Variable elimination strategies and construction of nonlinear polynomial invariant attacks on T-310

Nicolas T. Courtois[1] and Marios Georgiou[2]

[1]University College London, Gower Street, London, UK
[2]PricewaterhouseCoopers, Nicosia, Cyprus

Abstract. One of the major open problems in symmetric cryptanalysis is to discover new specific types of invariant properties for block ciphers. In this paper we study non-linear polynomial invariant attacks. The number of such attacks grows as 2^{2^n} and systematic exploration is not possible. The main question is HOW do we find such attacks? We have developed a constructive algebraic approach which is about making sure that a certain combination of polynomial equations is zero. We work by progressive elimination of specific variables in polynomial spaces and we show that one can totally eliminate big chunks of the cipher circuit. As an application we present several new attacks on the historical T-310 block cipher which has particularly large hardware complexity and a very large number of rounds compared to modern ciphers e.g. AES. However all this complexity is not that useful if we are able to construct new types of polynomial invariant attacks which work for any number of rounds.

Key Words: history of cryptography, Cold War, T-310, modern block ciphers, Boolean functions, ANF, Feistel ciphers, weak keys, backdoors, Generalized Linear Cryptanalysis, polynomial invariants, I/O sums, multivariate polynomials, Algebraic Cryptanalysis, Partitioning Cryptanalysis, higher-order correlation attacks, ciphertext-only attacks.

1 Introduction, Non-Linear Cryptanalysis

The concept of cryptanalysis with non-linear polynomials a.k.a. Generalized Linear Cryptanalysis (GLC) was introduced at Eurocrypt'95, cf. [14]. 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. [17] and there are extremely few positive results on this topic, cf. [20, 5]. Bi-Linear and Multi-Linear cryptanalysis were subsequently introduced [8, 9] for Feistel ciphers. Our work is rather disjoint compared to recent high-profile results in [20] dealing with SPN ciphers, we focus more on Feistel ciphers. Moreover unlike in [20] we focus on invariants which work for 100 % of the keys and we work on stronger invariants which hold with probability equal to 1 and eliminating all key bits (so that we can ignore the key schedule).

A classical open problem in cryptanalysis is discovery of invariant properties of complex type, cf. recent papers [20, 5]. The space of possible solutions is double exponential. One very general attack is called Partitioning Cryptanalysis (PC), cf. [1, 15, 16]. A more specific approach is to consider only partitions defined by the value (0 or 1) of a single Boolean polynomial. This is of course less general, yet it leads to a more illuminating approaches. Properties are described, discovered and studied with the tools of algebra rather than to happen by some incredible coincidence. The main question in this is paper how to explicitly construct interesting polynomial invariant attacks. First we code the problem as the problem of solving a surprisingly simple single equation of a limited degree which we will call FE which **guarantees** that we have a Boolean function and the polynomial invariant $\mathcal{P}$ which makes a block cipher weak. Our work is mainly about weak keys. Specific examples will be constructed based on a highly complex historical block cipher T-310. We construct numerous examples where the set of solutions to FE is not empty which demonstrates that our approach actually works.

Mathematical Theory of Invariants. There exists an extensive theory of multivariate polynomial algebraic invariants [13]. However mathematicians have studied primarily invariants w.r.t. linear transformations(!) and have rarely considered invariants with more than 5 variables or in finite fields of small size. In our work we study invariants w.r.t **non-linear** transformations (!!!) and up to 36+3 variables over $GF(2)$. A well-known polynomial invariant with applications in symmetric cryptography is the cross-ratio, cf. Sect. 4 in [9].

2 Notation and Methodology

We are looking for arbitrary polynomial invariants of type $\mathcal{P}(\text{Inputs}) = \mathcal{P}(\text{Output ANF})$ where $\mathcal{P}$ is some polynomial. The value of this polynomial (0 or 1) applied before and after one round of a block cipher should not change. In order to have notations, which are as compact as possible, in this paper the sign + will denote addition modulo 2, frequently we will omit the sign * in products and will frequently use short or single letter variable names. Initially the cipher state uses variables of type x_1 or y_{36} or e_1 which are binary variables $\in \{0, 1\}$. In polynomials $\mathcal{P}$ we will replace these by shorter one-letter abbreviations $a - z$, $M - V$, etc. Quite importantly, we consider, which is rarely done in cryptanalysis, that the Boolean function is an unknown denoted by a special variable Z. We will then postulate that Z may satisfy a certain algebraic equation [with additional variables] and then this equation will be solved in order to determine Z. Our Boolean function has 6 inputs.

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

where $e_1 \dots e_6$ 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 some other such as F, K, L. There will be 4 distinct copies of $Z()$ known as $Z1 - Z4$ or Z, Y, X, W which are later replaced be an ANF polynomial with 64 coefficients, e.g.:

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

Further capital letters $S1, S2, K, L, F, Z$ will be used to represent some very "special" sorts of variables which are placeholders for something more complex.

$S1, S2$ or in one-letter versions K, L, will be bits of the secret key used in a given round We then use the capital letter F to represent the bits which depends on the IV: a round-dependent constants (known to the attacker typically). Our notations would typically omit to specify in which round of encryption these bits are taken, as most of our work is about constructing **one round** invariants (which however do extend to an arbitrarily large numbers of rounds). 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 $S1, S2$ or K, L and unknown to the attacker. These bits will be different in different rounds. In order to construct an attack we start from any given cipher specs in forms of ANFs for one round, and we attempt to generate some complex polynomial invariant property. This framework covers most block ciphers ever made except that some ciphers would have more "secret" bits in one round.

2.1 Our Specific Cipher

In this paper we are going to work with one specific block cipher T-310. We do not provide a full description of how T-310 is initialized and used. We just concentrate on how one block cipher round operates (the ANFs). Below we show the internal structure of T-310, one of the most important block ciphers of the Cold War, massively used to encrypt all sorts of state communications, cf. [19]. T-310 is one of the most "paranoid" cipher designs we have ever seen. The cipher is iterated hundreds of times per one bit actually encrypted. The hardware complexity of T-310 is hundreds of times bigger than AES or 3DES, cf. [10]. Does it make this cipher very secure? Not quite, if we can construct algebraic invariants which work for any number of rounds.

