Algebraic Precomputations in Differential and Integral Cryptanalysis

Martin Albrecht[1], Carlos Cid[1], Thomas Dullien[2],
Jean-Charles Faugère[3], and Ludovic Perret[3]

[1] Information Security Group, Royal Holloway, University of London
Egham, Surrey TW20 0EX, United Kingdom
{M.R.Albrecht,carlos.cid}@rhul.ac.uk
[2] Lehrstuhl für Kryptologie und IT-Sicherheit, Ruhr-Universität Bochum
44780 Bochum, Germany
Thomas.Dullien@ruhr-uni-bochum.de
[3] SALSA Project - INRIA (Centre Paris-Rocquencourt)
UPMC, Univ Paris 06 - CNRS, UMR 7606, LIP6
104, avenue du Président Kennedy 75016 Paris, France
jean-charles.faugere@inria.fr, ludovic.perret@lip6.fr

Abstract. Algebraic cryptanalysis is a general tool which permits one to assess the security of a wide range of cryptographic schemes. Algebraic techniques have been successfully applied against a number of multivariate schemes and stream ciphers. Yet, their feasibility against block ciphers remains the source of much speculation. In this context, algebraic techniques have mainly been deployed in order to solve a system of equations arising from the cipher, so far with limited success. In this work we propose a different approach: to use Gröbner basis techniques to compute *structural* features of block ciphers, which may then be used to improve "classical" differential and integral attacks. We illustrate our techniques against the block ciphers PRESENT and KTANTAN32.

1 Introduction

Algebraic cryptanalysis is a general tool which permits one to assess the security of a wide range of cryptographic schemes [21, 20, 19, 17, 18, 23, 24, 22]. As pointed out in the report [13], "*the recent proposal and development of algebraic cryptanalysis is now widely considered an important breakthrough in the analysis of cryptographic primitives*". The basic principle of algebraic cryptanalysis is to model a cryptographic primitive by a set of algebraic equations. The system of equations is constructed in such a way as to have a correspondence between its solutions and some secret information of the cryptographic primitive (for instance, the secret key of a block cipher). The secret can thus be derived by solving the equation system.

Such algebraic techniques have been successfully applied against a number of multivariate schemes and in stream cipher cryptanalysis. On the other hand, their feasibility against block ciphers remains the source of much speculation [15, 14, 19]. One of the reasons is that the sizes of the resulting equation systems are usually beyond the capabilities of current solving algorithms. Furthermore, the complexity estimates are complicated as the algebraic systems are highly structured; a situation where known complexity bounds are no longer valid [4, 2, 3].

While it is currently infeasible to *cryptanalyse* a block cipher by algebraic means alone, these techniques nonetheless have practical applications for block cipher cryptanalysis. For instance, Albrecht and Cid proposed in [1] to combine differential cryptanalysis with algebraic attacks and demonstrated the feasibility of their techniques against reduced-round versions of the block cipher PRESENT [7]. In this approach, the key recovery was approached by solving (or showing lack of solutions in) equation systems that were much simpler than the one arising from the full cipher.

In this paper, we further shift the focus away from attempting to solve the full system of equations. Instead, we use Gröbner basis techniques to compute *structural features* of block ciphers. It turns out that significant information can be gained without solving the equation system in the classical sense. This information, computed via algebraic means, can in turn be potentially used to improve other, "non-algebraic" cryptanalytic methods. We illustrate our techniques by considering the differential cryptanalysis of reduced-round variants of PRESENT and KTANTAN32 [11], as well bit-pattern based integral attacks against PRESENT [31].

The paper is organised as follows. In Section 2 we briefly recall some of the cryptanalytic concepts of relevance to this paper. In Section 3 we provide a high-level description of the main idea behind this work, and briefly describe the ciphers that we use to demonstrate our ideas. These ideas are then applied to improve differential cryptanalysis (Section 4) and integral cryptanalysis (Section 5); experimental results are also presented in both sections.

2 Block Cipher Cryptanalysis

Differential cryptanalysis was formally introduced by Biham and Shamir in [6], and has since been successfully used to attack a wide range of block ciphers. By considering the distribution of output *differences* for the non-linear components of the cipher (e.g. the S-Box), the attacker may be able to construct *differential characteristics* $P^{'} \oplus P^{''} = \Delta P \rightarrow \Delta C = C^{'} \oplus C^{''}$ for a number of rounds N that are valid with non-negligible probability p. A plaintext pair $(P^{'}, P^{''})$ for which the characteristic holds is called a *right pair*, and this behaviour may be used to distinguish the cipher from a random permutation. By modifying the attack, one may use it to potentially recover key information: instead of characteristics for the full N-round cipher, the attacker considers characteristics valid for r rounds only ($r = N - R$, with $R > 0$). The attacker can partially decrypt the known ciphertexts and verify if the result matches the one predicted by the characteristic. Candidate (last round) keys are counted, and as random noise is expected for wrong key guesses, eventually a peak may be observed in the candidate key counters, pointing to the correct round key.

The chances of success and data requirements of differential attacks are typically estimated based on the idea of *signal to noise ratio*. Assume such a differential attack, making use of m plaintext pairs. If the attacker is attempting to recover k subkey bits, it can count the number of occurrences of the possible key values in 2^k counters. If β is the ratio of discarded pairs, based on some criteria to *filter* wrong pairs (e.g. ciphertext difference), and α is the average number of k-bit subkeys suggested by each pair, we expect the counters to contain on average $(m \cdot \alpha \cdot \beta)/2^k$ counts. The right subkey value is counted $m \cdot p$ times due to right pairs, plus the random counts for all the possible subkeys. The signal to noise ratio is therefore:

$$S/N = \frac{m \cdot p}{m \cdot \alpha \cdot \beta / 2^k} = \frac{2^k \cdot p}{\alpha \cdot \beta}.$$

Albrecht and Cid considered in [1] several ideas on how to use algebraic techniques to improve “classical” differential cryptanalysis. In the most promising method, named in [1] *Attack-C*, Gröbner basis computations (applied to the algebraic system arising from the outer rounds in differential cryptanalysis) are used to distinguish right pairs. These Gröbner basis computations could however only be performed during the online phase of the attack. This limitation prevented them from applying their techniques to Present-80 with more than 16 rounds, since computation time would exceed exhaustive key search. In this work, we extend the idea but take a different approach: we only perform Gröbner basis computations in a precomputation (or offline) phase. We show that these computations can also be used to improve the success of differential attacks (for instance, one can increase the signal to noise ratio S/N by using algebraic techniques).

Integral attacks were originally proposed for byte-oriented ciphers such as the AES, and can be viewed as a special form of higher-order differential attacks [27]. In such an attack, one uses sets of plaintexts that satisfy a particular structure (e.g. take on all possible values in one byte and a fixed arbitrary value in all other plaintext bytes). For some ciphers this leads to a predictable feature relating the ciphertexts after a few rounds, which in turn may be used to attack the cipher. In [31] Reza Z'Aba *et al.* extend the notion of integral attacks to bit-oriented ciphers, considering the block ciphers Present, Noekeon and Serpent.

