Specifying cycles of minimal length for commonly used linear layers in block ciphers

Guoqiang Deng * Yongzhuang Wei † Xuefeng Duan ‡ Enes Pasalic § Samir Hodžić ¶

Abstract

With the advances of Internet-of-Things (IoT) applications in smart cities and the pervasiveness of network devices with limited resources, lightweight block ciphers have achieved rapid development recently. Due to their relatively simple key schedule, nonlinear invariant attacks have been successfully applied to several families of lightweight block ciphers. This attack relies on the existence of a nonlinear invariant $g : \mathbb{F}_2^n \to \mathbb{F}_2$ for the round function F_k so that $g(x) + g(F_k(x))$ is constant for any input value x. Whereas invariants of the entire S-box layer has been studied in terms of the corresponding cycle structure [TLS16, WRP+20] (assuming the use of bijective S-boxes), a similar analysis for the linear layer has not been performed yet. In this article, we provide a theoretical analysis for specifying the minimal length of cycles for commonly used linear permutations (implementing linear layers) in lightweight block ciphers. Namely, using a suitable matrix representation, we exactly specify the minimal cycle lengths for those (efficiently implemented) linear layers that employ ShiftRows, Rotational-XOR and circular Boolean matrix operations which can be found in many well-known families of block ciphers. These results are practically useful for the purpose of finding nonlinear invariants of the entire encryption rounds since these can be specified using the intersection of cycles corresponding to the linear and S-box layer. We also apply our theoretical analysis practically and specify minimal cycle lengths of linear layers for certain families of block ciphers including some NIST candidates.

Keywords: Cyclic shift, XOR, Cycle of linear layer, Permutation matrix, Nonlinear invariant.

1 Introduction

Block ciphers are important cryptographic primitives whose security has been traditionally evaluated using some standard cryptanalytic techniques such as differential attacks [BS90], linear

*Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin, China, e-mail: d9801242@guet.edu.cn.

†Guilin University of Electronic Technology, Guilin, China, e-mail: walker_wei@msn.com.

‡Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, College of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin, China, e-mail: guidian520@126.com.

§University of Primorska, FAMNIT and IAM, Koper, Slovenia, e-mail: enes.pasalic6@gmail.com.

¶Technical University of Denmark, DTU Compute, Denmark, e-mail: saho@dtu.dk.

attacks [Mat93] , and their diverse variations. Certain requirements towards their efficient implementation in resource constrained environments have however given rise to a specific family of these primitives commonly known as lightweight block ciphers. Due to their reduced implementation cost, which induces greedy design rationales (commonly using a simplified key schedule), lightweight ciphers easily become a target for other cryptanalytic tools such as invariant attacks.

Subspace invariant attacks were introduced in [LAAZ11] (see also [LMR15]) and it was demonstrated that several lightweight block ciphers could be efficiently cryptanalyzed using this novel cryptanalytic approach. This attack basically relies on the property of having inputs and outputs that belong to the same affine subspace through (many) encryption rounds under the so-called *weak key assumption*. Their extension, a nonlinear invariant attack, was proposed by Todo *et al.* [TLS16] at ASIACRYPT 2016 which was successfully applied against the lightweight encryption algorithms SCREAM [GLSV15], iSCREAM [GLSV14] and Midori64 [BBI$^+$15]. The main idea behind nonlinear invariant attacks is to identify a nonlinear Boolean function $g : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2$ for which the evaluation of $g(x) + g(E_k(x))$ is constant for any x, where $E_k(x)$ is the encryption function of a considered n-bit block cipher performed using the secret key k. The function g is then called a nonlinear invariant for $E_k(x)$ and those keys $k \in \mathcal{K}$ for which g is a nonlinear invariant are called *weak keys*.

To extend the application range and possibly find even larger classes of weak keys, Y. Wei *et al.* [WYWP18] introduced the concept of generalized invariant: $g(x + a_1) + g(E(x) + a_2) = c$, where $c \in \mathbb{F}_2$, which enables the effects of round constant to be eliminated by introducing two n-bit vectors a_1, a_2. So far, only a few works have been conducted towards finding invariants of the S-box layer. Traditionally, the S-box layer can be viewed as a parallel application of a certain number of small S-boxes and invariants of the whole S-layer can be specified by concatenating invariants of constituent S-boxes [BCLR17]. Initially, a method of specifying invariants of S-boxes based on its cycle structure was mentioned in [TLS16]. More precisely, Todo *et al.* [TLS16] showed that $\#g = 2^{(\#\text{ cycles of } F)}$ when F (representing a bijective S-box) has at least one cycle of odd length; alternatively $\#g = 2^{(\#\text{ cycles of } F)+1}$ when F only has cycles of even length. Quite recently, an extensive theoretical treatment related to generalized and closed loop invariants of bijective S-boxes appeared in [WRP$^+$20]. It is important to notice that only in certain cases (for instance using quadratic invariant g and binary orthogonal matrix as a linear layer [WYWP18]) an invariant of the entire encryption round can be specified. Therefore, being at the same time a main motivation of this article, there is a necessity to further analyze invariants of commonly used linear layers.

