On the Existence of Non-Linear Invariants and Algebraic Polynomial Constructive Approach to Backdoors in Block Ciphers

Nicolas T. Courtois

University College London, Gower Street, London, UK

Abstract. In this paper we study cryptanalysis with non-linear polynomials cf. Eurocrypt'95 (adapted to Feistel ciphers at Crypto 2004). Previously researchers had serious difficulties in making such attacks work. Even though this is less general than a general space partitioning attack (FSE'97), a polynomial algebraic approach has enormous advantages. Properties are more intelligible and algebraic computational methods can be applied in order to discover or construct the suitable properties. In this paper we show **how** round invariants can be found for more or less any block cipher, by solving a certain surprisingly simple single algebraic equation (or two). Then if our equation has solutions, which is far from being obvious, it will guarantee that some polynomial invariant will work for an arbitrarily large number of encryption rounds. This paper is a proof of concept showing that it IS possible, for at least one specific quite complex real-life cipher to **construct in a systematic way**, a non-linear component and a variety of non-linear polynomial invariants holding with probability 1 for any number of rounds and any key/IV. Thus we are able to weaken a block cipher in a permanent and pervasive way. An example of a layered attack with two stages is also shown. Moreover we show that sometimes our equation reduces to zero, and this leads to yet stronger invariants, which work for **any** Boolean function including the original historical one used in 1970-1990.

Key Words: block ciphers, Boolean functions, Algebraic Normal Form, unbalanced Feistel ciphers, weak key attacks, backdoors, history of cryptography, T-310, Linear Cryptanalysis, Generalized Linear Cryptanalysis, partitioning cryptanalysis, polynomial invariants, triangular systems of equations, I/O sums, multivariate polynomials, symmetric polynomials, algebraic cryptanalysis.

1 Introduction

The concept of cryptanalysis with non-linear polynomials sometimes called Generalized Linear Cryptanalysis (GLC) was introduced by Harpes, Kramer, and Massey at Eurocrypt'95, cf [24]. It can also be described in terms of I/O sums or I/O relations (algebraic equations relating Inputs and Output variables), cf. [12]. Constructing such properties is in general a difficult combinatorial problem, and many researchers have in the past failed to find any such properties, cf. for example Knudsen and Robshaw at Eurocrypt'96 cf. [28] and there are extremely very few positive results on this topic. A later paper Crypto 2004 provides a

precious insight into this problem: the idea is that non-linear polynomial I/O sums (or I/O relations) can be eventually be constructed if we consider that the choice of polynomials in such relations will depend strongly on the internal wiring (connections) of the cipher. Concepts of Bi-Linear and Multi-Linear cryptanalysis were subsequently introduced [9, 10, 12] in order to work with Feistel ciphers with two and several branches specifically. In this paper we revisit the question of non-linear cryptanalysis and give it a fresh start. Can we construct a more substantial variety of round invariant properties, with a variety of polynomials of degree 2,3,4 etc.., working in a more natural real-life block cipher setting? This rather than building some very peculiar new ciphers [10, 11, 8, 3]?

1.1 Combinatorial or Algebraic

A major problem in cryptanalysis is discovery of invariant of semi-invariant properties of complex type. Some heuristics assumptions or some simplifications are needed as the space of solutions is very large and systematic exploration is not quite possible. There are two major approaches to our problem: combinatorial and algebraic. A combinatorial approach would be to try to modify the connections of the cipher so that to obtain a desired property, where the S-boxes or the Boolean functions would be given (for example modify the internal permutation inside DES). This leads to combinatorial problems which could be studied with formal coding and SAT solvers. An algebraic approach would be to consider the structure of the cipher fixed, even if very complex, and try to determine the solutions by solving a system of algebraic equations. Also a well known very general combinatorial approach considers arbitrary subsets of linear vectors spaces, and it is called partitioning cryptanalysis, cf. [3, 25, 26]. A more algebraic approach is to consider only specific forms of partitions, mainly those defined by the value (0 or 1) of a single Boolean polynomial. This is of course less general, BUT it leads to a more **effective** approach to the problem, effective in the sense that we expect that properties are described, discovered and studied with the tools of algebra, in particular many new very interesting questions can be asked, and we expect properties to be computed or derived rather than to happen by some incredible coincidence. In other terms we expect the algebraic approach to be more illuminating about what actually happens and also easier to study. More importantly In this paper we put forward an **algebraic approach of a new sort**. We show how the problem can be coded with a surprisingly simple single equation of a limited degree which we will call FE. Solving such equation(s) **guarantees** that we obtain a Boolean function and the polynomial invariant P which makes a block cipher weak. When the equation is simpler, the invariant become stronger and hold for a larger space of Boolean functions. It also avoids exploring the vast double-exponential space of possible Boolean functions (this again would be a combinatorial approach). Specific examples will be constructed based on a historical Cold War block cipher which has a highly complex irregular internal structure. With this cipher we will demonstrate that the set of solutions is sometimes not empty, and therefore our attack actually works.

1.2 Related Work

Our approach continues the research on non-linear cryptanalysis of block ciphers [24, 28, 9, 10] with a specific twist: we allow the attacker to manipulate the Boolean function. There have been numerous works on non-linear cryptanalysis and Generalized Linear Cryptanalysis (GLC) [24, 28, 9, 10, 27] and then Bi-Linear and Multi-Linear cryptanalysis [9, 10, 12] for Feistel ciphers. There have also been many constructions of weak ciphers in cryptographic literature, cf. for example work related to the AES S-box [10, 11], very recent work by Filiol *et al.* [3, 6], work on choice of constants in modern hash functions [1, 30]. Almost all research in this areas revolves around the fundamental notion of **partitioning**[1] **cryptanalysis** cf. [3, 25, 26] and Harpes will explain that partitioning cryptanalysis (PC) generalizes both LC and GLC [23, 25, 26]. All these works are closely related and also to the study of the groups generated by various cipher transformations and the question of primitive/imprimitive groups [34, 35, 32, 36, 2, 10]. A **serious theory** is nowadays being developed around what is possible or not to achieve in partitioning attacks, with new important notions such as strongly proper transformations, anti-invariant properties, hidden sums etc, cf. [8, 6]. In this paper we point out that the partitioning approach is simultaneously **too general and too obscure**. It is good for establishing numerous impossibility results cf. [6], but potentially it will obscure any **possibility results**. We can discover some invariant, properties but do we understand their nature? Can we manipulate the properties efficiently and compress them (represent them in a compact way)? Can we discover properties with some effective computational methods and see how various constraints will imply their existence or not? Can add and remove some complex invariants in block ciphers in a modular on-demand approach (keeping only more complex ones)? Yes we can, and our polynomial algebraic approach is what makes all these possible.

1.3 Low Degree Polynomials and the Cipher Structure

The idea to privilege a low degree multi-linear approach and more importantly that one needs to adapt to the high-level structure (wiring) of the block cipher was explicitly suggested in [9]. However none of the previous work has focused on explicitly constructing weak non-linear components in order to obtain invariants holding with probability 1. Also none of the previous works we are aware of, constructs invariant properties by solving a system of algebraic equations. Also in no cipher weakening research we are aware of, we can obtain a whole range of non-trivial solutions with increasing complexity, apparently without a limit.

[1] In fact what we do in this paper is ALSO partitioning cryptanalysis, except that our partitions are characterized by a [single] multivariate polynomial.

2 Notation and Methodology

In this paper we are going to work with one specific block cipher, in order to show what is possible or not. We are not however going to provide a full description of an encryption system and how it is initialized and used. We just concentrate on how one block cipher round operates and how it translates into relatively simple Boolean polynomials and eventually such a cipher could be strong or weak w.r.t. our attack. We will construct non-linear invariant properties which very substantially reduce the space of permutations which can be obtained by iterating our cipher. We will limit the number of variables used here to for example 20 out of 36. Our work is therefore very closely related to the so called partitioning attacks [3, 25, 6]. However we obtain and show the existence of specific polynomial invariants, rather than some obscure partition of a subspace the nature of which could be hard to apprehend.

Quite importantly, we are going to consider, which is very rarely done in symmetric cryptanalysis research, that the Boolean function is an unknown, yet to be determined. We will denote this function 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 for Z.

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. In general in this paper we will use small letters $a - z$, and x_1, x_{36} or e_1 for various binary variables $\in \{0, 1\}$. Capital letters such as $S1, S2, L, F, Z$ will be used to represent some very "special" sorts of variables which are placeholders for something more complex. In particular 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 $e_1 \dots e_6$ 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 other such as L, therefore at the end, our substitution will actually look like:

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

Other capital letters will be used to signify some bits which are also unknown which will be bits of the secret key used in a given round and such bits are in our work called by letters $S1, S2$, where S2 will be sometimes renamed L. We also use capital letters to represent some bits which depend on the IV, or a round-dependent variable constant. This sort of variable which is typically known to the attacker will be denoted by the letter F in this paper. In general we are going the 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 in general that each round of encryption will be 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 L and unknown to the attacker. This framework covers most block ciphers ever made and ever studied in human history.