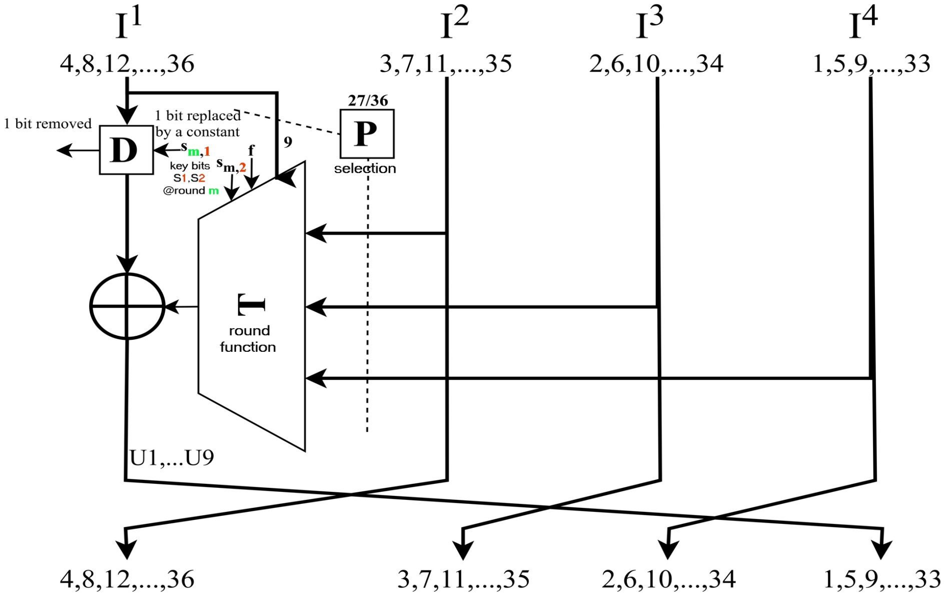

Fig. 1. T-310: a peculiar sort of Compressing Unbalanced Feistel scheme cf. [10].

The block size is 36 bits and the key has 240 bits. We are going to explore the space of invariants on 36 bits and study Boolean functions on 6 bits, some of which may lead to specific invariant attacks. There are $2^{2^6} = 2^{64}$ Boolean functions on 6 bits and an incredibly large number $2^{2^{36}}$ of possible invariants.

The cipher operates on 36-bit blocks and the state bits are numbered 1-36. The bit numbering in this compressing unbalanced Feistel cipher with 4 branches is such, cf. Fig 1, that bits $1, 5, 9 \ldots 33$ are those freshly created by this round, while ALL the input bits the numbers of which are NOT multiples of 4 are shifted by 1 position, i.e. bit 1 becomes 2 in the next round, 35 becomes 36, etc.

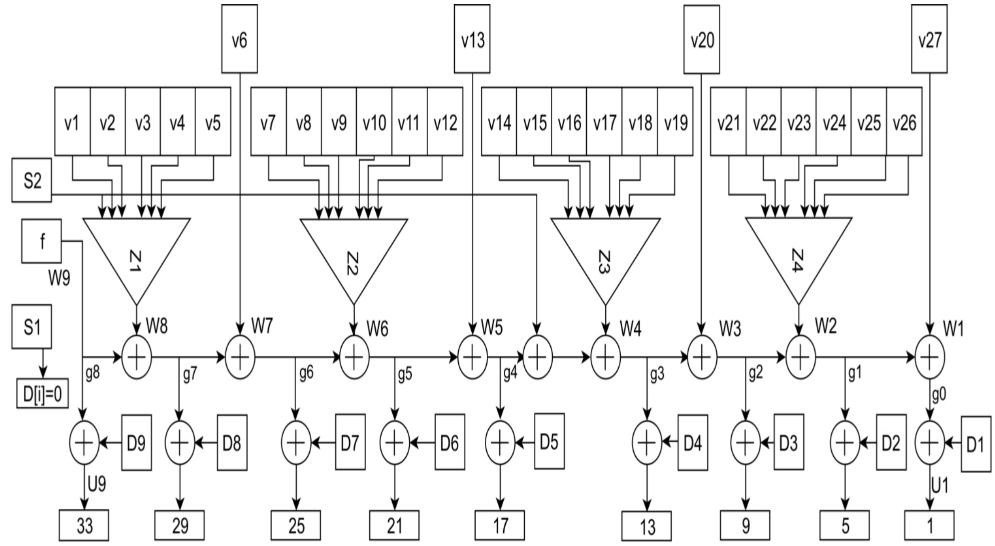

Fig. 2. The internal structure of one round of T-310 block cipher.

Here F is a public bit derived from an IV transmitted in the cleartext, S1 and S2 are bits of the secret key on 240 bits. S1 and S2 are repeated every 120 steps.

Few things remain unspecified in our picture: which bits and in which order are connected to D1-D9 and v1-v27. In T-310 this specification is called an LZS or *Langzeitschlüssel* which means a long-term key and which is distinct than the short-term key on 240 bits. We simply need to specify two functions[1] $D : \{1 \ldots 9\} \rightarrow \{0 \ldots 36\}$, $P : \{1 \ldots 27\} \rightarrow \{1 \ldots 36\}$. For example $D(5) = 36$ will mean that input bit 36 is connected to the wire D5 on our picture, and $P(1) = 25$ will mean that input 25 is connected as v1 or the 1st input of $Z1$. Finally the internal wiring LZS uses a special convention where the bit S1 is used instead of one of the Di by specifying that $D(i) = 0$. Overall one round can be described as 36 Boolean polynomials out of which only 9 are non-trivial.

A key step is rewriting the equations above using exact values for $D(i)$ and $P(j)$. Let $x_1, \ldots, x_{36}$ be the inputs and let $y_1, \ldots, y_{36}$ be the outputs. In this notation both variables x_i and y_i are treated "alike" and denoted by lowercase letters a-z backwards starting from x_{36} till x_{11} and y_{36} till y_{11}. Then we use capital letters M-V (avoiding some letters used elsewhere). For example $a = x_{36}$ and $t = x_{17}$, and $M = x_{10}$ and $V = x_1$.