The first work combining algebraic and higher-order differential attacks is [25] by Faugère and Perret. The authors used higher-order differentials to explain the improved runtime of their Gröbner basis algorithms against the Curry and Flurry families of block ciphers [10]. In this work, we also use algebraic techniques to improve integral cryptanalysis: we focus on recovering symbolic representations for relations that must hold on the output after a few rounds, illustrated on an attack against reduced-round variants of Present.

3 Symbolic Precomputation in Block Ciphers

The main idea explored in this paper involves shifting the emphasis of previous algebraic attacks away from attempting to solve an equation system towards using *ideal membership* *as implication*. In others words, instead of trying to solve an equation system arising from the cipher to recover secret key information, we use Gröbner basis methods to compute what a particular input pattern *implies*.

We use a small example to illustrate the main idea. Consider the block cipher PRESENT [7]. Its 4-bit S-Box can be completely described by a set of polynomials that express each output bit in terms of the input bits. Let $X_{i,j}$ and $Y_{i,j}$ denote the j^{th} input and output bits of the i^{th} S-Box, respectively. In differential cryptanalysis, one considers a *pair* of inputs $X'_{i,0}, \dots, X'_{i,3}$ and $X''_{i,0}, \dots, X''_{i,3}$ and the corresponding output bits $Y'_{i,0}, \dots, Y'_{i,3}$ and $Y''_{i,0}, \dots, Y''_{i,3}$. Since the output bits are described as polynomials in the input bits, it is easy to build a set of polynomials describing the parallel application of the S-Box to the pair of input bits. For example, assume the fixed input difference of $(0, 0, 0, 1)$ holds for this S-Box. This can be described algebraically by adding the equations $X'_{i,3} + X''_{i,3} = 1$, $X'_{i,j} + X''_{i,j} = 0$ for $0 \leq j < 3$ to the set. As usual, we add to this system (as well as in all calculations performed in this work) the field equations $X^2_{i,j} + X_{i,j} = 0$ and $Y^2_{i,j} + Y_{i,j} = 0$.

The set of equations now forms a description of the parallel application of the S-Box to two inputs with a fixed input difference. The ideal I spanned by these polynomials contains *all* polynomials that are *implied* by the set. If all polynomials in the generating set of the ideal evaluate to zero, it is clear that any element of I will also evaluate to zero. In particular *any polynomial in the ideal will vanish* if it is assigned values corresponding to the application of the S-Box with a pair of inputs with the above-mentioned input difference.

From a cryptographic point of view, it may be desirable to understand what relations between output bits will hold for a particular input difference. This can be done by considering the polynomials in *the output bits only* that are contained in I. Algebraically, we are trying to find elements in the ideal $I_Y = I \bigcap \mathbb{F}_2[Y'_{i,0}, \dots, Y'_{i,3}, Y''_{i,0}, \dots, Y''_{i,3}]$, where I is the ideal spanned by our original equations.

A *deglex* Gröbner basis G_Y of this ideal can be computed using standard elimination techniques.[1] For this, we can for example set up a block or product ordering where all output variables are lexicographically smaller than any other variable in the system. In addition, we fix the *deglex* ordering among the output variables. Computing the Gröbner basis with respect to such an ordering gives us the Gröbner basis G_Y of I_Y. We note that G_Y will contain the relations of lowest degree of I_Y due to the choice of term ordering. In our example we have:

[1] We refer the reader unfamiliar with Gröbner bases theory and techniques to [5] for the algebraic geometry concepts relevant to the remaining of this section.

$$
\begin{aligned}
G_Y = [&Y'_{i,3} + Y''_{i,3} + 1,\\
&Y'_{i,0} + Y'_{i,2} + Y''_{i,0} + Y''_{i,2} + 1,\\
&Y''_{i,0}Y''_{i,2} + Y'_{i,2} + Y''_{i,0} + Y''_{i,1} + Y''_{i,3},\\
&Y''_{i,0}Y''_{i,1} + Y''_{i,0}Y''_{i,3} + Y''_{i,1}Y''_{i,2} + Y''_{i,2}Y''_{i,3} + Y'_{i,1} + Y''_{i,0} + Y''_{i,1},\\
&Y'_{i,2}Y''_{i,2} + Y''_{i,1}Y''_{i,2} + Y''_{i,2}Y''_{i,3},\\
&Y'_{i,2}Y''_{i,0} + Y''_{i,1}Y''_{i,2} + Y''_{i,2}Y''_{i,3} + Y'_{i,1} + Y'_{i,2} + Y''_{i,0} + Y''_{i,3},\\
&Y'_{i,1}Y''_{i,2} + Y'_{i,2}Y''_{i,1} + Y'_{i,2}Y''_{i,3} + Y''_{i,1}Y''_{i,2} + Y'_{i,1} + Y'_{i,2} + Y''_{i,1},\\
&Y'_{i,1}Y''_{i,1} + Y'_{i,1}Y''_{i,3} + Y''_{i,1}Y''_{i,2} + Y''_{i,1}Y''_{i,3} + Y''_{i,2}Y''_{i,3} + Y''_{i,1},\\
&Y'_{i,1}Y''_{i,0} + Y'_{i,2}Y''_{i,1} + Y'_{i,2}Y''_{i,3} + Y''_{i,0}Y''_{i,3} + Y''_{i,1}Y''_{i,2} + Y''_{i,2}Y''_{i,3} + Y'_{i,1} + Y''_{i,3},\\
&Y'_{i,1}Y'_{i,2} + Y'_{i,2}Y''_{i,3} + Y''_{i,1}Y''_{i,2} + Y''_{i,2}Y''_{i,3} + Y'_{i,2}].
\end{aligned}
$$

There is no other linear or quadratic polynomial $p \in I_Y$ which is not a simple algebraic combination of the polynomials in G_Y. In other words, all *simple* relations involving only the output bits can be derived in a straightforward way from the set G_Y.

In order to formalise this idea, consider a function $\mathcal{E}$ (for example a block cipher), and assume $\mathcal{E}$ can be expressed as a set of algebraic equations F over a finite field $\mathbb{F}$. We can consider d parallel applications of $\mathcal{E}$, with inputs and outputs $P_0, \ldots, P_{d-1}$ and $C_0, \ldots, C_{d-1}$, respectively, and denote the corresponding polynomial systems by F_i. Now assume some property Λ holds on $P_0, \ldots, P_{d-1}$, and can be expressed by a set of algebraic equations F_Λ. A natural question to ask is: how do properties on the input set $P_0, \ldots, P_{d-1}$ affect properties on the output set $C_0, \ldots, C_{d-1}$?

We can simply combine the equation systems into the set $\overline{F} = F_\Lambda \cup (\bigcup_{i=0}^{d-1} F_i)$ and consider the ideal $I = \langle \overline{F} \rangle$. As discussed above, the unique reduced Gröbner basis G_C of the ideal $I_C = I \cap \mathbb{F}[C_0, \ldots, C_{d-1}]$ contains all "relevant" polynomials in $C_0, \ldots, C_{d-1}$, where "relevant" is determined by the term ordering. As soon as we compute the Gröbner basis G_C for the d parallel applications of the function $\mathcal{E}$, we only need to collect the right polynomials from G_C to obtain the properties on the output set $C_0, \ldots, C_{d-1}$ which are implied by Λ.