2.1 Our Specific Cipher

Too many block ciphers which appear in cryptographic literature are never used in any real-life situation. In this paper we are going however to work primarily on a particularly complex Feistel cipher on 4 branches known as T-310, which is how our proof of concept is going to be constructed. This choice implies that in our proof of concept, we are going to search for Boolean functions on 6 bits which lead to specific invariant attacks. There are $2^{2^6} = 2^{64}$ Boolean functions on 6 bits. The block size will be 36 bits. There is potentially 2^{36} possible non-linear invariants for 1 round of a cipher with 36-bit blocks.

2.2 The Challenge

One of the biggest open problems in symmetric cryptography, is to find new classes of invariant properties which hold for specific types of block cipher. For example, could that be possible that for a complex block cipher with no weakness w.r.t. simpler attacks such as Linear Cryptanalysis, we have that the following irreducible polynomial with 144 products the value of which is preserved after an arbitrary number of encryption rounds?

abehij+abcehij+abdehij+abcdehij+abefhij+abcefhij+abdefhij+abcdefhij+abeghij+abceghij+abdeghij+abcdeghij+bcefjk+abcefjk+bcdefjk+
abcdefjk+bcefgjk+abcefgjk+bcdefgjk+abcdefgjk+bcefhjk+abcefhjk+bcdefhjk+abcdefhjk+bcefijk+abcefijk+bcdefijk+abcdefijk+bcefgijk+
abcefgijk+bcdefgijk+abcdefgijk+abehijk+abcehijk+abdehijk+abcdehijk+abefhijk+bcefhijk+abdefhijk+bcdefhijk+abeghijk+abceghijk+
abdeghijk+abcdeghijk+adghil+abdghil+acdghil+abcdghil+adeghil+abdeghil+acdeghil+abcdeghil+adfghil+abdfghil+acdfghil+abcdfghil+
abehijl+abcehijl+abdehijl+abcdehijl+abefhijl+abcefhijl+abdefhijl+abcdefhijl+adghijl+abdghijl+acdghijl+abcdghijl+abeghijl+abceghijl+
adeghijl+acdeghijl+adfghijl+abdfghijl+acdfghijl+abcdfghijl+cdefgkl+acdefgkl+bcdefgkl+abcdefgkl+cdfghkl+acdfghkl+bcdfghkl+abcdfghkl+
cdefgikl+acdefgikl+bcdefgikl+abcdefgikl+adghikl+abdghikl+acdghikl+abcdghikl+adeghikl+abdeghikl+acdeghikl+abcdeghikl+adfghikl+
abdfghikl+cdfghikl+bcdfghikl+bcefjkl+abcefjkl+bcdefjkl+abcdefjkl+bcefgjkl+abcefgjkl+cdefgjkl+acdefgjkl+bcefhjkl+abcefhjkl+bcdefhjkl+abc

In this paper we make this happen, on demand, given an arbitrary specification of the internal wiring of the cipher. Our method is constructive-algebraic and works potentially for all known block ciphers and for Boolean functions and S-boxes of arbitrary size.

2.3 Polynomial Invariants

In general our methodology applies to any block cipher and we are looking for arbitrary polynomial invariants coded as formal polynomials. In particular however, the structure of the polynomial is provided or guessed by the attacker depending on "hints" or "heuristics" which are based on the high-level structure of the cipher. This is what is suggested from known examples of polynomial cryptanalysis [9, 10]. In particular for Feistel ciphers we will see a lot specific types of symmetric polynomials on subsets of variables. Therefore our polynomials will look like:

$$P(Za, b, \ldots) = P00 + P51 * (e + f + g + h) + P42 * (abc + abd + acd + bcd) + \ldots$$

this rather than completely general polynomials which would be too costly to consider and process efficiently. The goal is of course to avoid excessive generality and reduce the number of variables such as P51 which are determined at a later "algebraic solving" stage of the attack.

3 General Approach vs. One Specific Cipher

Most cryptanalytic attacks work, well, only in some cases. In this paper we try to strike a balance between a completely general constructive approach applicable to any block cipher, such as DES, AES, GOST, etc, and constructing simple examples dictated by the high-level structure of one specific Feistel cipher.

3.1 Polynomial Invariants vs. General Partitioning Attacks

In theory, our approach is totally general as every function could be written as a polynomial over a finite field, therefore in a very broad (and naive) sense Partitioning Cryptanalysis (PC) and Generalized Linear Cryptanalysis are equivalent. However our approach would not be practical if needed to study completely general polynomials with 36 variables In reality however, we will be working with sets characterized by one polynomial at a time, our ability to solve the equations is limited, and our approach will principally work when P is not too complex or when it is sparse and/or of reduced degree, and with specific very strong symmetries, cf. Section 2.3.

3.2 Constructive Approach Given the Cipher Wiring

We decided to execute our task on a cipher which offers great **flexibility** in the choice of the internal wiring, so that we can possibly make such adjustments if we do not find a property we are looking for. In theory most ciphers such as DES or AES do offer this level of flexibility (choice of P-boxes, arbitrary invertible matrices inside the S-box and inside the mixing layers, etc). Here we work the T-310 cipher from 1970s where such changes are officially allowed and are officially specified by the designers of the cipher. Here if we find a weak setup, it can be directly implemented with original historical hardware. Our work is at the antipodes compared to [10, 11] where the ciphers are really very special and have very strong hidden high-level structure (and then non-linear invariants propagating for 1 million rounds can be constructed). Our approach is really the opposite: we start from any given cipher spec in forms of ANFs for one round (possibly with some "chose permutation of wires" flexibility for connecting non-linear components) and we generate invariant properties on demand.

In general the attacker is allowed to manipulate the Boolean function, which variable has a large entropy (64 bits typically) and to a lesser extent also to somewhat but less[2] manipulate the cipher wiring, and also he is able to freely select an invariant polynomial P which choice however is restricted in several ways[3]. The approach we present is applicable to more or less[4] arbitrary block

[2] Only in the sense of 1 out of 1000: study of weaker and stronger cipher wirings.

[3] It will be restricted by 1) which kind of polynomial invariants maybe exit for this cipher, for example Feistel ciphers tend towards combinations of symmetric polynomials with subsets of variables, and 2) the computing power of the attacker and his ability to find invariants by solving increasingly complex systems of equations.

[4] This is if there is a sufficient amount of entropy inside these non-linear components, possibly not to Simon or TEA where S-boxes are too small or rather inexistent for us to manipulate.

ciphers which contain non-linear components. It is also applicable to hash functions and stream ciphers based on a core block cipher (a large permutation with few round-dependent bits which can be key bits, IV bits or message bits).

3.3 Representing One Encryption Round

Let us imagine that we are presented with an arbitrary complex block cipher designed by some very paranoid designers with large complexity and a sophisticated internal structure. For example the picture below shows the internal structure of T-310, one of the most important block ciphers of the Cold War, which was used at a massive scale in Eastern Europe to encrypt all sorts of state communications, cf. [37]. The cipher operates on 36-bit blocks and the state bits are numbered 1-36. Here is a glimpse on the internal structure:

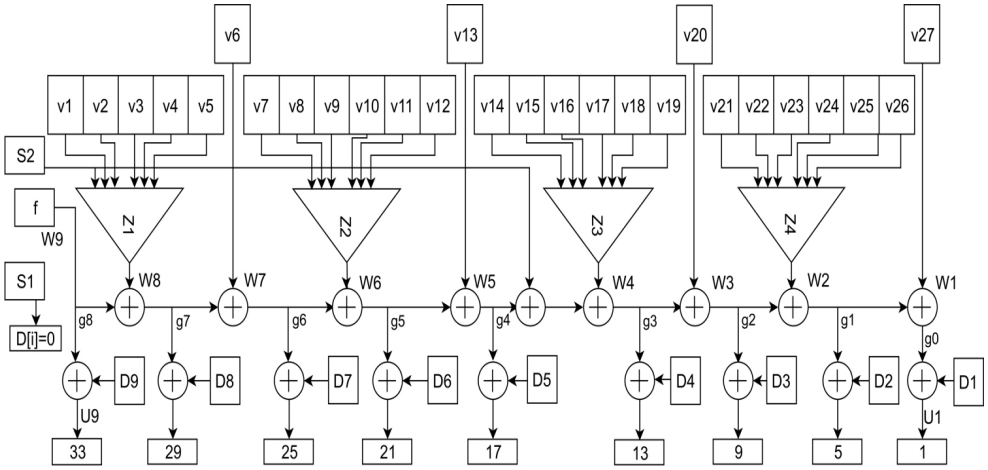

Fig. 1. A complex structure inside one round of T-310 block cipher.

