Block Cipher Invariants as Eigenvectors of Correlation Matrices (Full Version)*

Tim Beyne

imec-COSIC, KU Leuven
name.lastname@esat.kuleuven.be

Abstract. A new approach to invariant subspaces and nonlinear invariants is developed. This results in both theoretical insights and practical attacks on block ciphers. It is shown that, with minor modifications to some of the round constants, Midori-64 has a nonlinear invariant with 2^{96} corresponding weak keys. Furthermore, this invariant corresponds to a linear hull with maximal correlation. By combining the new invariant with integral cryptanalysis, a practical key-recovery attack on 10 rounds of unmodified Midori-64 is obtained. The attack works for 2^{96} weak keys and irrespective of the choice of round constants. The data complexity is $1.25 \cdot 2^{21}$ chosen plaintexts and the computational cost is dominated by 2^{56} block cipher calls. Finally, it is shown that similar techniques lead to a practical key-recovery attack on MANTIS-4. The full key is recovered using roughly 350 chosen plaintexts and the attack requires about 2^{56} block cipher calls. Furthermore, given less than 350 additional chosen ciphertexts under a related tweak, 2^{18} block cipher calls suffice to recover the full key.

Keywords: Invariant subspace attack · Nonlinear invariant attack · Linear cryptanalysis · Integral cryptanalysis · Correlation matrices · Midori-64 · MANTIS

1 Introduction

Block ciphers are an essential primitive for the construction of many cryptosystems. This leads to a natural desire to optimize them with respect to various application-dependent criteria. Examples include low-latency block ciphers such as PRINCE [7] and MANTIS [5], and the low-power design Midori-64 [3]. Biryukov and Perrin [6] give a broad overview of such *lightweight* primitives.

One requirement is shared by all applications: the block cipher must be secure – at the very least it must approximate a pseudorandom permutation. A common design decision that often helps to reduce latency, energy consumption and other cost measures is the simplification of the key-schedule. This, along with other aspects of lightweight designs, has led to the development of new cryptanalytic tools such as *invariant subspaces* [18] and *nonlinear invariants* [23]. These attacks are the subject of this paper.

* This work was supported by the Research Council KU Leuven: C16/18/004.

At CRYPTO 2017, it was shown by Beierle, Canteaut, Leander and Rotella that invariant attacks can often be averted by a careful choice of the round constants [4]. Their work, as well as the earlier work by Todo, Leander and Sasaki on nonlinear invariants [23], invites several questions. This paper will be concerned with three related problems that arise in this context.

1. In their future work sections, Todo *et al.* [23] and Beierle *et al.* [4] both express the desire to generalize the nonlinear invariant attack. One can argue that a deeper theoretical understanding of block cipher invariants is helpful, if not essential, to achieve this goal.
2. One potential generalization is the existence of block cipher invariants which are not invariants under all of the round transformations. It is important to investigate this possibility, because such cases are not covered by the techniques introduced by Beierle *et al.* for choosing the round constants.
3. The previous problem leads to a third question: do such (generalized) invariants *only* impact the security of the cipher for a specific choice of the round constants? The results in this paper suggest otherwise.

Contribution. The first of the problems listed above is addressed in Section 4, where the main contribution is Definition 2 and the discussion following it. It is shown that block cipher invariants have an effective description in terms of eigenvectors of *correlation matrices*. These matrices were first introduced by Daemen, Govaerts and Vandewalle [9] in the context of linear cryptanalysis [21]. As a side result, more insight into the relation between invariants and linear cryptanalysis is obtained.

Section 5 takes a closer look at the invariants of Midori-64, leading up to an example of an invariant of the type described in the second problem above. It will be shown in Section 5.3 that, with minor changes to the round constants, Midori-64 has an invariant which is not invariant under the round function. It applies to $2^{96} + 2^{64}$ weak keys. Note that this is a significantly larger class of weak keys compared to previous work, *i.e.* 2^{32} for the invariant subspace attack of Guo *et al.* and 2^{64} for the nonlinear invariant attack of Todo *et al.* [23]. In fact, it will be demonstrated that the invariant discussed in Section 5.3 corresponds to a linear hull with maximal correlation. This observation is of independent interest and will be briefly discussed in Section 5.4.

Finally, Sections 6 and 7 address the third question listed above. That is, two cryptanalytic results are given to demonstrate that block cipher invariants may impact the security of a block cipher regardless of the choice of round constants.

In Section 6, a practical attack on 10 rounds of Midori-64 – for any choice of round constants – will be given. The attack applies to 2^{96} weak keys and requires roughly $1.25 \cdot 2^{21}$ chosen plaintexts. The computational cost is dominated by 2^{56} block cipher calls. Note that the data complexity and especially the computational cost to determine whether a weak key is used, are significantly lower. As discussed by Luykx, Mennink and Paterson [20] in ASIACRYPT 2017, this has a significant impact on the multi-key security of the block cipher.

Section 7 shows that the full key of MANTIS-4 [5] can be recovered given 342 chosen plaintexts. This attack works for all keys provided that a weak tweak is used. The number of weak tweaks is 2^{32} (out of 2^{64}). The computational cost of this attack is dominated by 2^{56} block cipher calls. If 342 chosen ciphertexts under a related tweak are additionally available, the key can be recovered with a computational cost of 2^{18} block cipher calls.

2 Preliminaries and Related Work

Most of the notation used in this paper is standard, for instance $(\mathbb{F}_2, +, \cdot)$ denotes the field with two elements. Random variables are denoted in boldface.

Many of the results in this work can be compactly described by means of tensor products of real vector spaces. Let $V_1, \ldots, V_n$ be vector spaces over $\mathbb{R}$. Their tensor product is a real vector space $V_1 \otimes \cdots \otimes V_n$. Elements of $V_1 \otimes \cdots \otimes V_n$ will be called tensors. For $V = V_1 = \cdots = V_n$, the tensor product $V_1 \otimes \cdots \otimes V_n$ will be denoted by $V^{\otimes n}$. Knowledge of tensor products is not essential to understand this work.

The invariant subspace attack was introduced by Leander, Abdelraheem, AlKhzaimi and Zenner in the context of the PRINTCIPHER [18]. Let $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher. An affine subspace $a + V$ of $\mathbb{F}_2^n$ such that

$$E_k(a + V) = a + V, \tag{1}$$

is called an invariant subspace for E_k. The keys k for which (1) holds, will be called weak keys. At ASIACRYPT 2016, Todo *et al.* introduced the *nonlinear invariant attack* as an extension of this attack [23]. A Boolean function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is called a nonlinear invariant for E_k iff there exists a constant $c \in \mathbb{F}_2$ such that for all $x \in \mathbb{F}_2^n$,

$$f(x) + f(E_k(x)) = c.$$

Importantly, the constant c may depend on the key k, but not on x.

The description of block cipher invariants in this paper is based on *correlation matrices*, which were first introduced by Daemen *et al.* [9]. The definition of these matrices is postponed to Section 3, as they will be introduced from a novel point of view.

Finally, a brief description of Midori-64 is given here. This information will be used extensively in Sections 5 and 6. Midori-64 is an iterated block cipher with a block size of 64 bits and a key length of 128 bits [3]. It operates on a 64-bit state, which can be represented as a 4×4 array of 4-bit *cells*. The round function consists of the operations `SubCell` ($\mathfrak{S}$), `ShuffleCell` (P), `MixColumn` ($\mathfrak{M}$) and a key addition layer. This structure is shown in Figure 1.

The `SubCell` ($\mathfrak{S}$) mapping applies a 4-bit S-box S to each cell of the state. The fact that the S-box is an involution will be used in Section 5. The algebraic normal form of $S(x) = (S_1(x), S_2(x), S_3(x), S_4(x))$ is provided below. These expressions will not be used explicitly, but they can be helpful to verify the

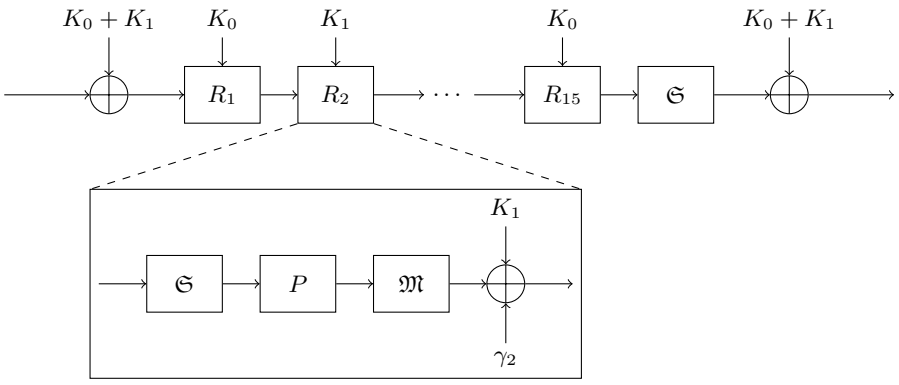

Fig. 1. The overall structure and round function of Midori-64.

calculations in Sections 6 and 7.

$$
\begin{aligned}
S_1(x_1, x_2, x_3, x_4) &= x_1x_2x_3 + x_1x_3x_4 + x_1x_2 + x_1x_3 + x_3x_4 + 1 \\
S_2(x_1, x_2, x_3, x_4) &= x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4 + x_1x_4 + x_1 + x_4 + 1 \\
S_3(x_1, x_2, x_3, x_4) &= x_1x_2 + x_1x_4 + x_2x_4 + x_2 + x_4 \\
S_4(x_1, x_2, x_3, x_4) &= x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4 + x_1x_4 + x_2x_4 + x_3.
\end{aligned}
$$

The permutation `ShuffleCell` (P) interchanges the cells of the state. It operates on the state as follows:

s_1	s_5	s_9	s_{13}
s_2	s_6	s_{10}	s_{14}
s_3	s_7	s_{11}	s_{15}
s_4	s_8	s_{12}	s_{16}

$\xrightarrow{P}$

s_1	s_{15}	s_{10}	s_8
s_{11}	s_5	s_4	s_{14}
s_6	s_{12}	s_{13}	s_3
s_{16}	s_2	s_7	s_9

The `MixColumn` ($\mathfrak{M}$) transformation acts on each state column independently by the following matrix over $\mathbb{F}_{2^4}$:

$$
M = \begin{pmatrix} 0\,1\,1\,1 \\ 1\,0\,1\,1 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \end{pmatrix}.
$$

That is, each cell of a column of the state is replaced by the exclusive or of the other elements in the same column. Finally, the round key in round i is alternatingly taken to be $K_0 + \gamma_i$ or $K_1 + \gamma_i$, where γ_i is a round constant.

Importantly, round constants are only added to the least significant (rightmost) bit of each cell, *i.e.* $\gamma_i \in \{0,1\}^{16}$.

The tweakable block cipher MANTIS [5] is quite similar to Midori-64, having nearly the same round function. Details will be given in Section 7.

3 Correlation Matrices

The cryptanalysis of symmetric-key primitives is generally based on properties of the plaintext that are reflected by the corresponding ciphertext. To every such property, one could associate a set of values satisfying it. A convenient way to work with sets of plaintexts, or more generally multisets, is to associate a probability space with the set of block cipher inputs. Let $\boldsymbol{x}$ be a random variable on $\mathbb{F}_2^n$ with probability mass function $p_{\boldsymbol{x}}$. The Fourier transform $\widehat{p}_{\boldsymbol{x}}$ of $p_{\boldsymbol{x}}$ is defined by

$$\widehat{p}_{\boldsymbol{x}}(\chi_u) = \sum_{x \in \mathbb{F}_2^n} p_{\boldsymbol{x}}(x)\chi_u(x),$$

where $\chi_u : x \mapsto (-1)^{u^\top x}$ is a character of $\mathbb{F}_2^n$. That is, the function $p_{\boldsymbol{x}}$ is expressed in the character basis of the algebra $\mathbb{C}[\mathbb{F}_2^n]$ of functions $\mathbb{F}_2^n \to \mathbb{C}$. Since the character group of $\mathbb{F}_2^n$ is isomorphic to $\mathbb{F}_2^n$, we may consider $\widehat{p}_{\boldsymbol{x}}$ to be a function on $\mathbb{F}_2^n$ instead. That is,

$$\widehat{p}_{\boldsymbol{x}}(u) = \mathbf{E}\left[(-1)^{u^\top \boldsymbol{x}}\right],$$

where $\mathbf{E}[\,\cdot\,]$ denotes the expected value. Additional information regarding the use of characters and, more generally, representations in the context of probability theory can be found in the references [8, 11].

Example 1. The Fourier transform of the uniform distribution on $\mathbb{F}_2^n$ is zero everywhere except at $u = 0$, *i.e.* it has coordinates $(1, 0, \ldots, 0)^\top$. Let $p(x) = 0$ for all $x \neq c$ and $p(c) = 1$, then $\widehat{p}(u) = (-1)^{u^\top c}$. To stress that $\widehat{p}$ is a vector, we will regularly use the notation $\widehat{p}_u = \widehat{p}(u)$. ▷

The following result is essential to the discussion of the invariants of Midori-64 in Section 5. Note that here, and further on, the vector spaces $\mathbb{F}_2^{mn}$ and $(\mathbb{F}_2^n)^m$ are treated as essentially the same. Recall that the symbol "$\otimes$" denotes the tensor product, which in this case coincides with the Kronecker product.