We note however that for many functions $\mathcal{E}$, computing G_C may be infeasible using current Gröbner basis techniques, implementations and computing power. Thus in practice, we may need to relax some conditions hoping that we still can recover useful information using a similar technique. We provide below a few heuristics and techniques that may still allow recovering *some* relevant equations.

Early Abort. To recover some properties we might not need to compute the complete Gröbner basis; instead we may opt to stop the computation at some degree D.

Replacing Symbols by Constants. It is possible to replace the symbols $P_0, \ldots, P_{d-1}$ by some constants (values) satisfying the constraint Λ which further simplifies the computation. Of course any polynomial recovered from such a computation would have to be checked against other values to verify whether it actually holds in general or with high probability.

Choosing a Different Term Ordering. Instead of computing with respect to an elimination ordering, which is usually more expensive than a degree compatible ordering, we may choose to perform our computations with respect to a more efficient ordering such as *degrevlex*. Used together with **Early Abort**, we have no assurances about the completeness of the recovered system; yet we might still be able to recover some useful information.

3.1 Block Ciphers

We briefly introduce the block ciphers used to demonstrate our techniques.

Present [7] was proposed at CHES 2007 as an ultra-lightweight block cipher, enabling a very compact implementation in hardware, and therefore particularly suitable for RFIDs and similar devices. There are two variants of Present: one for 80-bit keys and one for 128-bit keys, denoted as Present-80 and Present-128 respectively.

Present is an SP-network with a blocksize of 64 bits and both versions have 31 rounds. Each round of the cipher has three layers of operations: `keyAddLayer`, `sBoxLayer` and `pLayer`. The operation `keyAddLayer` is a simple subkey addition to the current state, while the `sBoxLayer` operation consists of 16 parallel applications of a 4-bit S-Box. The operation `pLayer` is a permutation of wires. In this work we consider round-reduced variants of Present denoted Present-Ks-Nr where $Ks \in \{80, 128\}$ and the number of rounds is $0 < Nr \leq 31$.

The designers of Present give a security analysis of their cipher by showing resistance against well-known attacks such as differential and linear cryptanalysis [7]. The best published differential attacks are for 16 rounds of Present-80 [30] and 17 (and possibly up to 19) rounds [1] for Present-128. Results on linear cryptanalysis for up to 26 rounds are available in [12, 26]. Bit-pattern based integral attacks [31] are successful up to seven rounds of Present. A new type of attack, called statistical saturation attack, was proposed in [16] and expected to be applicable to up to 24 rounds of Present.

KTANTAN32 was proposed at CHES 2009 and is the smallest cipher in a family of block ciphers proposed in [11]. It allows a very compact implementation in hardware. It has a blocksize of 32 bits and accepts an 80-bit key. The input is loaded into two registers L_2 and L_1 of 19 and 13 bit length respectively. A round transformation is then applied to these registers 254 times. This round function updates two bits using a quadratic function and performs rotations on the registers. After 254 rounds the content of L_2 and L_1 is output as the ciphertext.

The designers of KTANTAN consider a wide range of attacks in their security argument and show evidence that the cipher is secure against differential, linear, impossible differential, algebraic attacks, as well as some combined attacks. However strong cryptanalytic results against the cipher have recently been proposed in [8].

4 Algebraic Precomputation in Differential Cryptanalysis

In this section we show how to use the techniques discussed previously to improve the differential cryptanalysis of some block ciphers. More specifically, we attempt to increase the chances of success of such an attack by increasing the signal to noise ratio S/N; we illustrate the method against reduce-round versions of PRESENT and KTANTAN32.

4.1 Reducing the Noise

We briefly recall the basic principles of the main attack proposed in [1]. The proposed technique (referred to as *Attack-C*) was used to discard wrong pairs during a differential attack. The attacker would consider the equation systems modelling only the rounds $> r$ (the R outer rounds in the differential attack based on a characteristic valid for r rounds) for each plaintext–ciphertext pair. We denote these equation systems arising from the encryption of P' to C' and P'' to C'', by F'_R and F''_R respectively. The algebraic part of *Attack-C* of [1] consists of a Gröbner basis computation on the polynomial system

$$F = F'_R \cup F''_R \cup \{X'_{r+1,i} + X''_{r+1,i} + \delta X_{r+1,i}\},$$

where the last set refers to the (linear) polynomials arising from the output difference $\delta X_{r+1,i}$ predicted by the characteristic. Whenever the Gröbner basis of the ideal $\langle F \rangle$ is equivalent to $\{1\}$, we know that the system has no solution, and the pair $(P^{'}, P^{''})$ cannot be a right pair (it can thus be discarded). We note however that no strong assurances are given in [1] as to how many pairs are actually discarded by this technique (we refer the reader to [1] for a more detailed description of the proposed algebraic techniques in differential cryptanalysis).

In the present work, we consider the same system of equations as in *Attack-C* but replace the values of C' and C'' by symbols (i.e. variables). By computing a Gröbner basis for the right elimination ordering (cf. Section 3), we can recover relations in the variables C' and C'' that must evaluate to zero whenever the input difference for round $r + 1$ holds. We note that this computation can be done *offline*, as the actual values for the plaintexts and ciphertexts are not required. These equations may be used to improve the quality of the algebraic *filter* used to discard wrong pairs (in other words, to decrease the value of β in the expression of S/N). An estimate about the quality of this filter can calculated by computing the probability that the polynomials obtained evaluate to zero for random values of C' and C''.

4.2 Case Study: PRESENT

We consider the differential from [30] and construct filters for PRESENT reduced to $14+R$ rounds. The same filter also applies to $10+R$, $6+R$ and $2+R$ rounds since the characteristic is iterative with a period of four rounds. The explicit polynomials in this section do not differ for PRESENT-80 and PRESENT-128.

PRESENT 2R. We consider the polynomial ring

$$\begin{aligned} P = \mathbb{F}_2[\, & K_{0,0},\ldots,K_{0,79},\ K_{1,0},\ldots,K_{1,63},\\ & Y'_{1,0},\ldots,Y'_{1,63},\ \ Y''_{1,0},\ldots,Y''_{1,63},\ \ X'_{1,0},\ldots,X'_{1,63},\ \ X''_{1,0},\ldots,X''_{1,63},\\ & \ldots,\qquad\qquad\quad K_{15,0},\ldots,K_{15,3},\\ & Y'_{15,0},\ldots,Y'_{15,63},Y''_{15,0},\ldots,Y''_{15,63},X'_{15,0},\ldots,X'_{15,63},X''_{15,0},\ldots,X''_{15,63},\\ & Y'_{16,0},\ldots,Y'_{16,63},Y''_{16,0},\ldots,Y''_{16,63},X'_{16,0},\ldots,X'_{16,63},X''_{16,0},\ldots,X''_{16,63},\\ & C'_0,\ldots,C'_{63},\qquad C''_0,\ldots,C''_{63}] \end{aligned}$$

and use the following block ordering:

$$\underbrace{K_{0,0},\ldots,X''_{16,63}}_{\text{degrevlex}},\underbrace{C'_0,\ldots,C'_{63},C''_0,\ldots,C''_{63}}_{\text{degrevlex}}.$$

We set up an equation system as in [1], except that the ciphertext bits are symbols (C'_i and C''_i). Then, we compute the Gröbner basis up to degree $D = 3$ using PolyBoRi 0.6.3 [9, 29] with the option `deg_bound=3` and filter out any polynomial that contains non-ciphertext variables.

