Strong and Tight Security Guarantees against Integral Distinguishers

Phil Hebborn[1], Baptiste Lambin[1], Gregor Leander[1], and Yosuke Todo[2]

[1] Horst Görtz Institute for IT Security, Ruhr University Bochum, Bochum, Germany,
phil.hebborn@rub.de,baptiste.lambin@protonmail.com,gregor.leander@rub.de
[2] NTT Social Informatics Laboratories, Tokyo, Japan,
yosuke.todo.xt@hco.ntt.co.jp

Abstract. Integral attacks belong to the classical attack vectors against any given block ciphers. However, providing arguments that a given cipher is resistant against those attacks is notoriously difficult. In this paper, based solely on the assumption of independent round keys, we develop significantly stronger arguments than what was possible before: our main result is that we show how to argue that the sum of ciphertexts over any possible subset of plaintext is key-dependent, i.e., the non existence of integral distinguishers.

Keywords: block cipher · integral distinguisher

1 Introduction

As symmetric primitives, due to their performance advantages, are a vital part of our security building blocks, being able to assess their security is of great practical importance and theoretical interest. The security of block ciphers, and actually any symmetric primitive in use, is always the security against concrete attacks. Two of the most important attacks are certainly differential and linear attacks. Security arguments with respect to those attacks have been studied for quite some time already, leading to important concepts like the Markov model in [16]. Nowadays, we are usually able to bound, under the assumption of independent round-keys, the probability of a differential characteristic (or the correlation of a linear characteristic). Those are good, but certainly not fully satisfactory security arguments, as we often ignore the differential or linear-hull effect. Stronger arguments, like a bound of the expected differential probability, require a dedicated design, see e.g. [18] and [8].

Another classical attack vector is integral attacks, which can be traced back to high-order differentials by Lai [15], and then exploited by Knudsen to be used in actual attacks [14] as well as the so-called ”Square attack” [10]. In a nutshell, given a block cipher E_k, those attacks work by identifying a subset of plaintexts M such that summing over the corresponding ciphertexts results in a constant sum, i.e., $\sum_{x \in M} E_k(x)$ does not depend on the secret key k. Arguments for security against those attacks, i.e. arguments showing that such a set M should not exist for a given cipher, are very difficult to obtain. For most ciphers, we do not have any argument at all and if arguments are given, they only cover very specific sets M.

In most attacks, M is chosen as a subspace and more specifically by fixing some bits in the plaintext to constants. This specific choice of M is not at all necessary for a successful attack, and indeed there are examples of more involved plaintext sets being used for improved attacks. The main reason for this choice is the relation to the algebraic degree of the cipher. Indeed, for a cipher of algebraic degree at most d, taking M as any subspace of dimension larger than d leads to a successful distinguisher, as the sum is zero. Thus, a first step towards arguing the security of a block cipher against integral attacks is to show that its algebraic degree is maximal. However, even this special case was only settled very recently. For a long time, only upper bounds on the algebraic degree have been discussed. At ASIACRYPT'20 in [12], it was shown for the first time how to compute meaningful lower bounds on the degree of round-reduced block ciphers. Technically, this approach is based on recent progress on the division property initially introduced in [20].

While [12] demonstrated how to compute lower bounds on the degree for the first time, several drawbacks remained: It does not allow to exclude integral attacks and its applicability is limited due to a lack of efficiency. Further, only bounds for round-reduced variants could be computed.

Limited Arguments. As outlined above, even if the degree is high, there might still be integral distinguishers and attacks. An integral distinguisher in general makes use of a fixed set $M \subset \mathbb{F}_2^n$ of plaintext values and a bit-mask β such that, for any key k the value of $\sum_{x \in M} \langle \beta, E_k(x) \rangle$ is independent of the key. In [12], this was shown for a natural, but very limited, choice of sets M, where M consists of just fixing bits in the input. While this is, to the best of our knowledge, the best argument against integral attacks so far, it is far from being satisfactory. In particular, it does not capture integral distinguishers where M consists of fixing linear combination in the input, used in [17] or [19]. More generally, it does not capture integral distinguishers where M is not a linear subspace, as in [22].

Limited Number of Rounds. The arguments given in [12] allowed to compute lower bounds on the algebraic degree for a fixed number of rounds. When the number of rounds is increased, computing the bounds quickly becomes infeasible. This is in sharp contrast to the expected result. If r rounds of a given cipher have maximal degree $n-1$, it is naturally expected that more than r rounds have the same degree. While this intuition is probably true in most cases, it is of course not a sound security argument. Making this argument more precise is non-trivial. It is clear that in general, given F of degree $n-1$, representing the (fixed-key) first-part of the cipher there always exists a function G such that $G \circ F$ is of lower degree than F. Indeed, the easiest example is to choose G as the inverse of F, in case F is a permutation. One might hope that in the case of a keyed permutation, the situation is less bad, as at least the trivial example above is not applicable anymore. However, even in the keyed case, as we show in Example 4, there exist permutations F and G such that

$$\deg(G \circ (F(x) + k)) < \min(\deg(F), \deg(G)).$$

That is, the degree of the composition is actually smaller than the individual degrees, *for any key k*. This shows that it cannot be excluded simply by assuming independent round keys that the degree (as a keyed function) decreases.

Lack of Efficiency. The proof of the lower bound of degree proposed in [12] strongly ties to the division property [20], which is originally a tool to detect an integral distinguisher. To prove the lower bound, we need to generate a key pattern whose number of division trails is odd, and countable in practical time. As pointed out in [12], it is not easy because the number of trails exponentially increases unless a key pattern is generated in a clever manner. The so-called trail extension was used to generate such a key pattern and enabled to prove lower bounds on the degree (especially the number of rounds so that this lower bound is maximal, i.e. 63), for SKINNY-64, GIFT-64, and PRESENT. On the other hand, the applicability of the trail extension to other block ciphers is an open question. Interestingly, we faced potential difficulties of the trail extension when we tried expanding applications (the tweakable block cipher CRAFT as an example). The number of trails exceeds the practically countable range quickly, and it is unfeasible to prove a lower bound on the degree of CRAFT.

Our Contribution. In this paper we derive strong and tight bounds against integral distinguishers for several block ciphers. The only assumption on which we rely for our bounds is having independent-round key, i.e., independent round keys are XORed with the full state. Our bounds are *strong* as we show that for a cipher E_k, the sum $\sum_{x \in M} \langle \beta, E_k(x) \rangle$ is key-dependent for *any possible set M* (excluding only the whole input space and the empty set) and *any possible non-zero mask β*. We refer to this as the *integral-resistance property*. Our bounds are *tight* as (in most cases) the minimal number of rounds where we can show the non-existence *matches the best known distinguishers*. Our arguments extend to an arbitrary number of rounds greater than that.

First, we fix our notation and recall the basic techniques in Sect. 2. We develop the necessary theory to achieve the strong arguments in Sect. 3. To formalize the strong arguments, which means the guarantee of the non-existence of integral distinguishers under the sole assumption of independent round keys, we introduce the integral-resistance property. We develop the theoretical background to utilize the division property, for not only showing a lower bound of the degree of a cipher as in [12], but to show the integral-resistance property.

In Sect. 4 we study how adding more rounds (separated with a key addition) will affect the algebraic degree, the minimum degree, and the strong argument against integral distinguishers. For the minimum degree and the strong argument, we are able to show that adding (keyed or un-keyed) rounds is never a problem, as the minimum degree never decreases and the strong argument never vanishes when doing so. The algebraic degree on the other hand potentially decreases, as also sketched above. Here we are able to present efficiently computable criteria on the S-box that allow to exclude such undesirable behavior.

The drawback of the lack of efficiency is handled in Sect. 5. We show that, maybe counter-intuitively, a suitable rewriting of the cipher and in particular the S-box, can have a significant effect on the running time of the techniques used in [12]. We present (heuristic) conditions on how to choose a suitable description of the cipher that allows to keep the number of division trails reasonably low - a fact that is crucial as those have to be enumerated.

Finally, in Sect. 6 we apply the theory and tools developed to a set of ciphers. Besides the ciphers treated in [12], and which present a large fraction of the primitives used in the running NIST lightweight competition, we added a discussion of `CRAFT` [5], which was previously out of reach and a discussion of the ciphers `SIMON` and `Simeck` as examples of non-SPN ciphers. We assume independent round keys for all ciphers. Further, for `GIFT-64` and `SKINNY-64`, we assume, contrary to the specification, a key addition on the full state.

For all those applications we are able to show the non-existence of integral distinguishers. Interestingly, except for `GIFT-64` and `PRESENT`, our result matches the best known attacks tightly, as can be seen in Table 1.

We finally emphasize the meaning of our results. Our results guarantee that improving integral distinguishers is impossible under our assumption. This is strong claim compared to heuristic attack failure. For example, for `CRAFT`, we guarantee no integral distinguisher for 14 rounds and more. Thus any such distinguisher would have to violate our assumptions. In other words, it has to exploit the key scheduling. For `SKINNY-64` and `GIFT-64` our results are slightly weaker compared to the other applications because here round keys are not XORed with the whole state in both ciphers. Room of improvements still remains without exploiting key scheduling, but it must exploit the fact that the round key is XORed with the half of the state only.

2 Preliminaries

In this section, we are going to recall the definitions and the different notations of degree that are commonly used for Boolean functions. We also recall what was shown in [12] and briefly explain the necessary background on division property to explain how this was done technically.

2.1 Degree of Keyed Functions - Definitions and Results

A block cipher can be seen as a family of (keyed) vectorial boolean permutations, that is, bijective functions $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ with $k \in \mathbb{F}_2^m$. We can represent such functions with their algebraic normal form (ANF)

$$E_k(x) = \sum_{u \in \mathbb{F}_2^n} p_u(k) x^u$$

Table 1. Number of rounds of the best known integral distinguisher (together with a reference for this) vs. the number of rounds we need to ensure the integral-resistance property under the assumption that independent round keys are XORed to the full state. Numbers in red indicate are tight results.

Cipher	Known integral distinguisher	Integral-resistance property
SKINNY-64	12 [11]	13
CRAFT	13 [5]	14
GIFT-64	10 [2]	12
PRESENT	9 [23]	13
SIMON32	15 [21]	16
SIMON48	16 [23]	17
SIMON64	18 [23]	19
SIMON96	22 [23]	23
SIMON128	26 [23]	27
Simeck32	15 [21]	16
Simeck48	18 [23]	19
Simeck64	21 [23]	22

with $x^u = \prod_i x_i^{u_i}$ and $p_u(k)$ are functions $p_u : \mathbb{F}_2^m \to \mathbb{F}_2^n$ mapping keys to values in $\mathbb{F}_2^n$. We define the *algebraic degree* of E_k as the degree in the input variables x, that is, the algebraic degree $\deg(E_k)$ is defined as

$$\deg(E_k) := \max_u \{\mathrm{wt}(u) \mid p_u \neq 0\},$$

where $\mathrm{wt}(u)$ denotes the Hamming weight of u, i.e. the number of non-zero coordinates of u.