Theorem 1 (Independence) *Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ be independent random variables on $\mathbb{F}_2^n$. The Fourier transform of the joint probability mass function of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ is given by*

$$\widehat{p}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m} = \bigotimes_{i=1}^{m} \widehat{p}_{\boldsymbol{x}_i},$$

where $\widehat{p}_{\boldsymbol{x}_i}$ is the Fourier transform of the probability mass function of $\boldsymbol{x}_i$.

Proof. By the independence of $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$, we have

$$\widehat{p}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m}(u_1, \ldots, u_m) = \mathbf{E}\left[(-1)^{\sum_{i=1}^m u_i^\top \boldsymbol{x}_i}\right] = \prod_{i=1}^m \mathbf{E}\left[(-1)^{u_i^\top \boldsymbol{x}_i}\right].$$

□

In fact, Theorem 1 generalizes to arbitrary functions $f : (\mathbb{F}_2^n)^m \to \mathbb{C}$ such that $f(x_1, \ldots x_m) = \prod_{i=1}^m f_i(x_i)$ with $f_i \in \mathbb{C}[\mathbb{F}_2^n]$.

The reader who is familiar with tensors may find it intuitive to consider $\widehat{p}_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_m}$ in Theorem 1 to be a simple (*i.e.* rank one) tensor in $[\mathbb{R}^{2^n}]^{\otimes m}$. This fact is not essential to the remainder of the paper.

The discussion so far has been limited to probability distributions. The remainder of this section deals with transformations of these distributions. The relation between the probability distribution of $\boldsymbol{x}$ and $F(\boldsymbol{x})$ is in general given by a transition matrix. When represented in the basis of characters, such a matrix may be called a correlation matrix (not to be confused with a matrix of second moments).

Definition 1 (Correlation matrix over $\mathbb{F}_2^n$) *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function. The correlation matrix $C^F \in \mathbb{R}^{2^m \times 2^n}$ of F is the representation of the transition matrix of F with respect to the character basis of $\mathbb{C}[\mathbb{F}_2^n]$ and $\mathbb{C}[\mathbb{F}_2^m]$.*

Theorem 2 *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a vectorial Boolean function with correlation matrix C^F. Let $\boldsymbol{x}$ be a random variable on $\mathbb{F}_2^n$ with probability mass function $p_{\boldsymbol{x}}$, then*

$$\widehat{p}_{F(\boldsymbol{x})} = C^F \widehat{p}_{\boldsymbol{x}}.$$

Proof. This result is essentially a restatement of Definition 1. □

It is instructive to consider the coordinates of C^F. By the Fourier inversion formula, we have

$$p_{\boldsymbol{x}}(x) = \frac{1}{2^n} \sum_{u \in \mathbb{F}_2^n} (-1)^{u^\top x}\, \widehat{p}_{\boldsymbol{x}}(u).$$

By substituting the above into the definition of $\widehat{p}_{F(\boldsymbol{x})}$, and from Theorem 2, one obtains

$$\widehat{p}_{F(\boldsymbol{x})}(u) = \sum_{v \in \mathbb{F}_2^n} \left[\frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{u^\top F(x) + v^\top x}\right] \widehat{p}_{\boldsymbol{x}}(v) = \sum_{v \in \mathbb{F}_2^n} C^F_{u,v}\, \widehat{p}_{\boldsymbol{x}}(v).$$

Since this holds for all functions $\widehat{p}_{\boldsymbol{x}}$, the coordinates of C^F are

$$C^F_{u,v} = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{u^\top F(x) + v^\top x}. \tag{2}$$

This establishes the equivalence of Definition 1 and the definition due to Daemen *et al.* [9], which originates in the notion of *correlation* between Boolean functions. Note that (2) coincides with the Walsh-Hadamard transform of F, but since the result of this transformation is not typically interpreted as a linear operator, we will avoid this term.

To conclude this section, a few useful properties of correlation matrices will be listed. These results can also be found (some in a slightly different form) in [9]. In Theorem 5, δ denotes the Kronecker delta function.

Theorem 3 (Composition) *Let $F : \mathbb{F}_2^l \to \mathbb{F}_2^m$ and $G : \mathbb{F}_2^m \to \mathbb{F}_2^n$, then $C^{G \circ F} = C^G C^F$.*

Theorem 4 (Orthogonality) *Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$. If F is a bijection, then its correlation matrix C^F is orthogonal.*

Theorem 5 (Linear maps) *Let $L : \mathbb{F}_2^n \to \mathbb{F}_2^m$ be a linear map, then $C^L_{u,v} = \delta(v + L^\top u)$. Furthermore, if L is bijective, C^L is a permutation matrix.*

Theorem 6 (Boxed maps) *Let $F : \mathbb{F}_2^{sn} \to \mathbb{F}_2^{sm}$ be a vectorial Boolean function such that there exist functions $F_i : \mathbb{F}_2^n \to \mathbb{F}_2^m$, $i = 1, \ldots, s$ with the property that $F = (F_1, \ldots, F_s)$. Then*

$$C^F = \bigotimes_{i=1}^{s} C^{F_i}.$$

In light of Theorem 1, the property expressed by Theorem 6 is intuitively clear: a function satisfying the conditions of Theorem 6 preserves the independence of its inputs.

Example 2. Let C^K denote the correlation matrix corresponding to the function $x \mapsto x + K$ with $x, K \in \mathbb{F}_2^2$. Let $K = (\kappa_1, \kappa_2)$. By Theorem 6, $C^K = C^{\kappa_1} \otimes C^{\kappa_2}$. It follows that C^K is given by

$$C^K = \begin{pmatrix} 1 & 0 \\ 0 & (-1)^{\kappa_1} \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & (-1)^{\kappa_2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & (-1)^{\kappa_2} & 0 & 0 \\ 0 & 0 & (-1)^{\kappa_1} & 0 \\ 0 & 0 & 0 & (-1)^{\kappa_1 + \kappa_2} \end{pmatrix}.$$

The fact that the correlation matrix of a constant addition is diagonal will be essential to motivate our definition of block cipher invariants in Section 4. ▷

4 Block Cipher Invariants

The invariant subspace attack is based on the existence of an affine space which is mapped to itself by a block cipher. A nonlinear invariant is a set which is encrypted to itself or its complement. The purpose of this section is to define what it means for a "cryptanalytic property" to be invariant under a block

cipher, and then to show that this definition includes the nonlinear invariant and invariant subspace attacks as special cases.

Let $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an arbitrary function – in particular, F need not be bijective. With invariant subspace attacks in mind, it is reasonable to ask which probability distributions are invariant under F. This is equivalent to determining all multisets which are mapped to themselves by F. The solutions to this problem are precisely the eigenvectors of the transition matrix of F which are also probability distributions. The main issue with this formulation is that, even for a simple function such as the addition of a constant, computing the eigenvectors of the transition matrix is not as trivial as one might hope.

To simplify matters, we will make a change of basis to the character basis of $\mathbb{C}[\mathbb{F}_2^n]$, which was introduced in Section 3. That is, we consider the eigenvectors of correlation matrices instead of transition matrices. This has the important advantage that the correlation matrix of a constant addition is a diagonal matrix. This is helpful, because the columns of a diagonal matrix also form a basis of eigenvectors.

One final simplification can be made before stating Definition 2: there is no good reason to consider only probability distributions – one can simply allow all eigenvectors. It will be shown in Section 4.1 that nonlinear invariants are examples of eigenvectors that are not Fourier transformations of probability distributions.

Definition 2 (Block cipher invariant.) *A vector $v \in \mathbb{C}^{2^n}$ is an invariant for a block cipher $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ if it is an eigenvector of the correlation matrix C^{E_k}. If v is a multiple of $(1, 0, \ldots, 0)^\top$, it will be called a trivial invariant.*

This paper is only concerned with eigenvectors which correspond to real eigenvalues, *i.e.* ± 1 due to Theorem 4. More generally, one could also have eigenvalues which are complex roots of unity. This will be discussed briefly in Section 8, which covers future work.

Not all vectors satisfying Definition 2 can be used in cryptanalysis. A sufficient condition for an invariant to be useful is that it depends only on part of the key, and that it comes with an efficient way of testing whether it holds for a given set of plaintext/ciphertext pairs. Section 4.1 shows that the latter requirement is usually not a problem.

Finally, note that some work related to Definition 2 can be found in the literature. Abdelraheem *et al.* [1] have observed that invariant subspaces correspond to eigenvectors of a submatrix of C^{E_k}. This can be seen to be a special case of Definition 2. Dravie *et al.* [13] give several results related to the spectrum of correlation matrices (not in the context of invariant attacks).

4.1 Nonlinear Invariants

The goal of this section is to establish the relation between Definition 2 and nonlinear invariants. Theorem 7 provides a general result to this end, but the simpler Corollary 1 is sufficient to obtain the desired relation. For the following results, the notation $e_0 = (1, 0, \ldots, 0)^\top$ will be used.

Theorem 7 (Nonlinear invariant) *Let $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher with correlation matrix C^{E_k} and $f : \mathbb{F}_2^n \to \mathbb{F}_2$ a Boolean function with correlation matrix $(e_0\ v)^\top$. If v is an eigenvector of C^{E_k} with eigenvalue $\lambda = \pm 1$, then for any random variable $\boldsymbol{x}$ on $\mathbb{F}_2^n$, it holds that*

$$\Pr\left[f(E_k(\boldsymbol{x})) = 0\right] - \frac{1}{2} = \lambda \left(\Pr\left[f(\boldsymbol{x}) = 0\right] - \frac{1}{2}\right). \tag{3}$$

Conversely, suppose (3) *holds for a set of random variables $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ with probability distributions $p_{\boldsymbol{x}_1}, \ldots, p_{\boldsymbol{x}_m}$ such that* $\mathrm{Span}\left\{p_{\boldsymbol{x}_1}, \ldots, p_{\boldsymbol{x}_m}\right\} = \mathbb{R}^{2^n}$. *Then v is an eigenvector of C^{E_k} with eigenvalue λ.*

Proof. By the orthogonality of C^{E_k}, it holds that $\left[C^{E_k} v\right]^\top \left[C^{E_k} w\right] = v^\top w$. Since $C^{E_k} v = \lambda v$ with $\lambda = \pm 1$, it follows that $\lambda v^\top \left[C^{E_k} w\right] = v^\top w$ and hence $v^\top \left[C^{E_k} w\right] = \lambda v^\top w$.

For any $\boldsymbol{x}$, choose w as the Fourier transform of the probability mass function of $\boldsymbol{x}$. The equality $v^\top \left[C^{E_k} w\right] = \lambda v^\top w$ is then equivalent to (3). To show the converse, extract a basis $\{w_1, \ldots, w_{2^n}\}$ for $\mathbb{R}^{2^n}$ from the vectors $\widehat{p}_{\boldsymbol{x}_1}, \ldots, \widehat{p}_{\boldsymbol{x}_m}$. From $v^\top [C^{E_k} w_i] = \lambda v^\top w_i$, $i = 1, \ldots, 2^n$ it follows that $v^\top C^{E_k} = \lambda v^\top$. The result follows from the orthogonality of C^{E_k}. □

Theorem 7 has the following corollary, which gives the precise relation between the eigenvectors of C^{E_k} and the nonlinear invariants of E_k as defined by Todo, Leander and Sasaki [23].

Corollary 1 *Let $E_k : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a block cipher with correlation matrix C^{E_k} and $f : \mathbb{F}_2^n \to \mathbb{F}_2$ a Boolean function with correlation matrix $(e_0\ v)^\top$. Then v is an eigenvector of C^{E_k} with eigenvalue $(-1)^c$, $c \in \mathbb{F}_2$ if and only if for all $x \in \mathbb{F}_2^n$, it holds that*

$$f(x) + f(E_k(x)) = c.$$

Proof. For any x, apply Theorem 7 to a random variable $\boldsymbol{x}$ with probability distribution concentrated on x. For the converse, it suffices to note that the Fourier transforms of these probability distributions form a basis for $\mathbb{R}^{2^n}$. □

Finally, the following is a simple result that is useful to obtain the nonlinear invariant corresponding to an eigenvector v. Note that $\mathbf{1}_S$ denotes the indicator function of a set S.

Theorem 8 *Let S be any subset of $\mathbb{F}_2^n$ and let p_1, p_2 be functions*[1] *defined by $p_1(x) = 2^{-n}\mathbf{1}_S$ and $p_2(x) = 2^{-n}\mathbf{1}_{\mathbb{F}_2^n \setminus S}$ respectively. If $v \in \mathbb{F}_2^n$ is the difference of the Fourier transforms of p_1 and p_2, i.e., $v = \widehat{p}_2 - \widehat{p}_1$ then $\mathbf{1}_S$ has correlation matrix $(e_0\ v)^\top$.*

[1] Such functions may be called *defective* probability mass functions [15].