[1] Which are both assumed to be injective and $D(i)$ are always multiples of 4, this in order to avoid many degenerate cases and trivial attacks cf. [10].

Numbers	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Letters	V	U	T	S	R	Q	P	O	N	M	z	y	x	w	v	u	t	s	r	q	p	o	n	m	l	k	j	i	h	g	f	e	d	c	b	a

Fig. 3. Variable naming conventions

In the general case, one round of encryption is defined as follows:

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

Fig. 4. The specification of one round of T-310

Variable Renaming. When we manipulate concrete connections for some concrete cipher wiring (LZS) and in the study of all our later polynomial invariants $\mathcal{P}$ we can rewrite the above equations using our short one-letter notations, for example in one case we get exactly:

$$
\begin{aligned}
& a \leftarrow b \text{ then } b \leftarrow c \text{ then } c \leftarrow d \\
& d \leftarrow F + i \\
& e \leftarrow f \text{ then } f \leftarrow g \text{ then } g \leftarrow h \\
& h \leftarrow F + Z + e \text{ where } Z \text{ has 6 inputs } Z \leftarrow Z(L, j, h, f, p, d)) \\
& \qquad\qquad [\ldots] \\
& \qquad W \leftarrow Z(a, g, c, z, U, i) \\
& V \leftarrow F + Z + r + Y + m + L + X + W + x + w
\end{aligned}
$$

These expressions should be viewed as a set of 36 substitutions where each variable is replaced by a polynomial algebraic expression. The variables on the left hand side will be output variables after 1 round, and on the right hand side, we have ANF or polynomials in the input variables. In order to have shorter expressions to manipulate we replaced here $Z1 - Z4$ by shorter abbreviations Z, Y, X, W respectively. We also replaced S2 by a single letter L (used at 2 places). The other key bits $S1$ will only be used at one place if some $D(i) = 0$.

3 The Fundamental Equation

We want to **find** a polynomial expression $\mathcal{P}$ using any number between 1 and 36 variables such that it is an invariant after the substitutions in Section 2.1 above. For example if the polynomial $\mathcal{P}$ is fixed, and also in other cases, the attacker will write ONE SINGLE (or more) algebraic equation which he is going to solve to determine the unknown Boolean function Z, if a solution exists.

Definition 3.1 (Compact Uni/Quadri-variate FE). Our "Fundamental Equation (FE)" to solve is to make sure that sum of two polynomials like:

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

reduces to 0, or more precisely we are aiming at FE = 0 for any input, or in other words we want to achieve a formal equality of two Boolean polynomials like

$$\mathcal{P}(\text{Inputs}) + \mathcal{P}(\text{Transformed Outputs}) = 0$$

or even 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 $Z1-Z4$ will be later replaced by Boolean functions $Z(), Y(), X(), W()$.

Alternative Notation. There is also another notation which is more like notations used in classical invariant theory. Instead of writing

$$\text{FE} = \mathcal{P}(\text{Inputs}) + \mathcal{P}(\text{Transformed Outputs})$$

we can also write:

$$FE = \mathcal{P} + \mathcal{P}^{\phi}$$

where

$$\mathcal{P}^{\phi} \overset{def}{=} \mathcal{P}(\text{Inputs}^{\phi}) = \mathcal{P}(\text{Transformed Outputs})$$

which is the same as above, and we can also write:

$$\mathcal{P}^{i} \overset{def}{=} \mathcal{P}$$

$$\mathcal{P}^{o} \overset{def}{=} \mathcal{P}(inputs^{\phi})$$

where ϕ is the transformation induced by 1 round of encryption and where ϕ(Inputs) denotes a sequence of 36 polynomial expressions of output-side variables $(a, \ldots V)$ expressed as Boolean function of the 36 input-side variables [with some extra variables such as secret key variables]. For example the variable a is replaced by polynomial b and d by $F + i$. In other words they are written as formal polynomials in B_{36} corresponding to the ANF expressions of one round of encryption (and as a function of inputs of this round). Our usage of exponents is similar as in the mathematical (Hilbertian) invariant theory. Our exponents can be simply interpreted as transformations on polynomials, or more precisely as operations belonging to a certain group of transformations acting on a set of Boolean polynomials $\mathcal{P}$ or A or other say $(azM + b) \in B_{36}$ where B_{36} is the precise ring of all Boolean polynomials in 36 variables named $a - z$ and $M - V$ as in this paper. The notation $\mathcal{P}^{\phi}$ is very elegant and unhappily **ambiguous** in general, because in general ϕ depends also on F and various key bits. Then it happens that $\mathcal{P}^{\phi}$ is likely to be unique nevertheless: we are aiming at computing $\mathcal{P}^{\phi}$ primarily and precisely in cases where the result, the transformed and substituted polynomial $\mathcal{P}^{\phi}$ is such that the final result $\mathcal{P}^{\phi}$ does NOT depend on the variables F, K, L (!). This may and will become possible when our polynomial $\mathcal{P}$ is particularly well chosen.

In the next step, Z will be represented 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 Z:

Definition 3.2 (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 $Z1 - Z4$.

Some coefficients of $\mathcal{P}$ may be fixed, other will be variable. In all cases, all we need to do is to solve the equation above for Z, which is 64 binary unknowns for the ANF coefficients, plus some extra variables for $\mathcal{P}$. This formal algebraic approach, if it has a solution, still called Z for simplicity, or $(\mathcal{P}, Z)$ allows to **guarantee** a certain invariant $\mathcal{P}$ holds for 1 round.

A major problem is now **the existence of solutions**. Does this equation FE have a solution? In many interesting cases this equation will be unusually simple and sometimes it vanishes totally, all coefficients are equal to 0, one example of this can be found in Appendix and more in [5]. A previous draft paper on this topic [5] was initially about simple invariants of low degree and many of them has serious issues such as linear attacks also exist. In this paper we are looking for better attacks and how to construct attacks from scratch with strong properties such as elimination of certain variables e.g. F, K, L. Their presence would ruin the attack, it will not longer work for any key and any IV.