The *minimum degree* of E_k, is defined as the minimum degree over all non-zero component functions $\langle \beta, E_k \rangle$

$$\mathrm{minDeg}(E_k) = \min_{\beta \neq 0} \deg(\langle \beta, E_k \rangle).$$

Until recently, getting meaningful lower bounds for both the algebraic degree and the minimum degree was deemed essentially impossible for block ciphers of relevant size (i.e. at least 64-bit block size). However at ASIACRYPT'20, Hebborn et al. [12] managed to obtain such lower bounds at least for round-reduced variants. The main idea is that to show a lower bound d on the algebraic degree of E_k, one "simply" needs to show that there exists a $u \in \mathbb{F}_2^n$ such that $\mathrm{wt}(u) \geq d$ and $p_u \neq 0$. If we denote the coefficients of p_u by $\lambda_{u,v} \in \mathbb{F}_2^n$, that is,

$$E_k(x) = \sum_{u \in \mathbb{F}_2^n, v \in \mathbb{F}_2^m} \lambda_{u,v} x^u k^v,$$

this is the same as finding a $u \in \mathbb{F}_2^n$ with $\mathrm{wt}(u) \geq d$ and $v \in \mathbb{F}_2^m$ such that $\lambda_{u,v} \neq 0$, which is equivalent to $p_u \neq 0$.

Finally, they also introduced an even stronger notion, namely the appearance of *all* maximum-degree monomials, which is that for any given monomial x^u of algebraic degree $n-1$, and for any component function $\langle \beta, F\rangle$, there always exists at least one key k such that the monomial x^u appears in the ANF of $\langle \beta, E_k\rangle$.

However this paper comes with significant limitations. It was shown that having the all maximum-degree monomial property allows to rule out basic integral distinguishers. Indeed, it can only rule out distinguishers constructed with a set of plaintexts M built as an affine space of the form

$$M = \{x \in \mathbb{F}_2^n \text{ s.t. } \forall i \in I, x_i = c_i\},$$

where I is a subset of $\{1, \dots, n\}$ and c_i are fixed constants in $\mathbb{F}_2$. On the other hand, the case presented in [17], where the input set is an affine space with a more convoluted structure, is not. For example, already the affine space

$$M = \{x \in \mathbb{F}_2^n \text{ s.t. } x_0 + x_1 = 0\},$$

is out of scope, not to mention arbitrary subsets.

Despite these limitations, we can use the core idea of their work, which is, after explaining how to compute such a $\lambda_{u,v}$, to decide how to choose these u and v, allowing to actually compute $\lambda_{u,v}$ in practical time so that we can prove the various lower bounds and properties. We give more details about this in Sect. 2.3. Before that, in the next section we first give a high-level overview of the main tool used in their work, that is, division property.

2.2 High-Level Summary of Division Property

After the division property was first proposed in [20], many follow-up works have been proposed [7]. In [12], the various notations, definitions, and theorems about the division property were unified by using the parity set, which was used as another view of the division property in [7]. Here we briefly recall the main definitions and connections with the algebraic normal form.

Definition 1 (Parity Set). *Let $\mathbb{X} \subseteq \mathbb{F}_2^n$ be a set. We define the* parity set *of $\mathbb{X}$ as*

$$\mathcal{U}(\mathbb{X}) := \left\{ u \in \mathbb{F}_2^n \textit{ such that } \sum_{x \in \mathbb{X}} x^u = 1 \right\}.$$

The addition of two subsets $\mathbb{X}, \mathbb{Y} \subseteq \mathbb{F}_2^n$ is defined by

$$\mathbb{X} + \mathbb{Y} := (\mathbb{X} \cup \mathbb{Y}) \setminus (\mathbb{X} \cap \mathbb{Y}).$$

In other words, we view the set of all subsets of $\mathbb{F}_2^n$ as a binary vector space of dimension 2^n, and this addition is isomorphic to adding the binary indicator vectors of the sets. From this perspective, for $\mathbb{X}_i \subseteq \mathbb{F}_2^n$,

$$\mathcal{U}\left(\sum \mathbb{X}_i\right) = \sum \mathcal{U}\left(\mathbb{X}_i\right)$$

holds, i.e. $\mathcal{U}$ is a linear mapping. Moreover, it was shown in [7] that there is a one to one correspondence between sets and its parity set. That is the mapping $\mathcal{U} : \mathbb{X} \mapsto \mathcal{U}(\mathbb{X})$ is a bijection and actually its own inverse, i.e., $\mathcal{U}(\mathcal{U}(\mathbb{X})) = \mathbb{X}$.

We next define the propagation as follows.

Definition 2 (Propagation). *Given $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ and $a \in \mathbb{F}_2^n, b \in \mathbb{F}_2^m$, we say that* the division property a propagates to the division property b*, denoted by $a \xrightarrow{F} b$ if and only if $b \in \mathcal{U}(F(\mathcal{U}(\{a\})))$.*

Here the image of a set $\mathbb{X}$ under F is defined as

$$F(\mathbb{X}) := \sum_{a \in \mathbb{X}} \{F(a)\},$$

that is again using the addition of sets as defined above. Given $U_1 = \mathcal{U}(\mathbb{X})$, for any function F, $U_2 = \mathcal{U}(F(\mathbb{X}))$ is evaluated as

$$U_2 = \mathcal{U}(F(\mathbb{X})) = \sum_{x \in \mathbb{X}} \mathcal{U}(F(\{x\})) = \sum_{a \in \mathcal{U}(\mathbb{X})} \mathcal{U}(F(\mathcal{U}(\{a\}))) = \sum_{a \in U_1, a \xrightarrow{F} b} \{b\}. \quad (1)$$

To determine U_2 after applying the function F, it is enough to consider what happens with individual elements of U_1 to start with. Again, we emphasize that the sum in Equation 1 is modulo two, that is, if an element appears an even number of times on the right side, it actually does not appear in U_2. Note that the propagation rules shown in [21] can be proven by assigning concrete operation to F. More generally, the propagation for any function F is described as follows.

Proposition 1 ([12]). *Let $F \colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be defined as $F(x) = y$. For $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2^m$, it holds that $a \xrightarrow{F} b$ if and only if y^b contains the monomial x^a.*

We now generalize the definition above to the setting where F is actually given as the composition of many functions as $F = F_R \circ \cdots \circ F_2 \circ F_1$.

Definition 3 (Trail). *Given $F \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ as $F = F_R \circ \cdots \circ F_2 \circ F_1$, and $a_0, \ldots, a_R \in \mathbb{F}_2^n$ we call $(a_0, \ldots, a_R)$ a (division) trail for the compositions of F into the F_i if and only if*

$$\forall i \in \{1, \ldots, R\}, a_{i-1} \xrightarrow{F_i} a_i.$$

We denote such a trail by $a_0 \xrightarrow{F_1} a_1 \xrightarrow{F_2} \cdots \xrightarrow{F_R} a_R$.

Using the same considerations as in Equation 1, we can now state the main reason of why considering trails is useful.

Theorem 1 ([12]). *Given $F \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ as $F = F_R \circ \cdots \circ F_2 \circ F_1$, and $\mathbb{X} \subseteq \mathbb{F}_2^n$. Then*

$$\mathcal{U}(F(\mathbb{X})) = \sum_{a_0, \ldots, a_R, a_0 \in \mathcal{U}(\mathbb{X}), a_0 \xrightarrow{F_1} a_1 \xrightarrow{F_2} \cdots \xrightarrow{F_R} a_R} \{a_R\}$$

Finally, we show the link between the division property and the ANF.

Corollary 1 ([12]). *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a function with algebraic normal form*

$$F(x) = \sum_{u \in \mathbb{F}_2^n} \lambda_u x^u$$

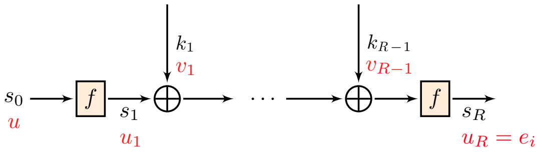


Fig. 1. Notations for the trails of a key-alternating block cipher, where the terms in red are the parity-set vector of the corresponding state

where $\lambda_u = (\lambda_u^{(1)}, \ldots, \lambda_u^{(n)}) \in \mathbb{F}_2^n$*. Furthermore, let* $\mathbb{X}$ *be the set such that* $\mathcal{U}(\mathbb{X}) = \{\ell\}$*. Then*

$$\lambda_\ell^{(i)} = 1 \Leftrightarrow e_i \in \mathcal{U}(F(\mathbb{X})) \Leftrightarrow \#\{a_1, \ldots, a_{R-1} | \ell \xrightarrow{F_1} a_1 \xrightarrow{F_2} \cdots \xrightarrow{F_R} e_i\} = 1 \bmod 2$$

2.3 Proof of a Lower Bound on the Degree and Finding Key Patterns

Now that we are equipped with the results from the previous section, we can give more details about the work from [12] where the authors gave lower bounds on the algebraic degree and minimum degree of block ciphers, as the techniques we use in the next section to give strong arguments against integral distinguishers strongly rely on their results. We put ourselves in the context of key-alternating block ciphers, depicted in Figure 1. We assume that we have a round function f, and the block cipher E is built by alternating applications of f with a round key addition (with an XOR) between them. As in [12], we assume that the round keys are independent from each other. The internal states are thus denoted by $s_0, \ldots, s_R$, where R is the number of rounds, s_0 is the input (plaintext) of the block cipher and s_R the output (ciphertext). The round keys are denoted by $k_1, \ldots, k_{r-1}$. The key length m is $(r-1)n$.

As mentioned in Sect. 2.1, showing a lower bound d on the algebraic degree of a block cipher is equivalent to exhibiting vectors $u \in \mathbb{F}_2^n$ of weight at least d and $v \in \mathbb{F}_2^m$ (where m is the key length) so that $\lambda_{u,v}$ is non-zero, which in particular means that one coordinate $\lambda_{u,v}^{(i)}$ is equal to 1. As each round key is independent, we can write v as $(v_1, \ldots, v_{R-1})$ so that $k^v = k_1^{v_1} k_2^{v_2} \ldots k_{R-1}^{v_{R-1}}$.

According to the previous section, proving that $\lambda_{u,v}^{(i)} = 1$ is equivalent to showing that the number of trails $(u, v) \xrightarrow{E} e_i$ is odd. We take the same denomination as in [12], and call u the *input pattern*, v the *key pattern* and $u_R = e_i$ the *output pattern*. As mentioned in Sect. 2.1, to get lower bounds on the minimum degree, instead of showing that a single $\lambda_{u,v}^{(i)}$ is equal to 1, we now need to compute the value of several of these $\lambda_{u,v}^{(i)}$ so that we end up with a set of $\lambda_{u,v}$ which spans a vector space of dimension n.