Most notably, apart from the above mentioned property related to orthogonal matrices and the general resistance of linear layers to invariant attacks discussed in [BCLR17], there is no theoretical analysis of invariants of linear layers that can be found in the literature. In particular, the properties of linear layers that employ cyclic shift and XOR operations as the most basic computer instructions, have not been considered so far. These operations are exclusively used in many well-known wireless sensor networks and video recognition systems. The same is true for lightweight authenticated encryption algorithms and hash functions submitted to the Round 2 Candidates of NIST LWC Competition, including SKINNY-AEAD/SKINNY-HASH [BJK$^+$16], Ascon [DEMS19], Knot [ZDY$^+$19], Pyjamask [GJK$^+$19], ForkAE [ALP$^+$19], mixFeed [CN19a], PHOTON-Beetle [BCD$^+$19], COMET [GJN19], etc.

The main objective of this article is to provide a rigour theoretical analysis of nonlinear invariants in terms of the *minimal cycle length* of those linear layers that employ most basic operations (suitable for lightweight design rationales for increasing the speed and reduction of the implementation cost). The notion of the *minimal cycle length* of a linear layer refers here to the least common multiple of all the cycles corresponding to different elements that suitably represent the entire linear layer. Our main contributions are given in the next section where we additionally, for clarity, emphasize the main motivation of this work through a theoretical result which provides the means of combining cycle structures of linear and nonlinear layer for the purpose of specifying invariants of the whole encryption round.

1.1 Motivation and contributions

The structure of a round function R of an SPN block cipher $E_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is given as a composition of an S-box layer $\mathcal{S}$ and a linear layer $\mathcal{L}$, i.e. $R(x) = (\mathcal{L} \circ \mathcal{S})(x)$ (for simplicity the notation for round keys is omitted). There exist various design approaches which utilize this structure, and in general, not all of them allow the application of the non-linear invariant attack in the way as it was presented in [TLS16].

The so-called LS-design of a block cipher, which is based on the bitslice construction of R, has been recently cryptanalyzed in [TLS16]. It has been shown that a quadratic invariant of $\mathcal{S}$ is actually an invariant for R, if $\mathcal{L}$ is an orthogonal matrix. This is mainly due to the fact that $\mathcal{L}$ is applied in a bitslice manner. On the other hand, the PRESENT-like round function does not employ a bitslice design, and thus it does NOT allow the application of the non-linear attack presented as in [TLS16] (nor its generalized version [WYWP18]). It turns out that the application of nonlinear invariant attacks to PRESENT-like round functions is a quite challenging task if one wants to find an invariant of a round function that holds with probability equal to 1.

On the other hand, there exist many lightweight block ciphers whose round functions have low degree (even quadratic), but still it is not clear how the non-linear invariant attack can be applied due to their structure. As low-degree of a round function seems to be a potential weakness against this attack, the main problem that needs to be to resolved in this context is an efficient specification of invariants of the encryption round.

A potential approach that one may consider is to find all cycles of $\mathcal{S}$ and $\mathcal{L}$, and if they contain a common cycle (say C), then it is a cycle of the mapping $R = \mathcal{L} \circ \mathcal{S}$. As pointed out in [TLS16], one then easily constructs a non-linear invariant of R (say a Boolean function g) such that g is constant on the cycle C, and has complementary value outside of C (the cycle C is a subset of $\mathbb{F}_2^n$ of cardinality less than 2^n). This observation, based on the ideas in [TLS16], is generalized with the following proposition.

Proposition 1 *Let $\mathcal{S}, \mathcal{L} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be arbitrary bijective mappings. Suppose that $\mathfrak{S} = \{C_1^{\mathcal{S}}, \ldots, C_t^{\mathcal{S}}\}$ and $\mathfrak{L} = \{C_1^{\mathcal{L}}, \ldots, C_r^{\mathcal{L}}\}$ be the sets of all cycles of $\mathcal{S}$ and $\mathcal{L}$ $(t, r \geq 1)$, where for instance $C_j^{\mathcal{S}} = \{\mathcal{S}^k(x^{(j)}) : k \in \mathbb{N}\}$, for some $x^{(j)} \in \mathbb{F}_2^n$ $(j = 1, \ldots, t)$. If $\mathcal{S}$ and $\mathcal{L}$ have common cycles, that is $\mathfrak{S} \cap \mathfrak{L} = \{D_1, \ldots, D_p\}$ (for some $p \leq min\{t, r\}$), then for a Boolean function $g : \mathbb{F}_2^n \to \mathbb{F}_2$ defined as*

$$g(x) = \begin{cases} c_i, & x \in D_i \in \mathfrak{S} \cap \mathfrak{L}, \ \ i = 1, \ldots, p, \\ d, & x \notin \bigcup_{z=1}^{p} D_z, \end{cases},$$

for some fixed values $c_i, d \in \mathbb{F}_2$, it holds that $g(\mathcal{L}(\mathcal{S}(x))) = g(x)$ for all $x \in \mathbb{F}_2^n$, i.e. g is an invariant function of the mapping $\mathcal{L} \circ \mathcal{S}$.