Proof. Clearly, the first row of the correlation matrix of S is given by $e_0^\top$. For the second row, remark that

$$v_u = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\mathbf{1}_S(x) + u^\top x} = \frac{1}{2^n} \left[\sum_{x \notin S} (-1)^{u^\top x} - \sum_{x \in S} (-1)^{u^\top x} \right] = \widehat{p}_2(u) - \widehat{p}_1(u).$$

□

Example 3. Consider the function $F : (x_1, x_2) \mapsto (x_2, x_1)$. It has correlation matrix

$$C^F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1. \end{pmatrix}.$$

The vector $2^{-1}(1, 1, 1, -1)^\top = 2^{-2}[(3, 1, 1, -1)^\top - (1, -1, -1, 1)^\top]$ is an eigenvector of C^F. The corresponding nonlinear invariant is $f(x_1, x_2) = x_1 x_2$. ▷

4.2 Computing Invariants

In general, it is nontrivial to compute the invariants of a block cipher. This is in part due to large block sizes, and in part due to the key-dependence of the invariants. To avoid dependencies on the key, one could attempt to find invariants for parts of the block cipher that do not involve the key. The influence of the key addition can easily be checked afterwards. In fact, when working in the character basis, it only depends on the nonzero pattern of the invariant.

The problem is then reduced to computing the invariants of an unkeyed permutation $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$. With Definition 2 in mind, one might consider using a standard numerical procedure to compute the eigenvectors of C^F. This is not a particularly efficient approach: the computational cost is $\mathcal{O}(2^{3n})$, which is of the same order as the ANF-based algorithm proposed by Todo *et al.* [23] to find nonlinear invariants.

In fact, due to the structure of the matrix C^F, its eigendecomposition can be computed using at most $\mathcal{O}(n2^{2n})$ operations. The following algorithm generalizes the cycle structure approach which is mentioned by Todo *et al.* [23] as "potentially applicable". One computes the cycle-decomposition of F. Then, for each cycle $(x_0, \ldots, x_{l-1})$ and for each $0 \leq j < l$, let $v^{(j)}$ be the Fourier transform of the uniform distribution on the singleton $\{x_j\}$. Let $\zeta = e^{2\pi\sqrt{-1}/l}$. For every $0 \leq k < l$, one obtains an eigenvector[2] $w = \sum_{j=0}^{l-1} \zeta^{-kj} v^{(j)}$ corresponding to the eigenvalue ζ^k:

$$C^F w = \sum_{j=0}^{l-1} \zeta^{-kj} C^F v^{(j)} = \sum_{j=0}^{l-1} \zeta^{-k(j-1)} v^{(j)} = \zeta^k w.$$

[2] It is not hard to see that it will be linearly independent from any previously computed eigenvectors.

This method obtains a complete eigenvector basis, since the sum of all cycle lengths is 2^n.

Unfortunately, even the algorithm above is impractical for $n = 64$. To obtain invariants, it is thus necessary to exploit structural properties of the block cipher. Here, Definition 2 will be of use by facilitating a convenient description of invariants. Theorem 9 in Section 5 provides an example in the context of Midori-64.

The main structural property that has been exploited in previous work such as [16,18,23] is the existence of non-trivial *simultaneous* invariants for the linear layer and the nonlinear layer of a block cipher. In the first part of Section 5, this approach is briefly revisited from the point of view of Definition 2. Then, more general (*i.e.* not requiring simultaneous eigenvectors) invariants will be discussed. Note that the discussion in Section 5 will be tailored to the block cipher Midori-64.

5 Invariants for Midori-64

In this section, the invariants of Midori-64 are discussed in the correlation matrix framework. As an example, in Section 5.2 we revisit the invariant subspace attack of Guo *et al.* [16] and the nonlinear invariant from Todo *et al.* [23]. Then, in Section 5.3, a more general invariant will be obtained. This invariant will be used in Sections 6 and 7 to obtain practical attacks on (round reduced) Midori-64 and MANTIS.

Before proceeding with the computation of the invariants, it is necessary to analyze the structure of Midori-64 in more detail. Section 5.1 provides the necessary preliminaries.

5.1 State Representation and Round Transformations

In its most general form, the Fourier-domain representation of the Midori-64 state is a vector $v \in \mathbb{C}^{2^{64}}$. Recall from Section 2 that it is convenient to represent the Midori-64 state as a 4×4 array of 4-bit cells. For this reason, we will denote coordinate $u = (u_1, \ldots, u_{16})$ with $u_i \in \mathbb{F}_2^4$ of v by $v_u = v_{u_1,\ldots,u_{16}}$. This notation reflects the fact that we can think of v as a tensor of order 16, *i.e.* $v \in [\mathbb{C}^{2^4}]^{\otimes 16}$.

From Figure 1, and by using Theorem 3, the correlation matrix of the Midori-64 round function is given by

$$C^{R_i} = C^{\kappa_i + \gamma_i} C^{\mathfrak{M}} C^P C^{\mathfrak{S}},$$

where $\kappa_i = K_0$ when i is odd and K_1 when i is even. Recall that $C^{\kappa_i+\gamma_i}$ is a diagonal matrix. It follows from Theorem 6 that $C^{\mathfrak{S}} = [C^S]^{\otimes 16}$ and $C^{\mathfrak{M}} = [C^M]^{\otimes 4}$. The matrix $C^S \in \mathbb{R}^{16\times 16}$ is a symmetric orthogonal matrix and $C^M \in \mathbb{R}^{2^{16}\times 2^{16}}$ is a symmetric permutation matrix. Specifically, we have $C^M_{u,v} = \delta(u + Mv)$ by Theorem 5. Finally, C^P is a permutation matrix such that $C^P v_{u_1,\ldots,u_{16}} = v_{u_{\pi^{-1}(1)},\ldots,u_{\pi^{-1}(16)}}$ with π the `ShuffleCell` permutation.[3]

[3] A transformation such as C^P may be called a *braiding map*.

It is convenient to look only for invariants with *independent cells* in the sense of Theorem 1 – but the reader should be reminded that the invariants need not be Fourier transforms of probability distributions. That is, we will assume that there exist vectors $v^{(1)}, \ldots, v^{(16)}$ such that

$$v_{u_1,\ldots,u_{16}} = \prod_{i=1}^{16} v^{(i)}_{u_i}. \tag{4}$$

Equivalently, $v = \otimes_{i=1}^{16} v^{(i)}$. Of course, this assumption imposes a serious restriction. However, assuming (4) greatly simplifies the theory and is sufficiently general to recover the invariant attacks of Guo *et al.* [16] and Todo *et al.* [23]. Furthermore, more general assumptions are not necessary to obtain the invariant that will be presented in Section 5.3.

The invariants considered in Section 5.2 will be required to be invariant under $\mathfrak{S}$, $\mathfrak{M}$ and P. Consider the last requirement, *i.e.* v is an eigenvector of C^P. Recall that C^P is a permutation matrix such that

$$C^P \bigotimes_{i=1}^{16} v^{(i)} = \bigotimes_{i=1}^{16} v^{(\pi^{-1}(i))}.$$

If v is symmetric, that is, $v^{(1)} = \cdots = v^{(16)} = \widetilde{v}$, then $\otimes_{i=1}^{16} v^{(i)} = \widetilde{v}^{\otimes 16}$ is clearly invariant under C^P. It turns out that for the purpose of this paper, it suffices to consider only invariants v such that there exists some $\widetilde{v} \in \mathbb{C}^{16}$ such that

$$v_{u_1,\ldots,u_{16}} = \prod_{i=1}^{16} \widetilde{v}_{u_i}. \tag{5}$$

That is, $v = \widetilde{v}^{\otimes 16}$ and v will be called symmetric, in line with standard terminology for such tensors. Note that assumption (5), is less restrictive than (4). Indeed, for any realistic choice of round constants, an asymmetric invariant tends to lead to conflicting requirements on the key after a sufficient number of rounds. Slightly more general invariants can be obtained by requiring that $v^{(i)}$ is constant on the cycles of π.

Computing an eigenvector basis for $C^{\mathfrak{S}}$ is not difficult. In the remainder of this section, the eigenvectors of $C^{\mathfrak{M}}$ satisfying (4) and (5) will be listed. In particular, it is not necessary to compute these eigenvectors numerically. We begin with the straightforward result in Lemma 1. The main result is stated in Theorem 9.

Lemma 1 *If $v^{\otimes 4}$ is a real eigenvector of C^M, then there exists a scalar $\alpha \in \mathbb{R}_0$ such that all coordinates of v in the standard basis are equal to 0 or $\pm\alpha$.*

Proof. The condition that $v^{\otimes 4}$ is an eigenvector of C^M is equivalent to

$$v^{\otimes 4}_{u_1,u_2,u_3,u_4} = \lambda v^{\otimes 4}_{M(u_1,u_2,u_3,u_4)^\top}.$$

Hence, we have for all $u_1, \ldots, u_4 \in \mathbb{F}_2^4$ that

$$\prod_{i=1}^{4} v_{u_i} = \lambda \prod_{i=1}^{4} v_{\Sigma_{j \neq i} u_j}. \tag{6}$$

Note that no vector of the form $v^{\otimes 4}$ can correspond to $\lambda = -1$, since it follows from (6) that $v_u^4 = \lambda v_u^4$. Suppose that at least one coordinate of v is nonzero, *i.e.* $v_u = \alpha$ for some u. By (6), this implies $\alpha v_{u'}^3 = \alpha^3 v_{u'}$ for any $u' \in \mathbb{F}_2^4$. Consequently, $v_{u'} \in \{0, \pm\alpha\}$. □

Theorem 9 *If $v^{\otimes 4}$ is a real eigenvector of C^M, then $\mathcal{A} = \{u \mid v_u \neq 0\}$ is an affine subspace of $\mathbb{F}_2^4$ and there exists a scalar $\alpha \in \mathbb{R}_0$ such that $v_u = \pm\alpha$ for all $u \in \mathcal{A}$. The converse is also true in the following cases:*

- *For $\dim \mathcal{A} = 0$, $\dim \mathcal{A} = 1$ and $\dim \mathcal{A} = 2$.*
- *For $\dim \mathcal{A} = 3$, provided that the number of negative coordinates of v is even.*

The condition for $\dim \mathcal{A} = 3$ is also necessary.

Proof. Suppose $v^{\otimes 4}$ is a real eigenvector of C^M. Let $a, u, u' \in \mathbb{F}_2^4$ such that $v_a \neq 0$, $v_{a+u} \neq 0$ and $v_{a+u'} \neq 0$. By (6), we have

$$v_{a+u+u'}^2 v_{a+u'} v_{a+u} = v_a^2 \, v_{a+u} v_{a+u'} \neq 0.$$

Hence, $v_{a+u+u'} \neq 0$. It follows that $\mathcal{A}$ is an affine space. Lemma 1 completes the argument.

To show the converse, first consider the case $\dim \mathcal{A} \in \{0, 1, 2\}$. It suffices to demonstrate that if $u_1, \ldots, u_4 \in \mathcal{A}$, then $\prod_{i=1}^4 v_{u_i} = \prod_{i=1}^4 v_{\Sigma_{j \neq i} u_j}$. Note that $\{u_1, \ldots, u_4\}$ and $\{\Sigma_{i \neq 1} u_i, \ldots, \Sigma_{i \neq 4} u_i\}$ generate the same affine space. Since the dimension of this space is at most two, it contains at most four elements. Hence, both products contain the same factors.

For $\dim \mathcal{A} = 3$, the previous argument no longer applies when $u_1, \ldots, u_4$ are linearly independent. In this case the left and right hand side of $\prod_{i=1}^4 v_{u_i} = \prod_{i=1}^4 v_{\Sigma_{j \neq i} u_j}$ involve different variables. Hence, since $\mathcal{A}$ contains eight elements, the products of these elements must be positive. □

The only symmetric rank one invariants which are not covered by Theorem 9 are those containing only nonzero entries. It would be possible to extend the result to cover this case as well, but this would have little practical value since such eigenvectors can never lead to a significant class of weak keys. This will become clear in Section 5.2.

5.2 Simultaneous Eigenvectors

As discussed in Section 4.2, it is not possible to find the eigenvectors of C^{E_k} directly and to subsequently identify those vectors that depend only on a limited portion of the key. A more realistic approach is to find joint eigenvectors for all

of the transformations in the round function. This corresponds to the strategy that is commonly used, and it is the strategy that will be applied in this section.

The problem considered in this section is thus to find vectors $v \in \mathbb{R}^{2^{64}}$ such that $[C^S]^{\otimes 16} v = \lambda v$ and $[C^M]^{\otimes 4} v = \mu v$ with $\lambda, \mu \in \{-1, 1\}$. Furthermore, v must be an eigenvector of C^P, but if v is symmetric, we need not separately consider this requirement. For each of these vectors v, we additionally require that they are eigenvectors of $C^{K+\gamma_i}$ for $i = 1, \ldots, 16$. In general, this is not possible without making some assumptions on the key K.

If $\{v_1, \ldots, v_{16}\}$ is a basis of eigenvectors of C^S, then the set of all vectors of the form $\otimes_{i=1}^{16} v_{\ell_i}$ with $\ell_i \in \{1, \ldots, 16\}$ is a basis of eigenvectors of $[C^S]^{\otimes 16}$. Suppose that E^S_{+1} is the eigenspace of C^S corresponding to eigenvalue 1, and E^S_{-1} likewise for eigenvalue -1. Any useful invariant must be an eigenvector of the diagonal matrices $C^{\kappa_i + \gamma_i}$ as well. That is, the invariants must be an element of one of the vector spaces listed in Table 1.

Table 1. Bases for the intersection of the eigenspaces of C^S and C^{γ_i}.