In both cases, the main goal is to find (several) u and v so that we can compute the coefficients $\lambda_{u,v}^{(i)}$ in practical time, which according to the previous section, means being able to enumerate all corresponding trails. The core of the work in [12] is thus to give a (heuristic) algorithm to determine which u and v to choose so that we can actually enumerate all of these trails in a reasonable time. As we also need to compute such $\lambda_{u,v}^{(i)}$ for our results, we give a quick overview of their algorithm and refer the reader to the full paper for more details. Their main observation is that having a key pattern with a high weight tends to lower the resulting number of trails, which is quite interesting since the lower this number is, the easier (and quicker) enumerating them should be. However, the naive idea of simply maximizing the weight of the key pattern is not enough, and thus they used the following strategy. We start by fixing the input pattern u and output pattern $u_R = e_i$, and focus on finding $v = (v_1, \ldots, v_{R-1})$ so that the number of trails is reasonably low. Starting from u_R, we first search for a (partial) key pattern v_{R-1} so that the number of trails $(u_{R-1}, v_{R-1}) \rightarrow e_i$ is odd and low (optimally, only one trail), maximizing the weight of the (partial) key pattern v_{R-1} as it should help minimize the number of trails. After finding such a v_{R-1}, we now search in the same way for a partial key pattern v_{R-2} so that again, the number of trails $(u_{R-2}, v_{R-2}, v_{R-1}) \rightarrow e_i$ is odd and low (again, optimally, only one trail). The authors observed that this ”local optimization” strategy seems to fail if we keep going too close to the first round. Thus, we only keep doing this up to some round ”in the middle” R_{mid}, leading us to a partial key pattern $(v_{R_{mid}}, \ldots, v_{R-1})$ so that the number of trails from $(u_{R_{mid}}, v_{R_{mid}}, \ldots, v_{R-1})$ to e_i is odd and low. After that, we directly search for the remaining parts of the key pattern $(v_1, \ldots, v_{R_{mid}-1})$ so that the number of trails from $(u, v_1, \ldots, v_{R_{mid}-1})$ to $u_{R_{mid}}$ is odd and low, and finally verify that the number of trails $(u, v_1, \ldots, v_{R-1}) \rightarrow e_i$ is still odd. If so, we proved that $\lambda_{u,v}^{(i)} = 1$ and keep using the same strategy until we found enough (u, v) as we need. One limitation is that for various technical reasons, the authors of [12] were limited to SPN block ciphers, so that they were able to exploit Super S-box representations, making ciphers like Feistel networks out of reach, and some ciphers (e.g. `CRAFT`) did not have a favorable behavior regarding the trail extension technique that we just summarized. Nonetheless, we actually managed to get results for the Feistel networks ciphers `SIMON` and `Simeck`, as shown in Sect. 6.5, as well as getting results on `CRAFT` with new techniques in Sect. 5.

In summary, from [12], we can efficiently compute the value of some coefficients $\lambda_{u,v}$, which we will use in the upcoming sections to prove our results.

3 Strong Arguments Against Integral Distinguishers

Here, we are going to derive necessary and sufficient conditions on when integral distinguishers are not possible. More precisely, we aim at conditions such that we can conclude that, for a cipher E_k, the sum

$$\sum_{x \in M} \langle \beta, E_k(x) \rangle$$

is key-dependent for *any possible set* M (excluding only the whole input space and the empty set) and *any possible non-zero mask* β. Note that this covers a much larger set of possible integral distinguishers than commonly used in previous works. Indeed, most classical integral distinguishers build the set M as an affine space by fixing some bits to a constant value, while the other bits take all possible values. Some recent works [17, 11] extended this further and built M still as an affine space, but now using constant *linear combinations* of bits instead of single bits. What we aim to show here is the most general case as we are considering *any* possible set M, including sets without an affine structure.

Before stating the general results and explaining how to verify those efficiently for specific ciphers (under the assumption of independent round keys), we are going to consider simple examples to clarify the approach beforehand.

For this, we consider Boolean functions only, i.e. only a single output bit. This can be thought of as investigating a single fixed β. All the examples will be key-dependent with a key consisting of the three-bit key $k = (k_0, k_1, k_2)$.

Example 1 (Missing High-Degree Terms). As a first example, let $f_k : \mathbb{F}_2^3 \to \mathbb{F}_2$ be given as

$$f_k(x_0, x_1, x_2) = k_1 x_0 x_1 + k_1 x_0 + x_1 + (k_1 k_2 + k_3) x_2$$

While this function reaches the maximal degree (for a balanced function), it clearly does not satisfy the condition that $\sum_{x \in M} f_k(x)$ is key-dependent for any non-trivial M. Indeed, considering simply $M = \{000\}$ leads to a key-independent sum, simply as the constant term of f_k is key-independent. When considering a version of f_k using an additional whitening key $h = (h_0, h_1, h_2)$ defined as $f_{h,k}(x) = f_k(x + h)$, we get the polynomial expression

$$\begin{aligned} f_{h,k}(x) &= k_1 x_0 x_1 + (k_1 h_1 + k_1) x_0 + (k_1 h_0 + 1) x_1 \\ &\quad + (k_1 k_2 + k_3) x_2 + (k_1 k_2 h_2 + k_1 h_0 h_1 + k_1 h_0 + k_3 h_2 + h_1) \end{aligned}$$

which now does contain a key-dependent constant term, but choosing M as $M' = \{000, 001, 100, 101\}$ leads to $\sum_{x \in M'} f_{h,k}(x) = 0$ again. So while lower degree integral attacks might be avoided by adding whitening keys, high degree attacks remain unchanged. This is due to the fact that whitening keys do not affect the coefficients of monomials of maximal degree.

Example 2 (Linearly Dependent High-Degree Terms). Consider now g_k as

$$g_k(x_0, x_1, x_2) = k_0 x_0 x_1 + k_1 x_0 x_2 + (k_0 + k_1) x_1 x_2.$$

Now, all quadratic terms are present. While for g_k itself, there are key-independent coefficients, e.g. the constant term, this is not the case for $g_{h,k}$ defined as $g_{h,k}(x) = g_k(x + h)$. The corresponding expression is

$$\begin{aligned} g_{h,k}(x) &= \sum_u \lambda_u(k) \\ &= k_0 x_0 x_1 + k_1 x_0 x_2 + (k_0 h_1 + k_1 h_2) x_0 + (k_0 + k_1) x_1 x_2 \\ &\quad + (k_0 h_0 + k_0 h_2 + k_1 h_2) x_1 + (k_0 h_1 + k_1 h_0 + k_1 h_1) x_2 \\ &\quad + k_0 h_0 h_1 + k_0 h_1 h_2 + k_1 h_0 h_2 + k_1 h_1 h_2 \end{aligned}$$

and contains every monomial of degree smaller than n with a key-dependent coefficient. However, there are still sets M such that the corresponding sum is key-independent. For this example, there are exactly two non-trivial sets namely

$$M_0 = \{000, 110, 011, 101\} \text{ and } M_1 = \mathbb{F}_2^3 \setminus M_0$$

which yield to constant sums. Concretely we have

$$\sum_{x \in M_0} g_{h,k}(x) = \sum_{x \in M_1} g_{h,k}(x) = 0.$$

The reason for this is that the coefficients of g_k (and thus of $g_{h,k}$) of the monomials x_0x_1, x_0x_2 and x_1x_2 are linearly dependent polynomials. Indeed, the set can be written as $M_0 = \mathcal{U}(\{110, 101, 011\})$. Thus, the sum can be written as

$$\sum_{x \in \mathcal{U}(\{110,101,011\})} g_{h,k}(x) = \sum_{u \in \{110,101,011\}} \lambda_u(k) = k_0 + k_1 + (k_0 + k_1),$$

that is the sum of the linearly dependent coefficients.

Example 3 (Linearly Independent High-Degree Terms). A slight modification of the second example is given by

$$\ell_k(x) = k_0x_0x_1 + k_1x_0x_2 + k_0k_1x_1x_2$$

and the version with whitening keys leads to

$$\begin{aligned}\ell_{h,k}(x) = {} & k_0x_0x_1 + k_1x_0x_2 + (k_0h_1 + k_1h_2)x_0 + k_0k_1x_1x_2 + (k_0k_1h_2 + k_0h_0)x_1 \\ & + (k_0k_1h_1 + k_1h_0)x_2 + k_0k_1h_1h_2 + k_0h_0h_1 + k_1h_0h_2.\end{aligned}$$

As can be checked by running through all possible non-empty sets of size less than eight, none of the corresponding sums will be key-independent.

The reason why the last example does not lead to any integral distinguishers is, as we will elaborate in general next, that $\ell_k(x)$ (i) contains all monomials of degree $n-1$ and (ii) the corresponding coefficients are linearly independent polynomials.

Considering a Single Output Bit. For two vectors $u, v \in \mathbb{F}_2^n$, we define (as usually in this context)

$$u \succeq v \quad \Leftrightarrow \quad (v_i = 1 \Rightarrow u_i = 1).$$

Lemma 1. *Let $f_k : \mathbb{F}_2^n \to \mathbb{F}_2$ be a family of functions with ANF*

$$f_k(x) = \sum_{u \in \mathbb{F}_2^n} p_u(k)x^u$$

and consider a version of f_k with an additional pre-whitening key k_0, i.e.

$$f_{k,k_0}(x) := f_k(x+k_0) = \sum_{v\in\mathbb{F}_2^n} q_v(k,k_0)x^v$$

Then we have

$$q_v(k,k_0) = \sum_{u\succeq v} p_u(k)k_0^{u\oplus v}$$

Proof. We express $q_v(k,k_0)$ in terms of p_u. We get

$$\begin{aligned} f_{k,k_0}(x) = f_k(x+k_0) &= \sum_{u\in\mathbb{F}_2^n} p_u(k)\,(x+k_0)^u \\ &= \sum_{u\in\mathbb{F}_2^n} p_u(k)\left(\sum_{v\preceq u} x^v k_0^{u\oplus v}\right) = \sum_{v\in\mathbb{F}_2^n}\left(\sum_{u\succeq v} p_u(k)k_0^{u\oplus v}\right)x^v \end{aligned}$$

□

Next, we show a sufficient criterion to ensure that all the polynomials $q_v(k,k_0)$ are linearly independent (for $v \neq (1,\dots,1)$). For this, we denote by u_i the vector in $\mathbb{F}_2^n$ of weight $n-1$ such that its ith position is zero. That is, u_i is the bitwise complement of the ith unit vector.

Theorem 2. *Let f_k and f_{k,k_0} be defined as above. If the polynomials $p_{u_i}(k)$ are linearly independent and $p_{(1,\dots,1)}(k) = 0$, then all polynomials*

$$\{q_v(k,k_0) \mid v \in \mathbb{F}_2^n \setminus \{1\}\}$$

are linearly independent.