Proof. Note that every cycle $C_j^{\mathcal{S}} = \{\mathcal{S}^k(x^{(j)}) : k \in \mathbb{N}\}$ $(j = 1, \ldots, t)$ also contains the vector $x^{(j)}$. For instance, if we assume that $\mathcal{S}(x^{(j)}) = \mathcal{S}^{\lambda}(x^{(j)}) \in C_j^{\mathcal{S}}$ for some $\lambda > 1$, then by applying $\mathcal{S}^{-1}$ to both sides one obtains $x^{(j)} = \mathcal{S}^{\lambda-1}(x^{(j)}) \in C_j^{\mathcal{S}}$, and thus $x^{(j)} \in C_j^{\mathcal{S}}$. On the other hand, recall that cycles of a mapping are pairwise disjoint subsets of $\mathbb{F}_2^n$, i.e. $C_i^{\mathcal{S}} \cap C_j^{\mathcal{S}} = \emptyset$ for $i \neq j$ and $\bigcup_{j=1}^{t} C_j^{\mathcal{S}} = \mathbb{F}_2^n$ (similarly for $\mathcal{L}$). Thus, the statement follows from the previous facts. ∎

Usually, the design of the S-box layer $\mathcal{S} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ at input $x = (\overline{x}_1, \ldots, \overline{x}_s) \in \mathbb{F}_2^m \times \ldots \times \mathbb{F}_2^m$ $(sm = n)$, is given as

$$\mathcal{S}(\overline{x}_1, \ldots, \overline{x}_s) = (S(\overline{x}_1), \ldots, S(\overline{x}_s)),$$

where an S-box $S : \mathbb{F}_2^m \to \mathbb{F}_2^m$ is a bijective non-linear mapping (in many cases $m \in \{4, 6, 8\}$). As the mapping S is defined on a low-dimensional domain, then it is easy to find its complete cyclic structure. Consequently, the complete cyclic structure of $\mathcal{S}$ can be found.

However, the cycle structure of a linear layer $\mathcal{L}$ (for which there exist various design approaches) is an open problem that we are addressing in this work. More precisely, we analyze several most common types of linear layers used in certain well-known block ciphers, and derive efficient algorithms for finding the minimal cycle length (see Definition 3) of underlying linear layers. The application of our results is clearly in the framework of Proposition 1 (which essentially establishes the motivation for this work), which we leave for further investigation due to its potential utilization in applying non-linear invariant attacks to PRESENT-like encryption schemes.

The problem of computing the cycle structure of linear layers that use the cyclic shift and XOR operations will be specifically considered with respect to the following three families of basic operations. We consider ShiftRows, Rotational-XOR and Cir-Boolean matrix operations which are denoted by $SR(x)$, $l - RX(x)$ and $CBM(x)$ (see Table 1 for their precise definition), respectively. The main contributions of this paper can be summarized as follows:

- We view the cyclic shift and XOR operation as a certain action of a suitably chosen permutation matrix P, which allows us to derive the connections between the $SR(x)$ and $l - RX(x)$ operation in terms of P.
- For the $SR(x)$ operation we derive the exact length of a minimal cycle, which is then applied and practically confirmed in the case of TANGRAM-128[ZJD+19].
- In the case of the $l - RX(x)$ operation, we show that when l is odd then there exists the minimum positive cycle for the $l - RX(x)$ operation and an explicit formula for its computation is derived. These results are then turned into an efficient algorithm for obtaining the cycle of minimum length for the $l - RX(x)$ operation, cf. Algorithm 2. On the other hand, when l is even, $l - RX(x)$ operation is never a permutation and therefore unsuitable for implementing a linear layer.
- We show that $CBM(x)$ and $l - RX(x)$ operation are equivalent (can be identified using suitable different forms) which is also illustrated in the case of Pyjamask-128[GJK+19], cf. Proposition 2.

- Using our algorithms and explicit formulas, we provide an extensive analysis of linear layers of many well-known lightweight block ciphers in terms of finding a cycle of minimum length.

1.2 Organization

The rest of this paper is organized as follows. Some useful notations are introduced in Section 2 and certain properties of circulant permutation matrices are given. In Section 3, we analyze the cycle structure of linear layers of ShiftRows, Rotational-XOR and Cir-Boolean matrix type, respectively. A detailed and rigorous theoretical analysis is provided and efficient algorithms for finding cycles of minimal length are given. In section 4, we apply our theoretical study and algorithms to a large collection of lightweight block ciphers and identify their cycles of minimal length. Some concluding remarks are given in Section 5.

2 Preliminaries

In this section we firstly establish more closely the motivation for this work (Section 1.1). Then, in Section 2.1 we recall the notion of permutation matrices, as well as their representation in terms of circulant Boolean matrices. We also point out a relation between two widely used operations (in the design of blocker ciphers), namely ShiftRows and Rotational-XOR. These operations, which will be considered in matrix representations, are respectively denoted by $SR(x)$ and $l-RX(x)$ when evaluated at input vector x. In general, the notation used throughout the paper is summarized in Table 1.

Table 1: Notations

Symbol	Meaning
x	n-bit binary row vector
O	Zero vector or zero matrix from the context
E	Identity matrix
M	Circulant Boolean matrix
P	Permutation matrix defined by (1)
$\lvert S\rvert$	Cardinality of set S
$[a,b]$	Least common multiple of a and b
(a,b)	Greatest common divisor of a and b
$a \mid b$	a divides b
$a \nmid b$	a does not divide b
$x <<< i$	i-bit left cyclic shift operation on x
$x >>> i$	i-bit right cyclic shift operation on x
$SR(x)$	$x \leftarrow (x <<< i)$
$l-RX(x)$	$x \leftarrow \Sigma(x) = (x <<< i_1) \oplus (x <<< i_2) \oplus \cdots \oplus (x <<< i_l)$
$CBM(x)$	$x^T \leftarrow Mx^T$
$SR^j(x)$	$SR(x)$ operation used iteratively j times
$l-RX^j(x)$	$l-RX(x)$ operation used iteratively j times

2.1 Properties of permutation matrix

In order to analyze the problem of finding the minimal cycle length of a linear layer, it is convenient to express the main parameters using matrix notation. We first recall the definitions of a fixed point of a mapping and circulant Boolean matrix.