T-310 happens to be one of the most "paranoid" cipher designs we have ever seen. The per-encrypted-bit hardware complexity of T-310 is absolutely staggering. It is hundreds of times more costly, even with modern software and hardware, than AES or triple DES, cf. [16]. Also typically and almost always the bits actually used in encryption come from the right end, see [21, 14], which is more complex than the other end. However one round is **intelligible** and by no means secure. Therefore the designers have mandated that incredibly large number of 1651 such rounds must be executed in order encrypt just one character on 5 bits. Does it make it cipher very secure as intended for a serious government security cipher in the middle of the Cold War? Not quite, if we are able to show that algebraic invariants can be constructed which work **no matter how many rounds** we apply, and not quite if such invariants exist which work without using all the state bits, and without using all the key bits either.

3.4 Internals of One Round

A lot remains unspecified on our picture: which bits and in which order are connected to D1-D6 and v1-v27. In T-310 this specification is called an LZS or *Langzeitschlüssel* which means a long-term key. It could be compared to the knowledge of rotors in Enigma or S-boxes in GOST. We need simply to specify two functions $D : \{1 \dots 9\} \to \{0 \dots 36\}$, $P : \{1 \dots 27\} \to \{1 \dots 36\}$. 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 $Z1$.

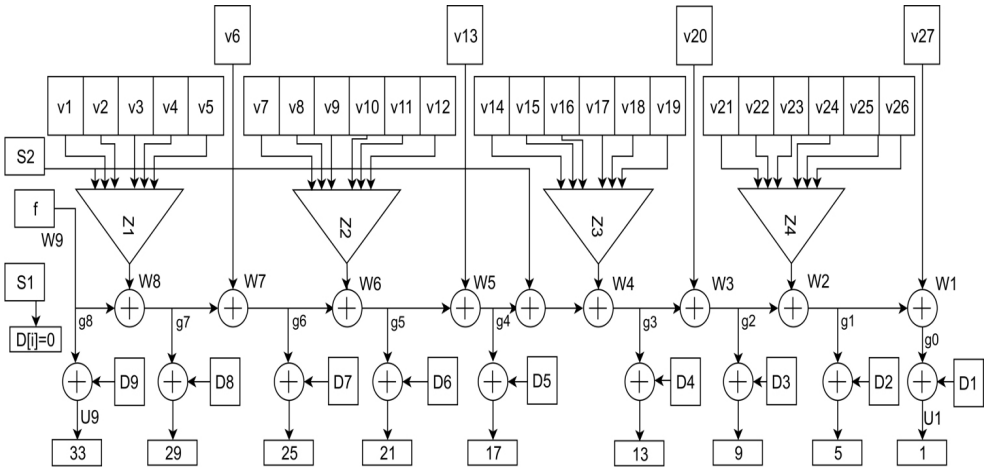

Fig. 2. Internal wiring of one round of T-310.

It remains to specify that each round, bits $1, 5, 9 \dots 33$ 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, and bit 35 becomes 36. this completes the specification of the internal connections of one round[5]. We will also note that F is a public bit derived from an IV transmitted in the cleartext, and S1 and S2 are bits of the secret key which has 240 bits Finally the internal wiring LZS has a special convention where the bit S1 is used as one of Di by specifying [6] that $D(i) = 0$. For the time being[7] we are going to assume that all the four Boolean functions Z1-Z4 are identical.

[5] In fact, it is possible to rewrite this cipher to make it look as a non-orthodox variant of an "Unbalanced Compressing Feistel" cipher with 4 branches, cf. [31, 14].

[6] Needless to say D and P are injective and have been specified and studied with serious amount of maths and analysis since the early 1970s [34]. Historical documents [34] show advanced combinatorial results about strength of the cipher after a few rounds, and contain explicit group-theoretic claims about the size of the group generated by the permutations of this cipher. A recent paper analyses the historical classes KT1 and KT2 of long-term keys from [34] and shows that they can be proven to be secure against a certain class of ciphertext-only attacks, see [15].

[7] This condition will be definitely relaxed in a future update of this paper and offers yet much larger freedom for the attacker, with 256 bit of entropy to manipulate.

3.5 ANF Coding of One Full Round

Overall the cipher can be described as 36 Boolean polynomials out of which only 9 are non-trivial. Let $x_1, \ldots, x_{36}$ be the inputs and let $y_1, \ldots, y_{36}$ be the outputs.

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

We observe that T-310 has a remarkable "triangular" structure with increasing complexity and we aim to benefit from this with invariants using a subset of all cipher state bits only.

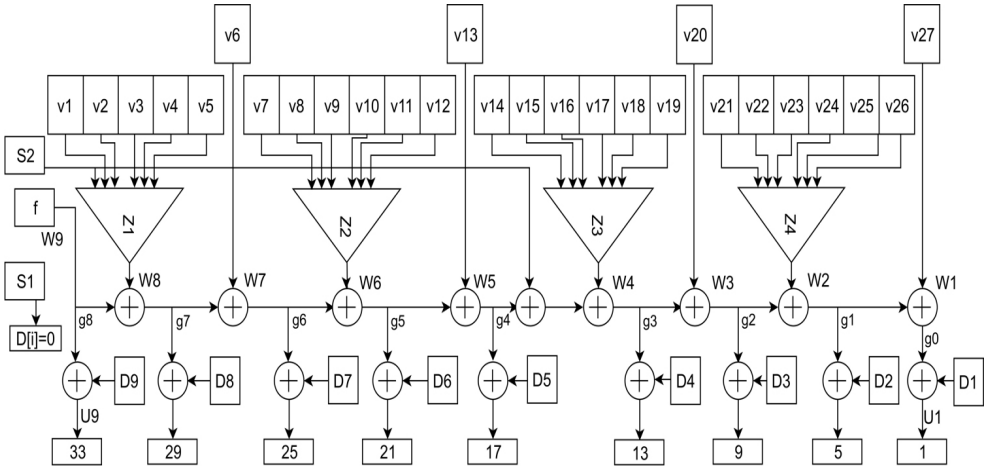

Fig. 3. Internal wiring of one round of T-310.

3.6 Variable Renaming

We recall that input of one round are denoted by $x_1, \ldots, x_{36}$. Now we are going to migrate towards more convenient and shorter notations. We will emphasise the fact that the variables x_i and y_1 are treated "alike" and we will call them by lowercase letters a-z backwards starting from x_{36} till x_{11} and y_{36} till y_{11}. Thus for example $a = x_{36}$ and $t = x_{17}$. This will allow our higher degree polynomial expressions to be take less space. Unhappily by doing this we deprive ourselves from the possibility of using variables x_{10} till x_1, however, quite interestingly, in our work we simply want to avoid using such variables. Basically due to the peculiar triangular structure of this cipher, variable x_1 has the most complex dependency on the highest possible number of input bits and is more likely to use key bits. We are going to avoid generating examples which would be too complex. Yet we have a scalable approach able to produce numerous non-trivial solutions with increasing complexity.

The reader may be surprised to hear that in our cryptanalysis efforts we will be able to map x_i to y_i (they are denoted by the same letter) and this for ALL the 36 variables, and this after exactly 1 round. This even though for almost all variables $x_i = y_{i-1}$ after one round. This is **not impossible**, not even in Linear Cryptanalysis (LC). For example if $a \stackrel{def}{=} x_{36}$ in input expressions, and also used to denote y_{36} for output expressions $b \stackrel{def}{=} y_{35}$ and also used to denote y_{35}, it is unthinkable that say a is an invariant property because actually b becomes a in the next round. However in fact a polynomial such as $a + b + c + d$ absolutely CAN be an invariant expression for one round (!). This is, only IF our cipher had ANY invariant linear property for 1 round true with probability w.r.t the usual Linear Cryptanalysis (LC). This is possible but not so common. It does happen for this is cipher, and a recent paper shows that linear invariants will occur for about 3% of so called LC-weak long term keys inside the so called KT1 specification of officially approved or "well-formed" long-term keys, cf. [16].