This computation returns 64 polynomials, 46 of which are linear. Forty linear polynomials are of the form $C'_i + C''_i$ and encode the information that the last round output difference of 10 S-Boxes must be zero (cf. [30]). The remaining 24 polynomials are split into two sets F_0, F_2 of 12 polynomials in 24 variables each; furthermore the sets F_j do not share any variables with each other or the first 40 linear polynomials. The systems F_j are listed in Figure 2 in the Appendix. The probability that all polynomials evaluate to zero for a random point is $\approx 2^{-50.669}$. We recall that Wang's filter from [30] passes with probability $2^{-40}\cdot(5/16)^6 \approx 2^{-50.07}$. Thus, our filter improves upon Wang's by a factor of $2^{0.59} \approx 1.51$.

In order to estimate how close to optimal our filter is, we construct random pairs C', C'' which pass our polynomial filter and notice that for *Attack-C* from [1] mounted using a SAT-solver, roughly every second such pair for Present-80 and 317 out of 512 for Present-128 will pass. Thus, the most precise filter that can be constructed only using the ciphtertext bits and the output difference of round r will accept a pair with probability $\approx 2^{-51.669}$ for Present-80 and with probability $\approx 2^{-51.361}$ for Present-128.

PRESENT 3R. We extend the ring and the block ordering in the obvious way and compute a Gröbner basis with degree bound 3. The computation returns 28 polynomials, 16 of which are linear. The linear polynomials have the form $C'_i + C''_i$ for

$$i \in \{3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, 63\}.$$

The remaining 12 polynomials are quadratic and cubic (cf. Figure 3 in the Appendix). The probability that all polynomials evaluate to zero on a random point is $\approx 2^{-18.296}$. In order to estimate how close to optimal this filter is, we construct random pairs C', C'' which pass this polynomial filter. *Attack-C* using a SAT-solver will accept roughly 6 in 1024 pairs for Present-80 and 9 out of 1024 pairs for Present-128. Thus, we expect an optimal filter – based on the output difference of round r and the ciphertext bits – to pass with probability $\approx 2^{-25.711}$ for PRESENT-80 and $2^{-25.126}$ for PRESENT-128. That is, there is a factor of $2^{7.4} \approx 168$ between our filter and this optimal filter.

PRESENT 4R. We again extend the ring and the block ordering in the obvious way and compute a Gröbner basis with degree bound 3, to we recover

$$(C'_{32+j} + C''_{32+j} + 1)(C'_j + C''_j + 1)(C'_{16+j} + C'_{48+j} + C''_{16+j} + C''_{48+j})$$

for $0 \leq j < 16$. The probability that all polynomials evaluate to zero on a random point is $\approx 2^{-3.082}$.

We verified experimentally that this bound is optimal by using the SAT solver CRYPTOMINISAT [28] on *Attack-C* systems in a 4R attack against PRESENT-80-14. The solver returned an assignment which satisfies the equation system with probability $\approx 2^{-3}$. Thus, we conclude that our filter is optimal among the filters which only consider only the output difference of round r and the ciphertext bits.

4.3 Case Study: KTANTAN

In Table 1 we give our results against KTANTAN32. We used the best characteristic for 42 rounds as provided by the designers and extended it to 71 rounds. The characteristic is valid with probability 2^{-31}. We present results for computation with degree bound $D = 4$ and 5. For each D we give the number of polynomials of degree 1 to 5 found (denoted as $d = *$). In the last column of each experiment we give the approximate probability that all the equations we found evaluate to zero for random values (denoted $\log_2 p$).

4.4 Increasing the Signal

In this section, we consider the problem of increasing the amount of correct data that has to agree with and is always suggested by a right pair. Increasing this value usually has considerable costs attached to it. First, more data needs to be managed and thus usually the counter tables become larger. On average, we can expect each additional bit considered to double the size of these tables. Second, in order to generate more data, more partial decryptions must be performed which in turn increases the computation time. Additionally, the number of key bits that can be trial decrypted may be limited by the number of rounds R we can consider because of the quality of the filter.

In this work we use (non-linear) relations available from the first few rounds instead of the last R rounds. Assume that we have an SP-network, a differential characteristic $\Delta = (\Delta P, \Delta Y_1, \ldots, \Delta Y_r)$ valid for r rounds with probability p, and (P', P'') a right pair for Δ (so that $\Delta P = P' \oplus P''$ and ΔY_r holds for the output of round r). For simplicity, let us assume that only one S-Box is active in round 1, and by abuse of notation, that X'_1,

Table 1. Decreasing the noise for KTANTAN32.

	degree bound = 4						degree bound = 5					
N	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$\log_2 p$	$d=1$	$d=2$	$d=3$	$d=4$	$d=5$	$\log_2 p$
72	32	0	0	0	0	−32.0	32	0	0	0	0	−32.0
74	32	0	0	0	0	−32.0	32	0	0	0	0	−32.0
76	32	0	0	0	0	−32.0	32	0	0	0	0	−32.0
78	31	3	0	0	0	−32.0	31	3	0	0	0	−32.0
80	28	11	0	0	0	−31.4	28	11	0	0	0	−31.4
82	25	23	0	0	0	−31.0	25	23	0	0	0	−31.0
84	20	32	4	8	0	−29.0	20	32	4	32	0	−29.0
86	16	44	19	8	0	−25.7	16	46	23	75	106	< -24
88	12	39	54	96	0	−24.0	12	51	103	371	745	< -23
90	8	41	129	287	0	−23.0	8	42	133	612	1762	< -22
92	4	28	113	285	0	−20.0	4	33	133	743	2646	−20.4
94	1	20	94	244	0	−16.3	1	25	124	662	2345	−18.5
96	0	8	38	96	0	−12.8	0	8	52	287	1264	−14.3
98	0	3	8	29	0	−7.0	0	3	10	46	156	−9.1
100	0	1	3	13	0	−3.7	0	1	3	18	47	−4.6
102	0	0	0	2	0	−0.8	0	0	0	4	9	−0.9
103	0	0	0	1	0	−0.4	0	0	0	2	4	−0.4
104	0	0	0	0	0	0.0	N/A	N/A	N/A	N/A	N/A	N/A

X_1'' and K_0 denote the S-Box input *vectors* corresponding to the plaintext *vectors* P_1', P_1'' (also restricted to the S-Box) and initial key whitening, respectively. Thus we have the relations

$$S(P_1' \oplus K_0) = S(X_1') = Y_1' \text{ and } S(P_1'' \oplus K_0) = S(X_1'') = Y_1''.$$

The S-Box operation S can be described by a (vectorial) Boolean function, expressing each bit of the output Y_1' as a polynomial function (over $\mathbb{F}_2$) on the input bits of X_1' and K_0. If (P', P'') is a right pair, then the polynomial equations arising from the relation $\Delta Y_1 = Y_1' \oplus Y_1'' = S(P_1' \oplus K_0) \oplus S(P_1'' \oplus K_0)$ give us a very simple equation system to solve, with only the key variables $K_{0,j}$ as unknowns (and which do not vanish identically because we are considering nonzero differences). Consequently, right pairs suggest additional information about the key from the first round difference. In particular, if ΔY_1 holds with probability 2^{-b} then we can recover b bits of information about the key, as soon as we have a right pair.