Definition 1 *Let $\mathcal{F} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be a linear mapping. If there exists a vector $x_0 \in \mathbb{F}_2^n$ such that $\mathcal{F}(x_0) = x_0$ holds, then x_0 is called a fixed point of $\mathcal{F}$.*

Definition 2 *A Boolean matrix M (of size $n \times n$) is said to be circulant if it is given as*

$$M = \begin{pmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \\ a_{n-1} & a_0 & a_1 & \cdots & a_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_2 & a_3 & a_4 & \cdots & a_1 \\ a_1 & a_2 & a_3 & \cdots & a_0 \end{pmatrix},$$

where the entries $a_i \in \mathbb{F}_2^n$, $0 \le i \le n-1$. The matrix M is shortly denoted by

$$M = cir(\begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix}).$$

When $a_1 = 1$ and $a_j = 0$ for $j \neq 1$, one obtains a special circulant Boolean matrix given by

$$P = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{1}$$

Moreover, the i-th power of the matrix P (denoted by P^i) is given as

$$P^i = \begin{pmatrix} 0 & 0 & \cdots & 0 & \overset{i}{\overset{\downarrow}{1}} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \leftarrow n-i$$

It turns out that the structure of P can be easily related to the cyclic shift and XOR operations used in linear layers of block ciphers. This connection is based on the following facts:

1) If P is of size $n \times n$, then the set $G = \{P^0 = E, P^1, P^2, \cdots, P^{n-1}\}$ forms a cyclic group

(with respect to matrix multiplication evaluated modulo 2). The order of the cyclic group G is n and thus $P^n = E$ (cf. [WS08]).

2) An arbitrary $n \times n$ circulant Boolean matrix M can be written as a linear combination of $P^0, P^1, \cdots, P^{n-1}$ with weights $c_0, c_1, \ldots, c_{n-1}$ [Sch74] , i.e. we have that

$$M = c_0 P^0 \oplus c_1 P^1 \oplus \cdots \oplus c_{n-2} P^{n-2} \oplus c_{n-1} P^{n-1}, \tag{2}$$

where $c_i \in \mathbb{F}_2$, $0 \leq i \leq n-1$, and vice versa. In other words, the space of circulant Boolean matrices is spanned by the set $\{P^0, P^1, \cdots, P^{n-2}, P^{n-1}\}$, where the linear combinations are taken over $\mathbb{F}_2$. Clearly, by omitting coefficients c_i which are equal to 0, one can write M in the simplified form as

$$M = P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l}, \quad 0 \leq i_1 < i_2 < \cdots < i_l < n.$$

3) Let $x = (x_1, \ldots, x_n)$ be a binary vector. The cyclic shift of its coordinates to the left for i positions is denoted by $x \lll i$, i.e. we have the operation

$$SR(x) : x \leftarrow (x_{i+1}, \ldots, x_n, x_1, \ldots, x_i) = (x \lll i).$$

This operation can also be expressed in terms of the matrices P^i as

$$SR(x) : x^T \leftarrow P^i x^T,$$

where x^T is the transpose of x, which is viewed as a row matrix.

4) For a binary vector x of length n, the combination of cyclic shifts and XOR operations forms the operation

$$l - RX(x) : x \leftarrow \Sigma(x) = (x \lll i_1) \oplus (x \lll i_2) \oplus \cdots \oplus (x \lll i_l),$$

where $0 \leq i_1 < i_2 < \cdots < n$. Similarly as in the case of the cyclic shift $SR(x)$, we have that $l - RX(x)$ can be expressed in terms of the matrices $P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l}$ as

$$l - RX(x) : x^T \leftarrow (P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l}) x^T.$$

3 The linear layer based on cyclic shift and XOR operations

In this section, we analyze the cyclic structure of linear layers, in terms of identifying the minimal cycle length specified as follows.

Definition 3 *Let $\mathcal{M} : \mathbb{F}_2^n \to \mathbb{F}_2^n$ be an arbitrary bijective mapping. The minimal positive integer $\mu \geq 1$ for which $\mathcal{M}^\mu(x) = x$ holds for all $x \in \mathbb{F}_2^n$, will be called the minimal cycle length of the mapping $\mathcal{M}$. In other words, the integer μ refers to the least common multiple of all possible cycle lengths of all different elements $x \in \mathbb{F}_2^n$ with respect to $\mathcal{M}$.*

In this context, we consider those linear layers that employ ShiftRows and Rotational-XOR operations (Sections 3.1 and 3.2), as well as those implementing circulant Boolean matrices (Section 3.3). In addition, we provide efficient algorithms for finding cycles of minimal length of these linear layers.

3.1 The ShiftRows type based on $SR(x)$ operation

In this section we analyze the ShiftRows-type linear layers which are widely used in the design of lightweight block ciphers. For instance, in the second round of the competition for lightweight block ciphers initiated by NIST, the schemes Knot [ZDY+19] and ORANGE [CN19b] utilize the left cyclic shift operation, whereas Romulus [IKMP19], ForkAE v.1 [ALP+19] and SKINNY-AEAD/SKINNY-HASH [BJK+19] use the right cyclic shift operation.