$\cap$	$\mathrm{Span}\{e_1, e_3, \ldots, e_{15}\}$	$\mathrm{Span}\{e_0, e_2, \ldots, e_{16}\}$
E^S_{+1}	$(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^\top$ $(0,0,1,0,1,0,1,0,-1,0,-1,0,-1,0,-1,0)^\top$	$(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^\top$
E^S_{-1}	$(0,1,0,0,0,1,0,0,0,-1,0,0,0,-1,0,-2)^\top$ $(0,0,0,1,0,0,0,1,0,0,0,-1,0,0,0,1)^\top$	$(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^\top$

The vectors $v^{\otimes 4}$ should additionally be eigenvectors of C^M. A necessary condition to this end is given by Theorem 9 (in fact, Lemma 1 is sufficient here). Using this result, only four nontrivial invariants of the form $v^{\otimes 16}$ remain. These are listed in Table 2. The first of these invariants satisfies the conditions of Theorem 8. It corresponds to the nonlinear invariant discovered by Todo, Leander and Sasaki [23]. The eigenvector in the second row of Table 2 corresponds to the invariant subspace obtained by Guo *et al.* [16].

Table 2. Invariants for Midori-64. Note that the last invariant is simply the nonlinear invariant corresponding to the second invariant (which is an invariant subspace).

Eigenvector (v for $v^{\otimes 16}$)	Weak-key class	Number of weak-keys
$(0,0,0,1,0,0,0,1,0,0,0,-1,0,0,0,1)^\top$	$\kappa_1 = \kappa_2 = 0$	2^{64}
$(1,0,1,0,1,0,1,0,-1,0,-1,0,-1,0,-1,0)^\top$	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	2^{32}
$(1,0,-1,0,-1,0,-1,0,1,0,1,0,1,0,1,0)^\top$	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	2^{32}
$(0,1,0,1,0,1,0,1,0,-1,0,-1,0,-1,0,-1)^\top$	$\kappa_1 = \kappa_2 = \kappa_3 = 0$	2^{32}

Note that the weak-key class corresponding to a given invariant (the second column in Table 2), is readily determined from the vector v. For instance, consider the vector $C^\kappa v$, with $\kappa = (\kappa_1, \ldots, \kappa_4)^\top \in \mathbb{F}_2^4$ a single nibble of the round key:

$$v = (0,0,0,1,0,0,0,1,0,0,0,-1,0,0,0,1)^\top,$$
$$C^\kappa v = (-1)^{\kappa_3+\kappa_4}(0,0,0,1,0,0,0,(-1)^{\kappa_2},0,0,0,(-1)^{1+\kappa_1},0,0,0,(-1)^{\kappa_1+\kappa_2})^\top.$$

Hence, v is invariant under C^κ provided that $\kappa_1 = \kappa_2 = 0$. Note that v is also invariant under the addition of the round constants – which has the same effect as modifying κ_4.

An alternative approach to finding invariants starts from the eigenvectors of C^M. Theorem 9 makes this method efficient. This will be the starting point to obtain more general invariants in Section 5.3.

5.3 Nonlinear Invariant for "Almost Midori-64"

In the previous section, a few eigenvectors of C^{R_i} were obtained by intersecting the eigenspaces of $C^{\mathfrak{M}}$, $C^{\mathfrak{S}}$ and $C^{K+\gamma_i}$. In general the eigenvectors of C^{R_i} are not eigenvectors of $C^{\mathfrak{M}}$ or $C^{\mathfrak{S}}$. Furthermore, the eigenvectors of C^{E_k} need not be eigenvectors of the round functions C^{R_i}. In order to find all invariants, then, it would be necessary to solve the eigenvalue problem of Definition 2 directly. As discussed before, tackling this problem is out of the scope of this paper, but a slightly more general type of invariant for Midori-64 is presented in this section.

Figure 2 shows the general idea: it may be possible to find a vector $u^{\otimes 16}$ which is mapped to a vector $v^{\otimes 16}$ by C^{R_i}, such that $C^{R_{i+1}} v^{\otimes 16} = u^{\otimes 16}$. Such a vector $u^{\otimes 16}$ would be an eigenvector of $C^{R_{i+1}} C^{R_i}$, but not of C^{R_i}.

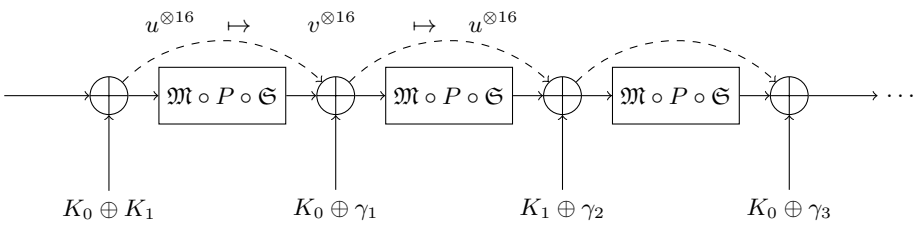

Fig. 2. If $u \neq v$, this figure depicts an invariant for two rounds which is not invariant under one round.

To find such an invariant, it suffices to obtain vectors u and $v = C^S u$ such that $C^M u^{\otimes 4} = u^{\otimes 4}$ and $C^M v^{\otimes 4} = v^{\otimes 4}$. Theorem 9 provides a complete list of possible choices for u and v. This approach is formalized in Algorithm 1[4]. This

[4] A SAGE implementation is available online at `https://homes.esat.kuleuven.be/~tbeyne/invariants/algorithm_1.html`.

algorithm requires a negligible amount of time, as the inner loop is only executed 5216 times – once for each symmetric rank one invariant of $C^{\mathfrak{M}}$. Note that it also returns invariants of the conventional type.

Algorithm 1 Finding symmetric rank-one invariants for two rounds of Midori-64.

for each affine subspace $\mathcal{A} \subseteq \mathbb{F}_2^4$ with $d := \dim \mathcal{A} \in \{0, 1, 2, 3\}$ **do**
  $S \leftarrow \{1\} \times \{1, -1\}^{2^d - 2}$
  if $d = 3$ **then**
    $S \leftarrow \{(s_1, \ldots, s_{2^d-1}, \prod_i s_i) \mid (s_1, \ldots, s_{2^d-1}) \in S\}$
  else
    $S \leftarrow S \times \{1, -1\}$
  end if
  for $(v_u)_{u \in \mathcal{A}} \in S$ **do**
    $w \leftarrow C^S v$
    $\mathcal{A}' \leftarrow \{u \in \mathbb{F}_2^4 \mid w_u \neq 0\}$
    if $\mathcal{A}'$ is affine **and** ($\dim \mathcal{A}' \neq 3$ **or** $|\{u \in \mathcal{A}' \mid w_u < 0\}|$ is even) **then**
      yield v ▷ $v^{\otimes 16}$ is invariant for some choice of round constants
    end if
  end for
end for

A list of invariants produced by Algorithm 1 is given in Appendix A. The most interesting pair of vectors u, v is given by

$$u = (0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0)^\top$$
$$v = (0,0,0,0,0,0,0,0,0,0,-1/2,-1/2,0,0,1/2,-1/2)^\top.$$

Clearly, u is invariant under the addition of any constant. For v, it holds that

$$C^\kappa v = (-1)^{\kappa_1+\kappa_3}/2 \cdot (0,0,0,0,0,0,0,0,0,0,-1,(-1)^{1+\kappa_4},0,0,(-1)^{\kappa_2},(-1)^{1+\kappa_2+\kappa_4})^\top,$$

which is a multiple of v provided that $\kappa_2 = \kappa_4 = 0$. For the usual choice of round constants of Midori-64, v is not invariant under the addition of the constants. However, had the round constants been chosen as $\gamma_i \in \{\mathtt{0}, \mathtt{2}, \mathtt{8}, \mathtt{A}\}^{16}$ rather than $\gamma_i \in \{0, 1\}^{16}$, the attack would apply. Moreover, such a restriction only applies to half of the rounds – the round constants of other rounds may be chosen arbitrarily.

The restriction $\kappa_2 = \kappa_4 = 0$ (which applies to K_0 or K_1, but not both) corresponds to a class of 2^{96} weak keys. By Theorem 8, v corresponds to the following nonlinear invariant:

$$f(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} [x_{4i}x_{4i-2} + x_{4i} + x_{4i-1} + x_{4i-3}] \tag{7}$$

That is, there exists a constant $c \in \mathbb{F}_2$ such that $f(E_k(x)) + f(x) = c$ for all x and for any even number of rounds. By Theorem 8, u corresponds to the following "nonlinear" invariant:

$$g(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} [x_{4i} + x_{4i-2}] \,. \tag{8}$$

Hence, for an even number of rounds, $g(E_k(x)) + g(x)$ is constant. Note that if the number of rounds is odd, the value $f(E_k(x)) + g(x)$ is constant instead. Appendix B provides test code for this property.

5.4 Trail Clustering in Midori-64

It is worthwhile to take a closer look at the invariant g given by (8) in Section 5.3. Since g is a linear function, it corresponds to a linear hull with correlation ± 1 (where the sign depends on the key). Considering the fact that Midori-64 has been designed with resistance to linear cryptanalysis in mind, this is remarkable.

Remark 1 *The correlation of any trail in "almost Midori-64" is (much) smaller than* 2^{-32}*, yet there is a linear hull with correlation* ± 1 *for* 2^{96} *keys.*

The correlation of a linear hull is equal to the sum of the correlations of all trails within the hull. It is well-established that, in theory, this sum could become large even if all terms are small. Such ideas go back to Nyberg [22]. Daemen and Rijmen [10] refer to this effect as *trail clustering*.

Remark 1 demonstrates an extreme case of trail clustering: the absolute correlation of the hull is not just large, it is maximal. This appears to be the first real-world observation of such behavior.

5.5 Additional Weak Keys for the Invariant from Section 5.3

This section shows that the invariant u from Section 5.3 is invariant under 2^{64} additional weak keys, under the same modifications of the round constants. Although 2^{64} is small compared to 2^{96}, the result is interesting because it provides an example of an invariant over four rounds which is not necessarily invariant over two rounds.

Let u and v be as defined at the end of Section 5.3. For any $\kappa \in \mathbb{F}_2^4$ with $\kappa_2 = \kappa_4 = 1$, we have

$$C^\kappa v = (-1)^{\kappa_1 + \kappa_3}/2 \cdot (0,0,0,0,0,0,0,0,0,0,-1,1,0,0,-1,-1)^\top .$$

Let $w = (-1)^{\kappa_1+\kappa_3} C^\kappa v$. By Theorem 9, $w^{\otimes 4}$ is an invariant of C^M. Furthermore, one can check that w is an eigenvector of C^S.

Hence, there exist 2^{32} keys K such that $C^K \, v^{\otimes 16} = \pm w^{\otimes 16}$ with $w^{\otimes 16}$ invariant under the round function. This observation can be used to show that $u^{\otimes 16}$ defines an invariant for $2^{96} + 2^{64}$ rather than 2^{96} weak keys. Figure 3 illustrates

this. The top branch in Figure 3 corresponds to the discussion in Section 5.3 and holds assuming that $K_{0,4i-2} = K_{0,4i} = 1$ for $i = 1, \ldots, 16$. The bottom branch corresponds to a different set of weak keys for which $K_{0,4i-2} = K_{0,4i} = 1$ and $K_{1,4i-2} = K_{1,4i} = 0$ for $i = 1, \ldots, 16$. Hence, the 4-round invariant in Figure 3 and its full-round extension hold for $2^{96} + 2^{64}$ weak keys.

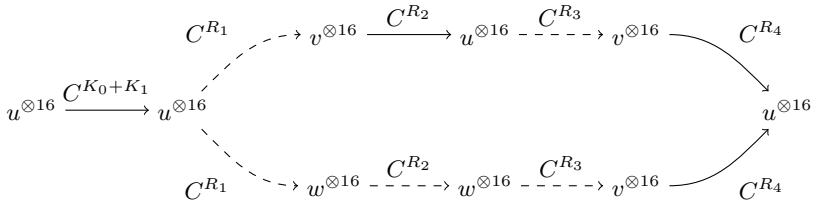

Fig. 3. The invariant from Section 5.3 holds for $2^{96} + 2^{64}$ weak keys. Dashed arrows indicate transitions for which an assumption on the round keys is necessary.

6 Practical Attack on 10 Rounds of Midori-64

The purpose of this section is to demonstrate that the invariant for "almost Midori-64" can be used even when the round constants are not modified. In fact, the attack in this section is valid for any choice of round constants.

Specifically, it will be shown that 10 rounds of Midori-64 are subject to a key-recovery attack that requires $1.25 \cdot 2^{21}$ chosen plaintexts and has a computational cost of 2^{56} block cipher calls. The downside of this attack is that it is limited to 2^{96} out of 2^{128} keys. Note that Midori-64 has been analyzed in several prior works. Lin and Wu [19] demonstrate meet-in-the-middle attacks on 10, 11 and 12 rounds of Midori-64. Chen and Wang [24] give a 10 round impossible differential cryptanalysis. The downside of those attacks is that they can not be executed in practice. Table 3 provides an overview of attacks on Midori-64.

Table 3. Overview of key-recovery attacks on Midori-64. Time is measured by the number of encryption operations. Memory is expressed in number of bytes.