4 Milestone Example - Eliminating Round Constants *F*

It is easy to see that every freshly created output in one round of T-310 depends on F. For this reason in early invariant attacks on T-310 eliminating F seemed quite difficult [5]. Being able to do this, is in some sense an interesting generic attack. A useful strategy is to aim at eliminating F completely at an early stage of our construction. Moreover if at all we actually can eliminate the constant F, we can also eliminate a lot more complex things[2] as we will see later.

4.1 A Construction of a Basic Multiple Invariant without *F*

We show how an invariant can be constructed in an ordered and systematic way. We assume that $D(9) = 32$ and $D(8) = 36$ which implies $d \leftarrow F + e$ and $h \leftarrow F + Z1 + a$ and which mandates a sort of imperfect cycle of length 8 on 8 bits:

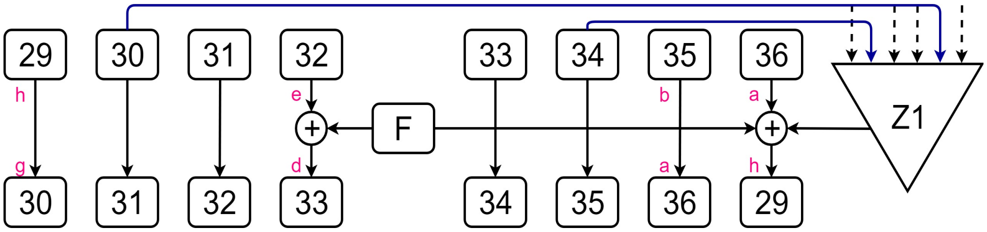

Fig. 5. Consequences of assuming that $D(9) = 32$ and $D(8) = 36$. F is used twice and will be eventually eliminated. The intention is that the invariant will not depend on F and neither on 4 additional bits which enter Z1.

[2] F is a constant known to the attacker. However **if** F could be totally eliminated for the purpose of finding a non-linear invariant operating on X bits, other more complex variables which depend on many key bits and on what happens in other parts of the cipher, also CAN be eliminated, cf. Section 4.2 below.

We will decide a bit later (after writing the FE) where different inputs of $Z1$ need to be connected. We get a series of obvious transitions such as ce becomes df which we would get in traditional Bi-Linear Cryptanalysis (BLC) [8], plus a series of less obvious transitions due to the 2 assumptions $D(9) = 32$ and $D(8) = 36$ such as bd becomes ce, hoping that the term Fc can be somewhat cancelled later.

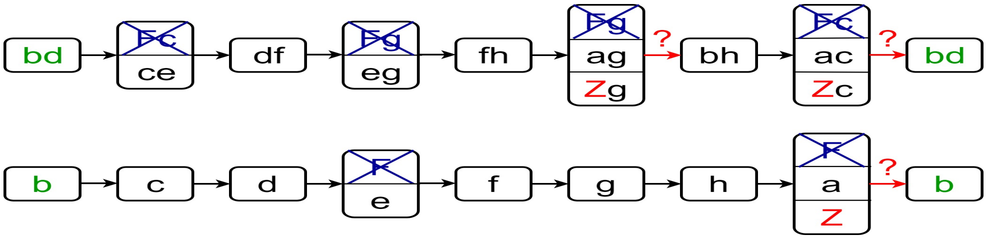

Fig. 6. A detailed explanation for our invariant which shows terms which cancel. This analysis is done under our initial ad-hoc assumption that $D(9) = 32$ and $D(8) = 36$.

This analysis of cycles on degree 2 monomials suggests to use the following irreducible polynomial defined as the union of terms in black in both cycles:

$$\mathcal{P} = a + b + c + ac + d + bd + e + ce + f + df + g + ag + eg + h + bh + fh$$

Knowing that each term in blue in Fig. 6 appears an even number of times, we have **already eliminated** all terms with F. Now we add all the parts with Z on both cycles, we expect that our FE will be: $Z + Zc + Zg \equiv 0$. This can only work with $Z \not\equiv 0$ if bits c and g are connected as inputs of $Z1$. This is the moment at which we need to decide which bits will be connected to become inputs of $Z1$, cf. earlier Fig. 5 and this can be done in any order, not necessarily following our figure. For example we can have $P(1) = 34$ where c corresponds to x_{34} and $P(4) = 30$ where g corresponds to x_{30}. We are now able to generate a long term key for which our invariant is going to work, for example:

```
827: P=34,32,25,30,19,28,18,35,31,33,23,36,24,22,5,1,
13,17,16,10,21,6,20,29,9,15,3 D=21,17,29,24,27,20,31,36,32
```

For this LZS we can now re-compute our FE which will have fewer unknowns:

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

the fundamental equation is then as expected:

$$Z = Z(c + g)$$

and because FE does not depend on either F or L, we do not need eight copies of it but just one. Here is one solution:

$$Z = e + be + ce + bce + bf + bcf + bef + bcef$$

This completes a construction of a non-linear round invariant. We have checked that there is no linear invariant in any of the eight cases depending on $F, S1, S2$, and therefore Linear Cryptanalysis (LC) does not work here. Our **non-linear invariant** $\mathcal{P}$ works in all eight cases and therefore it propagates **for an arbitrary number of rounds** for any key and for any IV.

We have obtained an invariant $\mathcal{P}$ on 8 bits 29-36 the key feature of which is that it completely eliminates 4 bits which come from other parts of the cipher. In fact the invariant obtained above can be constructed systematically.

Theorem 4.3 (Pre-conditions for Key 827). *The invariant* $\mathcal{P} = a+b+c+ac+d+bd+e+ce+f+df+g+ag+eg+h+bh+fh$ *will occur for* $L = 0$ *or* $L = 1$*, or for any* L*, each time the following set of conditions are satisfied:*

$$\begin{cases} D(9) = 32 \\ D(8) = 36 \\ (1+c+g)Z(L, P[1-5]) \equiv 0 \\ Z \not\equiv 0 \end{cases}$$

Proof: The constraints $D(9) = 32$ and $D(8) = 36$ already mandate a cycle between numbers 29-36 shown in Fig. 5 and they mandate all the transitions of 6 which do not have a red question mark (?) sign, which depend on the FE. Finally, we check that all terms in F are eliminated.