Commonly, the operations performed in an encryption round act on the state matrix, where the data being processed is suitably arranged in a matrix form. We assume that the linear layer acts as a transformation applied to each row separately. In this context, by a ShiftRows-type operation we will refer to a linear transformation performing (different) rotational shift operations on each row of the state matrix.

We firstly focus on the problem of finding cycles of minimal length for linear layers which utilize a single shift operation. This result then easily extends to linear layers that employ different shift operations for each row of the state matrix by finding the least common multiple of these minimal cycles. Notice that shifting i bits to the left cyclically is equivalent to shifting $(n-i)$ bits to the right, and therefore without loss of generality, we will only consider left cyclic shifts. We start with the following technical result.

Theorem 1 *Let $x \in \mathbb{F}_2^n$. If x is shifted cyclically i bits to the left, that is $SR(x) : x \leftarrow (x \lll i)$, then the minimal cycle length of the SR operation (in terms of Definition 3) is given by*

$$\mu = \frac{n}{(i,n)} = \frac{n}{\gcd(i,n)}. \tag{3}$$

Proof. Recall that the operation $SR(x)$ can be described in terms of the permutation P^i as

$$SR(x) : x^T \leftarrow P^i x^T.$$

In order to find the cycle of the $SR(x)$, we need to compute the minimum positive integer μ such that $\mathrm{SR}^\mu(x) = x$, which is equivalent to

$$\left(P^i\right)^\mu x^T = x^T, \ \ \forall x \in \mathbb{F}_2^n.$$

Here, we have that $\left(P^i\right)^\mu = E$, and thus $P^n = E$ implies that $n \mid i\mu$. As it is easy to verify $\frac{n}{(i,n)} \mid \frac{i\mu}{(i,n)}$, the greatest common divisor of $\frac{n}{(i,n)}$ and $\frac{i}{(i,n)}$ is equal to 1, and consequently $\frac{n}{(i,n)} \mid \mu$.

Hence, let $\mu = k \cdot \frac{n}{(i,n)}$, for $k \in \mathbb{N}$. Since we have that $\left(P^i\right)^\mu = \left(P^i\right)^{\frac{kn}{(i,n)}} = \left(P^n\right)^{\frac{ki}{(i,n)}} = (E)^{\frac{ki}{(i,n)}} = E$, and $\left(P^i\right)^\mu = \left(P^i\right)^{\frac{n}{(i,n)}} = E$ for $k = 1$, we conclude that the minimum cycle length of the $SR(x)$ operation is $\mu = \frac{n}{(i,n)}$, which corresponds to the case when $k = 1$. ∎

Remark 1 *If different cyclic shifts are applied to each row of the state matrix, then the minimal cycle length of $\mathcal{L}$ is the least common multiple of minimal cycle length which correspond to each row (determined by Theorem 1).*

To illustrate a concrete application of the above remark, we consider the problem of finding the minimal cycle of the linear layer of TANGRAM-128 [ZJD+19]. The 128-bit state of this cipher is represented as a matrix of size 4×32. The linear layer of TANGRAM-128 applies four different cyclic left shift operations to each row, as shown in Figure 1 below.

$$
\begin{aligned}
&(\; a_{0,31} \quad \cdots \quad a_{0,1} \quad a_{0,0} \;) \xrightarrow{\lll 0} (\; a_{0,31} \quad \cdots \quad a_{0,1} \quad a_{0,0} \;) \\
&(\; a_{1,31} \quad \cdots \quad a_{1,1} \quad a_{1,0} \;) \xrightarrow{\lll 1} (\; a_{1,30} \quad \cdots \quad a_{1,0} \quad a_{1,31} \;) \\
&(\; a_{2,31} \quad \cdots \quad a_{2,1} \quad a_{2,0} \;) \xrightarrow{\lll 8} (\; a_{2,23} \quad \cdots \quad a_{2,25} \quad a_{2,24} \;) \\
&(\; a_{3,31} \quad \cdots \quad a_{3,1} \quad a_{3,0} \;) \xrightarrow{\lll 11} (\; a_{3,20} \quad \cdots \quad a_{3,22} \quad a_{3,21} \;)
\end{aligned}
$$

Figure 1. The linear layer of TANGRAM-128.

Applying Theorem 1, we can easily deduce the minimal cycle lengths for each row of the state matrix. Using (3), we have that the minimal cycle length for the last three rows are 32, 4 and 32, respectively. Apparently, the least common multiple of these lengths, being 32, determines the minimal cycle length for SR operation employed in TANGRAM-128.

3.2 The Rotational-XOR type based on $l - RX(x)$ operation

In this section, we analyze the Rotational-XOR type linear layers, that is the case when a linear layer employs the combination of cyclic shifts and XOR operations (i.e., $l - RX(x)$ operation). These linear layers (among different useful properties) provide a great flexibility in the round function implementations [Guo17]. For instance, the linear layer of SM4 [SM412] scheme (GM/T 0002-2012) employs the Rotational-XOR operations. A similar structure can also be found in the algorithms Ascon [DEMS19] (entering the second round of lightweight block ciphers competition of NIST), FBC [FZZ+19](the second round of lightweight block ciphers competition of China), DBlock [WZY15] and RoadRunneR [BS15].