Proof. Assume there are coefficients $\alpha_v \in \mathbb{F}_2$ such that

$$T = \sum_{v\in\mathbb{F}_2^n\setminus\{1\}} \alpha_v q_v(k,k_0) = 0.$$

We have to show that this implies $\alpha_v = 0$ for all v. We first rewrite this as

$$\begin{aligned} T &= \sum_{v\in\mathbb{F}_2^n\setminus\{1\}} \alpha_v q_v(k,k_0) = \sum_{v\in\mathbb{F}_2^n\setminus\{1\}} \alpha_v\left(\sum_{u\succeq v} p_u(k)k_0^{u\oplus v}\right) \\ &= \sum_{v\in\mathbb{F}_2^n\setminus\{1\}} \alpha_v\left(\sum_{v\oplus w\succeq v} p_{v\oplus w}(k)k_0^w\right) = \sum_{v\in\mathbb{F}_2^n\setminus\{1\}} \alpha_v\left(\sum_{\substack{w\in\mathbb{F}_2^n \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} p_{v\oplus w}(k)k_0^w\right) \\ &= \sum_{w\in\mathbb{F}_2^n}\left(\sum_{\substack{v\in\mathbb{F}_2^n\setminus\{1\} \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} \alpha_v p_{v\oplus w}(k)\right)k_0^w \end{aligned}$$

Here, we denote by $\mathrm{Sup}(x)$ the set of non-zero bit positions, that is

$$\mathrm{Sup}(x) = \{i \mid x^{(i)} = 1\}.$$

The above implies that $T = 0$ if and only if for all $w \in \mathbb{F}_2^n$, we have

$$T(w) := \sum_{\substack{v\in\mathbb{F}_2^n\setminus\{1\} \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} \alpha_v p_{v\oplus w}(k) = 0.$$

We show that this implies $\alpha_v = 0$ by induction on the weight of v.

For $\mathrm{wt}(v) = 0$, that is v being the all-zero vector, consider a vector w with $\mathrm{wt}(w) = n-1$. That is, w is one of the vectors u_i. The set of vectors v such that $\mathrm{Sup}(w) \cap \mathrm{Sup}(v) = \emptyset$ contains only the all-zero vector and e_i. We thus get,

$$T((1,\dots,1)) = \alpha_{(0,\dots,0)} p_{u_i}(k) + \alpha_{e_i} p_{((1\dots,1))}(k) = 0.$$

By assumption $p_{((1\dots,1))}(k)$ is zero, while $p_{u_i}(k)$ is not, thus $\alpha_{(0,\dots,0)} = 0$.

$\mathrm{wt}(v) = t \leq n-2$*:* We now assume by induction that $\alpha_v = 0$ for all v of weight smaller than t. We consider a vector w of weight $\mathrm{wt}(w) = n-(t+1)$. Then, the set of vectors such that $\mathrm{Sup}(w)\cap\mathrm{Sup}(v) = \emptyset$ contains one vector of weight $t+1$, vectors of weight exactly t, and vectors of weight smaller than t. We split $T(w)$ accordingly as follows

$$T(w) = \sum_{\substack{\mathrm{wt}(v)=t+1 \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} \alpha_v p_{v\oplus w}(k) + \sum_{\substack{\mathrm{wt}(v)=t \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} \alpha_v p_{v\oplus w}(k) + \sum_{\substack{\mathrm{wt}(v)<t \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} \alpha_v p_{v\oplus w}(k)$$

By the induction hypothesis, the last part is zero, as $\alpha_v = 0$ for $\mathrm{wt}(v) < t$. Furthermore, the first part is zero as here $v \oplus w = (1,\dots,1)$ and $p_{((1\dots,1))}(k)$ is zero, which implies

$$T(w) = \sum_{\substack{\mathrm{wt}(v)=t \\ \mathrm{Sup}(w)\cap\mathrm{Sup}(v)=\emptyset}} \alpha_v p_{v\oplus w}(k) = 0.$$

Now here $v\oplus w$ is of weight $n-1$ and thus is one of the vectors u_i. By assumption, the polynomials p_{u_i} are linearly independent and thus $T(w) = 0$ implies $\alpha_v = 0$ for all v of weight t such that $\mathrm{Sup}(w)\cap\mathrm{Sup}(v) = \emptyset$. As w was arbitrary of weight $n-(t+1)$ this means that $\alpha_v = 0$ for all v of weight t. □

This finally implies, as a corollary, that there are no key-independent integral distinguishers in a very general sense. Any sum of output values, except for the empty sum and summing all outputs, is key-dependent. More precisely,

Corollary 2. *Let f_k and f_{k,k_0} be defined as above and assume that the polynomials $p_{u_i}(k)$ are non-constant linearly independent, and $p_{(1,\dots,1)}(k) = 0$. Then, for any proper non-empty subset $M \subset \mathbb{F}_2^n$ the sum*

$$\sum_{x\in M} f_{k,k_0}(x)$$

depends on the value of the key (k, k_0).

Proof. It holds that

$$\sum_{x \in M} f_{k,k_0}(x) = \sum_{\ell \in \mathcal{U}(M)} \left(\sum_{x \preceq \ell} f_{k,k_0}(x) \right) = \sum_{\ell \in \mathcal{U}(M)} q_\ell(k, k_0).$$

As M is non-empty and not the full space, $\mathcal{U}(M)$ contains elements of weight less than n. Then, the theorem above implies that the sum is non-zero viewed as a polynomial in k and k_0 and thus key-dependent as claimed. □

We like to remark that this property is not only sufficient but also necessary. Indeed, if the polynomials p_{u_i} are linearly dependent, there exist a linear combination that is constant zero. As the whitening key does not influence the value of the monomials of degree $n-1$, this directly leads to a set M corresponding to a constant, i.e. key-independent, sum.

Linear Combinations of Output Bits. Let us next consider a family of vectorial Boolean functions E_k, with the most important example being a block cipher. We want to extend the previous arguments to this case. Here, we want to guarantee that any non-trivial linear combination of output bits is key dependent. This can be done as follows.

Consider $E_k \colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a family of functions with ANF

$$E_k(x) = \sum_{u \in \mathbb{F}_2^n} P_u(k) x^u$$

where now $P_u(k)$ is a vector in $\mathbb{F}_2^n$. A linear combination of output bits is specified by fixing a $\beta \in \mathbb{F}_2^n$ and considering

$$\langle \beta, E_k(x) \rangle = \langle \beta, \sum_{u \in \mathbb{F}_2^n} P_u(k) x^u \rangle = \sum_{u \in \mathbb{F}_2^n} \langle \beta, P_u(k) \rangle x^u$$

If we can ensure that, for each fixed non-zero β, the polynomials $\langle \beta, P_u(k) \rangle$ fulfill the conditions of Corollary 2, we ensured that no integral distinguisher is possible on any linear combination of output bits.

So, for any non-zero β, the polynomials $\langle \beta, P_{u_i}(k) \rangle$ should be linearly independent and $\langle \beta, P_{(1,\dots,1)}(k) \rangle = 0$. The latter is true if and only if $P_{(1,\dots,1)}(k) = 0$. Note that in the case of a block cipher, since we need the block cipher to be invertible, it can be at most of degree $n-1$ and thus we are guaranteed to have $P_{(1,\dots,1)}(k) = 0$. For the former, we require that

$$\sum_i \alpha_i \langle \beta, P_{u_i}(k) \rangle = 0 \ , \ \alpha_i \in \mathbb{F}_2$$

implies that all α_i are equal to zero. This can be rewritten as

$$\begin{aligned} 0 = \sum_i \alpha_i \langle \beta, P_{u_i}(k) \rangle &= \sum_i \alpha_i \sum_j \beta^{(j)} P_{u_i}^{(j)}(k) \\ &= \sum_{i,j} \alpha_i \beta^{(j)} P_{u_i}^{(j)}(k) = \sum_{i,j} \gamma_{i,j} P_{u_i}^{(j)}(k) \end{aligned}$$

with $\gamma_{i,j} = \alpha_i \beta^{(j)} \in \mathbb{F}_2$. One way to simplify this equation is to require something (potentially significantly) stronger, namely that all $n \times n$ polynomials

$$p_{i,j}(k) := P^{(j)}_{u_i}$$

are linearly independent.

On Key-Patterns and Matrices. Asking that all the polynomials $p_{i,j}$ are linearly independent can be put into the following context for input-, output- and key-pattern. Consider the polynomials in its ANF

$$p_{i,j}(k) := P^{(j)}_{u_i} = \sum_{v \in \mathbb{F}_2^{\ell}} \lambda^{(j)}_{u_i,v} k^v .$$

The values of $\lambda^{(j)}_{u_i,v}$ are equal to the parity of the number of trails $(u_i, v) \to e_j$, that is trails with input pattern u_i, key-pattern v and output pattern e_j. If we want to show that all those polynomials are linearly independent, it is sufficient (and actually necessary) to find a set of key-patterns $v_1, \dots v_s$, with $s \geq n^2$ such that the *integral-resistance matrix*

$$\mathcal{I}(E) = \begin{pmatrix} \lambda^{(1)}_{u_1,v_1} & \lambda^{(2)}_{u_1,v_1} & & \lambda^{(n)}_{u_1,v_1} & \lambda^{(1)}_{u_2,v_1} & \lambda^{(2)}_{u_2,v_1} & & \lambda^{(j)}_{u_i,v_1} & & \lambda^{(n-1)}_{u_n,v_1} & \lambda^{(n)}_{u_n,v_1} \\ \lambda^{(1)}_{u_1,v_2} & \lambda^{(2)}_{u_1,v_2} & \cdots & \lambda^{(n)}_{u_1,v_2} & \lambda^{(1)}_{u_2,v_2} & \lambda^{(2)}_{u_2,v_2} & \cdots & \lambda^{(j)}_{u_i,v_2} & \cdots & \lambda^{(n-1)}_{u_n,v_2} & \lambda^{(n)}_{u_n,v_2} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & & \vdots & \vdots \\ \lambda^{(1)}_{u_1,v_s} & \lambda^{(2)}_{u_1,v_s} & & \lambda^{(n)}_{u_1,v_s} & \lambda^{(1)}_{u_2,v_s} & \lambda^{(2)}_{u_2,v_s} & & \lambda^{(j)}_{u_i,v_s} & & \lambda^{(n-1)}_{u_n,v_s} & \lambda^{(n)}_{u_n,v_s} \end{pmatrix}$$

has full rank. This brings us to the following proposition which we apply in Sect. 6.

Proposition 2 (INTEGRAL-RESISTANCE PROPERTY). *Let $E \colon \mathbb{F}_2^n \times \mathbb{F}_2^m \to \mathbb{F}_2^n$ be a block cipher and $\mathcal{I}(E)$ be a corresponding integral-resistance matrix. If $\mathcal{I}(E)$ has full rank and k_0 is an independent whitening key, $E_k(x + k_0)$ fulfills the integral-resistance property, i.e for every proper subset $M \subset \mathbb{F}_2^n$ and output mask $\beta \in \mathbb{F}_2^n$ the sum*

$$\sum_{x \in M} \langle \beta, E_k(x + k_0) \rangle$$

is key-dependent.

4 Guarantee for More Rounds

Even if the lower bound on the degree of an R-round block cipher is d, it is not clear that $R+1$ rounds have a degree at least d. Indeed, the next example shows that this is not only non-trivial but simply wrong in general.

