On the Existence of low-degree Equations for Algebraic Attacks

Frederik Armknecht[⋆]

Theoretische Informatik
Universität Mannheim
68131 Mannheim, Germany
Armknecht@th.informatik.uni-mannheim.de

Abstract. Algebraic attacks on block ciphers and stream ciphers have gained more and more attention in cryptography. The idea is to express a cipher by a system of equations whose solution reveals the secret key. The complexity of an algebraic attack is closely related to the degree of the equations. Hence, low-degree equations are crucial for algebraic attacks. So far, the existence of low-degree equations for simple combiners, combiners with memory and S-boxes was treated independently.

In this paper, we unify these approaches by reducing them to the same problem: finding low-degree annihilators. This enables a systematic treatment and implies a general criterion for the existence of low-degree equations.

The unification allows to extend former results to all three cases. Therefore, we repeat an algorithm for finding a generating set of all low-degree equations. Additionally, we introduce a new improved version, adapted to specific keystream generators (e.g., for the Bluetooth keystream generator).

Finally, we describe for certain cases an upper and a lower bound for the lowest possible degree. To the best of our knowledge, the upper bound has only been presented in the context of keystream generators before and the lower bound was not published previously.

Keywords: algebraic attacks, block ciphers, keystream generators, low-degree equations, annihilators, algorithms, bounds

1 Introduction

The idea of algebraic attacks is to attack a cipher by solving a system of equations. In this paper, we concentrate on algebraic attacks against block ciphers and LFSR-based keystream generators.

An algebraic attack on a block ciphers was discussed for the first time in [15]. The author reduced the security of DES to solving an (unknown) system of equations. In [9], it was proven that the Advanced Encryption Standard (AES) can be easily described by continuous fractions. In [7], the authors showed that AES can be attacked by solving a system of quadratic equations. The reason is that the only non-linear operation, the S-box, can be described by a system of quadratic Boolean equations. Later, it was shown in [14] that this attack can be improved by using quadratic equations over the finite field $GF(2^8)$. Both attacks are the only attacks known yet which may work for full AES. Although the correctness and the complexity require

[⋆] This work has been supported by grant 620307 of the DFG (German Research Foundation)

further examinations, the existence of a system of low-degree equations is a potential threat.

In [6], algebraic attacks on simple combiners were presented. For each observed keystream bit, an attacker has knowledge of one or several valid equations. If an attacker has enough equations at his disposal, the secret key can be recovered by solving the system of equations. For several keystream generators (e.g. LILI-128, Toyocrypt), the algebraic attacks are the fastest known attacks. Both the required number of known keystream bits and the complexity of the attack are polynomial in the key size but exponential in the degree of the equations. Therefore, the availability of low-degree equations is crucial for an efficient attack.

For several reasons, the extension of the attack to LFSR-based keystream generators with additional memory (e.g., the Bluetooth keystream generator) was not apparent. In [1], this question was finally solved. The authors showed that any LFSR-based keystream generator can be expressed by system of equations with a bounded degree. Also here, the effort grows exponentially with the degree.

An important improvement of algebraic attacks on LFSR-based keystream generators are fast algebraic attacks [4] which have been further examined in [2] and [12]. In fast algebraic attacks, some special properties of the system of equations are exploited to reduce the computation complexity. Nevertheless, the complexity remains exponential in the degree of the equations.

The three cases described above have in common that they require equations of low degree. For each of the cases, different methods resp. criteria for deciding the existence of low-degree equations were developed. In this paper, we unify these approaches into one theory. We show that the question can be reduced to the existence of low-degree annihilators of certain functions. This allows for the first time a systematic treatment of the existence of low-degree equations for several important cases. In particular, results and algorithms developed for one case can now be transferred to the other ones. Another advantage is that the theory is also applicable for ciphers which work over other fields than $GF(2)$.

Additionally, we present two different algorithms which compute a generating set of all low-degree equations (if any exist). The first one works in general and was discussed before in several papers (but never in its generality). The second one is adapted to certain LFSR-based keystream generators (e.g., the Bluetooth keystream generator) to avoid unnecessary computations. As far as we know, this improvement was unpublished before.

The practicability of these algorithms is limited to some cases. Hence, efficient criteria to decide the existence of low-degree equations are desirable. For certain cases we present an upper and a lower bound for the minimal degree. The upper bound occurred before in the context of LFSR-based keystream generators [6, 1]. To the best of our knowledge, the lower bound is new.

The paper is organized as follows: in section 2, we provide some facts about Boolean functions and annihilators. In section 3, algebraic attacks against combiners without memory, section 3.1, combiners with memory, section 3.2, and S-boxes used in block ciphers, section 3.3, are described. We show that in all three cases, the introduction of characteristic functions allows a general treatment of the existence of low-degree equations. We derive a general criterion over arbitrary fields in section 3.4. In section 4, we describe two algorithms for computing all low-degree equations. While the first one is applicable in general, the second one is adapted to certain keystream generators to avoid specific redundant operations. Both algorithms can be used to decide the existence of low-degree equations. However, their practical feasibility is limited. Therefore, 5 we present in section an upper and a lower bound for the minimal degree of the equations. Section 6 concludes the paper.

2 Basics on Boolean Functions

Definition 1. *For an integer $n \geq 1$, we define by $\mathbb{B}_n$ the set of all Boolean functions $f : \{0,1\}^n \to \{0,1\}$. For $f \in \mathbb{B}_n$ we define the sets $0_f := \{x \in \{0,1\}^n \mid f(x) = 0\}$ and $1_f := \{x \in \{0,1\}^n \mid f(x) = 1\}$.*

Obviously, it is $1_f \dot{\cup} 0_f = \{0,1\}^n$ and $1_f \cap 1_g = 1_{f \cdot g}$ for all $f, g \in \mathbb{B}_n$.

Definition 2. *Let $f \in \mathbb{B}_n$ be a Boolean function. We call $g \in \mathbb{B}_n$ an annihilator of f if $f \cdot g \equiv 0$. Further on, we define $An(f) := \{g \in \mathbb{B}_n \mid f \cdot g \equiv 0\}$.*

, the set of the annihilators of f.

Proposition 1. *Let $f, g \in \mathbb{B}_n$. Then g is an annihilator of f if and only if $1_f \subseteq 0_g$.*

Proof.

$$\Big[f(x) \cdot g(x) \equiv 0\Big] \Leftrightarrow \Big[\forall x : \ f(x) = 1 \Rightarrow g(x) = 0\Big] \Leftrightarrow \Big[1_f \subseteq 0_g\Big]$$

Proposition 2. *Let $f, g \in \mathbb{B}_n$ be arbitrary. Then g is a multiple of f if and only if $1_g \subseteq 1_f$.*

Proof. "$\Rightarrow$" Assume that g is a multiple of f. I.e., there exists $h \in \mathbb{B}_n$ such that $f \cdot h = g$. Then $1_g = 1_{f \cdot h} = 1_f \cap 1_h \subseteq 1_f$.
"$\Leftarrow$" Assume now that $1_g \subseteq 1_f$. Then $1_g = 1_g \cap 1_f = 1_{f \cdot g}$. Hence, it is $g = g \cdot f$ and therefore g is a multiple of f.

