 with special properties for Boolean functions with 6 variables.

In this paper we show that a block cipher can be insecure for **almost entirely structural reasons** and that a good choice of complex highly-nonlinear Boolean functions or S-boxes does not really help. A general attack (rich in possibilites) with these characteristics was not seen before in block[18] cipher cryptanalysis. It is easy to see that the exact attacks described in this paper will work also when the Boolean function is secret and unknown[19] to the attacker. This even though they work better if we know the Boolean function and we can and do adapt[20] to it. We have tried this with key 265: many of the degree 7 or 8 invariants work nevertheless for arbitrary Boolean functions chosen at random. The success probability is then lower, closer to 1 % than to 25 %. Now given the fact that Boolean functions on 6 variables have frequently more than one interesting annihilator, cf. Appendix I.19 in [20], we also systematically find numerous additional invariants with degrees 7 and 8, we conjecture that the cumulative power of the attack described in this paper is overall substantially[21] higher.

---

[18] In stream cipher cryptanalysis we have worst-case attacks mathematically proven to work for 100 % or all Boolean functions for certain parameter choices, cf. [10, 11].

[19] It is also able to work for several non-linear components used simultaneously, for example imagine that $Z1 \neq Z2$ of that they are key-dependent components [31].

[20] The difference lies in accidental vs. deliberate wiring of inputs of W and Y, in this paper we have adapted LZS 265 to our specific degree 8 invariant to work with one (large) class of Boolean functions covering 25 % of all Boolean functions in B_6.

[21] We expect to obtain even more invariants such that if one does not work for a cipher, another invariant may work, leading to a cumulative success rate closer to say 30 or maybe 80 % than to 1 %. We also anticipate to find attacks working with exact original encryption hardware, cf. Question 1 in Section 9.

References

1. Ange Albertini, Jean-Philippe Aumasson, Maria Eichlseder, Florian Mendel and Martin Schläffer: Malicious Hashing: Eves Variant of SHA-1, in SAC 2014, pp.1-19, Springer LNCS 8781.
2. Martin Albrecht: *Algebraic Attacks against the Courtois Toy Cipher,* In Cryptologia, Volume 32, Issue 3 July 2008 , pp. 220 - 276, 2008.
3. Arnaud Bannier, Nicolas Bodin, and Eric Filiol: *Partition-Based Trapdoor Ciphers,* `eprint.iacr.org/2016/493`.
4. Gregory V. Bard, Shaun V. Ault and Nicolas T. Courtois: *Statistics of Random Permutations and the Cryptanalysis Of Periodic Block Ciphers,* In Cryptologia, Vol. 36, issue 03, pp. 240-262, July 2012.
5. 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.
6. Joan Boyar, Magnus Find, René Peralta: *Four Measures of Nonlinearity,* In Algorithms and Complexity, CIAC 2013, LNCS 7878, pp. 61-72, Springer,
7. Nicolas Courtois, Gregory V. Bard, David Wagner: *Algebraic and Slide Attacks on KeeLoq,* In FSE 2008, pp. 97-115, LNCS 5086, Springer, 2008.
8. Marco Calderini: *A note on some algebraic trapdoors for block ciphers,* last revised 17 May 2018, `https://arxiv.org/abs/1705.08151`
9. Claude Carlet: *The complexity of Boolean functions from cryptographic viewpoint,* In Complexity of Boolean Functions (Dagstuhl, Germany), Dagstuhl Seminar Proceedings, no. 06111, 2006.
10. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback,* Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345–359, Springer. Extended version: `www.nicolascourtois.com/toyolili.pdf`.
11. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs,* ICISC 2004, LNCS 3506, pp. 3–20, Springer 2005. Extended version available on `http://eprint.iacr.org/2003/125/`.
12. Nicolas T. Courtois: *On the Existence of Non-Linear Invariants and Algebraic Polynomial Constructive Approach to Backdoors in Block Ciphers,* `https://eprint.iacr.org/2018/807.pdf`, revised 3 Dec 2018.
13. Nicolas Courtois: *Feistel Schemes and Bi-Linear Cryptanalysis,* in Crypto 2004, LNCS 3152, pp. 23–40, Springer, 2004.
14. Nicolas Courtois: *The Inverse S-box, Non-linear Polynomial Relations and Cryptanalysis of Block Ciphers,* in AES 4 Conference, Bonn May 10-12 2004, LNCS 3373, pp. 170–188, Springer, 2005.
15. Nicolas Courtois: *The Inverse S-box and Two Paradoxes of Whitening,* Long extended version of the Crypto 2004 rump session presentation, *Whitening the AES S-box,* Available at `http://www.minrank.org/invglc_rump_c04.zip`. Main theorem appears in Appendix B of the extended version of [14].
16. Nicolas Courtois, Guilhem Castagnos and Louis Goubin: *What do DES S-boxes Say to Each Other ?* Available on `eprint.iacr.org/2003/184/`.
17. Nicolas Courtois: *An Improved Differential Attack on Full GOST,* in "The New Codebreakers a Festschrift for David Kahn", LNCS 9100, pp. 278-299, Springer, 2016.
18. Nicolas Courtois: *An Improved Differential Attack on Full GOST,* In Cryptology ePrint Archive, Report 2012/138. 15 March 2012, updated December 2015, `http://eprint.iacr.org/2012/138`.