In our paper we will be of course looking at what happens for the remaining **stronger** 97 % of LZS keys and at more general **non-linear** polynomial expressions such as say $h + g + f + e + hgf + hge + hfe + gfe$. All examples we give in this paper are real-life examples which actually have been encountered in our research on cryptanalysis of this cipher. The polynomials we obtain are however not random, but rather they combinations of specific symmetric polynomials on subsets of variables, with additional terms which are not symmetric, and they do depend very strongly on the wiring of the cipher. For example this polynomial invariant with 74 monomials has been seen to work:

acdeghjk+abcdeghjk+acdefghjk+abcdefghjk+abdehijk+abcdehijk+abdefhijk+abcdefhijk+abdeghijk+acdeghijk+abdefghijk+acdefghijk+
acdgijl+abcdgijl+acdegijl+abcdegijl+bcfgijl+abcfgijl+acdfgijl+bcdfgijl+bcefgijl+abcefgijl+acdefgijl+bcdefgijl+acdghijl+
abcdghijl+acdeghijl+abcdeghijl+bcfghijl+abcfghijl+acdfghijl+bcdfghijl+bcefghijl+abcefghijl+acdefghijl+bcdefghijl+bcdfhikl+
abcdfhikl+abdefhikl+bcdefhikl+bcdfghikl+abcdfghikl+abdefghikl+bcdefghikl+abcdegjkl+abcdefgjkl+acdeghjkl+acdefghjkl+abcfijkl+
abcdfijkl+abcefijkl+abcdefijkl+acdgijkl+abcdgijkl+acdegijkl+bcfgijkl+acdfgijkl+bcdfgijkl+abcdfgijkl+bcefgijkl+acdefgijkl+
bcdefgijkl+abdehijkl+abcdehijkl+abcfhijkl+bcdfhijkl+abcefhijkl+bcdefhijkl+acdghijkl+abcdghijkl+abdeghijkl+bcfghijkl+acdfghijkl+bcefghijkl

3.7 The Result After Renaming

Overall, depending on the exact values of $D(i)$ and $P(j)$, we can rewrite the beginning of our equation system as follows, skipping the last few equations (our invariants will NOT use them!). 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.

$$
\begin{aligned}
a &\leftarrow b \\
b &\leftarrow c \\
c &\leftarrow d \\
d &\leftarrow F + i \\
e &\leftarrow f \\
f &\leftarrow g \\
g &\leftarrow h \\
h &\leftarrow F + Z1 + e \\
Z1 &\leftarrow Z(L, j, h, f, p, d)) \\
i &\leftarrow j \\
j &\leftarrow k \\
k &\leftarrow l \\
l &\leftarrow F + Z1 + r + g \\
m &\leftarrow n \\
n &\leftarrow o \\
o &\leftarrow p \\
p &\leftarrow F + Z1 + r + Z2 + c \\
Z2 &\leftarrow Z(k, l, o, e, n, t) \\
q &\leftarrow r \\
r &\leftarrow s \\
s &\leftarrow t \\
t &\leftarrow F + Z1 + r + Z2 + m + s \\
u &\leftarrow v \\
v &\leftarrow w \\
w &\leftarrow x \\
x &\leftarrow F + Z1 + r + Z2 + m + L + Z3 + b \\
Z3 &\leftarrow Z(u, s, ?, ?, ?, b) \\
y &\leftarrow z \\
&\ldots \text{ [few lines missing]}
\end{aligned}
$$

These expressions should be viewed as a sequence of substitutions where a variable is replaced by a polynomial algebraic expression.

3.8 Further Remarks and Observations

We will use the sign ? each time we run out of convenient 1-letter variable names. We will then aim at not using at all these variables in our invariants. Moreover in order to have shorter expressions to manipulate later on, we have replaced S2 by a single letter L. The other key bits $S1$ currently do not appear at all (the place where they would appear depends on $D(i)$). In general in our attacks we would represent key bits $S1$ by short[8] notation K. In practice will **avoid** using S1, as in real-life encryption it is almost always used at place D1, cf. [21, 14] and would appear in the very last equation $y_1 = F + \ldots + x_{D(1)}$ in Section 3.5 which we have omitted now. The lower side is the side we will be simply avoiding (for the time being) in our invariant attacks. This not only due to the shortage of lowercase letters, but also because equations tend to be more complex than at the upper end(!). Using only a subset of variables and avoiding K, or eliminating K is what we need, and it is a great idea form the cryptanalysis point of view. In this way we will in fact obtain I/O properties with fewer monomials and such that half of the secret key of type S1 (up to 120 bits) are **eliminated** from the start(!).

3.9 On ANF Degree

One important point is that the algebraic degree of all our ANF expressions is constant. The ANF expressions have degree at most 6 and it does not increase for the lower parts (parts with K and bits 1 to 11 we may want to avoid). In contrast the number of monomials actually used increases (linearly).

3.10 Search for Invariants

Now the only thing which remains to be done is to find a polynomial expression P say

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

using any number between 1 and 36 variables such that if we substitute in P all the variables by the substitutions defined above in Section 3.7, we would get exactly the same polynomial expression P .

For simplicity and in practice in our proof of concept examples in this paper we will be using only 20 variables a-t. and we will avoid Z3 and Z4, constructing only invariants using Z1 and Z2 with round-dependent constants L and F only, cf. Section 4.12. More generally what we do, applies also to Z3 and Z4 and to any block cipher, however there will be more than 2 round-dependent parameters [key bits or round-dependent constants] and the success rate will be lower.

[8] This is used when we are looking at 1-round invariant properties and there is no ambiguity about in which round this variable K is used.

4 The Fundamental Equation

We want to **find** a polynomial expression P using any number between 1 and 36 variables such that it is an invariant after the substitutions of Section 3.7. In this paper the idea is that P could potentially be any non-linear polynomial of a certain degree with a specific well-chosen set of say 17 variables (specifically avoiding using too many variables from the lower parts of the cipher) and with specific internal symmetries due to the structure of the cipher. One simple method is to select a specific short symmetric polynomial which have already been seen to work for this cipher. More generally the polynomial is not always symmetric (cf. Section 5.1 and will also have variables which are not yet known, and yet have a specific form, cf. example in Section 2.3.

Once the polynomial P is fixed, the attacker will write ONE SINGLE algebraic equation which he is going to solve to determine the unknown Boolean function Z, if it exists.

Definition 4.1 (Compact Univariate FE). Our "Fundamental Equation (FE)" to solve is simply a substitution like:

$$P(Inputs) = P(Outputs)$$

or more precisely

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

At this stage expressions of type Z1 or Z3 are placeholders for degree 6 polynomials yet to be specified fully.

The main unknown in FE is a Boolean function Z and in simple cases the FE can be of type $fZ = g$ where f and g are two polynomials[9]. More generally Z is represented by an Algebraic Normal Form (ANF) and 64 binary variables which are the coefficients of the ANF of Z, and there will be several equations, and several **instances** Z1-Z4 of the same Z:

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

$$Z1 \leftarrow Z00 + Z01 * L + Z02 * c + Z03 * Lc + \ldots + Z62 * cklfh + Z63 * Lcklfh$$

Likewise we will also replace:

$$Z2 \leftarrow Z00 + Z01 * k + Z02 * l + Z03 * kl + \ldots + Z62 * loent + Z63 * kloent$$

and the coefficients $Z00 \ldots Z63$ will be the same inside $Z2$, $Z3$ and $Z4$, however the sets of 6 variables chosen out of 36 will be different in Z1-Z4.

Note. Our compact notations omit the stars for products of small variables.

[9] Such equations have numerous non-trivial solutions, cf. [13].

4.3 On Choice of P in our FE

Here P is an arbitrary fixed polynomial, with degree say between 2 and 20. For example one we have chosen ourselves(!). Or the one of Section 2.3 with some unknowns. Then if we cannot find a solution, we will enlarge the space of solutions but making more or all coefficients of P variable. In all cases also, all we need to do is to solve the equation above for Z (plus a variable amount of extra variables). This formal algebraic approach, if it has a solution Z, will **guarantee** that our invariant P holds for 1 round.

4.4 Solving the Fundamental Equation

We recall our "Fundamental Equation (FE)".

$$P(Inputs) = P(Outputs)$$

or

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

The process is as we can see EXTREMELY SIMPLE: we assume that a certain equation holds for Z and we solve it for Z which is 64 binary unknowns for the ANF coefficients. The solution is found easily[10] either by standard Gröbner bases techniques, or by conversion to SAT problem as described in [7]. Our experience seems to show that this problem will rarely be actually computationally really hard. This depends on many factors such as size, degree and shape of P etc.

4.5 Case when P is fixed

In fact our problem is in many interesting cases trivial to solve. For example it is easy to see that if P fixed, and if L, F are not used, (or if they are fixed, or for a reduced set of equations after elimination of all monomials containing L, F) the problem boils down to solving a system of LINEAR of equations with 64 binary variables which are the coefficients Zii which define the ANF of Z.

4.6 Simultaneous Solving

In a major variant of our problem, the polynomial P is no longer fixed, instead some or all of its coefficients are variables. Moreover it is in general advisable to already assume that P has specific symmetries which decreases the number of coefficients to be determined, cf. example in Section 2.3. Then, ignoring special bits L, K, F etc, the equations are no longer linear, but rather bi-linear, with products of coefficients of P and of Z. More generally, the problem is multi-linear. Again, in practice this problem is solved by a SAT solver typically or by Gröbner basis algorithms.