Proposition 3. *Let $f \in \mathbb{B}_n$. Then $An(f) = \{(f \oplus 1) \cdot g \mid g \in \mathbb{B}_n\}$. I.e., each annihilator of f is a multiple of $f \oplus 1$.*

Proof. Due to proposition 1, g is an annihilator of f if and only if $1_f \subseteq 0_g$. Obviously, this is equivalent to $0_{f \oplus 1} \subseteq 0_g \Leftrightarrow 1_g \subseteq 1_{f \oplus 1}$. By proposition 2, this is equivalent to $(f \oplus 1) \mid g$.

An alternative proof has been given in [13].

Definition 3. *We define for a set $\mathcal{F} \subseteq \mathbb{B}_n$, $\mathcal{F} \neq \{0\}$, its minimal degree by*

$$\text{mindeg}(\mathcal{F}) := \min\{\deg(f) \mid f \in \mathcal{F}, \ f \neq 0\}.$$

3 A general criterion for low-degree equations

In this section, we describe algebraic attacks on combiners without memory, on combiners with memory and on block ciphers as the AES. In all three cases the existence of low-degree equations are a necessary precondition for algebraic attacks. We will show that by introducing the notion of characteristic functions it is possible to derive a common criterion for the existence of low-degree equations.

3.1 Simple combiner

We define a simple combiner to be a keystream generator which consists of the following components:

- An internal state $S \in \{0,1\}^n$
- A linear update function $L : \{0,1\}^n \to \{0,1\}^n$
- A projection $\pi : \{0,1\}^n \to \{0,1\}^k$
- An output function $f : \{0,1\}^k \to \{0,1\}$

Let the initial state S_0 be the secret key K. Then for each clock t, the keystream bit z_t is computed by $f(\pi(S_t)) = z_t$ and the internal state S_t is updated to $S_{t+1} := L(S_t)$ which is equal to $L^{t+1}(K)$.

The first step in an algebraic attack is to describe the secret key K by a system of equations in dependence of the observed keystream. This requires the knowledge of two functions g_0 and g_1 such that for all clocks $t \geq 0$ the following is true:

$$\begin{array}{l} \text{If } \mathrm{z_t} = 0 \Rightarrow g_0(\pi(L^t(K))) = 0 \\ \text{If } \mathrm{z_t} = 1 \Rightarrow g_1(\pi(L^t(K))) = 0 \end{array} \tag{1}$$

Actually, it suffices if at least one of the two functions is $\not\equiv 0$. In this case, One example is $g_0 = f$ and $g_1 = f \oplus 1$. Depending on the values of the observed keystream $z_0, z_1, \ldots$, an attacker can now set up the following system of equations:

$$\begin{array}{ll} g_{z_0}(\pi(K)) & = 0 \\ g_{z_1}(\pi(L(K))) & = 0 \\ g_{z_2}(\pi(L^2(K))) & = 0 \\ & \vdots \end{array} \tag{2}$$

Of course, if multiple functions $g_0, g_0', \ldots$ resp. $g_1, g_1', \ldots$ are known fulfilling the condition above, then all of them can be inserted in the system of equations. If an attacker has enough equations at his disposal, he can recover the secret key K by solving the system of equations. This is the idea of algebraic attacks. To find the solution, several algorithms were discussed (e.g., Linearization [6], XL, XSL [7], Gröbner bases [8]). All have in common that they benefit from a low degree of (2). Observe that the linearity of L implies that the degree of (2) is bounded by $\max\{\deg(g_0), \deg(g_1)\}$.[1] I.e., the lower the degree of g_0 resp. g_1, the faster the

[1] If the degrees are different, an attacker may use only those functions in (2) with the lower degree.

algebraic attack. Therefore, it is important for algebraic attacks to be able to decide whether functions g_0 resp. g_1 of low degree exist or not.

Algebraic attacks on simple combiners have been introduced in [6]. The authors proposed three different scenarios (S3a, S3b, S3c) under which functions g_0 or g_1 of low degree exist. In [13], it was showed that these scenarios can be reduced the following general criterion: Low-degree functions g_0 resp. g_1 do exist if and only f resp. $f \oplus 1$ has annihilators of low-degree.

Now we will embed this setting into a new description which implies immediately the criterion of [13]. This description has the advantage that it is extendable to other situations (e.g., combiners with memory and S-boxes). To motivate our approach, we rewrite (1) to

$$\begin{array}{l}\text{If f(X)} = 0 \Rightarrow g_0(X) = 0 \\ \text{If f(X)} = 1 \Rightarrow g_1(X) = 0\end{array} \tag{3}$$

I.e., the expressions on the left side characterizes all inputs for which the functions on the right side must be equal to zero.[2] We generalize this idea by introducing the notion of characteristic functions:

Definition 4. *Let $f \in \mathbb{B}_n$ be the output function of a simple combiner. For $z \in \{0,1\}$, we define the characteristic function $\mathcal{C}_z \in \mathbb{B}_n$ by*

$$\forall X \in \{0,1\}^n : \quad \mathcal{C}_z(X) = 1 \ \Leftrightarrow f(X) = z.$$

In the case of simple combiners, it is $\mathcal{C}_0 = f$ and $\mathcal{C}_1 = f \oplus 1$. In the case of combiners with memory and S-boxes, the form of the characteristic functions will not be that obvious. Now, we can reformulate (3) to

$$\begin{array}{l}\text{If } \mathcal{C}_0\text{(X)} = 1 \Rightarrow g_0(X) = 0 \\ \text{If } \mathcal{C}_1\text{(X)} = 1 \Rightarrow g_1(X) = 0\end{array} \tag{4}$$

We use now (4) to give a definition of the desired functions g_0 resp. g_1:

Definition 5. *Let $\mathcal{C}_0, \mathcal{C}_1 \in \mathbb{B}_n$ be the characteristic functions of a simple combiner. For $z \in \{0,1\}$, we call a function $g_z \in \mathbb{B}_n$ a z-function if*

$$\forall X \in \{0,1\}^n : \quad \mathcal{C}_z(X) = 1 \Rightarrow g_z(X) = 0.$$

The following proposition is the main result of this section.

Proposition 4. *Let $\mathcal{C}_0, \mathcal{C}_1 \in \mathbb{B}_n$ be the characteristic functions of a simple combiner. Then the following is true:*

1. *$g \in \mathbb{B}_n$ is a z-function for $z \in \{0,1\}$ if and only if $g \in An(\mathcal{C}_z)$*
2. *If $g \in \mathbb{B}_n$, $g \not\equiv 0$, is a z-function, then* $\deg(g) \geq \text{mindeg}(An(\mathcal{C}_z))$

Proof. We fix $z \in \{0,1\}$. By definition, $g \in \mathbb{B}_n$ is a z-function if and only if $\mathcal{C}_z(X) = 1$ implies $g_z(X) = 0$ for all $X \in \{0,1\}^n$. This is equivalent to $1_{\mathcal{C}_z} \subseteq 0_{g_z}$. By 1, this is the case if and only if $g_z \in An(\mathcal{C}_z)$. This proves the fist proposition. The second one is obvious from the definition of mindeg.

[2] Otherwise, (1) would not be true in general.

The proposition says that low-degree equations for an algebraic attack exist if and only if $\mathcal{C}_0$ or $\mathcal{C}_1$ has non-trivial low-degree annihilators. In this case, it is $\mathcal{C}_0 = f$ and $\mathcal{C}_1 = f \oplus 1$. Together with proposition 3, this is equivalent to that $f \oplus 1$ or f has low-degree multiples. The same criterion was derived in [13], but in a different way.