Recall that the operation $l - RX(x)$ is given by

$$l - RX(x) : x \leftarrow \Sigma(x) = (x \lll i_1) \oplus (x \lll i_2) \oplus \cdots \oplus (x \lll i_l).$$

In terms of the matrix representation, $l - RX(x)$ can be written as $l - RX(x) = Qx^T$, where the matrix Q is given by $Q = P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l}$.

In order to find a minimum positive integer $\mu \geq 1$ such that $l - RX^{\mu}(x) = x$ holds for all $x \in \mathbb{F}_2^n$, or equivalently

$$Q^{\mu}x^T = (P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l})^{\mu}x^T = x^T \Leftrightarrow Q^{\mu} = (P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l})^{\mu} = E,$$

we need to consider the equality $Q^{\mu} = E$. As Q is a sum of matrices P^{i_j}, the following lemma helps us to compute efficiently Q^{μ} in terms of powers of P^{i_j}.

Remark 2 *Notice that Lemma 1 only considers the case $n = 2^m$ which is however the most common case in practice. The analysis of a general case when n is an arbitrary positive integer is more complicated and does not provide a useful simplification compared to the case $n = 2^m$.*

Lemma 1 *Let P be an $n \times n$ binary permutation matrix. If n is of the form $n = 2^m$, then*

$$(P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l})^n = (P^{i_1})^n \oplus (P^{i_2})^n \oplus \cdots \oplus (P^{i_l})^n, \quad 0 \leq i_1 < i_2 < \cdots < n.$$

Proof. Since $2^m = n$, one has to prove that

$$(P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l})^{2^m} = (P^{i_1})^{2^m} \oplus (P^{i_2})^{2^m} \oplus \cdots \oplus (P^{i_l})^{2^m}.$$

We notice that all the cross terms in the expansion of $(P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l})^{2^m}$ have coefficients which are even positive numbers and the reduction modulo 2 gives the claim. ∎

In what follows, we utilize Lemma 1 in order to analyze the operation $l - RX(x)$ with respect to the parity of l.

3.2.1 The operation $l - RX(x)$ with l being odd

In practice, when l is odd, most of the block ciphers with a linear layer of the Rotational-XOR type apply the $3 - RX(x)$ operation on rows of the state matrix. Therefore, we first consider this specific case and then give a generalization for arbitrary odd l. The following result provides an explicit formula to compute the length of minimal cycle for the $3 - RX(x)$ operation.

Theorem 2 *Let x be a binary vector of length n. Suppose that the left cyclic shift is performed three times on x followed by the XOR operation on these shifts, i.e. let us consider the operation*

$$3 - RX(x) : x \leftarrow \Sigma(x) = (x \lll i_1) \oplus (x \lll i_2) \oplus (x \lll i_3), \quad 0 \leq i_1 < i_2 < i_3 < n.$$

If n is of the form $n = 2^k$, then:

i) It holds that $3 - RX^n(x) = x$, for any $x \in \mathbb{F}_2^n$.

ii) There exists a minimum cycle length μ of $3 - RX(x)$ such that $\mu \leq n$ and $\mu | n$. Moreover, μ is given by

$$\mu = \min\{\big[\frac{n}{(i_3 - i_1, n)}, \frac{n}{(i_2, n)}\big], \big[\frac{n}{(i_3 - i_2, n)}, \frac{n}{(i_1, n)}\big], \big[\frac{n}{(i_2 - i_1, n)}, \frac{n}{(i_3, n)}\big]\}. \tag{4}$$

Proof. *i)* Since $n = 2^k$, Lemma 1 implies that

$$Q^n = (P^{i_1} \oplus P^{i_2} \oplus P^{i_3})^n = P^{n \cdot i_1} \oplus P^{n \cdot i_2} \oplus P^{n \cdot i_3}.$$

Recall that for P it holds that $P^n = E$, and thus

$$Q^n = (P^n)^{i_1} \oplus (P^n)^{i_2} \oplus (P^n)^{i_3} = E^{i_1} \oplus E^{i_2} \oplus E^{i_3} = E.$$

We notice that n is not necessarily the minimal cycle length of Q.

ii) Denoting by μ the minimal cycle length of Q, we first prove that $\mu | n$ by contradiction. Assume that $\mu \nmid n$, so that there is a positive integer t such that $n = t\mu + m$, where $m < \mu$. Since $Q^n = Q^{t\mu + m} = Q^{t\mu} Q^m = E$ and $Q^\mu = E$, we conclude that $Q^m = E$. However, $m < \mu$

contradicts the fact that μ is the minimal cycle length of Q. Therefore, $\mu | n$ and consequently $1 \leq \mu \leq n$.