4.2 A Transposed Version

Until now we have constructed an invariant for $Z1$ which eliminates F which is known to the attacker (and it also ignores 4 more bits entering $Z1$). Now IF we can eliminate F, we can do a lot better. We can simply **transpose** our invariant from Z1 to Z4 and here it will eliminate g2, cf. Fig. 2 plus another 4 bits which depend on at least 17 bits and 2 key bits in EXACTLY the same way. Instead of eliminating a constant F known to the attacker we are now eliminating $g2$ which is a lot more complex to know, actually it depends on almost everything else (and the attacker could not possibly know or determine $g2$).

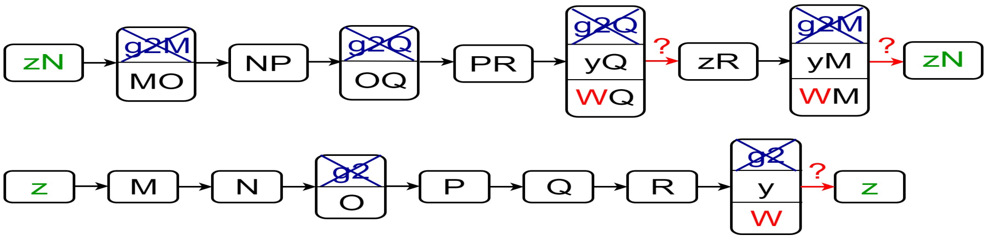

Fig. 7. The same invariant transposed to $Z4$.

By doing so we get a stronger invariant. For example we found the following LZS:

```
847: P=32,22,26,14,21,36,30,17,15,29,27,13,4,23,1,8,35,20,
5,16,24,9,10,6,7,28,12 D=24,12,8,16,36,4,20,28,32
```

Here everything is transposed: the FE is $WM + WQ + W = 0$ and is the same in all 8 cases for any F and any K, L and the transposed invariant is now

$$yM + zN + MO + NP + yQ + OQ + zR + PR + y + z + M + N + O + P + Q + R$$

This works for a variety of Boolean functions, and we cannot directly transpose the previous solution because the two key variables c, g are now M, Q at different positions. A correctly transposed solution is then for example $Z = d + cd + bd + bcd + cf + bcf + cdf + bcdf$. A crucial point here is that a very complex part of the cipher enters this component at $g2$ and yet it could be totally eliminated. Furthermore LZS 847 is a permutation on 36 bits secure against LC and all previously known attacks on T-310 [10].

5 Eliminating a Lot More - Construction of Better Simultaneous Invariants

We will now work through intersection of spaces of polynomials in order to see what invariants are possible and discover some yet better invariants. Previously we have eliminated F plus 4 more bits and then transposed this result to eliminate a quantity with even more complex dependencies. Here we go one step further.

We are going to show the existence of an invariant on $Z1$ and $Z4$ which mixes bits which sit at two opposite ends[3] of the cipher, cf. Fig 2. These parts are connected through $Z2$ and $Z3$ by a quantity called $g27$ and defined as $g27 = g2 + g7$ cf. Fig. 2 which depends on an excessively large number of round input bits (at least 19) plus the key bit $S2$. Without $g27$ none of the outputs on the right hand side we use can be computed. Yet this connection $g27$ gets totally eliminated (so does F and few other things). A lof of complexity is simply eliminated totally.

We focus on 8+8 bits 29-36 and 5-12 pertaining to $Z1$ and $Z4$ only and strictly avoiding anything between g2 and g7. We start by assuming the following four constraints which implements a basic sort of "exchange" connection between two opposite ends of the cipher.

$$\begin{cases} D(2) = 4 \cdot 9 \\ D(3) = 4 \cdot 8 \\ D(8) = 4 \cdot 2 \\ D(9) = 4 \cdot 3 \end{cases}$$

We can again generate cycles in the same way as in Section 4.1: transitions are either natural e.g. $bc \leftarrow cd$ or consequences of the 4 conditions on $D()$ above. Similarly we ignore the boxes with blue crosses which we hope might eventually be eliminated later inside the final FE which is not yet finalized. A detailed analysis of these natural cycles as shown in Fig. 8 (and few more) leads to 8 natural clusters of monomials for $\mathcal{P}$ which most likely work together:

5.1 Analysis of Polynomial Spaces

This gives 8 natural pre-FE equations, with the idea that a final $\mathcal{P}$ and final FE is a fixed linear combination of these 8, which are exactly:

[3] We call it "spooky interaction" the two distant (as remote as only possible) parts of the cipher "talk" to each other in terms of a polynomial invariant which mixes variables from both sides. However their principal connection a.k.a. $g27$ is eliminated.

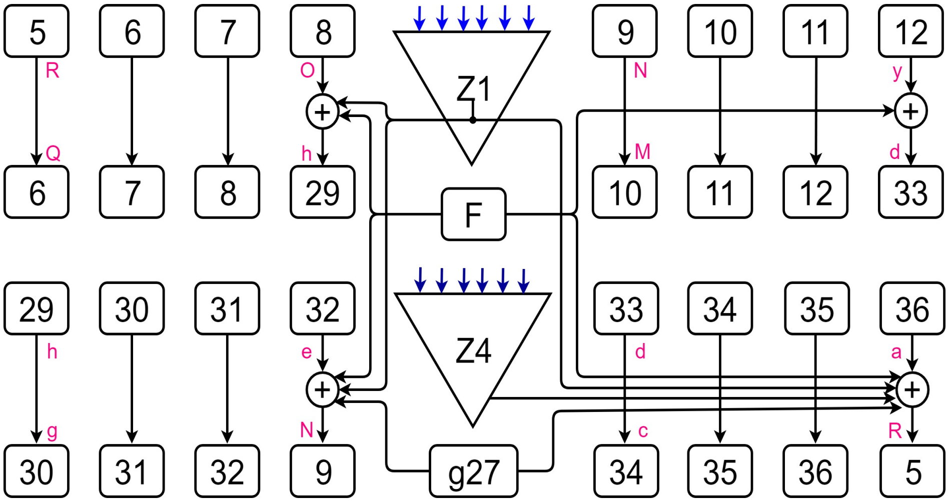

Fig. 8. Our ad-hoc assumption is $D(2) = 36$, $D(3) = 32$, $D(8) = 8$, $D(9) = 12$. Here $g27 \overset{def}{=} g2 + g7$ cf. Fig. 2. We aim at invariants on 8+8 bits which would not depend on $F, g27$ and few more inputs of $Z1, Z4$ in blue are not yet connected. Other inputs of $Z1, Z4$ will be used and their connections are decided at a later stage.