Remark 1. Proposition 4 makes only statements about the degree of equations of the type $g(\pi(L^t(K))) = 0$. Therefore, equations of the type

$$h(\pi(L^t(K)), \pi(L^{t+1}(K)), \ldots, \pi(L^{t+l}(K))) = 0$$

may exist with a degree lower than $\text{mindeg}(An(f))$ and $\text{mindeg}(An(f \oplus 1))$ (see for example the clever ideas used in fast algebraic attacks [4].) But until now, it is unclear how to find such equations without using z-functions.

3.2 Combiners with memory

A combiner with memory consists of the following components:

- An internal state $\tilde{S} = (S, M) \in \{0,1\}^n \times \{0,1\}^l$
- A projection $\pi : \{0,1\}^n \to \{0,1\}^k$
- A linear update function $L : \{0,1\}^n \to \{0,1\}^n$
- A non-linear update function $\Psi : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}^l$
- An output function $f : \{0,1\}^k \times \{0,1\}^l \to \{0,1\}$

At each clock t, the keystream bit z_t is computed by $z_t = f(\pi(S_t), M_t)$. The internal state $\tilde{S}_t = (S_t, M_t)$ is updated to $\tilde{S}_{t+1} := (L(S_t), \Psi(\pi(S_t), M_t))$. Again, an attackers goal is to recover $K = S_0$.

If an attacker uses the equations $f(\pi(L^t(K), M_t) = z_t$ for an algebraic attack, he faces two problems. Either he keeps the expressions M_t in the equations or he expresses $f(\pi(L^t(K), M_t)$ by $f'(\pi(K), \ldots, \pi(L^t(K)), M_0)$. In the first case, the number of unknowns increases with the number of equations which makes the system of equations unsolvable. In the second case, the degree can go up arbitrarily high. Hence, for an efficient algebraic attacks, a different approach is necessary.

Such an approach was introduced in [1]. The authors proved that functions g_Z for all $Z \in \{0,1\}^{l+1}$ exist with at least one function $\not\equiv 0$ such that the following equation is true

$$\text{If } (\mathrm{z_t}, \ldots, \mathrm{z_{t+l}}) = \mathrm{Z} \ \Rightarrow \mathrm{g_Z}(\pi(\mathrm{L^t(K)}), \ldots, \pi(\mathrm{L^{t+l}(K)})) = 0 \qquad (5)$$

The similarity to (1) is obvious. Again, the degree of equation (5) is bounded by $\deg(g)$. Therefore, it is possible to pursuit the same strategy as described in section 3.1. First, an attacker sets up a system of equations using (5). Then, he recovers the secret key K by computing the solution. Therefore, he can apply the same methods as described in section 3.1. For several combiners with memory (e.g., the Bluetooth keystream generator), the algebraic attack was faster than all previously known attacks. The efficiency of the attack depends again on the degrees in (5). We will show now that the criterion for low-degree equations from section 3.1 can be easily extended to this case. For this purpose, we adapt the definitions of characteristic functions and z-functions to this situation:

Definition 6. *For combiner with memory, an integer $r \geq 1$ and a value $Z = (Z_1, \ldots, Z_r) \in \{0,1\}^r$, we define the characteristic function $\mathcal{C}_Z : (\{0,1\}^k)^r \rightarrow \{0,1\}$ by $\mathcal{C}_Z((X_1, \ldots, X_r)) = 1$ if and only if $S_1, \ldots, S_r \in \{0,1\}^n$ and $M_1, \ldots, M_r \in \{0,1\}^l$ exist such that the following conditions are fulfilled:*

1. $X_i = \pi(S_i), i = 1, \ldots, r$
2. $S_{i+1} = L(S_i), i = 1, \ldots, r-1$
3. $z_i = f(\pi(S_i), M_i), i = 1, \ldots, r$
4. $M_{i+1} = \Psi(S_i, M_i), i = 1, \ldots, r-1$

In other words, $\mathcal{C}_Z(X) = 1$ if and only if $(X_1, \ldots, X_r)$ is a partial assignment of the secret key which does not contradict the observed part Z of the keystream. If r is too small, then $\mathcal{C}_Z \equiv 1$ for all $Z \in \{0,1\}^r$. In [1], the authors showed that for $r = l + 1$ at least one $Z \in \{0,1\}^r$ exists such that $\mathcal{C}_Z \not\equiv 1$. In fact, the set $NCrit(Z)$ defined in [1] is equal to the set $1_{\mathcal{C}_Z}$. Similar to the case of simple combiners, we introduce the notion of Z-functions by

Definition 7. *For a combiner with memory, an integer $r \geq 1$ and a value $Z = (Z_1, \ldots, Z_r) \in \{0,1\}^r$, we say that a function $g_Z \in B_{k \cdot r}$ is a Z-function if the following holds:*

$$\forall X = (X_1, \ldots, X_r) \in \{0,1\}^{k \cdot r} : \text{ If } \mathrm{C_Z(X) = 1} \ \Rightarrow \mathrm{g_Z(X) = 0}$$

I.e., the functions described in (5) are Z-functions. The proof of the following proposition is almost the same as the proof of proposition 4:

Proposition 5. *Let $r \geq 1$ and $\mathcal{C}_Z$ for $Z \in \{0,1\}^r$ be the characteristic functions of a combiner with memory. Then the following is true:*

1. *$g \in \mathbb{B}_{k \cdot r}$ is a Z-function for $Z \in \{0,1\}^r$ if and only if $g \in An(\mathcal{C}_Z)$*
2. *If $g \in \mathbb{B}_{k \cdot r}$, $g \not\equiv 0$, is a Z-function, then* $\deg(g) \geq \operatorname{mindeg}(An(\mathcal{C}_Z))$

Remark 2. There is one important difference between the case of simple combiners and combiners with memory. In the first case, we have only two characteristic functions: $\mathcal{C}_0$ and $\mathcal{C}_1 = \mathcal{C}_0 \oplus 1$. If $\operatorname{mindeg}(An(\mathcal{C}_0))$ is high but $\operatorname{mindeg}(An(\mathcal{C}_0 \oplus 1))$ is low, then the low-degree annihilators of $\mathcal{C}_0 \oplus 1$ can be used for an algebraic attack. The reason is that $\mathcal{C}_0 \oplus 1$ is equal to the **other** characteristic function $\mathcal{C}_1$.

If in the case of a combiner with memory, the values $\operatorname{mindeg}(An(\mathcal{C}_Z))$ are high for all $Z \in \{0,1\}^r$, then all equations of the form

$$g(\pi(L^t(K)), \ldots, \pi(L^{t+r-1}(K))) = 0$$

have a high degree. The values of $\operatorname{mindeg}(An(\mathcal{C}_Z \oplus 1))$ have absolutely **no** influence on this fact.[3]

[3] Unless, it is $\mathcal{C}_Z = \mathcal{C}_{Z'} \oplus 1$ for two different Z and Z'.

Remark 3. Assume that for a fixed r, the characteristic functions $\mathcal{C}_Z$ are known for all $Z \in \{0,1\}^r$. This knowledge provides information about $\mathcal{C}_{\hat{Z}}$ for all $\hat{Z} \in \{0,1\}^{r+1}$. Let $\hat{Z} = (Z_1, \ldots, Z_{r+1}) \in \{0,1\}^{r+1}$ be arbitrary. Then $(X_1, \ldots, X_{r+1}) \in 1_{\mathcal{C}_{(Z_1,\ldots,Z_{r+1})}}$ implies that $(X_1, \ldots, X_r) \in 1_{\mathcal{C}_{(Z_1,\ldots,Z_r)}}$ **and** $(X_2, \ldots, X_{r+1}) \in 1_{\mathcal{C}_{(Z_2,\ldots,Z_{r+1})}}$. Hence, it is $1_{\mathcal{C}_{(Z_1,\ldots,Z_{r+1})}} \subseteq 1_{\mathcal{C}_{(Z_1,\ldots,Z_r)} \cdot \mathcal{C}_{(Z_2,\ldots,Z_{r+1})}}$ and therefore $\mathcal{C}_{(Z_1,\ldots,Z_{r+1})}$ is a multiple of $\mathcal{C}_{(Z_1,\ldots,Z_r)} \cdot \mathcal{C}_{(Z_2,\ldots,Z_{r+1})}$.