19. Nicolas T. Courtois, Theodosis Mourouzis, Michał Misztal, Jean-Jacques Quisquater, Guangyan Song: *Can GOST Be Made Secure Against Differential Cryptanalysis?,* In Cryptologia, vol. 39, Iss. 2, 2015, pp. 145-156.

20. Nicolas T. Courtois, Klaus Schmeh, Jörg Drobick, Jacques Patarin, Maria-Bristena Oprisanu, Matteo Scarlata, Om Bhallamudi: *Cryptographic Security Analysis of T-310,* Monography study on the T-310 block cipher, 132 pages, received 20 May 2017, last revised 29 June 2018, `https://eprint.iacr.org/2017/440.pdf`

21. Nicolas T. Courtois, Maria-Bristena Oprisanu: *Ciphertext-only attacks and weak long-term keys in T-310,* in Cryptologia, vol 42, iss. 4, pp. 316–336, May 2018. `http://www.tandfonline.com/doi/full/10.1080/01611194.2017.1362065`.

22. Nicolas Courtois, Maria-Bristena Oprisanu and Klaus Schmeh: *Linear cryptanalysis and block cipher design in East Germany in the 1970s,* in Cryptologia, 2018.

23. Nicolas Courtois, Marios Georgiou and Matteo Scarlata: *Slide Attacks and LC-Weak Keys in T-310,* Accepted for publication, will appear in Cryptologia in 2019.

24. 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.

25. Tony Crilly: *The rise of Cayley's invariant theory (18411862),* In Historia Mathematica, Vol. 13, Iss. 3, August 1986, pp. 241–254

26. Nicolas Courtois: *On Multiple Symmetric Fixed Points in GOST,* in Cryptologia, Iss. 4, vol 39, 2015, pp. 322-334.

27. Orr Dunkelman and Nathan Keller: *Linear Cryptanalysis of CTC,* Available at `http://eprint.iacr.org/2006/250/`.

28. E. K. Grossman, B. Tuckerman: *Analysis of a Weakened Feistel-like Cipher,* 1978 International Conference on Communications, pp.46.3.1-46.3.5, Alger Press Limited, 1978.

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. C. Harpes, J. L. Massey: *Partitioning cryptanalysis,* In FSE 97, LNCS 1267, pp. 13–27, 1997.

31. Sandy Harris, Carlisle Adams: *Key-Dependent SBox Manipulations,* In SAC'98, LNCS 1556, pp. 15–26, Springer, 1999.

32. Lars R. Knudsen, Matthew J. B. Robshaw: *Non-Linear Characteristics in Linear Cryptoanalysis,* Eurocrypt'96, LNCS 1070, Springer, pp. 224–236, 1996.

33. Pawel Morawiecki: *Malicious Keccak,* `eprint.iacr.org/2015/1085`

34. *Referat 11: Kryptologische Analyse des Chiffriergerätes T-310/50. Central Cipher Organ, Ministry of State Security of the GDR, document referenced as 'ZCO 402/80', a.k.a. MfS-Abt-XI-594, 123 pages, Berlin, 1980.*

35. *Klaus Schmeh: The East German Encryption Machine T-310 and the Algorithm It Used,* In Cryptologia, vol. 30, iss. 3, pp. 251–257, 2006.

36. 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.