To show (4), we observe that the equality $Q^{\mu} = P^{\mu \cdot i_1} \oplus P^{\mu \cdot i_2} \oplus P^{\mu \cdot i_3} = E$ implies that at least two matrices among $P^{\mu \cdot i_1}, P^{\mu \cdot i_2}, P^{\mu \cdot i_3}$ have to be equal, and the third one has to be equal to E. This is due to the fact that the identity matrix has values 1 placed on the main diagonal and that Q^{μ} is a sum of odd number of matrices. Without loss of generality, we consider the case $P^{\mu \cdot i_1} \oplus P^{\mu \cdot i_3} = O$ and $P^{\mu \cdot i_2} = E$. Then, the equality $P^{\mu \cdot i_1} \oplus P^{\mu \cdot i_3} = O$ means that $\mu \cdot i_1 \equiv \mu \cdot i_3 \pmod{n}$, and thus we have $n \,|\mu(i_3 - i_1)$. It is obvious that

$$\frac{n}{(i_3 - i_1, n)} | \frac{\mu(i_3 - i_1)}{(i_3 - i_1, n)}.$$

From the fact that $(\frac{n}{(i_3 - i_1, n)}, \frac{(i_3 - i_1)}{(i_3 - i_1, n)}) = 1$, we conclude that $\frac{n}{(i_3 - i_1, n)} | \mu$. Hence the minimal value of μ is equal to $\frac{n}{(i_3 - i_1, n)}$. On the other hand, $P^{\mu \cdot i_2} = E$ implies that $n \,|\mu i_2$, and by Theorem 1 the minimal value of μ is $\frac{n}{(i_2, n)}$. As we require that $P^{\mu \cdot i_1} \oplus P^{\mu \cdot i_3} = O$ and $P^{\mu \cdot i_2} = E$ hold simultaneously, the minimum positive cycle μ is consequently given by $\mu = [\frac{n}{(i_3 - i_1, n)}, \frac{n}{(i_2, n)}]$.

Similarly, assuming $P^{\mu \cdot i_2} \oplus P^{\mu \cdot i_3} = O$ and $P^{\mu \cdot i_1} = E$ gives $\mu = [\frac{n}{(i_3 - i_2, n)}, \frac{n}{(i_1, n)}]$, whereas in the case $P^{\mu \cdot i_1} \oplus P^{\mu \cdot i_2} = O$ and $P^{\mu \cdot i_3} = E$ we have $\mu = [\frac{n}{(i_2 - i_1, n)}, \frac{n}{(i_3, n)}]$.

By summarizing these three cases, the minimal cycle length of Q is given by (4), which completes the proof. ∎

The following example illustrates the calculation of the minimal cycle length of the linear layer of Ascon [DEMS19] specified in Figure 2.

$$x_i \leftarrow \textstyle\sum_i (x_i), \quad 0 \leq i \leq 4.$$

$$\begin{aligned}
x_0 &\leftarrow \textstyle\sum_0 (x_0) = x_0 \oplus (x_0 >>> 19) \oplus (x_0 >>> 28) \\
x_1 &\leftarrow \textstyle\sum_1 (x_1) = x_1 \oplus (x_1 >>> 61) \oplus (x_1 >>> 39) \\
x_2 &\leftarrow \textstyle\sum_2 (x_2) = x_2 \oplus (x_2 >>> 1) \oplus (x_2 >>> 6) \\
x_3 &\leftarrow \textstyle\sum_3 (x_3) = x_3 \oplus (x_3 >>> 10) \oplus (x_3 >>> 17) \\
x_4 &\leftarrow \textstyle\sum_4 (x_4) = x_4 \oplus (x_4 >>> 7) \oplus (x_4 >>> 41)
\end{aligned}$$

Figure 2 Ascons linear layer with 64-bit functions $\sum_i (x_i)$.

Example 1 *The state of Ascon cipher consists of five state rows, where each row is of the size 64 bits. The linear layer is defined by applying $3 - RX(x)$ operation to each row, see Figure 2. To obtain the minimal cycle length of this linear layer, we need to compute the cycle of $3 - RX(x)$ operation for each row separately. Finally, the least common multiple of the obtained minimal cycle lengths (corresponding to each row of the state) is the minimal cycle length of the linear layer.*

Let us consider the first row of the state, which is given by

$$x_0 \leftarrow \Sigma_0(x_0) = x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28).$$

Since Theorem 2 assumes the left cyclic shift, the previous operation can be transformed to

$$x_0 \leftarrow \Sigma(x_0) = x_0 \oplus (x_0 \lll 45) \oplus (x_0 \lll 36).$$

Following the notation of Theorem 2, we have $i_1 = 0$, $i_2 = 36$, $i_3 = 45$. *By* (4), *we get that the minimal cycle length of this operation is 64. Similarly, the minimal cycle lengths with respect to the last four rows are* 32, 64, 64 *and* 32, *respectively. The least common multiple of these lengths is* 64, *which is the minimal cycle length of the linear layer of Ascon.*

The above example indicates that the linear layer of Ascon possess a sufficient robustness in terms of its minimal cycle length which equals to n in this case. Nevertheless, the following hypothetical example demonstrates that a proper choice of the shift values i_1, i_2, i_3 is crucial in this context.

Example 2 *Let the* $3 - RX(x)$ *operation be performed on a* 16*-bit vector* x *as*

$$x \leftarrow \Sigma(x) = (x \lll 1) \oplus (x \lll 4) \oplus (x \lll 7).$$

By Theorem 2, we know that a minimal cycle length μ *divides* 16 *(and also* $\mu < 16$*). In fact, by (4) we have that*

$$\begin{aligned} \mu = & \min\{\big[\frac{16}{(7-1,16)}, \frac{16}{(4,16)}\big], \big[\frac{16}{(7-4,16)}, \frac{16}{(1,16)}\big], \big[\frac{16}{(4-1,16)}, \frac{16}{(7,16)}\big]\} \\ = & \min\{[8,4],[16,16],[16,16]\} = \min\{8,16,16\} = 8. \end{aligned}$$

We notice that the proofs of Theorem 2-(i) and the first part of Theorem 2-(ii) do not depend on the fixed value $l = 3$, which gives us the following corollary.