3.3 S-boxes

An S-box is a Boolean mapping $S : \{0,1\}^n \to \{0,1\}^m$. In [7], the authors proposed an algebraic attack on the block cipher AES. The attack was based on the observation that the AES S-box $S : \{0,1\}^8 \to \{0,1\}^8$ can be expressed by a system of quadratic equations. I.e., multiple functions $g \in \mathbb{B}_{16}$ of degree 2 exist such that

$$\text{If S(X) = Y} \Rightarrow \text{g(X, Y) = 0} \tag{6}$$

They used this system of quadratic equations to derive an algebraic attack on the AES. Although the attack in [7] is still controversial discussed, the existence of low-degree equations for the S-box is a potential threat which should not be ignored. We will see that the existence of low-degree equations is again equivalent to the existence of low-degree annihilators of an appropriate characteristic function.

We observe the similarities between (6) and (1) and (5). Hence, the following definitions and proposition are obvious:

Definition 8. *Let $S : \{0,1\}^n \to \{0,1\}^m$ be an S-box. The corresponding characteristic function $\mathcal{C}_S : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ is defined by*

$$\mathcal{C}_S(X, Y) = 1 \Leftrightarrow S(X) = Y.$$

Definition 9. *Let $S : \{0,1\}^n \to \{0,1\}^m$ be an S-box and $\mathcal{C}_S$ its characteristic function. We call $g : \{0,1\}^{n+m} \to \{0,1\}$ an S-function if*

$$\mathcal{C}_S(X, Y) = 1 \Rightarrow g(X, Y) = 0.$$

Proposition 6. *et $S : \{0,1\}^n \to \{0,1\}^m$ be an S-box and $\mathcal{C}_S$ its characteristic function. Then*

1. *$g \in \mathbb{B}_{n+m}$ is an S-function if and only if $g \in An(\mathcal{C}_S)$*
2. *If $g \in \mathbb{B}_{n+m}$, $g \not\equiv 0$, is an S-function, then* $\deg(g) \geq \text{mindeg}(An(\mathcal{C}_S))$

The observation made in [7] can be reformulated to: The characteristic function $\mathcal{C}_S$ of the S-box used in the AES has quadratic annihilators.

3.4 A general criterion

In the three previous sections we have seen that three presumably different situations can be expressed by the same theory. This allows to set up the following general criterion for the existence of low-degree equations:

General criterion for low-degree equations
Let $C \in \mathbb{B}_n$ be a characteristic function as described in one of the sections 3.1, 3.2 and 3.3. Then, an equation of degree $\leq d$ for an algebraic attack exist if and only if $\mathcal{C}$ has an annihilator of degree $\leq d$.

4 Computing low-degree annihilators

In this section we present two algorithms to compute a generating set for all low-degree Boolean annihilators. The first one, *Algorithm 1* in section 4.1, is applicable in general. The second one, *Algorithm 2* described in 4.2, is adapted to certain keystream generators as E_0 or the summation generator where the output function f and the update function Ψ are invariant under certain permutations. This can be used to avoid redundant computations occurring *Algorithm 1.*

4.1 A general algorithm

We describe now a general algorithm which among other things can compute a generating set for all low-degree annihilators. The algorithm is described for Boolean functions. The reason is that it makes it easier to discuss the adapted version in the next section. A general algorithm for functions over arbitrary fields can be found in the proof of theorem 1.

For a set $\mathcal{F}$ of functions, $< \mathcal{F} >$ denotes its linear span.

Algorithm 1

```
Given: A set $\mathcal{F} \subseteq \cdot\;_n$, $\mathcal{X} \subseteq \{0,1\}^n$
Task: A set $\mathcal{F}' \subseteq \cdot\;_n$ such that $< \mathcal{F}' >$ is the vector space of all functions $f \in < \mathcal{F} >$ with
    $f(X) = 0$ for all $X \in \mathcal{X}$
Algorithm:
    Set $\mathcal{F}_0 := \mathcal{F}$ and $\mathcal{X} = \{X_1, \ldots, X_s\}$
    For i from 1 to $s$ do
        Set $\mathcal{F}_i^0 := \{f | f \in \mathcal{F}_{i-1}, f(X_i) = 0\}$ and $\mathcal{F}_i^1 := \{f | f \in \mathcal{F}_{i-1}, f(X_i) = 1\}$
        If $|\mathcal{F}_i^1| > 1$ then
            Choose $\tilde{f} \in \mathcal{F}_i^1$ and set $\mathcal{G}_i^0 := \{\tilde{f} \oplus f | f \in \mathcal{F}_i^1 \setminus \{\tilde{f}\}\}$
        else $\mathcal{G}_i^0 := \emptyset$
    $\mathcal{F}_i := \mathcal{F}_i^0 \cup \mathcal{G}_i^0$
Output: The set $\mathcal{F}_s$
```

The correctness is given by proposition 10 in appendix A.

Remark 4. If we set $\mathcal{X} := 1_{\mathcal{C}}$ and $\mathcal{F}$ to be the set of all monomials in $\mathbb{B}_n$ of degree $\leq d$, then *Algorithm 1* computes a generating set for all annihilators of degree $\leq d$.

Although *Algorithm 1* is described explicitly for Boolean functions, it can be easily reformulated to solving a system of linear equations. This implies a general algorithm over arbitrary fields (see the proof of theorem 1 in section 5.1). The general algorithm is not new. Probably, the quadratic equations for the AES S-box in [7] were derived with the same method. In [3], the authors used it to find quadratic equations for

S-boxes of other block ciphers. In [1], it was described to compute a basis for all ad-hoc equations for combiners with memory. Later, in [13], the same algorithm and several improvements were presented to compute equations for combiners without memory. With the theory of section 3, this coincidence is not a surprise, but rather a consequence of the similarity of all three cases.

4.2 An adapted algorithm

The theory in this section was motivated by the following fact: for some keystream generators (e.g., E_0 or the summation generator), the output function f and the update function Ψ depend only on the Hamming weight of $\pi(S_t)$. I.e., the set $1_{\mathcal{C}_Z}$ is invariant under certain permutations. We will discuss in this section how this property can be exploited to avoid unnecessary steps in *Algorithm 1*.

Definition 10. *Let $\mathcal{S}_n$ be the group of permutations on $\{1,\ldots,n\}$. For $X=(x_1,\ldots,x_n)$ and $\sigma\in\mathcal{S}_n$ we define $\sigma(X):=(x_{\sigma(1)},\ldots,x_{\sigma(r)})$ and $[X]:=\{\sigma(X)\mid\sigma\in S\}$. Further on, for $f\in\mathbb{B}_n$ we define $\sigma(f)=\sigma(f)(x_1,\ldots,x_n):=f(x_{\sigma(1)},\ldots,x_{\sigma(n)})$.*
Let $S\subseteq\mathcal{S}_n$. We say that a set $\mathcal{X}\subseteq\{0,1\}^n$ is S-invariant if $\sigma(x)\in\mathcal{X}$ for all $\sigma\in S$ and all $x\in\mathcal{X}$. Consequently, we say that a function $f\in\mathbb{B}_n$ is S-invariant if $\sigma(f)=f$ for all $\sigma\in S$.