Example 4. Let $F, G\colon \mathbb{F}_2^3 \to \mathbb{F}_2^3$ be permutations of degree 2 defined as follows:

$$F(x_1, x_2, x_3) \coloneqq \begin{pmatrix} x_1x_2 + x_3 \\ x_1 \\ x_2 \end{pmatrix}, \qquad G(x_1, x_2, x_3) \coloneqq \begin{pmatrix} x_1 + x_2x_3 \\ x_2 \\ x_3 \end{pmatrix}.$$

Then we can write the composition of F and G with a key addition in the middle as

$$G(F(x) + k) = G\begin{pmatrix} x_1x_2 + x_3 + k_1 \\ x_1 + k_2 \\ x_2 + k_3 \end{pmatrix} = \begin{pmatrix} x_3 + x_1k_3 + x_2k_2 + k_1 + k_2k_3 \\ x_1 + k_2 \\ x_2 + k_3 \end{pmatrix},$$

which has only degree 1 in x.

Thus, the algebraic degree can decrease if the highest-degree monomials are cancelled out by applying an additional one round. Although it is nontrivial in general, for some block ciphers (with independent round key assumption), we show that we can guarantee that the algebraic degree does not decrease. Intriguingly, as we will see in this section, this argument does not work for all choices of S-boxes. The case of minimal degree and for the strong arguments against integral distinguishers, the situation is more clear: Here, as we will detail later in this section, adding additional rounds never allows to decrease the minimal degree nor invalidates the strong argument.

4.1 More Rounds for the Algebraic Degree

We split the discussion of how to argue about the algebraic degree into parts, dealing step by step with the linear layer, a single Boolean function, and finally an entire round.

The linear layer does not change anything as both the algebraic degree as well as the minimal-degree are invariant under affine equivalence (see e.g. [9]).

Lemma 2. *Let $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be any function and $A\colon \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an affine, invertible function, then it holds that*

$$\begin{aligned} \deg(F \circ A) &= \deg(A \circ F) = \deg(F), \\ \mathrm{minDeg}(F \circ A) &= \mathrm{minDeg}(A \circ F) = \mathrm{minDeg}(F). \end{aligned}$$

To cover a layer of S-boxes, we consider the general situation of a parallel application of functions. Since the algebraic degree is the maximum degree of all output bits, it is enough to look at each output bit separately. Therefore, it is sufficient to consider the influence of an isolated S-box on the algebraic degree.

As a next step, the following theorem gives efficient to verify conditions on when appending a single Boolean function does not decrease the algebraic degree.

Theorem 3. *Let a Boolean function $f\colon \mathbb{F}_2^m \to \mathbb{F}_2$ be given and consider a round key $k \in \mathbb{F}_2^m$. Consider the algebraic normal form of the function*

$$f_k : \mathbb{F}_2^m \to \mathbb{F}_2, \quad f_k(x) = f(x + k)$$

be given as

$$f_k(x) = \sum_{u \in \mathbb{F}_2^m} p_u(k) x^u.$$

Assume that

$$p_{e_i}(k) \notin \operatorname{span}\{p_u(k) \mid u \neq e_i \text{ and } u \neq 0\},$$

that is p_{e_i} is not linearly dependent on the other coefficients when viewed as a polynomial in k. Then it holds that $\deg(f_k \circ F) \geq \deg(F_i)$ for any function $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$. That is, the degree of the ith coordinate of F is a lower bound of the algebraic degree of the composition of F with f_k.

Proof. We denote the algebraic normal form of F_i by

$$F_i(y) = \sum_{v \in \mathbb{F}_2^n} \lambda_v y^v$$

Let the degree of F_i be d. Then, there exists a vector w of weight d such that $\lambda_w = 1$. We now get

$$\begin{aligned} f_k(F(y)) &= \sum_{u \in \mathbb{F}_2^m} p_u(k) F(y)^u = p_{e_i}(k) F_i(y) + \sum_{u \neq e_i} p_u(k) F(y)^u \\ &= p_{e_i}(k) \left(\sum_{v \in \mathbb{F}_2^n} \lambda_v y^v \right) + \sum_{u \neq e_i} p_u(k) F(y)^u \\ &= p_{e_i}(k) \lambda_w y^w + \sum_{v \neq w} p_{e_i}(k) \lambda_v y^v + \sum_{u \neq e_i} p_u(k) F(y)^u. \end{aligned}$$

By expanding the expression of $F(y)^u$, the last sum can be rearranged into $\sum_{\ell \in \mathbb{F}_2^n} q_\ell(k) y^\ell$, where each $q_\ell(k)$ corresponds to a linear combination of the $p_u(k)$ for $u \neq e_i$.

$$\begin{aligned} f_k(F(y)) &= p_{e_i}(k) \lambda_w y^w + \sum_{v \neq w} p_{e_i}(k) \lambda_v y^v + \sum_{\ell \in \mathbb{F}_2^n} q_\ell(k) y^\ell \\ &= \left(p_{e_i}(k) \lambda_w + q_w(k) \right) y^w + \sum_{v \neq w} \left(p_{e_i}(k) \lambda_v + q_v \right) y^v. \end{aligned}$$

As $p_{e_i}(k)$ is linearly independent from the $p_u(k)$ for $u \neq e_i$ and q_w corresponds to such a sum, the key-dependent coefficient of y^w is non-zero. As $\operatorname{wt}(w)$ equals d we conclude that $\deg(f_k \circ F) \geq d$. □

If a given Boolean function f fulfills the conditions of the above theorem, we say that f *preserves the degree of its ith input component.*

Example 5. Consider the Boolean function

$$f(x) = x_0 x_1 x_2 + x_0 x_1 + x_0 x_2 + x_0 + x_1 x_2 + x_2 x_3 + x_3 + 1.$$

Then

$$\begin{aligned} f_k(x) = x_0x_1x_2 &+ (k_2+1)x_0x_1 + (k_1+1)x_0x_2 + (k_1k_2+k_1+k_2+1)x_0 \\ &+(k_0+1)x_1x_2 + (k_0k_2+k_0+k_2)x_1 + x_2x_3 + (k_0k_1+k_0+k_1+k_3)x_2 \\ &+(k_2+1)x_3 + k_0k_1k_2 + k_0k_1 + k_0k_2 + k_0 + k_1k_2 + k_2k_3 + k_3 + 1. \end{aligned}$$

The non-zero non constant coefficients to consider are

$$\begin{array}{lll} p_{1110}(k) = 1 & p_{1100}(k) = k_2+1 & p_{1010}(k) = k_1+1 \\ p_{0110}(k) = k_0+1 & p_{1000}(k) = k_1k_2+k_1+k_2+1 & p_{0100}(k) = k_0k_2+k_0+k_2 \\ p_{0011}(k) = 1 & p_{0010}(k) = k_0k_1+k_0+k_1+k_3 & p_{0001}(k) = k_2+1. \end{array}$$

While $p_{1000}, p_{0100}, p_{0010}$ cannot be expressed as linear combination of the others (as the quadratic term is unique in the non-constant terms) p_{0001} actually can (as it is simply equal to p_{1100}). So in this case we get that

$$\deg(f_k \circ F) \geq \max\{\deg(F_0), \deg(F_1), \deg(F_2)\},$$

and thus f preserves the degree of its first (x_0), second (x_1) and third (x_2) input. However, it does not always preserve the degree of its last input. Indeed, consider F on four inputs y_0, y_1, y_2, y_3 such that

$$x_0 = F_0 = y_0y_1 \quad x_1 = F_1 = y_2y_3 \quad x_2 = F_2 = y_2 \quad x_3 = F_3 = y_0y_1y_2y_3.$$

Then $\deg(F) = 4$ while $\deg(f_k \circ F) = 2$.

Now, this theorem can be used to bound the algebraic degree as summarized in the next corollary. For this, we denote by $\mathbb{S}_k^{(r)}$ an S-box layer (the r-fold parallel application of S) together with an independent round key addition k, i.e.

$$\mathbb{S}_k(x_1, \ldots, x_r) = (S(x_1+k_1), \ldots, S(x_r+k_r)).$$

Corollary 3. *Consider an S-box $S\colon \mathbb{F}_2^m \to \mathbb{F}_2^m$. If, for each $1 \leq i \leq m$ there exists a coordinate function S_j such that S_j preserves the degree of its ith input, then for all functions $F\colon \mathbb{F}_2^{mr} \to \mathbb{F}_2^{mr}$, we have* $\deg(\mathbb{S}_k^{(r)} \circ F) \geq \deg(F)$.

Results. Based on Corollary 3, we computed which S-boxes preserve the degree. For a single S-box, that can be checked efficiently for all practical relevant values of n. The sage code that automatically checks the properties is given as supplementary material. Especially, if we go through all 302 representatives of all affine equivalence classes for 4-bit bijective S-boxes, there are 244 such S-boxes that preserve the algebraic degree, while 58 do not. Some specific examples are that the S-box of `GIFT` and `PRESENT` preserve the algebraic degree, while it is not the case for the S-box of `CRAFT`, `SKINNY-64` and `SKINNY-128`. We also tested the inverse mapping over $\mathbb{F}_{2^n}$ for $n = 3$ to $n = 8$ (e.g. the AES S-box), and each of them also preserves the algebraic degree. So in particular we see that any bound on the algebraic degree of the AES implies the same bound for the full AES.

4.2 More Rounds for the Minimal Degree

To bound the algebraic degree, we had to show that for any input bit i, there is at least on output bit that preserves the degree of its ith input. For the minimal degree, the situation is different in two ways. On the one hand, we have to ensure more. As we want to bound the minimal degree, we have to bound the degree of any linear combination of output bits. On the other hand, as we are going to assume that F has a given minimal degree, we know that preserving the degree of any linear combination of its input is sufficient. Finally, there is no direct equivalence to Corollary 3 for minimal degree. However, the next theorem (and its proof) shows that this can be dealt with. Indeed the case for minimal degree is significantly easier.