[10] All we have to do is to, consider that equality above must hold for each coefficient of each possible monomial. Thus we can rewrite the equation above as a system of simultaneous non-linear equations which will be small degree sparse polynomials in the 64 unknown coefficients Zii which form the ANF of Z, $ii = 0, \ldots, 63$.

4.7 The Solvability Problem

A major problem is rather, **the existence of solutions**. Does this equation FE have a solution? Or, does it have solutions which would satisfy some additional requirements required to design any sort of meaningful cryptanalytic attack? This is going to be a major question to study.

There are, as the reader may guess, countless cases where this problem has **no solution** whatsoever. Our experience shows that in many cases contradictions in our equations are found very easily[11] without even examining all the parts of the equations. This means that the fact that the space for Z is very large does not necessarily help to make our system of equations solvable. Frequently we stop earlier.

There are also numerous cases where the equation becomes unusually simple and has low degree, or it disappears totally (it reduces to zero once P is fixed, see Appendix A). In such cases the space for the solutions is typically quite large which increases the applicability of the attack to real-life cases, for example when the attacker is not quite able to choose the Boolean function, but he might be able to manipulate the cipher wiring and increase the degree and complexity P.

4.8 More General 5-Linear Fundamental Equation

In general we write our Fundamental Equation (FE):

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

as sum of products where we can distinguish 5 distinct types of variables or terms:

1. Terms of type $S1, S2$ or L which are key bits unknown to the attacker.
2. Terms of type F which are IV or round-dependent constant bits known to the attacker.
3. Terms of type Zii which are coefficients inside Z.
4. Terms of type Pjj which are coefficients of P if it is not fixed.
5. Terms of type $abcdklm$ which are monomials in internal I/O variables of the cipher.

Again the equation is expected to hold for all values for the small 1-letter variables abcdef and all the variables which start with capital letters are placeholders for things which play a particular role in our solving process. In general, depending on circumstances, we fix some of these terms and we determine the other by solving the Fundamental Equation. For example if P is known and fixed, the equation becomes 4-Linear in the sense above. Or if Z is known and fixed, the equation is again 4-Linear and can be used to determine P. Again in special cases this equation will be linear and in most other cases, it will be nevertheless possible to solve in practice.

[11] For example if our Fundamental Equation after our substitutions has the property that if Z divides one monomial, Za also divides this monomial, and if there is a single non-zero monomial without a, then the system of equations has no solution. Thus the fact that space for Z is quite large does NOT guarantee that the set of solutions we are looking for is not empty.

4.9 Extended 5-Linear Fundamental Equation

More generally we can extend our Fundamental Equation (FE) to:

$$P(a, b, c, d, e, f, \ldots) = P(b, \ldots, F + Z1 + r + Z2 + m + s, \ldots) + Q(F, L)$$

where Q is an another polynomial which depends on key and IV bits. In short notation we call this type of 5-Linear Extended Fundamental Equation by the name 5EFE.

4.10 Even More General Iterative 6-Linear Fundamental Equation

In general, the approach can be **iterative**. The attacker may, for example due to a previous iteration and solving FE once, already know some specific polynomial invariant for both the input and the output of each round. We call M this number and M is a complex polynomial in all small-letter variables abcd.. however the value of M is known, the other values aren't.

Our Extended 6-Linear Fundamental Equation (FE) will be:

$$P(a, b, c, d, e, f, \ldots) = P(b, \ldots, F + Z1 + r + g, \ldots) + Q(F, L, M)$$

where we have added one extra term $Q()$ and additional inputs M:

6. Terms of type M are bits of the internal state known to the attacker.

We have seen countless examples when this happens, and there are also many keys with linear invariants which can be used here as M in the same way. A nice real-life example is given in Section 7.2.

4.11 Iterative Solving

This section should be treated as optional and is not central in this paper. The approach presented above can be iterated in order to generate further invariants. Potentially in some cases the number of equations generated will grow by iterating this approach, until the cipher is completely broken, in the same way as the so called ElimLin algorithm, cf. [5, 33]. We believe that the 6-Linear approach above can be considered as a stand-alone cryptanalysis method. We believe it will break some very simple ciphers on its own, through iteration, without the necessity to do anything else other than iteration of linear algebra steps on well-chosen sets of non-linear polynomials which will yield a monotone sequence of invariant polynomial spaces of increasing size. In addition, in ElimLin equations penetrate slowly inside the cipher. Here all equations penetrate inside the cipher and we can generate new polynomial relations on variables at any round inside the cipher. However, as in ElimLin, we expect that in most cases the number of equations will stabilize early on, and no new invariants will be generated.

4.12 Additional Steps and Dealing with Key Bits

We have not yet explained how to deal with key and IV bits in the cipher. This problem was already a major problem in [9] and general non-linear cryptanalysis is substantially more complex than Bi-Linear cryptanalysis. One obvious method, is for example to assume say $F = 0, L = 0$ where $L = S2$ and find invariants by solving the Fundamental Equation (FE). But then we have NOT yet broken any cipher, though we can distinguish any power of a certain complex iterated permutation from a random permutation, including a power (an integer) which is not known to the attacker, which is already a strong result. At the antipodes, we can simply try to eliminate ALL the monomials which contain they key and IV bits. As simple as that, and this solves our problem and allows the attacker to deduce things about the internal state of the cipher for arbitrarily large numbers of rounds. Another method, is to use the method above for $F = 0$, then for $F = 1$, combine the two systems FE equations, and solve a bigger system of simultaneous [linear] equations for the coefficients of Z. More specific non-trivial strategies are needed when dealing with $S2$ or L bits, cf. Section 6.1 below, while we try to avoid as much as possible dealing with $S1$ bits. But it is maybe too early to go into such fine technical cryptanalytic details.

We need first to study, not if all this is actually feasible, but more importantly, if the set of solutions is **not empty**. We are going to solve first some simplified cases, before we attack more complex situations.

5 Concrete Examples

An example is worth more than a thousand words. We present numerous examples chosen for their elegance and simplicity. In these examples the long-term key has been selected inside some 100 permutations we have considered. Our approach is to construct toy ciphers fully compliant with the spec of T-310 cipher which involve a reduced number of cipher bits, between 13 and 20 out of 36. This is precisely in order to generate examples which are simpler than in the general case and therefore easier to study. Three quite interesting things happen then: first of all, interesting invariants do exist frequently, and secondly, they ignore many of the 36 bits, and also many of our reduced set of say 15 or 20 key bits, and third, they do not involve any of S1 key bits. Initially in our research we found plenty of examples of very complex invariants of degree say 11. Then we looked for simpler low degree invariants which are easier to study.

Moreover and importantly, we focus on examples where invariants would be happening NOT for pure structural/wiring reasons, but such that the invariant holds for a larger[12] permutation due to circumstances which depend very **strongly** on the Boolean function Z and which allow all other 36-4 cipher state variables to be eliminated. In other terms at this moment we will rather discard all invariants which would work for any Boolean function Z. We consider that such attacks are rather too trivial[13] or we consider that they are maybe too strong[14] and also too easy to detect in terms of more fundamental immutable properties of the cipher wiring, in order to be actually used in real-life cryptanalysis.

5.1 A Toy Example with P of Degree 3 and 4 Variables and $F = 0$

We examined a number of possible permutations fully compliant with the spec of T-310 cipher for degree 2 GLC invariants true with probability 1. For example, let us start with a non-linear round invariant property P which is particularly simple and uses only 4 variables:

$$P = efg + efh + egh + fgh + fg$$

We are quite close to using a random bijective LZS on a reduced set of bits and $P - fg$ is a symmetric homogenous polynomial of degree 3 in 4 variables only. However it is important to note that overall P is **NOT** a symmetric polynomial.

Here is an example of a long term key where this invariant works:

```
317: P=27,29,31,21,33,19,26,25,22,32,23,17,24,16,18,9,5,
10,35,13,36,30,34,11,2,28,14 D=17,25,26,35,18,34,30,32,28
```

[12] It is very useful when the invariants do NOT use all the key bits and such attacks are our core focus. Such attacks are expected to scale to larger ciphers and to larger subsets of bits. We consider that invariants which use all the key bits will be very cumbersome and are less likely to ever lead to practical attacks on block ciphers.

[13] When for example a smaller permutation on say 4 bits, cf. Section 5.2, is embedded inside our permutation on 20 bits which is a trivial form of a weak cipher.

[14] In Appendix A we show a nice real-life example of such an invariant.

What is the solution Z? All we need to do is to write our Fundamental Equation (FE) and substitute four variables using the ANF round equations in page 11.