Attack	Rounds	Time	Memory	Data	Weak keys	Reference
Meet-in-the-middle	10	$2^{99.5}$	$2^{95.7}$	$2^{59.5}$	N/A	Lin and Wu [19]
Meet-in-the-middle	11	2^{122}	$2^{92.2}$	2^{53}	N/A	Lin and Wu [19]
Meet-in-the-middle	12	$2^{125.5}$	2^{109}	$2^{55.5}$	N/A	Lin and Wu [19]
Impossible differential	10	$2^{80.8}$	$2^{68.1}$	$2^{62.4}$	N/A	Chen and Wang [24]
Invariant subspace	16	2^{16}	–	2	2^{32}	Guo *et al.* [16]
Nonlinear invariant*	16	$2^{15}h$	–	$33h$	2^{64}	Todo *et al.* [23]
Integral/invariant	10	2^{56}	–	$2^{21.3}$	2^{96}	Section 6

* This is an attack on a mode of operation. It recovers $32h$ bits of h encrypted blocks.

The attack presented below is based on the observation that integral properties [17] and invariants can often be combined. However, since we allow arbitrary round constants in this section, the invariant can only be used once. In this regard the nonlinear invariant that was introduced in Section 5.3 has an important advantage: with one assumption on the key, it covers two rounds.

6.1 Nonlinear Property for 6 Rounds of Midori-64

This section shows that the two-round nonlinear invariant for Midori-64 can be extended to a six round nonlinear property. When a key which does not belong to the weak key class is added to the state, the vector corresponding to a nonlinear invariant will be mapped to another vector which only depends (up to a scale factor) on key bits that are already "known", *i.e.* that had to be fixed to obtain the invariant in the first place. This holds in both the forward and backward direction, leading to a 6-round nonlinear property. This is illustrated in Figure 4.

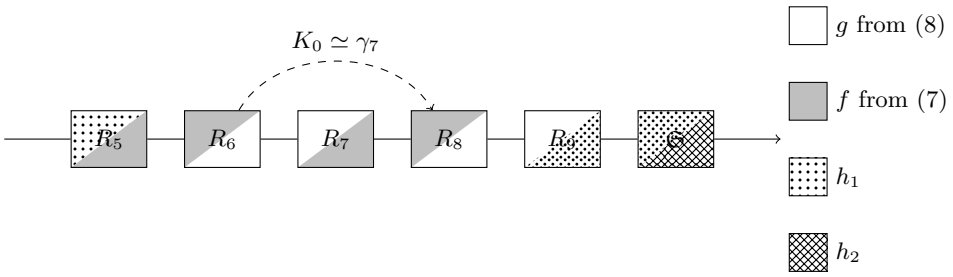

Fig. 4. Nonlinear property over six rounds of Midori-64. The notation "$\simeq$" is used to indicate equality in the second and fourth bits of every nibble of each of its arguments.

The functions h_1 and h_2 in Figure 4 depend on the choice of the round constants. Specifically, h_1 depends on $P^{-1}(\mathfrak{M}(\gamma_5+\gamma_7))$ and h_2 depends on $\gamma_7+\gamma_9$. For the purposes of this paper, a detailed description of h_1 is not necessary. For h_2, it holds that

$$h_2(x_1,\ldots,x_{64}) = \sum_{i=1}^{16} f(S(x_{4i-3},x_{4i-2},x_{4i-1},x_{4i}) + \gamma_{7,i} + \gamma_{9,i}).$$

In general, h_j can be written in the form

$$h_j(x_1,\ldots,x_{64}) = \sum_{i=1}^{16} h^{(\beta_{j,2i},\beta_{j,2i+1})}(x_{4i},x_{4i+1},x_{4i+2},x_{4i+3}), \tag{9}$$

where $\beta_j \in \mathbb{F}_2^{32}$ is a constant depending on the round constants. In particular, β_2 consists of the second and fourth bits of every nibble of $\gamma_7+\gamma_9$. For the default choice of round constants of Midori-64, $\beta_{j,2i} = 0$. Hence, only two different Boolean functions can occur as terms in (9):

$$h^{(00)}(x_1,x_2,x_3,x_4) = x_2 + x_4$$
$$h^{(01)}(x_1,x_2,x_3,x_4) = x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_3 + x_1x_4 + x_1 + x_2.$$

Since the functions h_1 and h_2 are balanced *on every cell* of the state, it holds that $\sum_{x\in S} h_i(x) = 0$ with S a set of state values such that every cell takes all values exactly once. This makes it possible to combine integral cryptanalysis with the 6-round nonlinear property described above.

6.2 Integral Property for 4 Rounds of Midori-64

An integral attack on Midori-64 that is suitable for our purposes will now be given. The following notation will be used: cells taking all values an equal number of times are denoted using the label "A", constant cells will be labeled by

"C". Subscripts are used to denote groups of values which jointly satisfy the "A" property. Note that cells can be part of several groups, *e.g.* a cell marked "$A_{i,j}$" is contained in groups i and j. The Midori-64 designers discuss the existence of a 3.5 round integral distinguisher. In fact, one can see that a 4-round integral property[5] exists. Note that the property is nearly identical to the Rijndael distinguisher discussed by Knudsen and Wagner [17], the difference being that the property works better than expected for Midori-64.

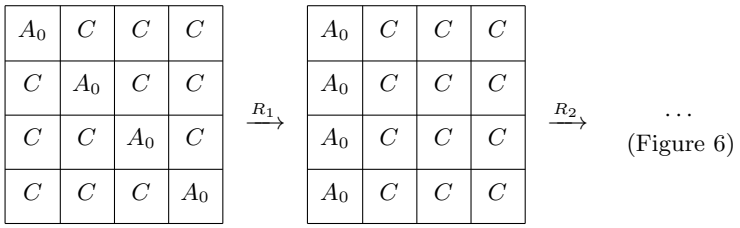

Fig. 5. First two rounds of the integral property for four rounds of Midori-64.

The integral property is based on a set of chosen plaintexts such that the diagonal cells take all possible values exactly once and all other cells are constant. After one round, the same property then holds for the first column whereas all other cells are constant. This is shown in Figure 5.

The effect of the remaining rounds is shown in Figure 6. Figure 6 shows that, before the last application of $\mathfrak{M}$, any four distinct cells in a column jointly satisfy the "A" property. This implies that all cells can be labeled "A" after four rounds.

The derivation in Figure 6 starts by forming appropriate groups of cells which are independent before the third round. Four (sometimes overlapping) groups of such cells are indicated using "A_i", $i = 0, \ldots, 3$ in Figure 6. The maps $\mathfrak{S}$ and P preserve the groups. Furthermore, one can see that four new groups can be obtained after the application of $\mathfrak{M}$. These groups can be chosen in such a way that they are aligned in different columns of the state after P has been applied. The four round property then follows.

6.3 Combination of the Nonlinear and Integral Properties

The final attack can now be described. Figure 7 provides an overview. Let $\mathcal{I}$ denote a set of plaintext/ciphertext pairs with the structure required by the integral property from Figure 5. Then, due to the nonlinear property from Figure 4, the following holds:

$$\sum_{(P,C)\in\mathcal{I}} h_2(C + K_0 + K_1) = \sum_{(P,C)\in\mathcal{I}} h_1((R_4 \circ \cdots \circ R_1)(P + K_0 + K_1)) = 0. \quad (10)$$

[5] If the zero-sum property can be used, this actually yields a 5-round property.

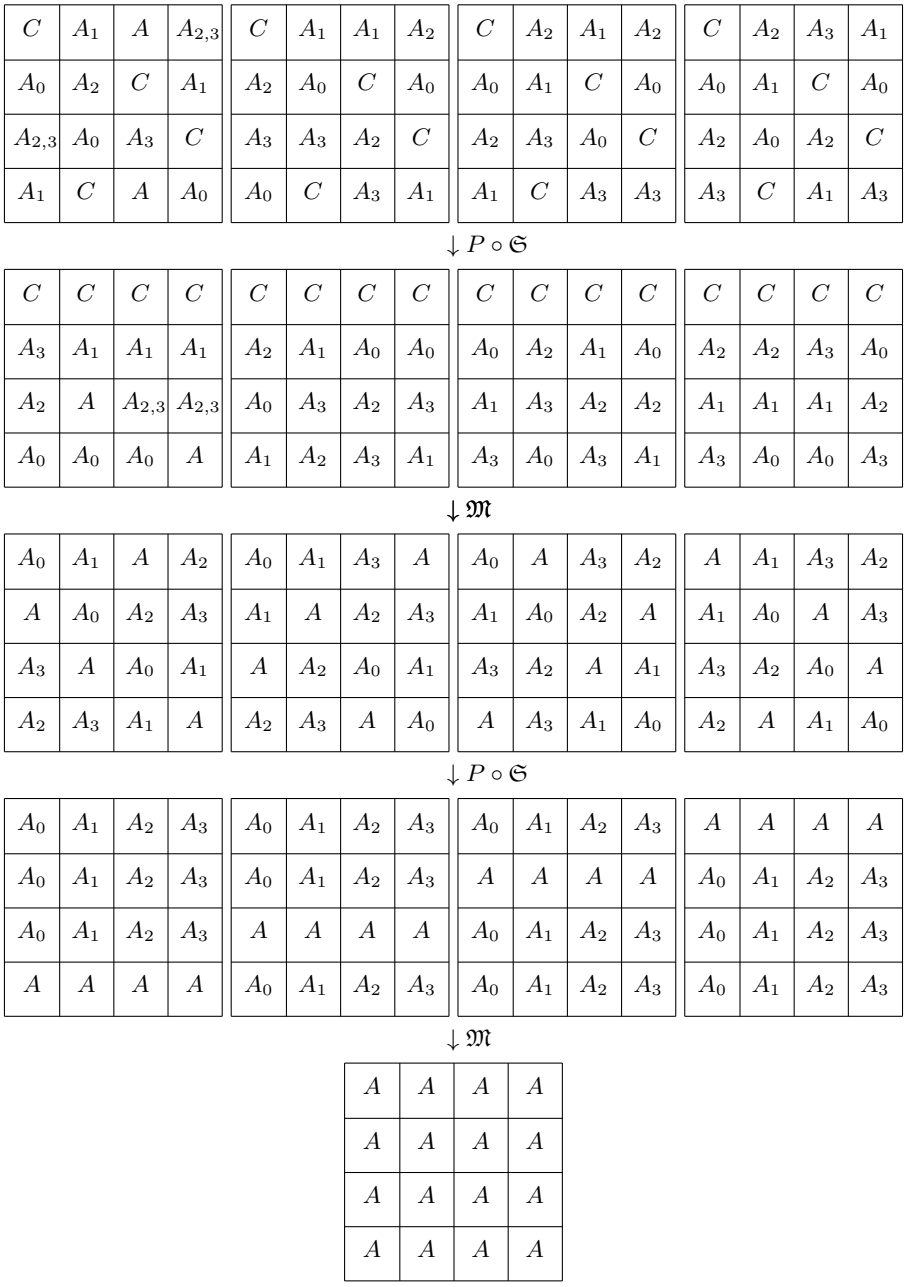

Fig. 6. Last two rounds of the integral property for four rounds of Midori-64.

Hence, every set $\mathcal{I}$ defines a low-degree nonlinear polynomial equation in (part of) $K_0 + K_1$. Given enough such equations, one observes that a Gröbner basis for the ideal generated by these polynomials can be efficiently (within a second on a regular computer) computed. Although computing Gröbner bases is hard in general, it is easy in this case due to the fact that key bits from different cells are never multiplied together.

Note that only those key bits which are involved in h_2 in a nonconstant way can be recovered by solving the system of polynomial equations. That is, the number of key bits recovered is four times the number of nonlinear terms in (9). For the default Midori-64 round constants, 40 key bits can be recovered. It was observed that these bits are often uniquely determined given 40 equations. This requires $40 \cdot 2^{16} = 1.25 \cdot 2^{21}$ chosen plaintexts. A more detailed analysis of the data requirements is provided in Section 6.4.

The remaining 24 bits of $K_0 + K_1$ can be guessed, along with the 32 unknown bits in K_0. This requires 2^{56} block cipher calls. Note that this additional work is only necessary after it has been established that a weak key is used. Hence, an attacker in the multi-key setting has a very efficient method to identify potential targets.

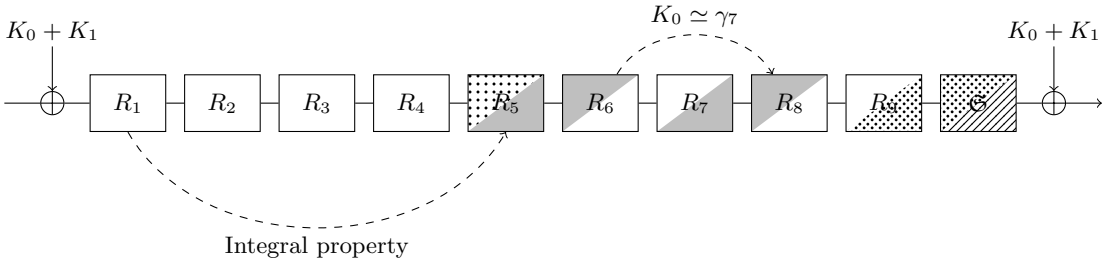

Fig. 7. Overview of the attack on 10 rounds of Midori-64.