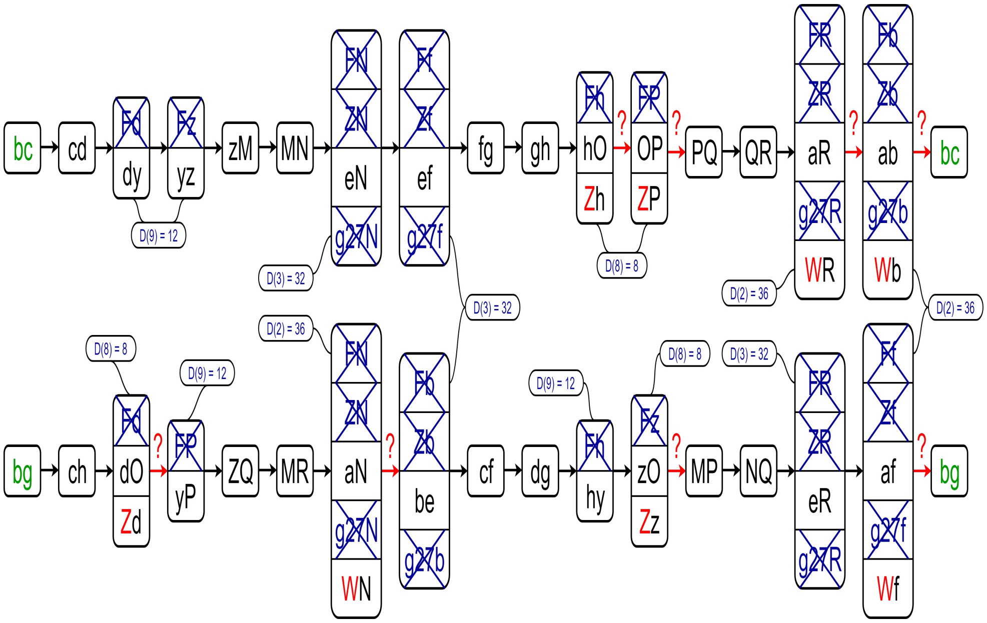

Fig. 9. A detailed analysis of transitions we aim at using in our invariant. The boxes with crosses are terms we hope to cancel later. Transitions in red with ? depend on Z and will eventually work only if our final FE equation has a solution.

```
Wb+Fd+(b+f+N+R)(F+Z+L+Y+X+P6+P13+P20)+h(F+Z)+Fz+ P(F+Z)+WR
(L+Y+X+P6+P13+P20)(c+g+M+Q)+c(Z+W)+ZM+WQ
Fb+FN+W(d+P)+ (F+Z)(f+R)+ (F+Z+L+Y+X+P6+P13+P20)(d+h+z+P)
F(Z+W+a+e+y+O)+We+Zy+(Z+L+Y+X+P6+P13+P20)(W+a+e+1)
Z(d+z)+W(f+N)+F(d+h+z+P)+(F+Z+L+Y+X+P6+P13+P20)(b+f+N+R)
W(g+M)+Z(g+Q)+(L+Y+X+P6+P13+P20)(c+g+M+Q)
Z(b+N)+W(h+z)+F(b+f+N+R)+(F+Z+L+Y+X+P6+P13+P20)(d+h+z+P)
ZW+Z(a+e)+W(y+O)
```

We start with a polynomial space of dimension 8. Now we observe that we need to eliminate Fb and other similar monomials. A standard row-echelon procedure forces us to XOR some equations and we obtain the following 6 linearly independent equations:

$$\begin{cases} XR+WR+(R+h+z+P+N+f+b+d)(P6+P13+P20)+Zb+Lb+ \\ \quad Xb+Lf+Yf+Xf+Lh+Yh+Xh+Zz+Lz+Yz+Xz+LP+YP+XP+ \\ \quad WP+Yb+YN+XN+LR+YR+Zd+Ld+Yd+Xd+Wd+ZN+Wb+LN \\ Zc+Lc+Yc+Xc+Wc+(c+g+M+Q)(P6+P13+P20)+ \\ \quad Lg+Yg+Xg+ZM+LM+YM+XM+LQ+YQ+XQ+WQ \\ WR+Wf+WN+Zh+Zz+ZP+Zd+Wb \\ Lc+Yc+Xc+(c+g+M+Q)(P6+P13+P20)+Lg+Yg+Xg+ \\ \quad +LM+YM+XM+LQ+YQ+XQ+Zg+Wg+WM+ZQ \\ Zb+Zf+WP+ZR+Wd+Wh+Wz+ZN \\ ZW+Za+Ze+Wy+WO \end{cases}$$

This stage is crucial for cipher designers, possibly one should be able to construct a block cipher in such a way, that the dimension of this set of polynomials is already 0 [here it is 6 so we can do cryptanalysis]. Then we are going to eliminate all products of $P6$ in the same way, which also leads to elimination of numerous other monomials and this leads to a dimension 3, still not 0:

$$\begin{cases} Zc+Wc+ZM+WQ+Zg+Wg+WM+ZQ \\ WR+Wf+WN+Zh+Zz+ZP+Zd+Wb \\ ZW+Za+Ze+Wy+WO \end{cases}$$