Theorem 4. *Let $F\colon \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a function and $S\colon \mathbb{F}_2^m \to \mathbb{F}_2^t$ be a function such that for any non-zero $\beta \in \mathbb{F}_2^t$ the component function $\langle \beta, S \rangle$ is non-constant. If we denote by S_k the function $S_k(x) = S(x+k)$ parameterized by a key $k \in \mathbb{F}_2^m$ then the minimal degree never decreases, that is*

$$\mathrm{minDeg}(S_k \circ F) \geq \mathrm{minDeg}(F).$$

Proof. Let $\beta \in \mathbb{F}_2^t$ be a non-zero vector. We consider the component function $f(x) = \langle \beta, S(x) \rangle$ and show that the minimal degree of $f_k \circ F$ is at least the minimal degree of F. For this, consider a monomial of maximal degree in the algebraic normal form of f, without loss of generality $x_0 \cdots x_{d-1}$. For f_k, this will in particular generate the term $k_1 \cdots k_{d-1} x_0$. This key-monomial $k_1 \dots k_{d-1}$ could also appear as part of the coefficients of *different linear* monomials x_i, but not in coefficients of non-linear monomials. Thus f_k can be written as

$$f_k(x) = k_1 \cdots k_{t-1} \langle \gamma, x \rangle + g_k(x)$$

with a non-zero $\gamma \in \mathbb{F}_2^m$ and a polynomial g such that $k_1 \cdots k_{t-1}$ cannot be expressed as a linear combination of its coefficients. That is, the term $k_1 \cdots k_{t-1} \langle \gamma, x \rangle$ cannot cancel in the algebraic normal form of $f_k \circ F$. Finally the degree of $\langle \gamma, x \rangle = \langle \gamma, F(y) \rangle$ is bounded by the minimal degree of F by definition. □

4.3 More Rounds for Strong Arguments

The strong arguments extends to more rounds automatically without any additional requirements. Indeed, consider the composition $E_k \circ F$ where E_k fulfills the strong arguments and F is a fixed permutation. Note that if F is actually key-dependent (using an independent key-value), the same argument applies. Considering any non-empty set $M \subset \mathbb{F}_2^n$, and any β, we get

$$\sum_{x \in M} \langle \beta, E_k(F(x)) \rangle = \sum_{y \in F(M)} \langle \beta, E_k(y) \rangle,$$

which depends on k as any sum does for E_k. Note that in the context of block ciphers where the round function is (most of the time) identical for each round

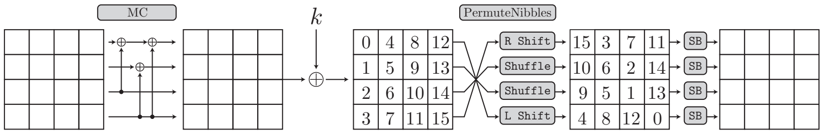


Fig. 2. Round function of `CRAFT`.

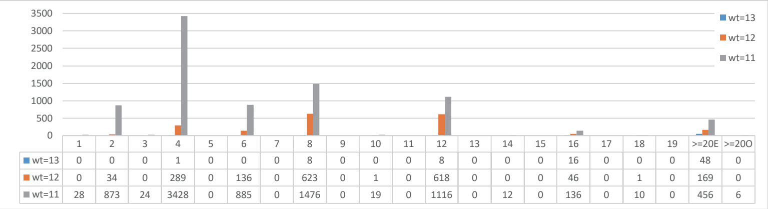
(assuming independent round keys and thus absorbing round constants), even though this arguments is valid when adding an arbitrary amount of rounds before E_k, it also includes the case where we add an arbitrary amount of rounds *after* E_k. Indeed in this context, $F^r \circ E_k = E_{k'} \circ F^r$, where F is the round function. If the previous arguments hold for $E_k \circ F^r$, it thus automatically holds for $F^r \circ E_k$, i.e. any sum $\sum_{x \in M} \langle \beta, F^r(E_k(x)) \rangle$ is key-dependent.

5 Improvements of Efficiency by using Equivalent S-boxes

The core part to guarantee the lower bound of degrees is to find a key pattern where the number of division trails from a plaintext to a ciphertext is odd. To find such a key pattern, the trail extension technique was proposed in [12]. A key pattern is generated from the ciphertext side round by round as outlined in Sect. 2.3. In the end, lower bounds on the (minimum) degree for round-reduced variants of `SKINNY-64`, `GIFT-64`, and `PRESENT` could be efficiently computed. On the other hand, it is open whether the trail extension technique can find such key patterns for other ciphers. We used the tool provided in [12] and modified it for the block cipher `CRAFT`. As a result, we failed to find key patterns in spite of the similarity to `SKINNY`. This is because the round function of `CRAFT` has fundamental problems to disturb trail extensions. In practice, the round function of `SKINNY-64` or `GIFT-64` is significantly suited to the trail extension, and the trail extension is unlikely succeeded in general.

We are only interested in the parity of the number of trails for a fixed pattern, but in practice, we cannot know the parity unless all trails are enumerated. Therefore, the feasibility highly depends on the number of trails, which we try to keep significantly small. When the key pattern is sequentially generated in the trail extension, the number of trails must be kept small throughout all iterations.

Let us focus on the `CRAFT` round function (see Fig. 2), in particular, a super S-box, which consists of four 4-bit S-boxes, MixColumns, and four 4-bit S-boxes. Let u, v, and w be the input, key, and output patterns on the super S-box, and the trail extension technique generates (u, v) from w such that the Hamming weights of u and v are as high as possible. As an example, we enumerated all (u, v) that can propagate to $w =$ `0x1200`, where the two S-boxes and one MixColumns are independently evaluated. Then, $wt(u) + wt(v) = 13$ is the maximum choice, e.g., ($u =$ `0x7777`$, v =$ `0x2000`) can propagate to $w =$ `0x1200`. Unfortunately, this trail is not available because there are 4 different trails satisfying $(u, v) \rightarrow w$.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	>=20E	>=20O
wt=13	0	0	0	1	0	0	0	8	0	0	0	8	0	0	0	16	0	0	0	48	0
wt=12	0	34	0	289	0	136	0	623	0	1	0	618	0	0	0	46	0	1	0	169	0
wt=11	28	873	24	3428	0	885	0	1476	0	19	0	1116	0	12	0	136	0	10	0	456	6

Fig. 3. The number of (u, v) and their numbers of trails, output pattern is `0x1200`. For example, when $wt(u) + wt(v) = 11$, there are 3428 (u, v) whose number of trails is 4. When the number of trails exceeds 20, the labels ≥20E and ≥20O are used for even parity and odd parity, respectively.

Figure 3 summarizes the number of (u, v) and their numbers of trails when $w =$ `0x1200`. The numbers of (u, v) with $wt(u) + wt(v) = 13$ and $wt(u) + wt(v) = 12$ are 81 and 1917, respectively, but there is no (u, v) whose number of trails is odd. When $wt(u) + wt(v) = 11$, there are 28 (u, v) whose number of trails is 1, but the number is very few. In other words, the choice of the trail extension is very limited among all trails. Even if such a rare propagation is adopted, after several rounds, it is unlikely to restrict the number of trails to a size that is able to handle in practice. This trend is not limited to $w =$ `0x1200`. Indeed, a preferable propagation is very rare for many output pattern. This is our heuristic explanation why the trail extension cannot find a key pattern for `CRAFT`.

5.1 Replacement to Equivalent S-box

We tackle the problem to expand the class of ciphers that we can prove a lower bound on the degree. The core of the problem is having too many trails. Therefore, we propose a new method to decrease the number of trails fundamentally. Generally, this method is based on rewriting the ciphers specification (potentially up to a linear change of plaintext and ciphertext and a different key-scheduling). We call such ciphers equivalent. More specifically, we replace the S-box in such a way that we get the exact same cipher. Thus, while we keep the cipher identical, its behaviour with respect to division trails might well change.

The first important remark is that even constant addition changes the propagation table of the division property unlike the differential distribution table or linear approximation table. Under the key-alternating ciphers, constant addition before/after S-boxes results in a different representation of the original cipher because such constant addition can be included in the round key addition.

Proposition 3 (Equivalent S-box for key-alternating ciphers). *Replacing an S-box S in a key-alternating cipher with an S-box S' : $S'(x) = S(x \oplus c_{in}) \oplus c_{out}$ results in an equivalent cipher under the independent round key.*

To demonstrate the effect of the equivalent S-box, we use the CRAFT S-box as an example. There are 76 possible transitions in the propagation table (see the left one in Table 2). On the other hand, in $S'(x) = S(x \oplus \texttt{0x7}) \oplus \texttt{0x7}$, there are only 56 possible transitions. The total number of trails decreases from 76 to 56. We can expect that the use of S' instead of S decreases the number of trails from a plaintext to a ciphertext.

For ciphers whose linear layer consists of word-wise XOR and word-wise shuffle such as CRAFT or SKINNY, there is a more wide equivalent class.

Proposition 4 (Equivalent S-box for ciphers with word-wise linear layer). *Replacing an S-box S in a key-alternating cipher whose linear layer consists of word-wise XOR and word-wise shuffle with an S-box S' : $S'(x) = A^{-1} \times S(A \times (x \oplus c_{in})) \oplus c_{out}$ results in an equivalent cipher under the independent round key.*

An invertible linear transformation, denoted by $A \times x$, is applied before the S-box, and its inverse A^{-1} is multiplied after the S-box. Unlike in the generally studied affine-equivalent class, we limit the second linear transformation to the inverse of the former linear transformation. Since multiplication of A^{-1} and word-wise XOR/shuffle are commutative, the multiplication of A^{-1} can be moved at the beginning of the next round. Then, as $A \times A^{-1}$ is the identity, we see that the ciphers are indeed equivalent.

The number of linear transformations is 20160 for 4-bit S-boxes. Therefore, there are (at most) $20160 \times 2^4 \times 2^4 \approx 2^{22.23}$ equivalent S-boxes, and we can choose a preferable S-box to prove the lower bound. Note that the target cipher also changes to the cipher whose plaintext and ciphertext is linearly transformed, but it never affects the algebraic degree, the minimum degree, and of course, the claim of no integral distinguisher[3].

5.2 Choice of Preferable Equivalent S-boxes

The most important problem is how to choose a preferable S-box from the equivalent class. Intuitively, the lower number of possible trails, the better. However, as far as we tried, this problem is not so simple, and choosing an S-box whose propagation table has the following property is better. In the following, let u and v be an input pattern and an output pattern, respectively.

- For any u with $wt(u) = n - 1$, the possible output pattern v is uniquely determined when $wt(u) - wt(v)$ is maximized.
- For any v with $wt(v) = 1$, the possible input pattern u is uniquely determined when $wt(u) - wt(v)$ is maximized.
- The number of possible transitions is as small as possible.

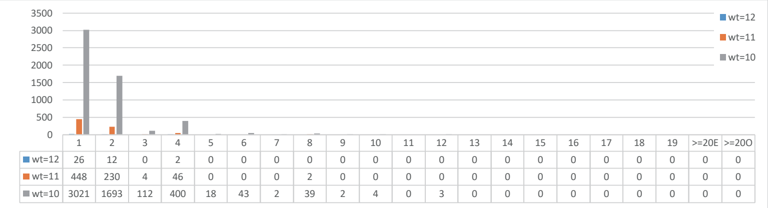
Note that these conditions are heuristically found, and whether adopting these conditions is optimal or not is an open question. As a consequence, the use of S-boxes satisfying these conditions allows us to prove the lower bound of CRAFT.

[3] Similar technique is already known in [17, 11], but there is significant difference. In previous works, a linear transformation is applied to S-boxes in the first and the last rounds only. In our proposal, it is applied to all S-boxes in the middle rounds.