There is no *a priori* reason to restrict this argument (which was considered in [1]) to the first round only. Let Δ, r, P', P'' be as before. We set up two equation systems F' and F'' involving P', C' and P'', C'' respectively and discard any polynomials from the rounds $> s$, where s is small (the discussion above refers to the case $s = 1$). We can then add linear equations as suggested by the characteristic up to s rounds and use this system to potentially recover information about the key from the first s rounds.

In order to avoid the potentially costly Gröbner basis computation for every candidate pair, we replace the vectors of constants P' and P'' by vectors of symbols. Using the idea from Section 3 we can compute polynomials involving only key variables and the newly introduced plaintext variables P' and P''. Assume that we can indeed compute the Gröbner basis, with P' and P'' as symbols, for the first s rounds combined with the linear equations arising from the characteristic. Assume further that the characteristic restricted to s rounds holds with a probability 2^{-b} and that we computed m_s polynomials in the variables K_0, P' and P''. This means that we can recover b bits of information when we evaluate all m_s polynomials, by replacing the variables in P' and P'' by their actual values.

This means that we have b bits of extra information and thus can write $S/N = \frac{2^{k+b} \cdot p}{\alpha \cdot \beta}$ without the overhead of performing any partial decryptions. However, we have to perform m_s polynomial evaluations (where we replace P' and P'' by their actual values) of relatively small low degree polynomials.

Case Study: PRESENT. We consider the first two encryption rounds and the characteristic from [30]. We set up a polynomial ring with two blocks such that the variables P_i and K_i are lexicographically smaller than any other variable. Within the blocks we chose a degree lexicographical term ordering. We set up an equation system covering the first two encryption rounds and added the linear equations suggested by the characteristic. Then, we eliminated all linear leading terms which are not in the variables P_i and K_i and computed a Gröbner basis up to degree five. This computation returned 22 linear and quadratic polynomials (we give the Gröbner basis for these polynomials in Figure 4). This system gives 8 bits of information about the key. Note that the first two rounds of the characteristic is valid with probability 2^{-8}.

Case Study: KTANTAN32. We consider the first 24 rounds of KTANTAN32 and compute the full Gröbner basis. This computation recovers 39 polynomials. We list an excerpt in Figure 1 in the Appendix. As expected we observe that the characteristic also imposes restrictions on the plaintext. These eight equations allow us to recover up to four bits (depending on the value of P'_{19}) of information about the key.

5 Algebraic Precomputation in Integral Cryptanalysis

In [31] *bit-pattern based integral attacks* against up to 7 rounds of Present were proposed. These attacks are based on a 3.5 round distinguisher. The attacker prepares 16 chosen plaintexts which agree in all bit values except the bits at the positions 51, 55, 59, 63. These four bits take all possible values $(0, 0, 0, 0), (0, 0, 0, 1), \dots, (1, 1, 1, 1)$. The authors of [31] show that the input bits to the 4th round are then balanced. That is, the sum of all bits at the same bit position across all 16 encryptions is zero. If $X_{i,j,k}$ denotes the k-th input bit of the j-th round of the i-th encryption, we have that $0 = \sum_{i=0}^{15} X_{i,4,k}$ for $0 \leq k < 64$.

We show below that more algebraic structure can be found. For this purpose we set up an equation system for Present-80-4 for 16 plaintexts of the form given above. We also added all information about relations between encryptions from [31] to the system in algebraic form. These relations are of the form $\sum_{i\in I} X_{i,j,k}$ for $I \subset \{0\ldots,15\}$. These relations would be found by the Gröbner basis algorithm eventually, but adding them directly can speed up the computation. Then we computed a Gröbner basis up to degree 2 only using PolyBoRi. This computation takes about 5 minutes and returns more than 500 linear polynomials in the input variables to the fourth round. All these polynomials relate bits from different encryptions, that is they contain $X_{i,j,k}$ and $X_{i',j',k'}$ with $i \neq i'$. In Figure 5 of the Appendix we provide a selection in order to illustrate the form of these polynomials.

The exact number of subkey bits we can recover using these polynomials varies with the values of the ciphertext bits. On average we can recover 50 subkey bits from the last round key of Present-80-4 using 2^4 chosen plaintexts by performing trial decryptions and comparing the relations between the inputs of the 4th round with the expected relations[2].

The same strategy for finding algebraic relations can be applied to Present-80-5 where we look for polynomials which relate the input variables for the 5th round. Using PolyBoRi with the same options as above, we found 26 linear polynomials. We can represent 12 of them as

$$X_{i,5,k} + X_{i+1,5,k} + X_{6,5,k} + X_{7,5,k} + X_{8,5,k} + X_{9,5,k} + X_{14,5,k} + X_{15,5,k},$$

with $i \in \{0, 2, 4\}$ and $k \in \{51, 55, 59, 63\}$.

Another 12 polynomials are of the form

$$\begin{aligned}&X_{i,5,k} + X_{i,5,k+32} + X_{i+1,5,k} + X_{i+1,5,k+32} + X_{i+8,5,k} + X_{i+8,5,k+32}+\\&X_{i+9,5,k} + X_{i+9,5,k+32} + X_{6,5,k} + X_{6,5,k+32} + X_{7,5,k} + X_{7,5,k+32}+\\&X_{14,5,k} + X_{14,5,k+32} + X_{15,5,k} + X_{15,5,k+32}.\end{aligned}$$

for $i \in \{0, 2, 4\}$ and $k \in \{3, 7, 11, 15\}$.

The remaining two polynomials can be represented by

$$\begin{aligned}&X_{4,5,k} + X_{4,5,k+32} + X_{4,5,k+48} + X_{5,5,k} + X_{5,5,k+32} + X_{5,5,k+48}+\\&X_{6,5,k} + X_{6,5,k+32} + X_{6,5,k+48} + X_{7,5,k} + X_{7,5,k+32} + X_{7,5,k+48}+\\&X_{12,5,k} + X_{12,5,k+32} + X_{12,5,k+48} + X_{13,5,k} + X_{13,5,k+32} + X_{13,5,k+48}+\\&X_{14,5,k} + X_{14,5,k+32} + X_{14,5,k+48} + X_{15,5,k} + X_{15,5,k+32} + X_{15,5,k+48}\end{aligned}$$

for $k \in \{3, 7\}$.