Finally it is possible to show that out of $2^3-1=7$ possible linear combinations of these $\mathcal{P}$, only the first 2 out of 7 lead to solutions, this under the condition that the cipher wiring $P()$ is injective, or we get FE equations for which the set of solutions is empty due to the simple fact that a Boolean function cannot annihilate variables which are not inputs of this function. This leads to only 2 possibilities, out of which we have chosen to work with one:

$$\mathcal{P}=bc+cd+dy+yz+zM+MN+eN+ef+fg+gh+hO+OP+PQ+QR+aR+ab+bg+ch+$$
$$dO+yP+zQ+MR+aN+be+cf+dg+hy+zO+MP+NQ+eR+af.$$

This $\mathcal{P}$ is irreducible and the FE is obtained by repeating the very same linear transformations by which we have reduced the dimension from 8 to 2 above:

$$NW+PZ+RW+Wb+Wf+Zd+Zh+Zz$$

All we have now to do is to ensure through $P()$ that various inputs which appear in the FE above are connected to $Z1=Z$ or $Z4=W$ respectively, for example f must by an input of W, therefore we need $P(i)=31$ for some $i\in\{21,\ldots,26\}$. This is exactly the moment at which we can decide all the connections in blue in Fig. 8. Here is an example of a LZS where this $\mathcal{P}$ works:

```
714: P=11,7,30,29,33,1,20,17,2,15,14,27,36,24,18,8,19,
23,28,32,4,16,31,9,35,5,13 D=16,36,32,24,4,28,20,8,12
```

This example was found by the exact steps we enumerate below and by feeding the resulting set of constraints on $D()$ and $P()$ to a SAT solver at the end. This is done many times until we find a valid permutation on 36 bits.

Solving the FE. It remains to find a solution to $NW + PZ + RW + Wb + Wf + Zd + Zh + Zz \equiv 0$. There is still some degree of freedom here in selecting which bits will be inputs of W and Z though function $P()$. With LZS 714 above, one possible solution is

$$Z = 1 + dc + cb + fb + b + c + de + df + db + e + f + d + eb.$$

We get another invariant which works for any number of rounds and any key.

6 From Invariants to Ciphertext-Only Attacks on T-310

We are able to construct a certain polynomial invariant property for our block cipher. Does it allow one to decrypt communications encrypted with T-310? A key observation cf. is that typically our polynomial invariants $\mathcal{P}$ will lead to partitioning the space of say 2^{16} elements into two sets of rather **unequal** sizes. We are able to produce **a strong pervasive bias**. For example we consider the invariant from Section 5.1 with 16 variables. A quick computer simulation checking all 2^{16} possible cases shows that we have the partition in two sets with 36864 and 28672 elements. Then even though any individual variable say a or N is typically not biased, neither are pairs of variables, we observed that in each case there exists a relatively small N such that for ANY subset of N out of 16, the joint probability distribution of these N variables is not uniform. In fact $N = 5$ seems to suffice. For example when $\mathcal{P} = 0$ we observed that the event $abcde = 1$ happens 1280 times and the event $abcd(e+1) = 1$ happens 1024 times. Moreover the event $abcdef = 1$ never happens.

In addition we have also checked that the polynomial $\mathcal{P}$ is irreducible and that there are no linear invariants true with probability 1. We found a non-trivial higher-order correlation attack where the bias does **not** depend on the number of rounds.

6.1 Decrypting T-310 Communcations

Now it is sufficient that some Boolean function say Z3 in the next round takes some subset of 5 bits out of 16 as an input. Then Z3 will take 1 more bits (presumably not one of the 16) and with a high probability, the output of this Boolean function will be biased at **every** encryption round. This will work if sufficiently many of the 16 bits are used with $Z3$ or with $Z2$. Such permanently biased bits are expected to lead to correlation attacks where $S1$ and the special output bit x_α actually used in the encryption [10] will be connected through just a few biased bits. This, given the fact that the same key bit $S1$ is repeated every 120 rounds, will inevitably lead to a ciphertext-only correlation attack on T-310 and on a key recovery attack on 120 bits of $S1$ key (recovering $S2$ could be harder). Knowing that bits in any natural language (not only German) in any reasonable encoding are **always** strongly biased, cf. [11], from any such a bias we can infer concrete values of individual key bits $S1$ by majority voting. A full description and study of the best possible attack of this type requires substantial amount of extra work and is beyond the scope of this paper.

Important Remark - Avoiding Trivial Cases. NOT every partitioning in two spaces of unequal sizes will work here. There exist some trivial cases entirely due to Linear Cryptanalysis. Some such degenerated examples are given in Appendix A of [5] for example where $\mathcal{P} = 0$ in 3/4 of cases.

Observations. We obtain an attack of the sort which is extremely rare in cryptanalysis: a correlation attack with recovery for at least 120 bits of the key, where the correlation does **NOT** degrade as the number of encryption rounds increases. It is also unique in another way: our biases are higher order biases on joint probability distributions of a certain dimension N which originate from a non-linear invariant. They appear essentially ex-nihilo: they are NOT biases which could be produced or understood as combinations of simpler biases or within the strict framework of Linear Cryptanalysis.

7 Conclusion

In this paper we study cryptanalysis with non-linear polynomial invariants. We show how a specific structure and internal wiring of more or less any block cipher, starting from round ANFs, can be translated into a relatively simple "Fundamental Equation" (FE), which can be used to study which specific non-linear invariants may exist (or not) for this cipher. In current research in Partitioning Cryptanalysis (PC) [15] there are some impossibility results [3, 21, 2] but extremely few possibility results [9]. Partitioning properties are extremely hard to find. Polynomial invariants are way more intelligible. Our main contribution is to show that the attacker does not need to randomly search for an interesting invariant $\mathcal{P}$ and a vulnerable non-linear component Z. Specific polynomial invariants $\mathcal{P}$ and weak Boolean functions which work together can now be **determined** – by solving our FE equation(s). Our approach is constructive, completely general and can be applied to almost any block cipher: we first write the FE then based on ad-hoc heuristics we determine a space of polynomials with a reduced dimension for $\mathcal{P}$, we substitute variables inside the FE(s), and we attempt to solve our FE(s). We have constructed numerous concrete examples of non-trivial non-linear invariants which propagate for any number of rounds, and for any key and IV. We anticipate that the success rate of this approach will be very different for different families of ciphers. If just one round function is very complex and uses many key bits, with too many constraints to satisfy simultaneously, our approach is likely to fail. Or solving FE will become computationally difficult.