Table 2. Propagation table for the `CRAFT` S-box. The left is the table of the original S-box. The right is the table of the equivalent S-box described in Example 6

	0	1	2	4	8	3	5	6	9	A	C	7	B	D	E	F
0	x			x	x						x					
1			x	x						x	x					
2		x					x		x					x		
4			x					x		x					x	
8				x							x					
3					x	x	x		x	x				x		
5		x	x			x		x	x	x			x		x	
6						x						x	x			x
9		x	x	x					x	x	x					
A					x		x		x					x		
C			x		x			x		x					x	
7		x		x			x	x	x							x
B		x		x	x	x				x				x		
D						x		x		x			x		x	
E		x		x	x	x						x	x			x
F																x

	0	1	2	4	8	3	5	6	9	A	C	7	B	D	E	F
0	x															
1				x	x						x					
2			x		x					x						
4		x			x				x							
8					x											
3		x				x	x	x		x	x	x				
5		x	x					x	x		x					
6			x						x	x						
9			x		x			x		x					x	
A			x	x						x						
C		x	x						x							
7								x					x	x	x	x
B							x					x			x	
D			x				x		x	x					x	
E		x					x			x	x			x		
F																x



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	>=20E	>=20O
wt=12	26	12	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
wt=11	448	230	4	46	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
wt=10	3021	1693	112	400	18	43	2	39	2	4	0	3	0	0	0	0	0	0	0	0	0

Fig. 4. The number of (u, v) and their numbers of trails in the `CRAFT` super S-box using the equivalent S-box described in Example 6. Output pattern is `0x1200`.

Example 6. The following S-box

$$S' : \texttt{0x0}, \texttt{0xC}, \texttt{0xA}, \texttt{0x7}, \texttt{0x9}, \texttt{0x6}, \texttt{0x1}, \texttt{0xF}, \texttt{0x8}, \texttt{0xE}, \texttt{0x4}, \texttt{0x3}, \texttt{0x2}, \texttt{0x5}, \texttt{0xD}, \texttt{0xB}$$

is equivalent to the original S-box of `CRAFT` and satisfies the three conditions. Note that S' is generated from the original S-box S as follows:

$$S'(x) = A^{-1} \times S(A \times (x \oplus \texttt{0x5})) \oplus \texttt{0xD}, \qquad A = \begin{pmatrix} 1&0&1&1 \\ 0&0&0&1 \\ 0&1&0&0 \\ 1&0&0&1 \end{pmatrix},$$

where $x = x_4 \| x_3 \| x_2 \| x_1$ is identical to the transpose of (x_4, x_3, x_2, x_1), i.e., $(x_4, x_3, x_2, x_1)^T$. Table 2 shows the comparison between the propagation tables of the original S-box and the equivalent S-box. The table of the equivalent S-box is more sparse than that of the original one. Six propagations labeled in red color correspond to the first and second conditions. For example, when $v =$ `0x4`, u with maximum Hamming weight is uniquely determined to `0xA`. As another example, when $u =$ `0xD`, v with minimum Hamming weight is uniquely determined to `0x2`.

Finally, we test the same experiment using the super S-box, i.e., we enumerate all (u, v) that can propagate to $w =$ `0x1200`. Figure 4 summarizes the number of (u, v) and their numbers of trails. Unlike the original one shown in Fig. 3, the majority of possible propagations has only one trail.

6 Applications

In this section we are going to apply our results, i.e. how to give stronger arguments to a set of ciphers. For each cipher, we briefly explain some specific observations and improvements. The results for all ciphers are given in Table 1.

In all ciphers whose block length is n, we need n^2 key patterns to guarantee no integral distinguisher, and the integral-resistance matrix has n^4 entries, i.e., 2^{24} and 2^{28} on 64-bit and 128-bit block ciphers, respectively. To compute these entries efficiently, we use a key pattern in which key patterns for the 1st and last rounds are non-zero if it is possible. Then, almost all entries in the integral-resistance matrix must be 0, and the integral-resistance matrix has the form of a diagonal block matrix. When all block matrices have full rank, the integral-resistance matrix has full rank. Another important remark is that even if some entries are not determined, we can still prove that the integral-resistance matrix has full rank. For example, since the following matrix

$$\begin{pmatrix} 1 & 0 & \star \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

has full rank independent of $\star$, we do not need to determine the entry $\star$. Note that [1] observed that using the Convex Hull technique to modelize S-boxes [23] in MILP can sometimes leads to inconsistencies. We double checked and made sure that this phenomenon does not happen for the S-boxes that we use.

6.1 Applications to CRAFT

CRAFT is a lightweight tweakable block cipher published in ToSC 2019 [5]. The block length is 64 bits, and 4-bit S-boxes are used as the nonlinear operation. On our proof, we assume independent tweakeys each round.

The designers of CRAFT showed an 13-round integral distinguisher [5] in the single tweak-key setting. For the tightness of results, our goal is therefore to show that 14 rounds (and more) do not have any integral distinguisher under the independent-tweakey assumption.

Figure 2 shows the round function of CRAFT. The propagation of the division property for the SB and MixColumns are independently modeled, where MixColumns is regarded as 16 parallel applications of 4-bit to 4-bit linear transformation. Moreover, the first round consists of the SB only. As shown in Sect. 5, we use an equivalent S-box instead of the original S-box to improve the efficiency. Note that using the original S-box did neither allow to extend trails nor to find key patterns whose number of trails is odd. Therefore, our new proposal using an equivalent S-box is necessary to handle CRAFT.

We start with the proof of having no integral distinguisher in 14 rounds. To prove it, we need at least 4096 key patterns whose corresponding 4096 × 4096 integral-resistance matrix has full rank. Generating 4096 key patterns is time consuming. Besides, it is unlikely that the integral-resistance matrix becomes full rank when 4096 key patterns are generated without care. A systematic strategy is required to efficiently generate such key patterns.

As a first improvement, we exploit the symmetry property of CRAFT. Denote

$$T\left(\begin{pmatrix} s_0 & s_4 & s_8 & s_{12} \\ s_1 & s_5 & s_9 & s_{13} \\ s_2 & s_6 & s_{10} & s_{14} \\ s_3 & s_7 & s_{11} & s_{15} \end{pmatrix}\right) = \begin{pmatrix} s_8 & s_{12} & s_0 & s_4 \\ s_9 & s_{13} & s_1 & s_5 \\ s_{10} & s_{14} & s_2 & s_6 \\ s_{11} & s_{15} & s_3 & s_7 \end{pmatrix},$$

then the round function R of CRAFT (excluding constant and tweakey additions) fulfills $R(T(s)) = T(R(s))$. As we can ignore the impact of keys and constant addition under the independent tweakey assumption, rotating by two columns is thus invariant for the computation of key patterns. Therefore, once we find a key pattern, the key pattern transformed by the symmetry property is also available. Thanks to this property, $4096/2 = 2048$ key patterns are enough.

As a second improvement, we use key patterns that share the same division trail in the middle part. Concretely, we first construct 12-round input/key/output patterns whose number of trails is odd, and then, the trail is systematically extended both forward and backward direction by 1 round, respectively. Then, we can generate 2048 key patterns only from 32 12-round patterns. As a consequence, we can generate 4096 key patterns whose corresponding integral-resistance matrix has full rank. As shown in Sect. 4.3, once we can generate an integral-resistance matrix of full rank for 14 rounds, it also guarantees no integral distinguisher in 14 rounds and higher.

6.2 Applications to SKINNY-64

SKINNY is a lightweight block cipher published at CRYPTO'16 [4]. There are two different version of SKINNY (64-bit block SKINNY-64 and 128-bit block SKINNY-128).

In [12], the lower bounds of the algebraic degree and the minimum degree reach the maximum, i.e., 63, in 10-round SKINNY-64 and 11-round SKINNY-64, respectively. It also shows that 13-round SKINNY-64 has 64 maximum degree monomials. On the contrary, the best integral distinguisher reaches 11 rounds [11][4]. Note that SKINNY does not have the pre-whitening key, the 11-round integral distinguisher can be extended to a 12-round one for free.

Our goal is to show that 13 rounds and more never have integral distinguishers under the assumption that each round-tweakey is independent and they are XORed to the full state. Again, SKINNY does not have a pre-whitening key.

[4] In the 11-round distinguisher shown in [11], each tweakey is not XORed to the full state. However, we confirmed that there are 11-round distinguishers even when each tweakey is XORed to the full state.

Therefore, to prove no integral distinguisher in 13 rounds, we need to construct a full-rank integral-resistance matrix for 12 rounds.

Similarly to CRAFT, the SC and MixColumns are independently modeled, where MixColumns is regarded as 16 parallel applications of 4-bit to 4-bit linear transformation. Moreover, the last round consists of the SC only. Unlike the CRAFT S-box, the division property table of the SKINNY-64 S-box is relatively sparse. Therefore, the trail extension is possible without the equivalent S-box technique [12]. However, using the equivalent S-box technique increases the efficiency significantly. The following S-box

$$S' : \texttt{0x1}, \texttt{0xA}, \texttt{0x2}, \texttt{0xB}, \texttt{0x3}, \texttt{0xC}, \texttt{0x4}, \texttt{0x9}, \texttt{0x6}, \texttt{0xE}, \texttt{0x5}, \texttt{0xF}, \texttt{0x8}, \texttt{0x0}, \texttt{0xD}, \texttt{0x7}$$

is equivalent to the original S-box of SKINNY-64 and satisfies the three conditions shown in Sect. 5. We also use two improvements which are similar to CRAFT to generate a full-rank integral-resistance matrix efficiently. The first improvement is the so-called *column rotation equivalence* [12]. Once we find a key pattern, three key patterns whose columns are rotated by 1, 2, and 3 are also available. Thanks to this property, $4096/4 = 1024$ key patterns are enough.

The second improvement used for CRAFT, i.e., to first generate division trails which cover only 10 rounds and then extend it to 12 rounds, is also applicable here. Unfortunately, we cannot use this trick for all of 1024 key patterns because there is no 10-round division trail from specific input pattern to specific output pattern. For key patterns where this is not possible, we need to generate key patterns for 12 rounds directly.

As a consequence, we can generate 4096 key patterns whose corresponding integral-resistance matrix has full rank. Again, once we generate a full rank integral-resistance matrix for 12 rounds, it also guarantees that there is no integral distinguisher for 12 rounds and more.

6.3 Applications to GIFT-64

GIFT is a lightweight block cipher published as CHES'17 by Banik et al. [2], with a 128-bit key and two variants depending on the block size : GIFT-64 and GIFT-128 for 64-bit and 128-bit block size respectively. Its round function is very simple and only consists of the key-addition, an S-box layer with 4-bit S-boxes and a bit permutation layer. Note that in the original design, the round key is only added to half of the state. As in [12] we are here considering a slightly different variant where the round key is added to the full state, as well as assuming that each round key is independent (as in the rest of this paper).

We reused the key patterns given by [12] for their proof of the "all maximal degree monomial" property, leading to key patterns $v_1, \ldots, v_s$ with $s = n^2 = 4096$ for 11 rounds. Note that these key patterns already have the special property mentionned at the start of Sect. 6.

We set a time limit of one minute for the computation of each coefficient in the integral-resistance matrix. That is, if we could not compute the total number of trails in less than one minute, we replace the coefficient by $\star$. By doing this

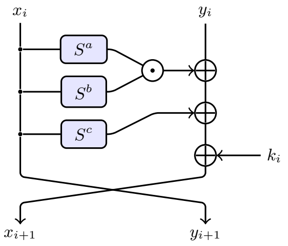


Fig. 5. SIMON/Simeck round function [13].