Proposition 7. *Let $S\subseteq\mathcal{S}_n$ and $<S>$ denote the subgroup of $\mathcal{S}_n$ generated by the elements of S. A set $\mathcal{X}\subseteq\{0,1\}^n$ is S-invariant if and only if it is $<S>$-invariant.*

Proof. Because of $S\subseteq<S>$, a $<S>$-invariant set is always S-invariant also.
Let $\mathcal{X}$ denote now an S-invariant set. We have to show that for all $\sigma,\tilde{\sigma}\in S$ it is $(\sigma\circ\tilde{\sigma})(\mathcal{X})\subseteq\mathcal{X}$ and $\sigma^{-1}(\mathcal{X})\subseteq\mathcal{X}$. The first proposition is obvious because of $\sigma(\mathcal{X})\subseteq\mathcal{X}$ and $\tilde{\sigma}(\mathcal{X})\subseteq\mathcal{X}$ by assumption. The reason for the second proposition is that $|\mathcal{S}_n|$ is finite and therefore σ^{-1} can be expressed by σ^m for an appropriate m.

Proposition 8. *For $S\subseteq\mathcal{S}_n$ a function $f\in\mathbb{B}_n$ is S-invariant if and only if $1_f\subseteq\{0,1\}^n$ is S-invariant.*

Proof. Obviously, 1_f is S invariant if and only if 1_f **and** 0_f are both S-invariant. Otherwise there would exist a $x\in 0_f$ and $\sigma\in S$ with $\tilde{x}:=\sigma(x)\in 1_f$. But this implies $\sigma^{-1}(\tilde{x})\in 0_f$ with $\sigma^{-1}\in<S>$ and $\tilde{x}\in 1_f$. Therefore 1_f is not $<S>$-invariant. With proposition 7, this contradicts that 1_f is S-invariant.
The S-invariance of 1_f and 0_f is equivalent to $f(x)=f(\sigma(x))=\sigma(f)(x)$ for all $x\in\{0,1\}^n$ and $\sigma\in S$. This is exactly the definition of the S-invariance of f.

Proposition 9. *Consider a combiner with memory as defined in section 3.2. Assume that for a set $S\subseteq\mathcal{S}_k$ the following properties hold:*

(I) For any choice of $X_1,\ldots,X_r\in\{0,1\}^k$ do $S_1,\ldots,S_r\in\{0,1\}^n$ exist such that $\pi(S_i)=X_i$ and $S_{i+1}=L(S_i)$.
(II) It is $f(X,M)=f(\sigma(X),M)$ for all $X\in\{0,1\}^k$, $M\in\{0,1\}^l$ and $\sigma\in S$.

(III) It is $\Psi(X, M) = \Psi(\sigma(X), M)$ for all $X \in \{0,1\}^k$, $M \in \{0,1\}^l$ and $\sigma \in S$.

Then $\mathcal{C}_Z$ is $S^r = S \times \ldots \times S$ invariant for all $Z \in \{0,1\}^r$.

Proof. We fix $Z \in \{0,1\}^r$ and $X \in 1_{\mathcal{C}_Z}$. I.e., there exist $S_1, \ldots, S_r \in \{0,1\}^n$ and $M_1, \ldots, M_r \in \{0,1\}^l$ such that

1. $X_i = \pi(S_i), i = 1, \ldots, r$
2. $S_{i+1} = L(S_i), i = 1, \ldots, r-1$
3. $z_i = f(\pi(S_i), M_i), i = 1, \ldots, r$
4. $M_{i+1} = \Psi(\pi(S_i), M_i), i = 1, \ldots, r-1$

Let $\sigma \in S$. We show that $\sigma(X) \in 1_{\mathcal{C}_z}$. I.e., we have to prove that $\hat{S}_1, \ldots, \hat{S}_r \in \{0,1\}^n$ and $\hat{M}_1, \ldots, \hat{M}_r \in \{0,1\}^l$ exist such that the conditions above are true. Conditions 1. and 2. are fulfilled by assumption *(I)* in the proposition for some $\hat{S}_i$. We argue now that 3 and 4 are also true if we additionally set $\hat{M}_i := M_i$.
It is

$$f(\pi(\hat{S}_i), \hat{M}_i) = f(\sigma(\pi(S_i), \hat{M}_i) \overset{(II)}{=} f(\pi(S_i), M_i) \overset{3.}{=} z_i,$$

$$\hat{M}_{i+1} = M_{i+1} \overset{4.}{=} \Psi(\pi(S_i), M_i) \overset{(III)}{=} \Psi(\sigma(\pi(S_i)), M_i) = \Psi(\pi(\hat{S}_i), \hat{M}_i).$$

Hence, $\sigma(X) \in 1_{\mathcal{C}_z}$.

The proposition was motivated by combiners with memory as the summation generator or the E_0 keystream generator. For $S := \mathcal{S}_k$ and r smaller than the shortest LFSR, the assumptions of proposition 9 are fulfilled. Hence, we know that $\mathcal{C}_Z$ is $\mathcal{S}_k^r$-invariant which is by proposition 8 equivalent to that $1_{\mathcal{C}_Z}$ is $\mathcal{S}_k^r$-invariant.
In fact, this can be used to compute $\mathcal{C}_Z$ faster. It is known that each $\mathcal{S}_k$-invariant function $f \in \mathbb{B}_k$ can be expressed by a linear combination of the elementary symmetric polynomials π_i for $i = 0, \ldots, k$. Hereby it is

$$\pi_i = \bigoplus_{1 \le j_1 < \ldots < j_i < k} x_{j_1} \cdot \ldots \cdot x_{j_1}$$

the i-th elementary symmetric polynomial and $\pi_0 := 1$. This implies that a $\mathcal{S}_k^r$-invariant function $f \in \mathbb{B}_{r \cdot k}$ can be uniquely described by

$$f = \bigoplus_{0 \le i_1, \ldots, i_r \le k} c_{i_1, \ldots, i_r} \cdot \pi_{i_1}^{(1)} \cdot \ldots \cdot \pi_{i_r}^{(r)}$$

where $\pi_i^{(j)}$ denotes the i-th elementary symmetric polynomials in the variables $x_{(j-1)\cdot k+1}, \ldots, x_{j \cdot k}$.
In the following we introduce an improvement of *Algorithm 1* which exploits the S-invariance of $\mathcal{X}$:

Algorithm 2

```
Given: A set $\mathcal{F} \subseteq . \;{}_n$ such that $< F >$ is $S$-invariant, $\mathcal{X} \subseteq \{0,1\}^n$, $S \subseteq \mathcal{S}_k$
Task: A generating set for all $f \in < F >$ with $f(X) = 0$ for all $X \in \mathcal{X}$
Algorithm:
    Set $\mathcal{F}_0 := \mathcal{F}$ and $\mathcal{X} = \{[X_1], \ldots, [X_s]\}$
    For i from 1 to $s$ do
        Chose $X \in [X_i]$ arbitrary
        Set $\mathcal{F}_i^0 := \{f | f \in \mathcal{F}_{i-1}, f(X) = 0\}$ and $\mathcal{F}_i^1 := \{f | f \in \mathcal{F}_{i-1}, f(X) = 1\}$
        If $\mathcal{F}_{i-1} = \mathcal{F}_i^0$
            then $\mathcal{F}_i := \mathcal{F}_{i-1}$
            else $\mathcal{F}_i$:=(Algorithm 1)($\mathcal{F}_{i-1}$,[$X_i$])
Output: The set $\mathcal{F}_s$
```

The correctness is given by proposition 11 in appendix A. For $S = \{id\}$, *Algorithm 2* is exactly *Algorithm 1*. The improvement is that in *Algorithm 1*, the sets $\mathcal{F}^0$ and G^0 are computed for <u>all</u> $X \in \mathcal{X}$.