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

On the right hand side we replace $a \leftarrow b$, etc, up to $h \leftarrow F + Z1 + e$.
Our Fundamental Equation (FE) becomes:

$$F(fg+fh+gh)+Z(fg+fh+gh)+gh+fg$$

We will now assume $F = 0$. We get:

$$Z(fg+fh+gh)+fg+gh$$

Due to absence of variables p and t in P, this equation FE contains only Z and not Y. More generally Z and Y are just two instances of the SAME Boolean function on two disjoint sets of variables.

We recall our Fundamental Equation (FE) which becomes:

$$Z(fg+fh+gh) = fg+gh$$

Then we need to substitute Z by an expression of type

$$Z00 + Z01 * L + Z02 * j + Z03 * L * j + Z04 * h + Z05 * L * h + \ldots$$

with the correct 6 variables L, j, h, f, p, d in the correct order. And if needed, we also replace Y by

$$Z00 + Z01 * k + Z02 * l + Z03 * k * l + \ldots$$

where the coefficients Zii will be the same.

The FE becomes then:

```
(LZ33+LZ41+Z32+Z40)*dfg + (LZ37+LZ41+Z36+Z40)*dfgh +
(LZ39+LZ43+Z38+Z42)*dfghj + (LZ55+LZ59+Z54+Z58)*dfghjp +
(LZ53+LZ57+Z52+Z56)*dfghp + (LZ35+LZ43+Z34+Z42)*dfgj +
(LZ51+LZ59+Z50+Z58)*dfgjp + (LZ49+LZ57+Z48+Z56)*dfgp +
(LZ33+LZ37+LZ41+LZ45+Z32+Z36+Z40+Z44)*dfh +
...
(LZ17+LZ21+Z16+Z20)*ghp
```

From here we obtain a system of simultaneous almost-linear[15] equations starting with:

```
L*Z33+L*Z41+Z32+Z40=0
L*Z37+L*Z41+Z36+Z40=0
...
L*Z17+L*Z21+Z16+Z20=0
```

[15] Here it is nearly linear because P is fixed, and most parts of it are linear. Except that some variables Zii are multiplied by L.

There are many solutions to this equation. One example solution is:

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

with $L = 1$ which gives in our case:

$$Z1 = Z(L,j,h,f,p,d) = h + f$$

and it easy to check that our FE in Z holds:

$$(h+f)(fg+fh+gh) = (fgh+fh+gh) + (fg+fh+fgh) = fg+gh$$

This solution is linear and no one would buy a backdoor cipher without non-linear functions Z. Such a cipher actually have additional issues, for example $P = e + h$ is also an invariant. However there are also plenty of **non-linear** solutions(!) which work all the same. For example:

$$Z(a,b,c,d,e,f) = a + d + ad + cd + f + af$$

which becomes

$$Z1 = Z(L,j,h,f,p,d) = L + f + Lf + hf + d + Ld$$

this one works only when $L = 1$. This completes the problem of constructing a non-linear invariant attack in our toy example. We have two concrete examples of Z and many other exist. However we are only able to construct an invariant in 1 case: here with $F = 0$ and $L = 1$. This invariant does not break T-310 cipher as F and L vary in different rounds.

5.2 A Very Important Observation

The example above is one of the simplest we have seen and is a bit degenerate: there are plenty of Boolean functions which work and many are not very interesting. However this example has an interesting feature: it is highly modular and we can add or remove invariants (potentially) one by one. Here is a simple example. It is easy to see that if in our example, we put $Z = 1 + a + c + d$ which becomes $Z1 = h + f$ when $L = 1$, we get FE of the form:

$$P(e,f,g,h) = P(f,g,h,F+h+f+e)$$

and we have bits e, f, g, h which do not depend on what happens in other parts of the cipher. A little toy cipher on 4 bits embedded inside a toy cipher on 15 bits. With other Boolean functions however, this very strong weakness goes away, the bits e, f, g, h **will** again depend on what happens in other parts of the cipher, and yet we keep the SAME non-linear invariant $P = efg+efh+egh+fgh+fg$.

Furthermore it is easy to see that when $Z = c + d$ our cipher has another really bad linear invariant $P = e + h$ when $F = 0$ and for any L. By setting $Z \neq c + d$, for example $Z = a + d + ad + cd + f + af$ we have **removed** a linear invariant $e + h$ from our cipher, but we have **kept** many non-linear invariants. This is an essential feature of our approach, a potential and ability to **remove some invariants while keeping other**, independently, by manipulating Z.

5.3 Resistance to Linear Cryptanalysis

Moreover, here the reader will have to believe us, this is a bit harder to check, the key 317 with $Z = a + d + ad + cd + f + af$ has no linear invariants. This cipher setting is **not vulnerable to Linear Cryptanalysis (LC)** in none of the four cases such as $L =?$ and $F =?$.

5.4 Example with $F = 1$

It is possible to see that when F, the same LZS 317 also has a non-linear invariant for the same $Z = a + d + ad + cd + f + af$ which is also actually of degree 2:

$$P = ef + fg + eh + gh$$

this is only when $F = 1$ and $L = 1$. Here the FE is $(Z + 1)(f + h) = 0$. A "slight" problem is that P is not the same as when $F = 0$.

5.5 Can We Have the Same P when $F = 0$ and $F = 1$?

An interesting question is, whether it is possible to find invariants which work in several cases simultaneously, for example when $F = 0$ and when $F = 1$, i.e. we want the SAME invariant P to apply to two different permutations with the same key. The Fundamental Equation method applies all the same. We will just obtain two FE equations instead of 1, leading to fewer solutions for Z and frequently leading to no solutions. Making sure that this equation is actually solvable is one of the main problems in this research. Several examples of how this can be achieved are provided in the following sections.

5.6 Impossibility Results and Provable Security

If this cannot be done, we would like to be to be able to prove mathematically that FE has no solution and this attack is impossible for our cipher. Thus we can hope to obtain for certain cipher a security proof against our method of making a block cipher deliberately weak. There exist numerous negative general results of this type in cryptanalysis, cf. [6, 8, 15] some of which will also apply here (in particular all those which also apply to partitioning cryptanalysis cf. [23, 6]. This paper leads indeed to a new well-defined way to **prove** a security of a cipher against a malicious Boolean function attack, where we will prove that FE has no solution. Such a proof can be done mathematically, or in a more automated way through formal algebra and known results in theory of polynomial ideals, using some Gröbner basis computations as a tool, or in a more obscure way with software such as a SAT solvers (which will output UNSAT, and some SAT solvers are also able to output a undeniable proof of UNSAT). Such a proof will exclude a very large number of attacks with a variety of Boolean functions and polynomials P, which space can hardly be explored systematically.

Application to T-310. Interesting questions are: is the KT1 class of keys for T-310 cipher specified in 1970s provably secure against non-linear invariants? A recent paper shows that it is **not** secure already against linear invariants [16]. In a future paper we expect to be able to show it is NOT secure against non-linear invariants either, and this for a strictly larger percentage of long-term keys

inside the class KT1. At this moment (for technical reasons due to reducing the number of variables the size of equations we solve and the number of key bits involved) all long-term key examples in this paper are not of type KT1.

6 Preliminary Remarks About Multiple Invariants

Our objective is to find a property which propagates for an arbitrarily large number of rounds in T-310, a real-life historical block cipher. This means that we need to find a **simultaneous** invariant which works in four cases $F = 0, L = 0$ up to $F = 1, L = 1$. This is not a small problem and it appears to be the most difficult task to accomplish here. All other steps are comparatively quite easy and the method remains the same: we just need up to four copies of our FE equations and solve these combined equations for a simultaneous solution. Specific examples will be shown below.

6.1 Dealing with L

Only 1 key bit is used per round. We need to (and we will be able to) find invariants early on, before the permutation becomes too complex. Moreover, as already explained, the key bits S1 we aim to eliminate them totally by using specific sets of variables on one side. Then it is possible to see that for the bits S2 a.k.a. L are not always excessively hard to deal with if we are allowed to manipulate the Boolean function. This because of the very specific way (not very strong) in which the key bit L is used in T-310. We can observe that if we restrict our attention to non-linear invariants which concern all the bits $a - t$, and Z1 non-linear function only, then P can use up to the whole 28 bits, which is plenty to explore for the attacker, and yet L is not used anymore[16]. We can then write and solve two FE equations for each case $L = 0$ and $L = 1$, and it is possible to see that IF both FE have solutions, we can produce a solution Z which will work for every L as follows.