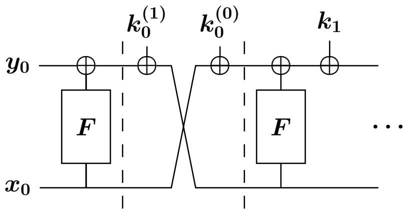


Fig. 6. SIMON/ Simeck one round extension for pre-whitening key.

we are still able to compute enough coefficients so that we can prove that the matrix is always full rank. Note that since GIFT doesn't have a whitening key, having the integral-resistance matrix at full rank for 11 rounds means that we prove the resistance against integral distinguishers for 12 rounds (as in the case of SKINNY-64). Thus with this, and assuming independent round keys, we were able to prove that 12 rounds and higher of GIFT64 has no integral distinguisher, according to Corollary 2.

6.4 Applications to PRESENT

PRESENT is another lightweight block cipher, published at CHES'07 [6], with a 64-bit block size and the option between 80-bit and 128-bit key. Its round function is also very simple, built with only a key addition, S-box layer of 4-bit S-boxes and a bit permutation. Similarly to CRAFT, we first build input/output/key-patterns leading to an odd number of trails for 11 rounds, and extend them by one round both in the forward and backward direction using affine equivalent S-boxes for the first and last round. In the end this allowed us to prove that 13 rounds and higher of PRESENT, assuming independent round keys, does not have any integral distinguisher, according to Corollary 2.

6.5 Applications to SIMON/Simeck

SIMON is a Feistel cipher which was published in 2013 [3]. Figure 5 shows the round function where S^i is the left circular shift by i positions. The shift constants for SIMON are $a = 8$, $b = 1$, and $c = 2$. While the cipher operates on n-bit words, the block length is $2n$. SIMON supports the block lengths 32, 48, 64, 96, and 128. The cipher Simeck [24] is very similar, it just replaces the shifts constants by $a = 0$, $b = 5$, $c = 1$ (to allow an even more efficient implementation) and supports the block lengths 32, 48, and 64.

For the division property of SIMON/Simeck we can observe the following *word rotation equivalence*: let $F(x_i, y_i, k_i) = (x_{i+1}, y_{i+1})$ be one round of SIMON/Simeck. Then the round function is shift invariant, that is, for all $0 \leq l < n$

$$(a_1, a_2, a_3) \xrightarrow{F} (b_1, b_2) \Leftrightarrow (S^l(a_1), S^l(a_2), S^l(a_3)) \xrightarrow{F} (S^l(b_1), S^l(b_2)).$$

Showing the integral-resistance property for SIMON and Simeck requires, with a naive approach, $4n^2$ key patterns. Based on the word rotation equivalence, we can reduce this number down to $4n$. Finding a key pattern where only one combination of a monomial and an output bit leads to an odd number of trails and all other combinations to an even number of trails leads to a unit vector as a column in the integral-resistance matrix. In this row where the unit vector has the 1 entry, we do not need to count the number of trails for other key pattern and just insert a 0 there. If the number of trails is odd for the specific monomial/output bit combination for another key pattern, we can just add the unit vector to this column to create a 0 entry there. Unlike the other block ciphers examined in this paper, the trail extension was not used, we simply maximized the weight of the key patterns.

We iterated over all monomial/output bit combinations and try to find a key pattern, so that the number of trails is odd. We parallelized the search of key patterns, which leads to a program behaviour where for ”easy” monomial/output bit combinations key patterns are found very fast and often give unit vectors. When the program found key patterns for ”complicated” monomial/output bit combinations, for many positions in the column the trail counting can be omitted, which drastically improves the performance.

When we can compute the integral-resistance matrix of full rank for r rounds of SIMON resp. Simeck, we still need a pre-whitening key on the full state to ensure the integral-resistance property. For that we add one additional round in the beginning (see Fig. 6). As we already assume independent round keys, we can (virtually, similarly to masking or secret sharing) split the second round key in two independent parts (k_1 and $k_0^{(0)}$), and add $k_0^{(0)}$ on the right word. Then $(k_0^{(0)}, k_0^{(1)})$ is our pre-whitening key on the full state. The application of F before does not change the integral-resistance property as shown in Sect. 4.3. This leads to a proof of the integral-resistance property for $r + 1$ rounds.

The best known integral distinguishers for SIMON32 and Simeck32 cover 15 rounds [21]. We can show that the integral-resistance property holds for 16 rounds of both ciphers which is tight in terms of number of rounds. The integral distinguishers shown in [23] for SIMON48/64/96/128 and SIMECK48/64 can be extended by one round with the technique shown in [22]. Adding another round gives our bounds in Table 1 where we show the integral-resistance property.

7 Conclusion and Future Work

In this paper, we were able to show strong security arguments against integral distinguishers for several block ciphers following the SPN and Feistel designs. Although these are the best security guarantees against integral distinguishers so far, in theory, it could still be that it is easy to mount integral attacks. Our result shows that any sum is key-dependent, however, this does not exclude the case where the sum is simply one key bit, which could be exploited in an attack.

A wider application to more (block) ciphers would be interesting, especially under the aspect of more automatization and less optimization by hand. We see further research directions in closing the gap for GIFT-64 and PRESENT between the best known integral distinguisher and the integral-resistance property, and a more intuitive understanding of what allows the degree to be extended to more rounds. For the question of covering more rounds, in the case of an algebraic degree, it would be interesting to better understand which S-boxes, or family of S-boxes, allow to preserve the degree for more rounds.

Our results are inherently non applicable to cryptographic permutations because (i) the key addition is crucial for our results and to reduce the complexity of the MILP models and (ii) the conditions for the integral distinguisher cannot be fulfilled. Deriving similar results for permutations or permutation based schemes would be interesting.

Acknowledgment. This work was partially funded by the DFG, (German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA – 390781972 and within the project Analysis and Protection of Lightweight Cryptographic Algorithms (432878529).

References

1. Baksi, A.: New insights on differential and linear bounds using mixed integer linear programming. In: SecITC 2020. pp. 41–54 (2020). https://doi.org/10.1007/978-3-030-69255-1_4
2. Banik, S., Pandey, S.K., Peyrin, T., Sasaki, Y., Sim, S.M., Todo, Y.: GIFT: A small present - towards reaching the limit of lightweight encryption. In: Fischer, W., Homma, N. (eds.) CHES 2017. LNCS, vol. 10529, pp. 321–345. Springer (2017). https://doi.org/10.1007/978-3-319-66787-4_16
3. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK families of lightweight block ciphers. IACR Cryptol. ePrint Arch. **2013**, 404 (2013)
4. Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., Sasaki, Y., Sasdrich, P., Sim, S.M.: The SKINNY family of block ciphers and its low-latency variant MANTIS. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part II. LNCS, vol. 9815, pp. 123–153. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_5
5. Beierle, C., Leander, G., Moradi, A., Rasoolzadeh, S.: CRAFT: lightweight tweakable block cipher with efficient protection against DFA attacks. IACR Trans. Symmetric Cryptol. **2019**(1), 5–45 (2019). https://doi.org/10.13154/tosc.v2019.i1.5-45
6. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: an ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer (2007). https://doi.org/10.1007/978-3-540-74735-2_31, https://doi.org/10.1007/978-3-540-74735-2_31
7. Boura, C., Canteaut, A.: Another view of the division property. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 654–682. Springer (2016). https://doi.org/10.1007/978-3-662-53018-4_24
8. Canteaut, A., Lallemand, V., Leander, G., Neumann, P., Wiemer, F.: bison instantiating the whitened swap-or-not construction. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478,

pp. 585–616. Springer (2019). https://doi.org/10.1007/978-3-030-17659-4_20, https://doi.org/10.1007/978-3-030-17659-4_20

9. Carlet, C., Crama, Y., Hammer, P.L.: Vectorial boolean functions for cryptography. In: Crama, Y., Hammer, P.L. (eds.) Boolean Models and Methods in Mathematics, Computer Science, and Engineering, pp. 398–470. Cambridge University Press (2010). https://doi.org/10.1017/cbo9780511780448.012
10. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher square. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer (1997). https://doi.org/10.1007/BFb0052343
11. Derbez, P., Fouque, P.: Increasing precision of division property. IACR Trans. Symmetric Cryptol. **2020**(4), 173–194 (2020). https://doi.org/10.46586/tosc.v2020.i4.173-194
12. Hebborn, P., Lambin, B., Leander, G., Todo, Y.: Lower bounds on the degree of block ciphers. In: Moriai, S., Wang, H. (eds.) ASIACRYPT 2020, Part I. LNCS, vol. 12491, pp. 537–566. Springer (2020). https://doi.org/10.1007/978-3-030-64837-4_18
13. Jean, J.: TikZ for Cryptographers. https://www.iacr.org/authors/tikz/ (2016)
14. Knudsen, L.R.: Truncated and higher order differentials. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 196–211. Springer (1994). https://doi.org/10.1007/3-540-60590-8_16
15. Lai, X.: Higher order derivatives and differential cryptanalysis. In: Communications and cryptography, pp. 227–233. Springer (1994)
16. Lai, X., Massey, J.L., Murphy, S.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT '91. LNCS, vol. 547, pp. 17–38. Springer (1991). https://doi.org/10.1007/3-540-46416-6_2, https://doi.org/10.1007/3-540-46416-6_2
17. Lambin, B., Derbez, P., Fouque, P.: Linearly equivalent s-boxes and the division property. Des. Codes Cryptogr. **88**(10), 2207–2231 (2020). https://doi.org/10.1007/s10623-020-00773-4
18. Nyberg, K., Knudsen, L.R.: Provable security against a differential attack. J. Cryptol. **8**(1), 27–37 (1995). https://doi.org/10.1007/BF00204800, https://doi.org/10.1007/BF00204800
19. Shibayama, N., Kaneko, T.: A new higher order differential of CLEFIA. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **97-A**(1), 118–126 (2014). https://doi.org/10.1587/transfun.E97.A.118
20. Todo, Y.: Structural evaluation by generalized integral property. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015, Part I. LNCS, vol. 9056, pp. 287–314. Springer (2015). https://doi.org/10.1007/978-3-662-46800-5_12
21. Todo, Y., Morii, M.: Bit-based division property and application to simon family. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 357–377. Springer (2016). https://doi.org/10.1007/978-3-662-52993-5_18
22. Wang, Q., Liu, Z., Varici, K., Sasaki, Y., Rijmen, V., Todo, Y.: Cryptanalysis of reduced-round SIMON32 and SIMON48. In: Meier, W., Mukhopadhyay, D. (eds.) INDOCRYPT 2014. LNCS, vol. 8885, pp. 143–160. Springer (2014). https://doi.org/10.1007/978-3-319-13039-2_9
23. Xiang, Z., Zhang, W., Bao, Z., Lin, D.: Applying MILP method to searching integral distinguishers based on division property for 6 lightweight block ciphers. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part I. LNCS, vol. 10031, pp. 648–678 (2016). https://doi.org/10.1007/978-3-662-53887-6_24
24. Yang, G., Zhu, B., Suder, V., Aagaard, M.D., Gong, G.: The simeck family of lightweight block ciphers. IACR Cryptol. ePrint Arch. **2015**, 612 (2015)