6.4 Detailed Analysis of the Data Requirements

The data requirements of the attack are determined by the number of equations that are necessary to recover the 40 bits of $K_0 + K_1$ that can occur as indeterminates in (10). If the constant cells of each integral plaintext set are selected independently and uniformly at random, then the probability that the system of equations has a unique solution may be computed. Figure 8 provides an estimate of this probability based on a sample of 200 key-recovery experiments.

For 40 equations – *i.e.* $1.25 \cdot 2^{21}$ chosen plaintexts – Figure 8 shows that the probability of recovering all 40 bits of the key is roughly 35%. With one additional equation, a probability of nearly 60% is obtained.

Note that even if the system does not have a unique solution, typically only a few additional bits of $K_0 + K_1$ will have to be guessed in the second phase

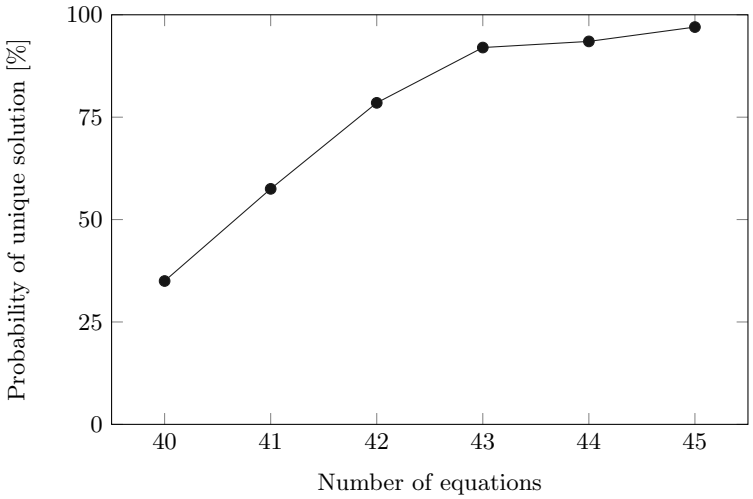

Fig. 8. Probability that the system of equations for key-recovery has a unique solution. The equations are constructed from (10) by selecting the constant cells in the integral plaintext sets independently and uniformly at random.

of the attack. In order to minimize the required amount of chosen plaintexts, additional equations may be constructed only when necessary.

7 Practical Attack on MANTIS-4

This section presents an attack on the block cipher MANTIS [5], which is closely related to Midori-64. Dobraunig, Eichlseder, Kales and Mendel give a practical attack against MANTIS-5 in the chosen tweak setting [12]. This attack has been extended to six rounds by Eichlseder and Kales [14]. The attack presented in this section is limited to MANTIS-4, but the assumptions about the capabilities of the attacker are different. The attacker is not allowed to choose the tweak, but it is assumed that a *weak tweak* is used. It will be shown that for every choice of the key, there are 2^{32} (out of 2^{64}) weak tweaks. When a weak tweak is used, the full key can be recovered from (on average) 346 chosen plaintexts and with a computational cost of approximately 2^{56} block cipher calls. If, in addition, 346 chosen plaintexts for a single related tweak are available, the computational cost reduces to roughly 2^{18} block cipher calls. Table 4 contains an overview of attacks on MANTIS.

Table 4. Overview of key-recovery attacks on MANTIS–r. Time is measured by the number of encryption operations.

Attack	r	Time	Memory	Data	Weak tweaks	Reference
Truncated differential*	5	2^{28}	–	2^{38}	N/A	Dobraunig *et al.* [12]
Truncated differential*	6	$2^{53.5}$	–	$2^{53.5}$	N/A	Eichlseder *et al.* [14]
Zero-correlation/integral*†	3/7	$2^{66.2}$	$2^{48.4}$	$2^{53.7}$	N/A	Ankele *et al.* [2]
Integral/invariant	4	2^{56}	–	346	2^{96}	Section 7.1
Integral/invariant*	4	2^{18}	–	692	2^{96}	Section 7.4

$*$ These attacks rely on related tweaks.
† This attack applies to a version of MANTIS with an asymmetric number of rounds in the inbound (3) and outbound (7) direction. Such attacks are not considered in this paper, but the techniques from this section could be used to obtain key-recovery attacks for MANTIS-6/4.

Figure 9 illustrates the overall structure of MANTIS-4. Unlike in Midori-64, the round key K_1 is the same in all rounds. Additional whitening keys K_0 and $K_0' = (K_0 \ggg 1) + (K_0 \gg 63)$ are added before the first round and after the last round. The round function is nearly identical to the Midori-64 round function, the difference being that the round keys and constants are added before rather than after the application of $\mathfrak{M}$. Hence, the 2-round nonlinear invariant for Midori-64 also applies to MANTIS-4. Note that the values of the round constants $\mathrm{RC}_1, \ldots, \mathrm{RC}_4$ are not essential to the attack described here.

Structurally, MANTIS differs from Midori-64 in two major aspects: it takes an additional tweak as an input, and it is a reflection cipher. In every round, the tweak is permuted cellwise by a permutation σ. In all other aspects, the tweak is treated in the same way as the round key K_1. The reflection property

enables extending the 6-round nonlinear property of Midori-64 to eight rounds. The presence of a tweak allows mounting a weak tweak rather than a weak key attack, which corresponds to a significantly weaker adversarial model.

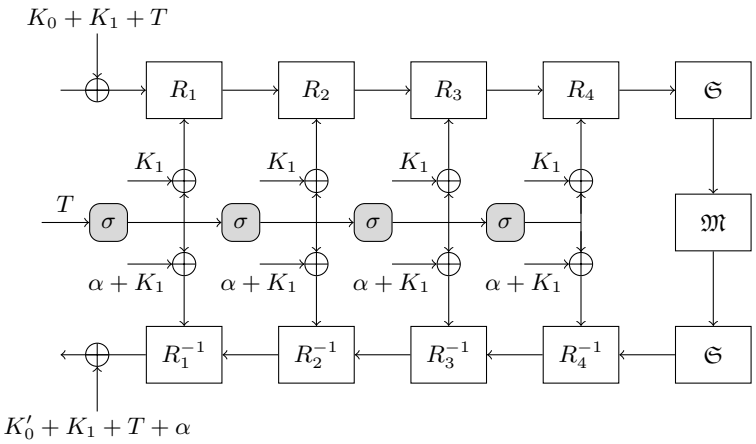

Fig. 9. Overview of MANTIS-4.

7.1 Description of the Attack

An overview of the attack is shown in Figure 10. As in the attack on Midori-64 from Section 6, a few initial rounds are covered by an integral property. Since the nonlinear property extends over eight rounds for MANTIS, it suffices to use a weaker integral property. Figure 11 shows the property that will be used. It requires 16 chosen plaintexts.

The nonlinear property is similar to the property that was discussed in Section 6, but slightly more complicated. Specifically, due to the tweak-key schedule, the functions h_1 and h_2 can depend on the tweak. As for Midori-64, h_1 and h_2 can be written in the form

$$h_j(x_1, \ldots, x_{64}) = \sum_{i=1}^{16} h^{(\beta_{j,2i}, \beta_{j,2i+1})}(x_{4i}, x_{4i+1}, x_{4i+2}, x_{4i+3}), \tag{11}$$

where $\beta_j = (\beta_{j,1}, \ldots, \beta_{j,32}) \in \mathbb{F}_2^{32}$ is a constant that possibly depends on the tweak and the functions $h^{(\beta_{j,2i}, \beta_{j,2i+1})}$ are given by

$$h^{(00)}(x_1, x_2, x_3, x_4) = x_2 + x_4$$
$$h^{(11)}(x_1, x_2, x_3, x_4) = x_2x_4 + x_1 + x_2 + x_3$$
$$h^{(01)}(x_1, x_2, x_3, x_4) = x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_3 + x_1x_4 + x_1 + x_2$$
$$h^{(10)}(x_1, x_2, x_3, x_4) = x_1x_2x_3 + x_1x_3x_4 + x_2x_3x_4 + x_1x_4 + x_2x_4 + x_2 + x_3 + x_4.$$

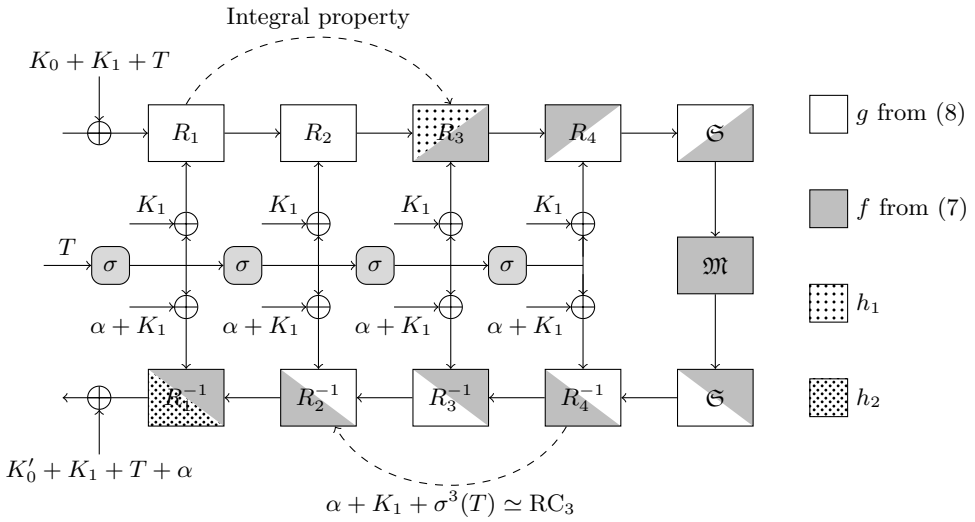

Fig. 10. Nonlinear property over eight rounds of MANTIS-4. The notation "$\simeq$" is used to indicate equality in the second and fourth bits of every nibble of each of its arguments.

Note that all of these functions are balanced. The constant β_1 consists of the second and fourth bits of every nibble of α. For convenience, this will be denoted by $\beta_1 \simeq \alpha$. For β_2, we have $\beta_2 \simeq \mathrm{RC}_1 + \alpha + K_1 + \sigma(T)$. This implies that

$$\beta_2 \simeq \mathrm{RC}_1 + \mathrm{RC}_3 + \sigma(T) + \sigma^3(T).$$

Let $\mathcal{I}$ denote a set of plaintext/ciphertext pairs such that the plaintexts have the structure required by the integral property, then

$$\sum_{(P,C)\in\mathcal{I}} h_2(C + K_0' + K_1 + T + \alpha) = \sum_{(P,C)\in\mathcal{I}} h_1(R_1(R_2(P + K_0 + K_1 + T))) = 0.$$

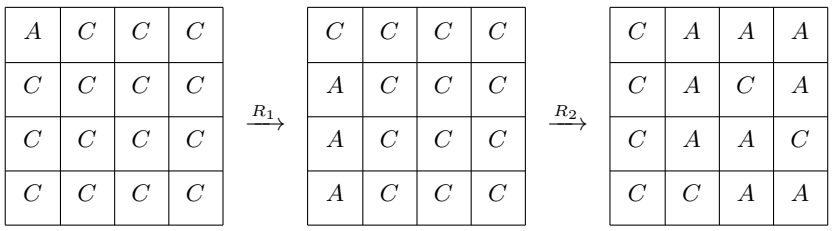

Fig. 11. Integral property for two rounds of MANTIS.

Hence, each set $\mathcal{I}$ corresponds to a low-degree polynomial equation in (part of) the key. As in Section 6, a Gröbner basis for the ideal generated by these polynomials can be efficiently computed.

As in the attack on Midori-64, only those key bits which are involved in h_2 in a nonconstant way can be recovered by solving the system of polynomial equations. For simplicity, assume that the functions $h^{(00)}$, $h^{(01)}$, $h^{(10)}$ and $h^{(11)}$ all occur as terms in (11) in the same proportion. Then the expected number of key bits that can be recovered by solving the system of polynomial equations is equal to 40.[6] For obtaining 40 key bits, it was observed that 40 equations are sufficient. This requires $2^4 \cdot 40 = 640$ chosen plaintexts.

The remaining bits of the whitening key $K_0' + K_1$ (24 bits on average) can then be guessed, along with the 32 unknown bits of K_1. For each such guess, it is possible to compute K_0' (since $K_0' + K_1$ is already known) and hence K_0. No additional plaintext/ciphertext pairs are necessary to carry out this process. Hence, the work required for the entire key-recovery attack is then roughly 2^{56} block cipher calls.

7.2 Reducing the Data Requirements by Overlapping Integral Sets

Figure 11 shows one possible integral property for two rounds of MANTIS, but many alternatives exist. One example is shown in Figure 12. Since the input sets for the integral properties in Figures 11 and 12 overlap for an equal choice of the constant cells, the data requirements can be reduced.

For example, to obtain 40 distinct integral sets from only 316 chosen plaintexts, one proceeds as follows. First, choose 2^8 plaintexts such that the first byte of the state takes all possible values. This yields a total of 32 overlapping integral sets of size 16: half of these correspond to the integral property in Figure 11, the other half to that in Figure 12. For the eight remaining integral sets, choose one of the already queried plaintexts and build the integral set by letting the third cell take all possible values – this corresponds to yet another integral property similar to that in Figures 11 and 12. Overall, this requires $2^8 + 8 \cdot 15 = 316$ chosen plaintexts.