Theorem 6.1.1 (Combination Theorem). We assume that in our FE problem Z2 is not used. We assume that we can solve the FE problem for Z independently in the case $L = 0$, and we obtain at least one Boolean function $Z_0(b, c, d, e, f)$, and also that we can solve the FE problem with the same P for $L = 1$ and obtain another Boolean function $Z_1(b, c, d, e, f)$. Furthermore we assume that at least **one** of these Boolean functions is non-zero. Then we can COMBINE the two solutions as follows:

$$Z(a, b, c, d, e, f) = a \cdot Z_1(b, c, d, e, f) + (a - 1) \cdot Z_0(b, c, d, e, f)$$

Proof: It is sufficient to see that with the combined Boolean function P is an invariant property for our permutation on 36 bits, for both $L = 0$ and $L = 1$.
Remark: this method is of limited interest and does not work when $Z2$ is used.

[16] Moreover there could be further properties which eliminate L with more bits

7 Construction of Multiple Simultaneous Invariants

7.1 A Strong Invariant Example with $F = 0$ and $F = 1$

On the algebraic front, very frequently the set of solutions is an empty set. We need therefore to examine a larger variety of LZS. For example we found the following 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
```

The substitutions are therefore, omitting the trivial ones of type $a \leftarrow b$, as follows:

$$\begin{aligned} d &\leftarrow F + e \\ h &\leftarrow F + Z1 + a \\ Z1 &\leftarrow Z(L, c, e, l, g, r)) \end{aligned}$$

Consider the following polynomial:

$$P = ae + bf + cg + dh + e + f + g + h$$

again we do not use Z2 and the fundamental equation is a remarkably simple set of 2 equations, for $F = 0$ and $F = 1$:

$$Z + Ze + a + e = 0$$
$$Ze = 0$$

One interesting solution, specifically avoiding solutions which involve a is: $Z(a, b, c, d, e, f) = bde + bcde$ which becomes

$$Z1 = Z(L, c, e, l, g, r) = (e + 1)clg$$

This is an educational example chosen for its simplicity, the FE is very short and composed of two equations which do not contradict each other.

Discussion. This setup is not excessively good yet. It is weak w.r.t Linear Cryptanalysis (LC) in some cases. We have found numerous examples not weak w.r.t. LC, however the FE is more complex. Moreover in general we are still not quite happy: our quadratic invariant works only in 2 out of 4 cases, when L is fixed. We have NOT YET broken a block cipher. We need to construct polynomial invariants which work in 4 cases simultaneously.

7.2 A Tentative Construction of an Advanced Quadruple Invariant

On this page we show that our key 827 allows a yet stronger form of attack to be constructed. Consider the following very simple polynomial:

$$P = ae + bf + cg + dh$$

Let $F = 0$, then we write our FE which becomes extremely simple $Ze = 0$ and it does NOT depend on L. Therefore when $F = 0$ our invariant P works for any L. It remains to see what happens for $F = 1$. In this case it is possible to see that we get another invariant which is the one from the previous section:

$$P = ae + bf + cg + dh + e + f + g + h$$

From this we can construct a more general invariant, 6-linear attack in the spirit of Section 4.10 with FE being of the form:

$$P(a, b, c, d, e, f, \ldots) = P(b, \ldots, F + Z1 + r + g, \ldots) + Q(F, L, M)$$

and

$$Q(F, L, M) = F \cdot M$$

More precisely, combining the two results, we have the following **quadruple** invariant which works for any F and any L:

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

where

$$P(a, b, c, d, e, f, g, h) = ae + bf + cg + dh$$

7.3 A Combined Invariant Attack with 2 Stages

In the example above $M = e + f + g + h$ is linear and the attacker could predict the value of $P(a, b, c, d, e, f, g, h) = ae + bf + cg + dh$ after an arbitrarily large number of rounds IF he can predict the values of $M = e+f+g+h$ at every round. Now, there are good chances that BOTH properties CAN be combined(!) in one single vulnerable long-term key (and polynomial M does not have to be linear). It is possible to see that both the property $P(left) = P(right)+F\cdot(e+f+g+h)$ occurs IF AN ONLY if a certain number of constraints C1 on P, D are satisfied, and moreover if another set of constraints C2 on P, D are satisfied, the attacker can also predict bits M at every round [possibly with a partial key guess]. The properties C1 and C2 are not always compatible and can be studied through machine learning, or through exact mathematical theorems[17]. In the case when pre-conditions C1 and C2 are compatible, which will happen sometimes[18], then we have found a way to **predict 2 bits for an arbitrary number of rounds** of our cipher and long term keys which satisfies both conditions C1 and C2 can easily be constructed[19].

[17] Numerous examples can be found in Section 21.22 page 87 in [14] and numerous examples of exact theorems which show how pre-conditions of P, D lead to a invariant property can be found in [14], see for example Theorem J.1.1. on page 173.

[18] In Section 21.14. [14] we see one example when two such conditions are compatible.

[19] In general this is done using a SAT solver, the solving complexity of this problem is very low, the coding is however very complicated.

7.4 A Concrete Example of a Quadruple Invariant or How to Backdoor T-310

In the previous example we have not shown if conditions C1 and C2 can be made to be compatible. An interesting question is, can we do better, and find a simpler example when an invariant can be constructed. Moreover, can this be done on demand say for ANY key including the same key 827 which will be a good indication that this sort of invariants are quite common. The answer is yes and in order to approach this problem we add a slight amount of arbitrary constraints to the FE equation: for example we select two or three higher degree coefficients of the polynomial P and fix them to zero. This allow us to generate a variety of solutions to the FE equation. In particular there exist a number of degenerate cases where there is no need for four FE equations, because some of them identical(!). Such cases are very nice as a proof of concept because of their simplicity. For example we consider the following very simple polynomial:

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

Again we replace $a \leftarrow b$, etc, with $d \leftarrow F + e$ and

$$h \leftarrow F + Z1 + a$$
$$Z1 \leftarrow Z(L, c, e, l, g, r))$$

the fundamental equation is then particularly simple:

$$Z + Zc + Zg$$

and because it does not depend on neither F not L we do not need four 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 4 cases and therefore Linear Cryptanalysis (LC) does not work here. Our **non-linear invariant** P works in all four cases and therefore it propagates **for an arbitrary number of rounds** for any key and for any IV.

7.5 Preliminary Conclusion

This completes our proof of concept for embedding a backdoor inside the T-310 cipher by solving the Fundamental Equation for Z. Now the ONLY difference between our cipher and the original government cipher is that the LZS is not standard. It is sufficient now to change[20] the LZS printed circuit board inside the cipher, in order to obtain a weak cipher with a non-linear property valid for an arbitrarily large number of rounds. We should add that no other cipher we heard of, uses such excessively large numbers of rounds as T-310 in the actual encryption process, leading to excessively a large hardware cost, cf. [17]. Our attack shows that in some case, all this does not help (!).

Very few attacks in symmetric cryptanalysis work when the number of rounds is very large, for example slide attacks specifically, cf. for example [4] or previous attacks with algebraic/polynomial invariants [10, 11].

[20] Such an upgrade would typically be done once per year, cf. [14].

8 Conclusion

One of the major open problems in block cipher cryptanalysis is to discover new **specific** types of invariant properties which can hold for a larger number of rounds. In this paper we study non-linear Boolean polynomial invariants. Previous researchers have had great difficulties to make this approach work and the space of the possibilities is too large for systematic exploration. In this paper we have turned the problem of non-linear cryptanalysis upside-down. We fix the combinatorial structure of the cipher, look at the round ANF formulas, and try to find a Boolean function Z and also simultaneously determine a suitable polynomial invariant P, which allows to achieve the desired property. This is done by solving the so called Fundamental Equation (FE) or several such equations combined. Such invariants have the capacity to avoid more complex parts of the cipher and also many key bits. Then, we encounter a major problem: several FE equations need to have a common solution. This can eventually be achieved(!), cf. Sect. 7.1 and 7.4. Iterative attacks are also possible, cf. Sect. 4.11 and 7.3.