Using the 26 polynomials listed above we expect to recover the round-key for the last round of Present-80-5 using $3 \cdot 2^4$ chosen plaintexts. For each S-box we have to guess the four subkey bits which separate the S-box output from the ciphertext. For each of

[2] We note that considering the full equation system for all rounds instead of only the equations of the 4th round we can recover the full encryption key using 2^4 chosen plaintext by performing a classical algebraic attack. The overall Gröbner basis computation for this task takes only a few minutes but the running time varies between instances.

the S-Boxes $12, 13, 14$ and 15, we have 3 linear equations to filter out wrong guesses on four bits. For each pair of S-boxes $(0, 8)$, $(1, 9)$, $(2, 10)$ and $(3, 11)$ we have again three linear equations to filter out wrong guesses, however this time we are filtering on eight bits. Thus, we need $2 \cdot 2^4$ chosen plaintexts to recover 16 bits and $3 \cdot 2^4$ chosen plaintext to recover 64 subkey bits. In [31], one required $5 \cdot 2^4$ chosen plaintexts. We mention that we can reduce the number of required texts further to 2^4 if we consider the polynomials from Present-80-4 and Present-80-5 together.

We were unable to obtain any polynomials for the input variables of the sixth round. However, just as in [31] we can extend our attack on Present-80-5 to an attack on Present-80-6 by guessing bits in the first round. Our improvements for Present-80-5 translate directly into an improvement for Present-80-6, dropping the data complexity from $2^{22.4}$ to 2^{21} chosen plaintexts (or 2^{20} if we consider the relations arising for the 4th round as well). Similarly, this additional information can be exploited for the Present-128-7 attack from [31].

6 Conclusion

In this work, we have introduced a novel application for algebraic cryptanalysis of block ciphers. We propose a method which can improve "classical" differential and integral cryptanalysis, by applying algebraic tools in a pre-computation phase. As such, we shift the focus from attempting to solve large systems of polynomial equations to recovering symbolic information about the underlying cipher. We note that the use of algebraic techniques in general, and Gröbner basis methods in particular, in block cipher cryptanalysis has received some criticism within the cryptographic community, as it has been often the case that "simpler" techniques can perform favourably in many situations. However in this paper we showed that the rich algebraic structure of Gröbner basis can offer many advantages and may give one a more subtle insight of the cipher structure. This can in turn be used in the cryptanalysis of the cipher. We note that *in principle* our techniques can recover an optimal amount of information and that in most cases considered in this work we were (almost) able to accomplish this. We expect that this approach is applicable to other cryptanalytical techniques and consider applying it as an area of future work.

Acknowledgements

The work described in this paper has been supported in part by the Royal Society grant JP090728 and by the European Commission through the ICT programme under contract ICT-2007-216676 ECRYPT II. The first author was supported by the Royal Holloway Valerie Myerscough Scholarship. The two last authors were also supported in part by the french ANR under the Computer Algebra and Cryptography (CAC) project ANR-09-JCJCJ-0064-01. We would like to thank Stanislav Bulygin for helpful comments. We would also like to thank William Stein for access to his computers (purchased under National Science Foundation Grant No. DMS-0821725).

References

1. Martin Albrecht and Carlos Cid. Algebraic Techniques in Differential Cryptanalysis. In *Fast Software Encryption 2009*, Lecture Notes in Computer Science, Berlin, Heidelberg, New York, 2009. Springer Verlag.
2. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. Complexity of Gröbner basis computation for semi-regular overdetermined sequences over F_2 with solutions in F_2. Technical Report 5049, INRIA, December 2003. Available at `http://www.inria.fr/rrrt/rr-5049.html`.
3. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In *Proc. International Conference on Polynomial System Solving (ICPSS)*, pages 71–75, 2004.
4. Magali Bardet, Jean-Charles Faugère, Bruno Salvy, and Bo-Yin Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Proc. of MEGA 2005, Eighth International Symposium on Effective Methods in Algebraic Geometry*, 2005.
5. Thomas Becker and Volker Weispfenning. *Gröbner Bases - A Computational Approach to Commutative Algebra*. Springer Verlag, Berlin, Heidelberg, New York, 1991.
6. Eli Biham and Adi Shamir. Differential Cryptanalysis of DES-like Cryptosystems. In *Advances in Cryptology — CRYPTO 1990*, volume 537 of *Lecture Notes in Computer Science*, pages 3–72, Berlin, Heidelberg, New York, 1991. Springer Verlag.
7. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 7427 of *Lecture Notes in Computer Science*, pages 450–466, Berlin, Heidelberg, New York, 2007. Springer Verlag. Available at `http://www.crypto.rub.de/imperia/md/content/texte/publications/conferences/present_ches2007.pdf`.
8. Andrey Bogdanov and Christian Rechberger. Generalizing meet-in-the-middle attacks: Cryptanalysis of the lightweight block cipher ktantan. In *Proceedings of Selected Areas in Cryptography 2010*, 2010.
9. Michael Brickenstein and Alexander Dreyer. PolyBoRi: A framework for Gröbner basis computations with Boolean polynomials. In *Electronic Proceedings of MEGA 2007*, 2007. Available at `http://www.ricam.oeaw.ac.at/mega2007/electronic/26.pdf`.
10. Johannes Buchmann, Andrei Pychkine, and Ralf-Philipp Weinmann. Block Ciphers Sensitive to Gröbner Basis Attacks. In *Topics in Cryptology – CT RSA'06*, volume 3860 of *Lecture Notes in Computer Science*, pages 313–331, Berlin, Heidelberg, New York, 2006. Springer Verlag. pre-print available at: `http://eprint.iacr.org/2005/200`.
11. Christophe De Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN — a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer Verlag, 2009.
12. Joo Yeon Cho. Linear cryptanalysis of reduced-round PRESENT. Cryptology ePrint Archive, Report 2009/397, 2009. available at `http://eprint.iacr.org/2009/397`.
13. Carlos Cid. D.STVL.7 algebraic cryptanalysis of symmetric primitives, 2008. available at `http://www.ecrypt.eu.org/ecrypt1/documents/D.STVL.7.pdf`.
14. Carlos Cid and Gaëtan Leurent. An Analysis of the XSL Algorithm. In *Advances in Cryptology — ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Computer Science*, pages 333–352, Berlin, Heidelberg, New York, 2005. Springer Verlag.
15. Carlos Cid, Sean Murphy, and Matthew Robshaw. Small Scale Variants of the AES. In *Fast Software Encryption 2005*, volume 3557 of *Lecture Notes in Computer Science*, pages 145–162, Berlin, Heidelberg, New York, 2005. Springer Verlag. Available at `http://www.isg.rhul.ac.uk/~sean/smallAES-fse05.pdf`.
16. Baudoin Collard and Francois-Xavier Standaert. A Statistical Saturation Attack against the Block Cipher PRESENT. In *Topics in Cryptology – CT-RSA 2009*, pages 195–210, 2009.