Note that the same technique can be applied to the attack on Midori-64 from Section 6, but it only reduces the data requirements by 40 chosen plaintexts.

7.3 Detailed Analysis of the Data Requirements

As remarked in Section 7.1, the number of whitening key bits that may be recovered depends on the value of the tweak. Specifically, it depends on the value of β_2 in (11). Recall that β_2 consists of the second and forth bits of each nibble of $\mathrm{RC}_1 + \mathrm{RC}_3 + \sigma(T) + \sigma^3(T)$. Indeed, every term of the form $h^{(01)}$ or $h^{(10)}$ may contribute four unknowns to the system of equations in the key. A

[6] For some tweaks, many more key bits can be recovered, and for others only a small number of key bits can be recovered. A detailed analysis is provided in Section 7.3.

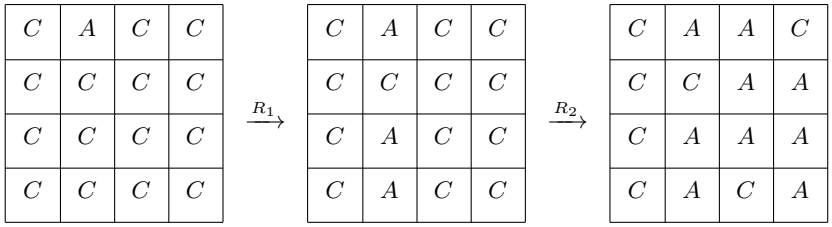

Fig. 12. Alternative integral property for two rounds of MANTIS.

term of the form $h^{(11)}$ contributes at most two unknown key bits, whereas $h^{(00)}$ is linear and hence does not supply any key bits.

In Section 7.1, it was estimated that 40 bits of the key may be recovered. This corresponds to the average value for a uniform random choice of round constants. For a fixed choice of RC_1 and RC_3, the average number of recoverable key bits may be computed as follows. Clearly, $\sigma(\boldsymbol{T}) + \sigma^3(\boldsymbol{T})$ and $\boldsymbol{x} + \sigma^2(\boldsymbol{x})$ have the same probability distribution when $\boldsymbol{T}$ and $\boldsymbol{x}$ are uniformly distributed random variables. Figure 13 illustrates the values of the nibbles of $\boldsymbol{x} + \sigma^2(\boldsymbol{x})$. The value of two cells, corresponding to fixed points of σ^2, is fixed whereas the other cells are individually – but not jointly – uniformly distributed.

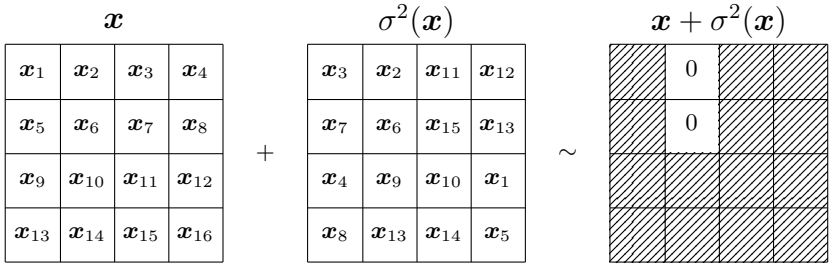

Fig. 13. Illustration of the distribution of $\boldsymbol{x} + \sigma^2(\boldsymbol{x})$ with $\boldsymbol{x}$ uniformly distributed. The hatched cells are individually uniformly distributed, but their joint distribution is not uniform.

Hence, the average number of recoverable key bits depends only on the part of $\mathrm{RC}_1 + \mathrm{RC}_3$ corresponding to the two unhatched cells in the right part of Figure 13. Specifically, since these cells contribute terms of the form $h^{(01)}$ and $h^{(00)}$, it follows by linearity of expectation that the average number of key bits that can be recovered equals $4 + 14\,(2 + 1/2) = 39$. Figure 14 shows a histogram of the number of recovered key bits for 100000 tweaks sampled uniformly at random (with replacement). Remark that the distribution is right-skewed. In

particular, while the mean number of recovered bits is 39, the median is in fact 40. The probability that at least 40 key bits can be recovered was estimated as 50.4%.

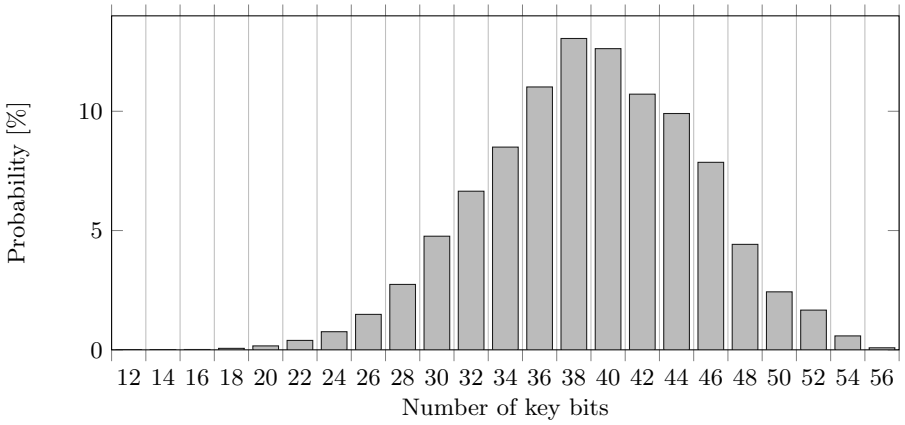

Fig. 14. Estimated probability distribution of the number of key bits that can be recovered given a sufficiently large number of equations, for a randomly chosen tweak.

As for the attack on Midori-64, the data requirements of the attack depend on the number of equations that are needed to uniquely recover the relevant key bits. The analysis is similar to that in Section 6.4. Figure 15 shows an estimate of the probability that the system of equations, when constructed from uniform random (overlapping) integral sets, has a unique solution. It was observed that if the integral sets overlap, the probability of recovering all key bits is lower so that an additional equation is typically necessary.

To recover all 40 bits of the key with a success probability greater than 50%, 42 equations suffice. This corresponds to $2^8 + 6 \cdot 15 = 346$ chosen plaintexts.

7.4 Improved Attack using Related-Tweak Chosen Ciphertexts

If a small number of additional chosen ciphertexts under a single related tweak are available, the computational cost of the attack can be significantly reduced. Specifically, given 346 chosen ciphertexts, the key-recovery cost can be reduced to 2^{18} block cipher calls. The basic idea is to perform the attack from Section 7.1 (without the brute-force phase) on the inverse cipher. An overview of the inverse attack is shown in Figure 16. Remark that the condition on the round key differs from that in Figure 10. Hence, in order to ensure that the property works for the same key K_1, a related tweak T' must be used. The only requirement on T' is that

$$T' \simeq T + \sigma^{-3}(\alpha),$$

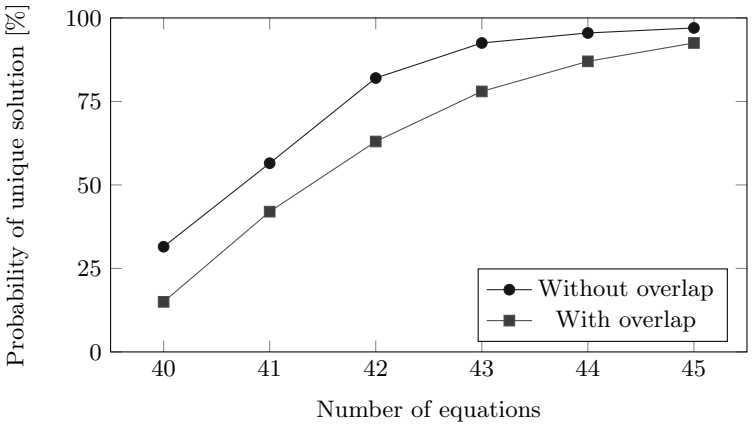

Fig. 15. Probability that the system of equations for key-recovery on MANTIS-4 has a unique solution, estimated based on a sample of 200 key-recovery experiments.

where the symbol "$\simeq$" indicates equality in the second and fourth bits of every nibble. Hence, there are 2^{32} valid choices for the related tweak T'.

As in Section 7.1, an eight round nonlinear approximation is combined with a two round integral property. Each integral set $\mathcal{I}$ defines an equation

$$\sum_{(P,C)\in\mathcal{I}} h_2'(P + K_0 + K_1 + T') = 0,$$

where h_2' is defined as in (11) but with a different constant $\beta_2' \simeq \beta_2 + \alpha + \sigma^{-2}(\alpha)$.

Since the bits of β_2' corresponding to the unhatched cells in Figure 13 are zero, the expected number of bits of $K_0 + K_1$ that can be recovered is the same as for the forward attack. Remark that, in the forward attack, one recovers bits of $K_0' + K_1$ with $K_0' = (K_0 \ggg 1) + (K_0 \gg 63)$ instead. One thus obtains a system of linear equations in K_0 and K_1. By linearity of expectation, the average number of equations is equal to $2 \cdot 39 = 78$. An estimate of the actual distribution of the number of equations is given in Figure 17. Since K_0 and K_0' are related by an invertible linear transformation, the equations in the system will be linearly independent.

In conclusion, given 346 chosen plaintexts and 346 chosen ciphertexts for a related tweak, the full key can usually be recovered at a cost of 2^{18} block cipher calls. The cost of the Gröbner basis computations appears to be significantly smaller than 2^{18} encryption operations, but this may depend on the details of the implementation.

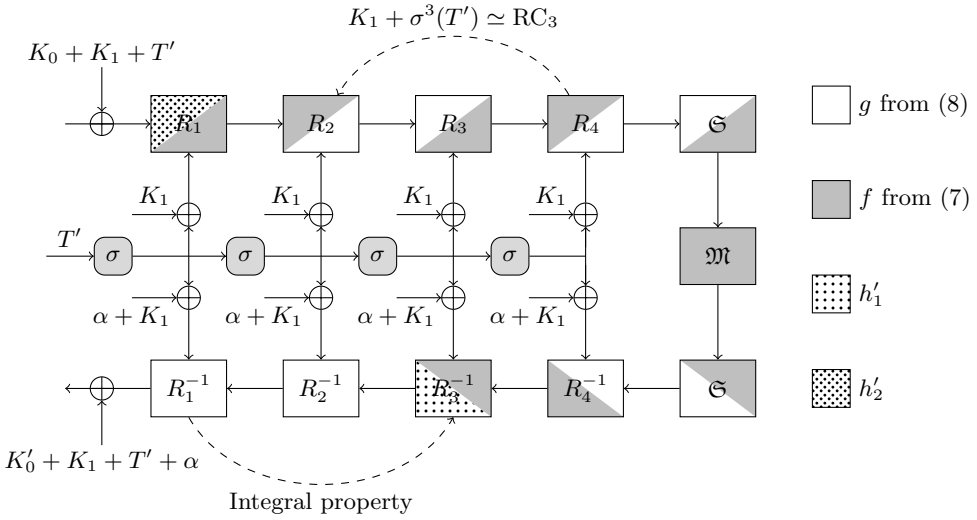

Fig. 16. Attack on MANTIS-4 in the reverse direction. The notation "$\simeq$" is used to indicate equality in the second and fourth bits of every nibble of each of its arguments.

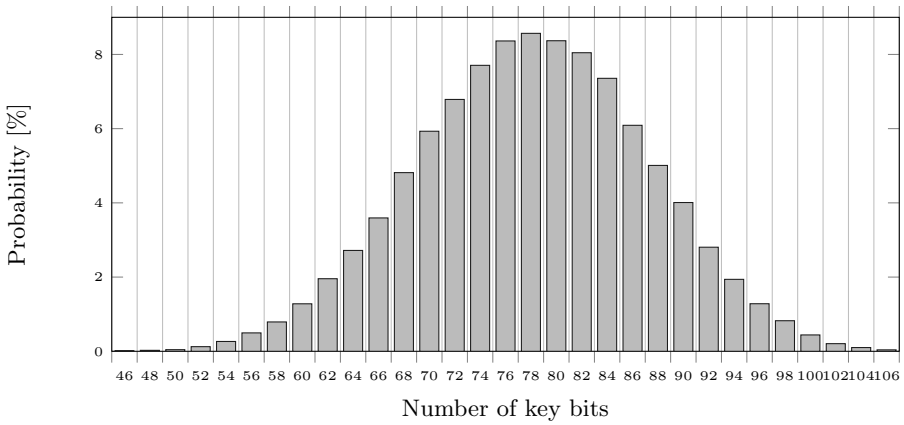

Fig. 17. Estimated probability distribution of the number of key bits that can be recovered given a sufficiently large number of equations, for a randomly chosen tweak. Note that the distribution is more symmetric than the distribution from Figure 14.

8 Future Work

Returning to Definition 2, one potentially interesting direction for future work is the use of complex eigenvalues. The corresponding eigenvectors are related to real invariants of $[C^{E_k}]^l$ with l the order of the corresponding eigenvalue. If l is not too large, then such invariants might lead to additional attacks.

Another topic that deserves more attention is the development of practical methods to compute an eigenvector basis for the correlation matrix of the entire round function. Even if this does not lead to new attacks, it could be a tool for designers to demonstrate security with respect to attacks based on invariants.