This paper shows how a specific structure and internal wiring of a block cipher can be translated into a relatively simple FE equation, which can be used to study which specific non-linear invariants may exist (or not) for this cipher. Stronger invariants can now be defined and characterized algebraically, P must be such that most or all the coefficients of FE reduce to zero, see Appendix A. Our main contribution is to show that the attacker does not need to randomly search for a vulnerable non-linear component Z (and for P). Weak Boolean functions and specific polynomial invariants P can be **determined** – by solving our FE equation(s). Our approach is constructive, completely general and can be applied to almost any block cipher: we directly write the Fundamental Equation from the ANFs, substitute variables inside, and attempt to solve our FE(s).

Future research. 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 difficult.

Positive Results. In current research on backdoors in block ciphers there are many impossibility results [6] but extremely few possibility results [10, 8]. Partitioning cryptanalysis [26] properties can be quite obscure, (weak ciphers seem to occur accidentally, and complex ciphers seem secure for no reason [36]). Polynomial invariants are way more intelligible. We discover that weak ciphers follow clear rules and a whole range from simple to increasingly complex invariant properties can now be characterized, studied and computed explicitly.

On Our Specific Cipher. What is incredible is that this approach works, **at all**, for at least one real-life block cipher T-310. Several factors help to make this happen: extremely few key and IV bits are used in one round, there is some freedom in the choice of the internal wiring with a strong triangular structure, and the degree of the Boolean polynomials is limited to 6. In contrast the space of possible polynomials Z and P is extremely large. In future research we expect to show that the proportion of keys for which FE has a solution can be computed exactly cf. [16], and that it is strictly increasing as the degree of P grows.

References

1. Ange Albertini, Jean-Philippe Aumasson, Maria Eichlseder, Florian Mendel and Martin Schläffer: Malicious Hashing: Eves Variant of SHA-1
2. R. Aragona, A. Caranti, M. Sala: *The group generated by the round functions of a GOST-like cipher, Ann. Mat. Pura Appl., 196 (2016), 117.*
3. *Arnaud Bannier, Nicolas Bodin, and Eric Filiol: Partition-Based Trapdoor Ciphers,* `eprint.iacr.org/2016/493`
4. Nicolas Courtois, Gregory V. Bard, David Wagner: *Algebraic and Slide Attacks on KeeLoq,* In FSE 2008, pp. 97-115, LNCS 5086, Springer, 2008.
5. Nicolas T. Courtois, Iason Papapanagiotakis-Bousy, Pouyan Sepehrdad and Guangyan Song: *Predicting Outcomes of ElimLin Attack on Lightweight Block Cipher Simon,* In proc. of Secrypt 2016, `http://discovery.ucl.ac.uk/1521419/1/Courtois_SECRYPT_2016_104.pdf`
6. Marco Calderini: *A note on some algebraic trapdoors for block ciphers,* last revised 17 May 2018, `https://arxiv.org/abs/1705.08151`
7. Nicolas Courtois, Gregory V. Bard: *Algebraic Cryptanalysis of the Data Encryption Standard,* In Cryptography and Coding, 11^{th} IMA Conference, pp. 152-169, LNCS 4887, Springer, 2007.
8. Marco Calderini: *On Boolean functions, symmetric cryptography and algebraic coding theory,* Ph.D. in Mathematics, Supervisor: Prof. Massimiliano Sala, University of Trento, Italy, April 2015
9. Nicolas Courtois: *Feistel Schemes and Bi-Linear Cryptanalysis,* in Crypto 2004, LNCS 3152, pp. 23-40, Springer, 2004.
10. 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. Extended version: `https://eprint.iacr.org/2005/251.pdf`
11. 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`. Also explained in Appendix B of the extended version of [10].
12. Nicolas Courtois: *Data Encryption Standard (DES),* slides used in GA03 Introduction to Cryptography and later in GA18 course Cryptanalysis taught at University College London, 2006-2016, `http://www.nicolascourtois.com/papers/des_course_6.pdf`
13. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback,* Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345-359, Springer. A long, extended version of this paper is available from `www.nicolascourtois.com`.
14. 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`
15. 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`.
16. Nicolas Courtois, Maria-Bristena Oprisanu and Klaus Schmeh: *Linear cryptanalysis and block cipher design in East Germany in the 1970s,* will appear in Cryptologia in 2018.

17. Nicolas Courtois, Jörg Drobick and Klaus Schmeh: *Feistel ciphers in East Germany in the communist era,* In Cryptologia March 2018, https://www.tandfonline.com/doi/full/10.1080/01611194.2018.1428835
18. Nicolas T. Courtois: *The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime,* In SECRYPT 2009 International Conference on Security and Cryptography: pp. 331-338. INSTICC Press 2009, ISBN 978-989-674-005-4.
19. Nicolas Courtois: *Algebraic Complexity Reduction and Cryptanalysis of GOST,* Monograph study on GOST cipher, 2010-2014, 224 pages, available at http://eprint.iacr.org/2011/626.
20. Jörg Drobick: *T-310/50 ARGON,* a web page about T-310 cipher machines http://scz.bplaced.net/t310.html
21. Jörg Drobick: *T-310 Schlüsselunterlagen,* a web page which enumerates several different known long-term keys for T-310 from 1973-1990, consulted 21 January 2017, http://scz.bplaced.net/t310-schluessel.html
22. H. Feistel, W.A. Notz, J.L. Smith, *Cryptographic Techniques for Machine to Machine Data Communications,* Dec. 27, 1971, Report RC-3663, IBM T.J.Watson Research.
23. Carlo Harpes: *Partitioning Cryptanalysis,* Post-Diploma Thesis, Signal and Information Processing Lab., Swiss Federal Institute of Technology, Zurich, March 1995.
24. 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.
25. Carlo Harpes: *Cryptanalysis of iterated block ciphers,* PhD thesis, No 11625, Swiss Federal Int. of Tech., ETH Series in Information Processing, Ed. J. L. Massey, Hartung-Gorre Verlag Konstanz, 1996, ISBN 3-89649-079-6, ISSN 0942-3044.
26. C. Harpes, J. L. Massey: *Partitioning cryptanalysis,* In FSE 97, LNCS 1267, pp. 1327, 1997.
27. Thomas Jakobsen: *Higher-Order Cryptanalysis of Block Ciphers.* Ph.D. thesis, Dept. of Math., Technical University of Denmark, 1999.
28. Lars R. Knudsen, Matthew J. B. Robshaw: *Non-Linear Characteristics in Linear Cryptoanalysis,* Eurocrypt'96, LNCS 1070, Springer, pp. 224-236, 1996.
29. Mitsuru Matsui: *Linear Cryptanalysis Method for DES Cipher,* Eurocrypt'93, LNCS 765, Springer, pp. 386-397, 1993.
30. Pawel Morawiecki: *Malicious Keccak,* eprint.iacr.org/2015/1085
31. *Jacques Patarin, Valérie Nachef, Côme Berbain: Generic Attacks on Unbalanced Feistel Schemes with Contracting Functions,* in Asiacrypt 2006, pp. 396-411, LNCS 4284, Springer 2006.
32. Kenneth G. Paterson: *Imprimitive Permutation Groups and Trapdoors in Iterated Block Ciphers,* In FSE 1999, pp. 201-214, Springer 1999.
33. Petr Susil, Pouyan Sepehrdad, Serge Vaudenay, Nicolas Courtois: *On selection of samples in algebraic attacks and a new technique to find hidden low degree equations.* In International Journal of Information Security vol. 15 iss. 1, pp. 51-65, Springer, 2016.
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. *Ralph Wernsdorf: The one-round functions of the DES generate the alternating group,* In proc. of EUROCRYPT92, LNCS 658, pp. 99112, Springer, 1993.

36. R. Sparr and R. Wernsdorf: *Group theoretic properties of Rijndael-like ciphers,* In Discrete Appl. Math, 156 (2008), 31393149.
37. Klaus Schmeh: *The East German Encryption Machine T-310 and the Algorithm It Used,* In Cryptologia, vol. 30, iss. 3, pp. 251 257, 2006.

A FE with Reduction to Zero and Structural Invariants

In this paper we show that a Boolean function can be chosen in order to create a specific invariant. An interesting question is whether the Fundamental Equation can be reduced to zero (!), i.e. given a certain fixed P it is simply equal to zero and it holds for any Z. When this happens, we will obtain a polynomial invariant which **works for any Z** and therefore it also works also for the original Boolean function used in the T-310 cipher in 1970s-1990. Here below we give one example when this happens. In this example a quadruple FE equation can be solved for 4 cases simultaneously, because all the 4 equations are empty. We define key 898 by:

```
898: P=35,17,19,25,17,25,22,19,31,30,29,26,20,36,23,5,
2,27,16,11,28,33,7,15,21,12,3 D=33,23,27,26,28,24,32,28,36
```

A computer simulation shows that the following polynomial with 43 terms is an invariant in this case:

$$P = eg+fh+gi+hi+ej+hj+ij+ek+fk+jk+fl+il+jl+hm+km+lm+en+in+fo+io+jo+$$

$$mo+gp+jp+kp+np+hq+kq+lq+oq+er+ir+pr+fs+is+js+ms+qs+gt+jt+kt+nt+rt$$

and a quick computation shows that the FE is indeed reduced to zero. This invariant is more complex than previously, it involves Z1, Z2 and a larger number of variables. It works for T-310 for any key, any IV and for an arbitrary number of rounds. In this example a linear invariant also exists: $e + f + g + h + l + m + n + o + p + q + r + s + t$ and both linear and non-linear invariants can be characterized (and computed!) from the fact that our FE reduces to zero. Both linear and non-linear invariants exist here and penetrate deeply inside the cipher. Now because the FE reduces to zero, **no Boolean function can make this cipher setup secure** against round invariant attacks (and also against partitioning attacks in general).