17. Nicolas T. Courtois. Fast algebraic attacks on stream ciphers with linear feedback. In *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 176–194, 2003.
18. Nicolas T. Courtois. Higher order correlation attacks, XL algorithm and cryptanalysis of Toyocrypt. In P.J. Lee and C.H. Lim, editors, *Information Security and Cryptology - ICISC 2002: 5th International Conference*, volume 2587 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 2003. Springer Verlag.
19. Nicolas T. Courtois and Gregory V. Bard. Algebraic Cryptanalysis of the Data Encryption Standard. In Steven D. Galbraith, editor, *Cryptography and Coding – 11th IMA International Conference*, volume 4887 of *Lecture Notes in Computer Science*, pages 152–169, Berlin, Heidelberg, New York, 2007. Springer Verlag. pre-print available at `http://eprint.iacr.org/2006/402`.
20. Nicolas T. Courtois and Willi Meier. Algebraic attacks on stream ciphers with linear feedback. In Eli Biham, editor, *Advances in Cryptology — EUROCRYPT 2003*, volume 2656 of *Lecture Notes in Computer Science*, pages 345–359, Berlin, Heidelberg, New York, 2003. Springer Verlag.
21. Nicolas T. Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In Yuliang Zheng, editor, *Advances in Cryptology — ASIACRYPT 2002*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287, Berlin, Heidelberg, New York, 2002. Springer Verlag.
22. Jean-Charles Faugère, Françoise Levy dit Vehel, and Ludovic Perret. Cryptanalysis of MinRank. In *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 280–296, 2008.
23. Jean-Charles Faugère and Antoine Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. In Dan Boneh, editor, *Advances in Cryptology — CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 2003. Springer Verlag.
24. Jean-Charles Faugère and Ludovic Perret. Cryptanalysis of 2R$^-$ schemes. In *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 357–372, 2006.
25. Jean-Charles Faugère and Ludovic Perret. Algebraic cryptanalysis of Curry and Flurry using correlated messages. Cryptology ePrint Archive, Report 2008/402, 2008. available at `http://eprint.iacr.org/2008/402.pdf`.
26. Jorge Nakahara Jr, Pouyan Seperhdad, Bingsheng Zhang, and Meiqin Wang. Linear (hull) and algebraic cryptanalysis of the block cipher PRESENT. In *The 8th International Conference on Cryptoplogy and Network Security - CANS 2009*, 2009.
27. Lars R. Knudsen. Truncated and higher order differentials. In *Fast Software Encryption 1995*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211, Berlin, Heidelberg, New York, 1995. Springer Verlag.
28. Mate Soos, Karsten Nohl, and Claude Castelluccia. Extending SAT solvers to cryptographic problems. In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing - SAT 2009*, volume 5584 of *Lecture Notes in Computer Science*, pages 244–257, Berlin, Heidelberg, New York, 2009. Springer Verlag.
29. William Stein et al. *SAGE Mathematics Software*. The Sage Development Team, 2008. Available at `http://www.sagemath.org`.
30. Meiqin Wang. Differential Cryptanalysis of reduced-round PRESENT. In Serge Vaudenay, editor, *Africacrypt 2008*, volume 5023 of *Lecture Notes in Computer Science*, pages 40–49, Berlin, Heidelberg, New York, 2008. Springer Verlag.
31. Muhammad Reza Z'Aba, Håvard Raddum, Matt Henricksen, and Ed Dawson. Bit-pattern based integral attacks. In Kaisa Nyberg, editor, *Fast Software Encryption 2008*, number 5086 in Lecture Notes In Computer Science, pages 363–381, Berlin, Heidelberg, New York, 2008. Springer Verlag.

A Explicit Polynomials

$$\begin{aligned}
&(P'_{19}+1)(P'_3P'_8+P'_{10}P'_{12}+K_3+K_{53}+P'_7+P'_{18}+P'_{23}),\\
&P'_8P'_{10}P'_{19}+K_8P'_{19}+P'_3P'_8+P'_6P'_{19}+P'_{10}P'_{12}+\\
&P'_{16}P'_{19}+K_3+K_{53}+P'_7+P'_{18}+P'_{19}+P'_{23},\\
&P'_{19}P'_{22}+K_1+K_{11}+P'_6+P'_{11}+P'_{17}+P'_{21}+P'_{26},\\
&P'_{23}P'_{26}+K_{65}+P'_{21}+P'_{25}+P'_{30},\\
&P'_1+1, P'_2, P'_5+1, P'_9+1
\end{aligned}$$

Fig. 1. Polynomials for the first two rounds of KTANTAN32.

$$\begin{aligned}
&(C'_{57+j}+C''_{57+j})(C'_{53+j}+C''_{53+j}+1)(C'_{17+j}+C''_{17+j}),\\
&(C'_{57+j}+C''_{57+j})(C'_{53+j}+C''_{53+j}+1)(C'_{33+j}+C''_{33+j}),\\
&(C'_{57+j}+C''_{57+j}+1)(C'_{25+j}+C''_{25+j}),\\
&(C'_{57+j}+C''_{57+j}+1)(C'_{41+j}+C''_{41+j}),\\
&(C'_{53+j}+C''_{53+j}+1)(C'_{21+j}+C''_{21+j}),\\
&(C'_{53+j}+C''_{53+j}+1)(C'_{37+j}+C''_{37+j}),\\
&(C'_{53+j}+C''_{53+j}+1)(C'_{49+j}+C'_{57+j}+C''_{49+j}+C''_{57+j}+1),\\
&(C'_{49+j}+C''_{49+j}+1)(C'_{17+j}+C''_{17+j}),\\
&(C'_{49+j}+C''_{49+j}+1)(C'_{33+j}+C''_{33+j}),\\
&C'_{1+j}+C'_{33+j}+C'_{49+j}+C''_{1+j}+C''_{33+j}+C''_{49+j},\\
&C'_{5+j}+C'_{37+j}+C'_{53+j}+C''_{5+j}+C''_{37+j}+C''_{53+j},\\
&C'_{9+j}+C'_{41+j}+C'_{57+j}+C''_{9+j}+C''_{41+j}+C''_{57+j},
\end{aligned}$$

Fig. 2. 2R polynomials for PRESENT with $j \in \{0, 2\}$.