Yet another direction for future work is to improve and extend the attack on 10 rounds of Midori-64 from Section 6 and the attack on MANTIS-4 from Section 7.

9 Conclusion

The three problems mentioned in the introduction have been addressed. In Section 4, a new theory of block cipher invariants was developed. Beside providing the foundation for the remainder of the paper, Definition 2 provides insight and uncovers several directions for future research. Section 5 provides a detailed analysis of invariants in Midori-64, leading to a new class of 2^{96} weak keys when minor modifications to the round constants are made. It was shown that this invariant is equivalent to a linear hull with maximal correlation. Finally, Sections 6 and 7 illustrate the importance of invariants, even when round constants initially seem to limit their applicability. Two practical attacks were described: (1) a key-recovery attack on 10-round Midori-64 for 2^{96} weak keys, requiring $1.25 \cdot 2^{21}$ chosen plaintexts (2) a key-recovery attack on MANTIS-4 with an average data complexity of 346 chosen plaintexts.

Acknowledgments. I acknowledge the anonymous referees for their comments and corrections. In addition, I thank Tomer Ashur and Yunwen Liu for discussions related to this work. Finally, I am especially grateful to Vincent Rijmen for his comments on a draft version of this paper, and for his support.

References

1. Abdelraheem, M.A., Ågren, M., Beelen, P., Leander, G.: On the distribution of linear biases: Three instructive examples. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 50–67. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2012)
2. Ankele, R., Dobraunig, C., Guo, J., Lambooij, E., Leander, G., Todo, Y.: Zero-correlation attacks on tweakable block ciphers with linear tweakey expansion. IACR Trans. Symmetric Cryptol. **2019**(1), 192–235 (2019). https://doi.org/10.13154/tosc.v2019.i1.192-235, `https://doi.org/10.13154/tosc.v2019.i1.192-235`

3. Banik, S., Bogdanov, A., Isobe, T., Shibutani, K., Hiwatari, H., Akishita, T., Regazzoni, F.: Midori: A block cipher for low energy. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015, Part II. LNCS, vol. 9453, pp. 411–436. Springer, Heidelberg, Germany, Auckland, New Zealand (Nov 30 – Dec 3, 2015). https://doi.org/10.1007/978-3-662-48800-3_17
4. Beierle, C., Canteaut, A., Leander, G., Rotella, Y.: Proving resistance against invariant attacks: How to choose the round constants. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017, Part II. LNCS, vol. 10402, pp. 647–678. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 20–24, 2017)
5. 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, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53008-5_5
6. Biryukov, A., Perrin, L.: State of the art in lightweight symmetric cryptography. Cryptology ePrint Archive, Report 2017/511 (2017), http://eprint.iacr.org/2017/511
7. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knežević, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçin, T.: PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg, Germany, Beijing, China (Dec 2–6, 2012). https://doi.org/10.1007/978-3-642-34961-4_14
8. Ceccherini-Silberstein, T., Scarabotti, F., Tolli, F.: Harmonic Analysis on Finite Groups. Cambridge University Press (2008)
9. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Preneel, B. (ed.) FSE'94. LNCS, vol. 1008, pp. 275–285. Springer, Heidelberg, Germany, Leuven, Belgium (Dec 14–16, 1995)
10. Daemen, J., Rijmen, V.: The wide trail design strategy. In: Honary, B. (ed.) 8th IMA International Conference on Cryptography and Coding. LNCS, vol. 2260, pp. 222–238. Springer, Heidelberg, Germany, Cirencester, UK (Dec 17–19, 2001)
11. Diaconis, P.: Group representations in probability and statistics, Lecture Notes–Monograph Series, vol. 11. Institute of Mathematical Statistics, Hayward, CA (1988). https://doi.org/10.1214/lnms/1215467418
12. Dobraunig, C., Eichlseder, M., Kales, D., Mendel, F.: Practical key-recovery attack on MANTIS5. IACR Trans. Symm. Cryptol. **2016**(2), 248–260 (2016). https://doi.org/10.13154/tosc.v2016.i2.248-260, http://tosc.iacr.org/index.php/ToSC/article/view/573
13. Dravie, B., Parriaux, J., Guillot, P., Millérioux, G.: Matrix representations of vectorial boolean functions and eigenanalysis. Cryptography and Communications - Discrete Structures, Boolean Functions and Sequences **8**(4), 555–577 (Oct 2016). https://doi.org/10.1007/s12095-015-0160-7, https://hal.archives-ouvertes.fr/hal-01259921
14. Eichlseder, M., Kales, D.: Clustering related-tweak characteristics: Application to MANTIS-6. IACR Trans. Symm. Cryptol. **2018**(2), 111–132 (2018). https://doi.org/10.13154/tosc.v2018.i2.111-132
15. Feller, W.: An Introduction to Probability Theory and Its Applicatons, vol. 2. John Wiley & Sons (1971)
16. Guo, J., Jean, J., Nikolic, I., Qiao, K., Sasaki, Y., Sim, S.M.: Invariant subspace attack against Midori64 and the resistance criteria for S-box designs. IACR Trans.

Symm. Cryptol. **2016**(1), 33–56 (2016). https://doi.org/10.13154/tosc.v2016.i1.33-56, http://tosc.iacr.org/index.php/ToSC/article/view/534
17. Knudsen, L.R., Wagner, D.: Integral cryptanalysis. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 112–127. Springer, Heidelberg, Germany, Leuven, Belgium (Feb 4–6, 2002)
18. Leander, G., Abdelraheem, M.A., AlKhzaimi, H., Zenner, E.: A cryptanalysis of PRINTcipher: The invariant subspace attack. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 206–221. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2011)
19. Lin, L., Wu, W.: Meet-in-the-middle attacks on reduced-round Midori64. IACR Trans. Symm. Cryptol. **2017**(1), 215–239 (2017). https://doi.org/10.13154/tosc.v2017.i1.215-239
20. Luykx, A., Mennink, B., Paterson, K.G.: Analyzing multi-key security degradation. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017, Part II. LNCS, vol. 10625, pp. 575–605. Springer, Heidelberg, Germany, Hong Kong, China (Dec 3–7, 2017)
21. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT'93. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg, Germany, Lofthus, Norway (May 23–27, 1994)
22. Nyberg, K.: Linear approximation of block ciphers (rump session). In: Santis, A.D. (ed.) EUROCRYPT'94. LNCS, vol. 950, pp. 439–444. Springer, Heidelberg, Germany, Perugia, Italy (May 9–12, 1995)
23. Todo, Y., Leander, G., Sasaki, Y.: Nonlinear invariant attack - practical attack on full SCREAM, iSCREAM, and Midori64. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016, Part II. LNCS, vol. 10032, pp. 3–33. Springer, Heidelberg, Germany, Hanoi, Vietnam (Dec 4–8, 2016). https://doi.org/10.1007/978-3-662-53890-6_1
24. Zhan, C., Xiaoyun, W.: Impossible differential cryptanalysis of Midori. Cryptology ePrint Archive, Report 2016/535 (2016), http://eprint.iacr.org/2016/535

A List of Invariants Produced by Algorithm 1

Table 5. Invariants for two rounds of (modified) Midori-64, as obtained using Algorithm 1. Only invariants with at least 2^{64} weak keys are listed. Note that these invariants are not valid for all choices of the round constants. The label "type I" refers to invariants with $u = v$, whereas "type II" indicates that $u \neq v$. Note that not all of these invariants are linearly independent.

Correlation vector (v for $v^{\otimes 16}$)	Amount of weak-keys	Type
$(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)^\top$	2^{128}	Trivial
$(0,0,0,0,0,0,0,0,0,0,1,1,0,0,-1,1)^\top$	2^{96}	Type II
$(0,1,0,0,1,0,0,0,-1,0,0,0,0,1,0,0)^\top$	2^{80}	Type II
$(0,0,0,1,0,0,-1,0,0,0,0,-1,0,0,-1,0)^\top$	2^{80}	Type II
$(1,-1,0,0,0,0,0,0,-1,-1,0,0,0,0,0,0)^\top$	2^{64}	Type II
$(0,0,0,0,0,0,1,1,0,0,0,0,0,0,1,1)^\top$	2^{64}	Type II
$(0,0,0,0,1,1,0,0,0,0,0,0,1,1,0,0)^\top$	2^{64}	Type II
$(0,0,1,1,0,0,0,0,0,0,1,1,0,0,0,0)^\top$	2^{64}	Type II
$(1,1,0,0,0,0,0,0,1,1,0,0,0,0,0,0)^\top$	2^{64}	Type II
$(0,0,0,0,0,0,0,0,0,0,1,-1,0,0,1,1)^\top$	2^{64}	Type I
$(0,0,0,0,0,0,1,-1,0,0,0,0,0,0,1,1)^\top$	2^{64}	Type I
$(0,0,0,1,0,0,0,1,0,0,0,-1,0,0,0,1)^\top$	2^{64}	Type I

B Test Code for Nonlinear Invariant from Section 5.3

The following code was tested using Sage 8.1.

```
import random
from operator import xor
from sage.crypto.sboxes import Midori_Sb0 as Sb0
from sage.crypto.boolean_function import BooleanFunction

def xor3(a, b, c):
    return xor(a, xor(b, c))

def mixColumn(nibbles):
    return [
        xor3(nibbles[1], nibbles[2], nibbles[3]),
        xor3(nibbles[0], nibbles[2], nibbles[3]),
        xor3(nibbles[0], nibbles[1], nibbles[3]),
        xor3(nibbles[0], nibbles[1], nibbles[2])
    ]
```

```
16
17 def subCell(nibbles):
18     for i in range(16):
19         nibbles[i] = Sb0(nibbles[i])
20
21 def addKey(nibbles, key):
22     for i in range(16):
23         nibbles[i] = xor(nibbles[i], key[i])
24
25 RC = [
26     [0,0,0,1,0,1,0,1,1,0,1,1,0,0,1,1], [0,1,1,1,1,0,0,0,1,1,0,0,0,0,0,0],
27     [1,0,1,0,0,1,0,0,0,0,1,1,0,1,0,1], [0,1,1,0,0,0,1,0,0,0,0,1,0,0,1,1],
28     [0,0,0,1,0,0,0,0,0,0,1,0,0,1,1,1], [1,1,0,1,0,0,0,1,0,1,1,1,0,0,0,0],
29     [0,0,0,0,0,0,1,0,0,1,1,0,0,1,1,0], [0,0,0,0,1,0,1,1,1,1,0,0,1,1,0,0],
30     [1,0,0,1,0,1,0,0,1,0,0,0,0,0,0,1], [0,1,0,0,0,0,0,0,1,0,1,1,1,0,0,0],
31     [0,1,1,1,0,0,0,1,1,0,0,1,0,1,1,1], [0,0,1,0,0,0,1,0,1,0,0,0,1,1,1,0],
32     [0,1,0,1,0,0,0,1,0,0,1,1,0,0,0,0], [1,1,1,1,1,0,0,0,1,1,0,0,1,0,1,0],
33     [1,1,0,1,1,1,1,1,1,0,0,1,0,0,0,0]
34 ]
35
36 def addRoundConstants(nibbles, r, b):
37     for i in range(16):
38         nibbles[i] = xor(nibbles[i], RC[r][i] << b)
39
40 ShuffleCell = [0, 10, 5, 15, 14, 4, 11, 1, 9, 3, 12, 6, 7, 13, 2, 8]
41 def shuffleCells(nibbles):
42     result = [0] * 16
43     for i in range(16):
44         result[i] = nibbles[ShuffleCell[i]]
45     return result
46
47 def midori64(nibbles, rounds, key, b = 0):
48     whitening_key = [xor(key[0][i], key[1][i]) for i in range(16)]
49     addKey(nibbles, whitening_key)
50     for i in range(rounds - 1):
51         subCell(nibbles)
52         nibbles = shuffleCells(nibbles)
53         for j in range(4):
54             result = mixColumn(nibbles[4*j:4*j+4])
55             for k in range(4):
56                 nibbles[4*j + k] = result[k]
57         addRoundConstants(nibbles, i, b)
58         addKey(nibbles, key[i % 2])
59     subCell(nibbles)
60     addKey(nibbles, whitening_key)
```

```
61      return nibbles
62
63
64  R.<x0, x1, x2, x3> = BooleanPolynomialRing(4)
65  f = BooleanFunction(x0*x2 + x0 + x1 + x3)
66  g = BooleanFunction(x0 + x2)
67
68  key = [[0] * 16, [0] * 16]
69
70  # Test vector
71  assert midori64([0] * 16, 16, key) == \
72      [3, 12, 9, 12, 12, 14, 13, 10, 2, 11, 11, 13, 4, 4, 9, 10]
73
74  # key = [[15] * 16, [0] * 16] # This also works (see Section 5.3)
75
76  nb_tests = 100
77  b = 1 # Add RC to bit b
78
79  counts = [0, 0]
80  for i in range(nb_tests):
81      input_value = [random.randint(0, 15) for i in range(16)]
82      input_projection = reduce(xor, map(g, input_value))
83      output_value = midori64(input_value, 16, key, b)
84      output_projection = reduce(xor, map(g, output_value))
85      counts[xor(input_projection, output_projection)] += 1
86
87  print("Correlation: ", 2 * counts[1] / sum(counts) - 1)
```