If $f(X_i) = 0$ for all $f \in \mathcal{F}_i$ then proposition 11 shows that $f(X) = 0$ for all $f \in \mathcal{F}_i$ and <u>all</u> $X \in [X_i]$. The algorithm *Algorithm 2* uses this fact to skip the computation of the sets $\mathcal{F}^0$ and $\mathcal{G}^0$ for the remaining elements in $[X_i]$. Therefore, the number of operations in *Algorithm 2* is in the worst case the same as in *Algorithm 1* but in presumably many cases significantly less.

Remark 5. With the theory of this section, one may be tempted to look only for S-invariant annihilators of an S-invariant characteristic function $\mathcal{C}$. Then, however, low-degree annihilators may be overlooked.

One example is the $\mathcal{S}_3$-invariant characteristic function $\mathcal{C} := 1 \oplus \pi_2 \oplus \pi_3$. It is $1_{\mathcal{C}} = \{[0,0,0],[1,0,0],[1,1,1]\}$ and $\text{mindeg}(An(\mathcal{C})) = 2$. A possible basis for all degree-2-annihilators is the set $\{x_1 \cdot (x_2 \oplus x_3), x_3 \cdot (x_1 \oplus x_2)\}$. On the other hand, $An(\mathcal{C})$ contains no $\mathcal{S}_3$-invariant functions of degree ≤ 2. The reason is that for an $\mathcal{S}_3$-invariant annihilator $f := c_0 \oplus c_1 \cdot \pi_1 \oplus c_2 \cdot \pi_2$ of degree ≤ 2 it must be

$$\begin{aligned}
0 &= f([0,0,0]) = c_0 \\
0 &= f([1,0,0]) = c_0 \oplus c_1 \\
0 &= f([1,1,1]) = c_0 \oplus c_1 \oplus c_2
\end{aligned}$$

This implies $c_0 = c_1 = c_2 = 0$ and therefore $f \equiv 0$. I.e., the $\mathcal{S}_3$-invariant characteristic function has quadratic annihilators which are all **not** $\mathcal{S}_3$-invariant.

5 Upper and lower bounds for mindeg($An(\mathcal{C})$)

If n resp. d are too large, the algorithms of section 4 become quickly impractical. Hence, more efficient methods to decide the existence of low-degree annihilators are desired. For certain cases, such criteria are presented in the following sections.

5.1 An upper bounds for mindeg($An(\mathcal{C})$)

Theorem 1. *Let $\mathbb{F}$ be an arbitrary field and $\mathbb{F}_n$ be the set of functions $\mathbb{F}^n \to \mathbb{F}$. Furthermore, let $\mathcal{F} := \{f_1, \ldots, f_s\} \subseteq \mathbb{F}_n$ be a set of linearly independent functions and $\mathcal{C} \in \mathbb{F}_n$ be given. We extend the definition of $1_{\mathcal{C}}$ to $1_{\mathcal{C}} := \{X | \mathcal{C}(X) \neq 0\}$. If $|1_{\mathcal{C}}| < s$ then $f \in < \mathcal{F} >$, $f \not\equiv 0$, exists with $f \in An(\mathcal{C})$.*

Proof. Assume that $1_{\mathcal{C}} = \{X_1, \ldots, X_r\}$. We set up the $r \times s$ matrix $M = (m_{ij})$ by $m_{ij} := f_j(X_i)$. Because of $s > |1_{\mathcal{C}}| = r$ the number of columns is larger than the number of rows. Therefore the columns are linearly dependent and at least one non-zero vector $c := (c_1, \ldots, c_s)$ exists such that $\sum_{j=1}^{s} c_j \cdot f_j(X_i) = 0$ for all i. This is equivalent to $f(x) = 0$ for all $x \in 1_{\mathcal{C}}$ for $f := \sum_{j=1}^{s} c_j \cdot f_j$. I.e., $f \in An(\mathcal{C})$.

The proof shows that the search for annihilators is equivalent to computing the kernel of the matrix M. This implies an algorithm for computing a generating set of all low-degree equations over arbitrary fields.

Corollary 1. *For integers $0 \leq d \leq n$, we set $\mu(n,d) := \binom{n}{0} + \ldots + \binom{n}{d}$. Let $\mathcal{C} \in \mathbb{B}_n$ be a Boolean function and d be such that $\mu(n,d) > |1_{\mathcal{C}}|$. Then* $\text{mindeg}(An(\mathcal{C})) \leq d$.

Proof. We use theorem 1 and define $\mathcal{F} \subseteq \mathbb{B}_n$ to be the set of all monomials in n variables of degree $\leq d$ (including the constant 1). Obviously, $\mathcal{F}$ consists of $\mu(n,d)$ linearly independent functions, all of them of degree $\leq d$. Due to $\mu(n,d) > |1_{\mathcal{C}}|$, theorem 1 guarantees the existence of a function $f \in < \mathcal{F} >$ with $f(X) = 0$ for all $X \in 1_{\mathcal{C}}$. Therefore, $f \in An(\mathcal{C})$. By definition of $\mathcal{F}$, f has a degree $\leq d$.

Remark 6. Let S be an arbitrary S-box $S : \{0,1\}^n \to \{0,1\}^m$. Then, $|1_{\mathcal{C}_S}| = 2^n$. If $2^n < \mu(n+m,d)$, then corollary 1 guarantees the existence of degree-d-equations. Figure 1 depicts an upper bound for mindeg($An(\mathcal{C}_S)$) for all values $1 \leq n, m \leq 32$. For example, for $n = m = 8$ (as it is the case for the AES S-box), an upper bound is 3. I.e., equations of degree have to exist, whatever the definition of S. Hence, the existence of degree-2-equations for the AES-S-box (see [7]) is not optimal with respect to algebraic attacks but not as surprising as it may appear. For the DES S-boxes with $n = 6$ and $m = 4$, an upper bound is 3. This has been showed before in [16]. The S-boxes used in Sober-t16 [10] ($n = 8$, $m = 16$) resp. in Sober-t32 [11] ($n = 8$, $m = 32$) have both quadratic equations.

Another interesting point is that for a fixed number n of inputs, the upper bound decreases if the number of outputs increases. A similar observation has been made for stream ciphers in [5].