$$\begin{gathered}
(C'_{36}+C''_{36})((C'_4+C''_4)(C'_{20}+C'_{52}+C''_{20}+C''_{52}+1)+(C'_{20}+C''_{20}+1)(C'_{52}+C''_{52}+1)),\\
(C'_{37}+C''_{37})((C'_5+C''_5)(C'_{21}+C'_{53}+C''_{21}+C''_{53}+1)+(C'_{21}+C''_{21}+1)(C'_{53}+C''_{53}+1)),\\
(C'_{40}+C''_{40})((C'_8+C''_8)(C'_{24}+C'_{56}+C''_{24}+C''_{56}+1)+(C'_{24}+C''_{24}+1)(C'_{56}+C''_{56}+1)),\\
(C'_{41}+C''_{41})((C'_9+C''_9)(C'_{25}+C'_{57}+C''_{25}+C''_{57}+1)+(C'_{25}+C''_{25}+1)(C'_{57}+C''_{57}+1)),\\
(C'_{45}+C''_{45})((C'_{13}+C''_{13})(C'_{29}+C'_{61}+C''_{29}+C''_{61}+1)+(C'_{29}+C''_{29}+1)(C'_{61}+C''_{61}+1)),\\
(C'_{46}+C''_{46})((C'_{14}+C''_{14})(C'_{30}+C'_{62}+C''_{30}+C''_{62}+1)+(C'_{30}+C''_{30}+1)(C'_{62}+C''_{62}+1)),\\
(C'_{06}+C''_{06})((C'_{22}+C''_{22})(C'_{38}+C'_{54}+C''_{38}+C''_{54}+1)+(C'_{38}+C''_{38}+1)(C'_{54}+C''_{54}+1)),\\
(C'_{10}+C''_{10})((C'_{26}+C''_{26})(C'_{42}+C'_{58}+C''_{42}+C''_{58}+1)+(C'_{42}+C''_{42}+1)(C'_{58}+C''_{58}+1)),\\
(C'_{12}+C''_{12})((C'_{28}+C''_{28})(C'_{44}+C'_{60}+C''_{44}+C''_{60}+1)+(C'_{44}+C''_{44}+1)(C'_{60}+C''_{60}+1)),\\
(C'_{52}+C''_{52}+1)(C'_{20}+C''_{20}+1)(C'_4+C'_{36}+C''_4+C''_{36}),\\
(C'_{60}+C''_{60}+1)(C'_{28}+C''_{28}+1)(C'_{12}+C'_{44}+C''_{12}+C''_{44}),\\
(C'_{10}+C'_{42}+C'_{58}+C''_{10}+C''_{42}+C''_{58})(C'_2+C'_{34}+C'_{50}+C''_2+C''_{34}+C''_{50}).
\end{gathered}$$

Fig. 3. 3R polynomials for Present.

$$\begin{aligned}
&(K_1+P'_1+1)(K_0+K_3+K_{29}+P'_0+P'_3),\\
&(K_2+P'_2)(K_0+K_3+K_{29}+P'_0+P'_3),\\
&K_1K_2+K_1P'_2+K_2P'_1+P'_1P'_2+K_0+K_1+K_3+K_{29}+P'_0+P'_1+P'_3,\\
&(K_9+P'_9+1)(K_8+K_{11}+K_{31}+P'_8+P'_{11}),\\
&(K_{10}+P'_{10})(K_8+K_{11}+K_{31}+P'_8+P'_{11}),\\
&K_9K_{10}+K_9P'_{10}+K_{10}P'_9+P'_9P'_{10}+K_8+K_9+K_{11}+K_{31}+P'_8+P'_9+P'_{11},\\
&(K_{49}+P'_{49}+1)(K_{41}+K_{48}+K_{51}+P'_{48}+P'_{51}),\\
&(K_{50}+P'_{50})(K_{41}+K_{48}+K_{51}+P'_{48}+P'_{51}),\\
&K_{49}K_{50}+K_{49}P'_{50}+K_{50}P'_{49}+P'_{49}P'_{50}+K_{41}+K_{48}+K_{49}+K_{51}+P'_{48}+P'_{49}+P'_{51},\\
&(K_{57}+P'_{57}+1)(K_{43}+K_{56}+K_{59}+P'_{56}+P'_{59}),\\
&(K_{58}+P'_{58})(K_{43}+K_{56}+K_{59}+P'_{56}+P'_{59}),\\
&K_{57}K_{58}+K_{57}P'_{58}+K_{58}P'_{57}+P'_{57}P'_{58}+K_{43}+K_{56}+K_{57}+K_{59}+P'_{56}+P'_{57}+P'_{59},\\
&K_5+K_7+P'_5+P'_7,\\
&K_6+K_7+P'_6+P'_7,\\
&K_{53}+K_{55}+P'_{53}+P'_{55},\\
&K_{54}+K_{55}+P'_{54}+P'_{55}
\end{aligned}$$

Fig. 4. Polynomials for the first two rounds of Present.

$$
\begin{array}{l}
X_{14,4,0} + X_{14,4,32} + X_{14,4,56} + X_{14,4,62} + X_{15,4,0} + X_{15,4,32} + X_{15,4,56} + X_{15,4,62} + 1, \\
X_{14,4,1} + X_{14,4,33} + X_{14,4,49} + X_{15,4,1} + X_{15,4,33} + X_{15,4,49}, \\
X_{14,4,2} + X_{14,4,34} + X_{14,4,58} + X_{14,4,62} + X_{15,4,2} + X_{15,4,34} + X_{15,4,58} + X_{15,4,62}, \\
X_{14,4,3} + X_{14,4,35} + X_{14,4,51} + X_{15,4,3} + X_{15,4,35} + X_{15,4,51}, \\
X_{14,4,4} + X_{14,4,36} + X_{14,4,52} + X_{15,4,4} + X_{15,4,36} + X_{15,4,52}, \\
X_{14,4,5} + X_{14,4,37} + X_{14,4,53} + X_{15,4,5} + X_{15,4,37} + X_{15,4,53}, \\
X_{14,4,6} + X_{14,4,38} + X_{14,4,54} + X_{15,4,6} + X_{15,4,38} + X_{15,4,54}, \\
X_{14,4,7} + X_{14,4,39} + X_{14,4,55} + X_{15,4,7} + X_{15,4,39} + X_{15,4,55}, \\
X_{14,4,8} + X_{14,4,40} + X_{14,4,56} + X_{15,4,8} + X_{15,4,40} + X_{15,4,56}, \\
X_{14,4,9} + X_{14,4,41} + X_{14,4,57} + X_{15,4,9} + X_{15,4,41} + X_{15,4,57}, \\
X_{14,4,10} + X_{14,4,42} + X_{14,4,58} + X_{15,4,10} + X_{15,4,42} + X_{15,4,58}, \\
X_{14,4,11} + X_{14,4,43} + X_{14,4,59} + X_{15,4,11} + X_{15,4,43} + X_{15,4,59}, \\
X_{14,4,12} + X_{14,4,44} + X_{14,4,62} + X_{15,4,12} + X_{15,4,44} + X_{15,4,62} + 1, \\
X_{14,4,13} + X_{14,4,45} + X_{14,4,61} + X_{15,4,13} + X_{15,4,45} + X_{15,4,61}, \\
X_{14,4,14} + X_{14,4,46} + X_{14,4,62} + X_{15,4,14} + X_{15,4,46} + X_{15,4,62}, \\
X_{14,4,15} + X_{14,4,47} + X_{14,4,63} + X_{15,4,15} + X_{15,4,47} + X_{15,4,63}, \\
X_{14,4,48} + X_{14,4,56} + X_{14,4,62} + X_{15,4,48} + X_{15,4,56} + X_{15,4,62} + 1, \\
X_{14,4,49} + X_{14,4,57} + X_{14,4,61} + X_{15,4,49} + X_{15,4,57} + X_{15,4,61}, \\
X_{14,4,50} + X_{14,4,58} + X_{14,4,62} + X_{15,4,50} + X_{15,4,58} + X_{15,4,62}, \\
X_{14,4,51} + X_{14,4,59} + X_{15,4,51} + X_{15,4,59} + 1, \\
X_{14,4,60} + X_{14,4,62} + X_{15,4,60} + X_{15,4,62} + 1, \\
X_{14,4,63} + X_{15,4,63} + 1.
\end{array}
$$

Fig. 5. Polynomials for four round integral attack against Present