7.1 Applications: Biased Partitioning and Decryption Attacks

In many $\mathcal{P}$ creates a partition into two sets with similar yet **unequal** sizes. Our invariants introduce a **permanent and pervasive bias** inside the cipher which is not degraded with iteration of the cipher. This is expected to lead to ciphertext-only higher-order correlation attacks with key recovery such that their complexity does NOT depend on the number of rounds, cf. Section 6.

7.2 Beyond Simple and Vulnerable Boolean Functions

In our proof of concept examples the Boolean function is very special and has a lower degree than expected. This is due to the fact that we imposed some specific constraints and our examples have been chosen for elegance and simplicity. In general, when the degree of $\mathcal{P}$ increases we expect to find many more/stronger invariants $\mathcal{P}$ which work for a larger proportion of the space of all Boolean functions on 6 variables. A first attack and proof of concept that when the degree of $\mathcal{P}$ increases one can attack indeed more or less arbitrarily strong Boolean functions can be found in [7].

References

1. Arnaud Bannier, Nicolas Bodin, and Eric Filiol: *Partition-Based Trapdoor Ciphers,* `https://ia.cr/2016/493`

2. Marco Calderini: *A note on some algebraic trapdoors for block ciphers,* last revised 17 May 2018, `https://arxiv.org/abs/1705.08151`
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. 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`, received 1 Sep 2018, last revised 27 Mar 2019.
6. Nicolas T. Courtois, Marios Georgiou: *Constructive Non-Linear Polynomial Cryptanalysis of a Historical Block Cipher,* At `http://arxiv.org/abs/1902.02748`.
7. Nicolas T. Courtois: *Structural Nonlinear Invariant Attacks on T-310: Attacking Arbitrary Boolean Functions,* `https://ia.cr/2018/1242`, revised 12 Sep 2019.
8. Nicolas Courtois: *Feistel Schemes and Bi-Linear Cryptanalysis,* in Crypto 2004, LNCS 3152, pp. 23–40, Springer, 2004.
9. 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. `https://www.researchgate.net/publication/221005723/_The_Inverse_S-Box_Non-linear_Polynomial_Relations_and_Cryptanalysis_of_Block_Ciphers`
10. 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://ia.cr/2017/440.pdf`
11. 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`.
12. Nicolas Courtois, Maria-Bristena Oprisanu and Klaus Schmeh: *Linear cryptanalysis and block cipher design in East Germany in the 1970s,* in Cryptologia, 05 Dec 2018, `https://www.tandfonline.com/doi/abs/10.1080/01611194.2018.1483981`
13. Tony Crilly: *The rise of Cayley's invariant theory (1841-1862),* In Historia Mathematica, Vol. 13, Iss. 3, August 1986, pp. 241–254
14. 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.
15. C. Harpes, J. L. Massey: *Partitioning cryptanalysis,* In FSE 97, LNCS 1267, pp. 13–27, 1997.
16. Thomas Jakobsen, Carlo Harpes: *Non-Uniformity Measures for Generalized Linear Cryptanalysis and Partitioning Cryptanalysis,* in Pragocrypt'96, 1996.
17. Lars R. Knudsen, Matthew J. B. Robshaw: *Non-Linear Characteristics in Linear Cryptoanalysis,* Eurocrypt'96, LNCS 1070, Springer, pp. 224–236, 1996.
18. 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.*
19. *Klaus Schmeh: The East German Encryption Machine T-310 and the Algorithm It Used,* In Cryptologia, vol. 30, iss. 3, pp. 251–257, 2006.
20. 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.

21. Yongzhuang Wei, Tao Ye, Wenling Wu, Enes Pasalic: *Generalized Nonlinear Invariant Attack and a New Design Criterion for Round Constants,* https://pdfs.semanticscholar.org/bd9f/fb5ae863ee15ce07c71eab8fb84ee9e810d1.pdf In IACR Tr. on Symm. Crypt. Vol. 2018, No. 4, pp. 62-79. `https://tosc.iacr.org/index.php/ToSC/article/view/7361/6531`

A A Degenerate Invariant with FE Reduced to Zero

Not all non-linear invariants are good. We consider the following degenerate case:

```
881: P=4,20,33,8,1,28,5,19,9,32,11,17,24,13,21,18,15,
25,12,16,35,22,23,29,36,30,34 D=0,36,4,8,12,20,24,28,32
```

Here the invariant $\mathcal{P}$ is a homogenous polynomial of degree 2 with $169 = 13^2$ terms and 26 variables which is highly symmetric and not at all irreducible:

$$(n+b+p+r+t+v+x+z+N+P+R+T+V)(a+m+o+q+s+u+w+y+M+O+Q+S+U)$$

Furthermore it is possible to verify that if we call $\mathcal{A}$ the first sum, and if we call $\mathcal{B}$ the second sum, $\mathcal{AB}$ is a non-linear invariant and also $\mathcal{A} + \mathcal{B}$ is a linear invariant for 1 round and the exact same cipher setup LZS 881. The linear invariant produces a partition with two sets of equal sizes. For the non-linear invariant the sizes are not equal and the probability that $\mathcal{P} = \mathcal{AB} = 0$ is exactly 3/4. Moreover, it is easy to see that this invariant works also for the original Boolean function (actually the FE is reduced to 0 here) and also for any other Z. This example remains however very special. Both $\mathcal{A}$ and $\mathcal{B}$ one at a time are 2-round invariants with $\mathcal{A} \mapsto \mathcal{B}$ after 1 round. Then after one round $\mathcal{A} \cdot \mathcal{B}$ becomes $\mathcal{B} \cdot \mathcal{A}$ which is the same, hence a quadratic invariant for 1 round exists. This is not a very good invariant: the 3/4 bias obtained here is something which we would already be constructed by combining the two linear approximations.