(n,m)	1				5				9				13				17				21				25				29			
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	3	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1
	4	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	4	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
	5	4	4	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
9	5	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2
	6	5	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	6	5	5	5	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
	7	6	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
13	7	6	6	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	3
	8	7	6	6	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	8	7	7	6	6	6	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	9	8	7	7	6	6	6	6	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4	4	4	4	4	4	4
17	9	8	7	7	7	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	4	4	4	4	4	4	4
	10	8	8	7	7	7	7	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	10	9	8	8	7	7	7	7	7	6	6	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5
	11	9	9	8	8	8	7	7	7	7	7	6	6	6	6	6	6	6	6	6	6	6	5	5	5	5	5	5	5	5	5	5
21	11	10	9	9	8	8	8	7	7	7	7	7	7	7	6	6	6	6	6	6	6	6	6	6	6	6	6	5	5	5	5	5
	12	10	10	9	9	8	8	8	8	7	7	7	7	7	7	7	7	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
	12	11	10	10	9	9	8	8	8	8	8	7	7	7	7	7	7	7	7	7	7	6	6	6	6	6	6	6	6	6	6	6
	13	11	11	10	10	9	9	9	8	8	8	8	8	8	7	7	7	7	7	7	7	7	7	7	7	6	6	6	6	6	6	6
25	13	12	11	10	10	10	9	9	9	9	8	8	8	8	8	8	8	7	7	7	7	7	7	7	7	7	7	7	7	7	6	6
	14	12	11	11	10	10	10	9	9	9	9	9	8	8	8	8	8	8	8	8	7	7	7	7	7	7	7	7	7	7	7	7
	14	13	12	11	11	10	10	10	9	9	9	9	9	9	8	8	8	8	8	8	8	8	8	7	7	7	7	7	7	7	7	7
	15	13	12	12	11	11	10	10	10	10	9	9	9	9	9	9	8	8	8	8	8	8	8	8	8	8	8	7	7	7	7	7
29	15	14	13	12	12	11	11	11	10	10	10	10	9	9	9	9	9	9	9	8	8	8	8	8	8	8	8	8	8	8	8	8
	16	14	13	13	12	12	11	11	11	10	10	10	10	10	9	9	9	9	9	9	9	9	9	8	8	8	8	8	8	8	8	8
	16	15	14	13	12	12	12	11	11	11	11	10	10	10	10	10	10	9	9	9	9	9	9	9	9	9	8	8	8	8	8	8
	17	15	14	13	13	12	12	12	11	11	11	11	10	10	10	10	10	10	10	9	9	9	9	9	9	9	9	9	9	9	8	8

Fig. 1. Upper bounds for mindeg($An(\mathcal{C}_S)$) for different S-boxes $S : \{0,1\}^n \rightarrow \{0,1\}^m$

5.2 A lower Bound for mindeg($An(\mathcal{C})$)

Theorem 2. *Let $f \in \mathbb{B}_n$ be a non-zero Boolean function of degree $\leq d$. Then*

$$2^{n-d} \leq |0_f| \leq (2^d - 1) \cdot 2^{n-d}.$$

Both bounds can be achieved.

Proof. We prove the theorem by induction over n for a fixed degree d. Because of $2^n - \left(2^d - 1\right) \cdot 2^{n-d} = 2^{n-d}$, it suffices to show the upper bound.

For $n = d$, it is $|0_f| = 2^d - 1$ for $f = x_1 \cdot \ldots \cdot x_n$. Suppose now that the proposition is true for some $n \geq d$. I.e., $|0_f| \leq (2^d - 1) \cdot 2^{n-d}$ for all non-zero $f \in \mathbb{B}_n$ of degree $\leq d$, and this bound is sharp. Let $f \in \mathbb{B}_{n+1}$ be an arbitrary non-zero Boolean function of degree $\leq d$. Then, f can be written as follows

$$f(x_1, \ldots, x_{n+1}) = f'(x_1, \ldots, x_n) \oplus x_{n+1} \cdot f''(x_1, \ldots, x_n)$$

where $f', f'' \in \mathbb{B}_n$, $\deg(f') \leq d$, $\deg(f'') \leq d-1$ and at least one of them is non-zero. We have to distinguish three cases:

$f' \neq 0$ **and** $f'' = 0$: Then $f \in \mathbb{B}_n$ and $|O_f| \leq (2^d - 1)2^{n-d} \leq (2^d - 1)2^{n+1-d}$ by assumption.

$f' = 0$ **and** $f'' \neq 0$: Then $0_f = \{(x, 0)\} \dot\cup \{(x, 1) | x \in 0_{f''}\}$. Because of $\deg(f'') \leq d - 1$, it is $|0_f| \leq 2^n + (2^{d-1} - 1) \cdot 2^{n-(d-1)} = (2^d - 1)2^{n-d}$.

$f' \neq 0$ **and** $f'' \neq 0$: Then 0_f can be expressed by

$$0_f = \{(x, 0) | x \in 0_{f'}\} \dot\cup \{(x, 1) | x \in 0_{f' \oplus f''}\}$$

If $f' \oplus f'' = 0$ then $\deg f' = \deg f'' \leq d - 1$ and

$$|0_f| = |0_{f'}| + |0_{f' \oplus f''}| \leq (2^{d-1} - 1) \cdot 2^{n-(d-1)} + 2^n = (2^d - 1) \cdot 2^{n+1-d}$$

If $f' \oplus f'' \neq 0$ then

$$|0_f| = |0_{f'}| + |0_{f' \oplus f''}| \leq (2^d - 1) \cdot 2^{n-d} + (2^d - 1) \cdot 2^{n-d} = (2^d - 1) \cdot 2^{n+1-d}$$

If we chose f' such that $|0_{f'}| = (2^d - 1) \cdot 2^{n-d}$ and $f'' = 0$ then the bound $(2^d - 1) \cdot 2^{n+1-d}$ is achieved by f.

Theorem 2 can be used to exclude the existence of annihilators of degree $\leq d$:

Corollary 2. *Let $f \in \mathbb{B}_n$. If $|1_f| > (2^d - 1)2^{n-d}$ then* mindeg($An(f)$) $> d$.

Proof. Let $g \in An(f)$. Proposition 1 implies that $1_f \subseteq 0_g$ and therefore $|1_f| \leq |0_g|$. Hence, the assumption $\deg(g) \leq d$ would lead to the following contradiction:

$$(2^d - 1)2^{n-d} < |1_f| \leq |0_g| \overset{Th.2}{\leq} (2^d - 1)2^{n-d}.$$

Example 1. The E_0 keystream generator uses $k = 4$ LFSRs. In [1], characteristic functions $\mathcal{C}_Z \not\equiv 1$ for all $Z \in \{0, 1\}^4$ were described. We've checked that $|1_{\mathcal{C}_Z}| = 53.248$ for all $Z \in \{0, 1\}^4$. Because of $|1_{\mathcal{C}_Z}| = 53.248 > 49.152 = (2^2 - 1)2^{4 \cdot 4 - 2}$, all annihilators $f \in An(\mathcal{C}_Z)$ have a degree ≥ 3.

6 Conclusion

In this paper, we examined the existence of low-degree equations for three important cases (combiners without memory, combiners with memory and S-boxes). For the first time, the question of low-degree equations is reduced in all three cases to the same problem, the search for low-degree annihilators. We assume that methods from algebraic geometry may turn out to be useful for further research.

Further on, we discussed two different algorithms for computing a generating set of all low-degree equations. The first one is generally applicable. It has been described before but never in its generality. The second one is adapted to certain keystream generators (e.g., the Bluetooth keystream generator) to avoid dispensable computations. To the best of our knowledge, it has not been published before.

Finally, we proved for certain cases an upper and a lower bound for the minimal degree. The upper bound has been discussed before in other papers but only in the context of LFSR-based keystream generators. The lower bound was unknown in the context of algebraic attacks.

References

1. Frederik Armknecht, Matthias Krause: *Algebraic attacks on Combiners with Memory*, Proceedings of Crypto 2003, LNCS 2729, pp. 162-176, Springer, 2003.
2. Frederik Armknecht: *Improving fast algebraic Attacks*, Fast Software Encryption 2004, LNCS 3017, pp. 65 - 82, Springer, 2004.
3. Alex Biryukov, Christoph De Cannière: *Block Ciphers and Systems of Quadratic Equations*, Fast Software Encryption 2003, pp. 274-289, Springer, 2003.
4. Nicolas Courtois: *Fast Algebraic Attacks on Stream Ciphers with Linear Feedback*, Proceedings of Crypto 2003, LNCS 2729, pp. 177-194, Springer, 2003.
5. Nicolas Courtois: *Algebraic Attacks on Combiners with Memory and Several Outputs*, Cryptology ePrint Archive, Report 2003/125, 2003. http://eprint.iacr.org/2003/125.
6. Nicolas Courtois, Willi Meier: *Algebraic attacks on Stream Ciphers with Linear Feedback*, Proceedings of Eurocrypt 2003, LNCS 2656, pp. 345-359, Springer, 2003. An extended version is available at http://www.cryptosystem.net/stream/
7. Nicolas Courtois, Josef Pieprzyk: *Cryptanalysis of block ciphers with overdefined systems of equations*, Proceedings of Asiacrypt 2002, LNCS 2501, pp. 267-287, Springer, 2002.
8. Jean-Charles Faugère, Gwenole Ars: *An algebraic cryptanalysis of nonlinear filter generators using Gröbner bases*, 2003. Available at http://www.inria.fr/rrrt/rr-4739.html.
9. Niels Ferguson, Richard Schroeppel, Doug Whiting: *A simple algebraic representation of Rijndael*, Proceedings of Seleceted Areas in Crpytography 2001, LNCS 2259, pp. 103 - 111, Springer, 2001.
10. Philip Hawkes, Gregory G. Rose: *Primitive specification and supporting documentation for Sober-t16 submission to NESSIE*, Proceedings of the first NESSIE workshop, Belgium, 2000.
11. Philip Hawkes, Gregory G. Rose: *Primitive specification and supporting documentation for Sober-t32 submission to NESSIE*, Proceedings of the first NESSIE workshop, Belgium, 2000.
12. Philip Hawkes, Gregory G. Rose: *Rewriting Variables: the Complexity of Fast Algebraic Attacks on Stream Ciphers*, will be presented at Crypto 2004. Available at http://eprint.iacr.org/2004/081/.
13. Willi Meier, Enes Pasalic, Claude Carlet: *Algebraic attacks and decomposition of Boolean functions*, Proceeding of Eurocrypt 2004, LNCS 3027, pp. 474-491, Springer, 2004.
14. Sean Murphy, Matthew Robshaw: *Comments on the Security of the AES and the XSL Technique*, Electronic Letters, 39:26-38, 2003.
15. Ingrid Schaumüller-Bichl: *Cryptanalysis of the Data Encryption Standard by the Method of Formal Coding*, Proceeding of Eurocrypt 1982, LNCS 149, pp. 235-255, Springer, 1983.

16. Takeshi Shimoyama, Toshinobu Kaneko: *Quadratic Relation of S-box and Its Application to the Linear Attack of Full Round DES*, Proceedings of Crypto 1998, LNCS 1462, pp. 200-211, Springer, 1998.

A Proofs of Correctness

In this section, we provide the proofs of correctness for *Algorithm 1* and *Algorithm 2*. The following proposition shows that *Algorithm 1* works properly:

Proposition 10. *Let the identifiers be as described in Algorithm 1. Then $< \mathcal{F}_i >$ is the vector space of all functions $f \in < \mathcal{F}_{i-1} >$ such that $f(X_1) = \ldots = f(X_i) = 0$.*

Proof. We show this by induction over i. The case $i = 0$ is trivial. Assume now that the proposition is true for $i \geq 1$. Because of $< \mathcal{F}_i > \subseteq < \mathcal{F}_{i-1} >$ it is $f(X_j) = 0$ for all $f \in < \mathcal{F}_i >$ and $j = 1, \ldots, i-1$. By the definition of $\mathcal{F}_i^0$ and $\mathcal{G}_i^0$, it is $f(X_i) = 0$ also. What remains is to show that all $f \in < \mathcal{F}_{i-1} >$ with $f(X_i) = 0$ are in $< \mathcal{F}_i >$ too.

Let $f \in < \mathcal{F}_{i-1} >$ with $f(X_i) = 0$. There exist $f_1, \ldots, f_r \in \mathcal{F}_{i-1}$ with $f = \bigoplus_{j=1}^r f_j$. W.l.o.g., we can assume that there is an r' with $1 \leq r' \leq r$ such that $f_j(X_i) = 0$ for $1 \leq j \leq r'$ and $f_j(X_i) = 1$ for $r' + 1 \leq j \leq r$. The equation

$$0 = f(X_i) = \bigoplus_{j=1}^{r'} \underbrace{f_j(X_i)}_{=0} \oplus \bigoplus_{j=r'+1}^{r} \underbrace{f_j(X_i)}_{=1}$$

shows that the value $r' - r$ is even. Because of $\tilde{f} \oplus \tilde{f} = 0$, the function f can be equivalently expressed by

$$f = \bigoplus_{j=1}^{r'} f_j \oplus \bigoplus_{j=r'+1}^{r} (f_j \oplus \tilde{f})$$

which is in $< \mathcal{F}_i^0 \cup \mathcal{G}_i^0 > = < \mathcal{F}_i >$.

The correctness of *Algorithm 2* is given by:

Proposition 11. *Let the identifiers be as in Algorithm 2 and $i \geq 0$. If $f(X) = 0$ for all $f \in \mathcal{F}_{i-1}$ and $\underline{\text{one}}$ $X \in [X_i]$ then $f(X) = 0$ for all $f \in \mathcal{F}_{i-1}$ and $\underline{\text{all}}$ $X \in [X_i]$.*

Proof. First we show that $< \mathcal{F}_i >$ is S-invariant if $< \mathcal{F}_{i-1} >$ is S-invariant. It is either $\mathcal{F}_i = \mathcal{F}_{i-1}$ or $\mathcal{F}_i =$(Algorithm 2)($\mathcal{F}_{i-1}$,$[X_i]$). In the first case, the proposition is trivial. In the second case, $< \mathcal{F}_i >$ is the vector space of all $f \in \mathcal{F}_{i-1}$ with $f(X) = 0$ for all $X \in [X_i]$. Let $f \in < \mathcal{F}_i >$ and $\sigma \in S$ be arbitrary. Then $\sigma(f)(X) = f(\sigma(X)) = 0$ for all $X \in [X_i]$. Therefore, $\sigma(f) \in < \mathcal{F}_i >$. This shows that $< \mathcal{F}_i >$ is S-invariant.

Now assume that $f(X) = 0$ for all $f \in \mathcal{F}_{i-1}$ and $\underline{\text{one}}$ $X \in [X_i]$. Let $f \in \mathcal{F}_{i-1}$ and $\sigma \in S$ be arbitrary. We have to show that $f(\sigma(X)) = 0$ also. By the observation above, $< \mathcal{F}_{i-1} >$ is S-invariant. I.e., there exist $f_1, \ldots, f_r$ with $\sigma(f) = f_1 \oplus \ldots \oplus f_r$. This implies

$$f(\sigma(X)) = \sigma(f)(X) = \underbrace{f_1(X)}_{=0} \oplus \ldots \oplus \underbrace{f_r(X)}_{=0} = 0.$$