Corollary 1 *Let* x *be a binary vector of length* n, *and let the operation* $l - RX(x)$, *for odd* l, *be given as*

$$l - RX(x) : x \leftarrow \Sigma(x) = (x \lll i_1) \oplus (x \lll i_2) \oplus \ldots \oplus (x \lll i_l), 0 \le i_1 < i_2 < \cdots i_l < n.$$

If n *is of the form* $n = 2^k$, *then for the minimal cycle length* μ *of* $l - RX(x)$ *it holds that* $\mu \le n$ *and* $\mu \,|n$.

Remark 3 *In the case when* $l > 5$, *the formula for the minimal cycle length of* $l - RX(x)$ *becomes more complicated since there are many different cases to be considered when finding the minimum of different terms. In general, we note that* $\mu \,|n$ *and* $\mu \le n$ *imply that* μ *has to be a power of* 2, *in the case when* $n = 2^k$.

In the context of Remark 3, we further elaborate the computation of a minimal cycle length μ in the case when $n = 2^k$. Hence, when the $l - RX(x)$ operation is performed on an n-bit vector x ($n = 2^k$), then the only candidate values of μ are $2^k, 2^{k-1}, \ldots, 2^1$. An efficient approach to find μ among these values is to substitute $2^{k-1}, 2^{k-2}, \ldots, 2^1$ into the equation

$$(P^{i_1})^\mu \oplus (P^{i_2})^\mu \oplus \cdots \oplus (P^{i_l})^\mu = E. \tag{5}$$

Algorithm 1: Finding a minimal cycle length of $l - RX(x)$ (l is odd) operating on $x \in \mathbb{F}_2^n$.

Input: $i_1, i_2, \ldots, i_l$, n, k $(2^k = n)$.
Output: A minimal cycle length μ of $l - RX(x)$.

1 Let $\mu = 2^{k-1}$.
2 For $j = 1, 2, \cdots, l$: let $m_j = \mu i_j \pmod n$ and $v_j = (\mu/2) i_j \pmod n$.
3 If $P^{m_1} \oplus P^{m_2} \oplus \cdots \oplus P^{m_l} = E$ and $P^{v_1} \oplus P^{v_2} \oplus \cdots \oplus P^{v_l} \neq E$, Output μ and terminate.
4 Let $\mu = \mu/2$. Go to Step 2.

Recall that we are looking for a minimal value of $\mu \in \{2^{k-1}, 2^{k-2}, \ldots, 2^1\}$ for which (5) holds.

For this purpose, one can use Algorithm 1 to compute the minimal cycle length of $l - RX(x)$ (l odd) to a single row of the state matrix. When Algorithm 1 is applied to each row separately, then the least common multiple of obtained minimal cycle lengths (of all rows) is a minimal cycle length of the whole linear layer. By observing the relation (5), we notice the following facts:

1) The exponents $i_1\mu, i_2\mu, \cdots, i_l\mu$ should be evaluated modulo n, due to the fact that $P^n = E$.

2) In relation (5), one actually does not need to verify the minimality of $\mu = 2^i$ for all $i = 1, 2, \ldots, k-1$. Namely, if there exists an integer $t_0 \in \{1, 2, \ldots, k-1\}$ such that $(P^{i_1})^{2^{t_0+1}} \oplus \cdots \oplus (P^{i_l})^{2^{t_0+1}} = E$ and $(P^{i_1})^{2^{t_0}} \oplus \cdots \oplus (P^{i_l})^{2^{t_0}} \neq E$, then 2^{t_0+1} is the minimum cycle length of $l - RX(x)$.

3) When checking the validity of (5), one does not need to compute the matrices $(P^{i_j})^{\mu}$. Instead, one can simply utilize the XOR operation and the fact that for any powers $i_j\mu$ and $i_k\mu$, we have that $(P^{i_j})^{\mu} \oplus (P^{i_k})^{\mu} = O$ holds if and only if $i_j\mu \equiv i_k\mu \pmod n$.

With respect to the previous facts, we also give Algorithm 2 as an efficient improvement of Algorithm 1.

Remark 4 *The case l being even is given in the Appendix. Due to the fact that $l - RX(x)$ is not a permutation for even l makes its use inappropriate in the implementation of linear layers which are supposedly bijective mappings. It can be also shown that in this case there are no cycles of minimal length.*

3.3 The Cir-Boolean matrix type based on $CBM(x)$ operation

In addition to the ShiftRows and Rotational-XOR operations, many block ciphers employ circulant Boolean matrices (shortly CBM) for implementing linear layers. This type of linear layers will be called Cir-Boolean matrix type. Compared to random matrices, the linear layer based on a circulant matrix is more likely to be optimal [DKR97]. We start with the following result, which regards linear layers based on the CBM operation.

Algorithm 2: An efficient algorithm for finding a minimal cycle length of $l - RX(x)$ (l is odd) operating on $x \in \mathbb{F}_2^n$.

Input: $i_1, i_2, \ldots, i_l$, n, k $(2^k = n)$.
Output: A minimal cycle length μ of $l - RX(x)$.

1 Let $\mu = 2^{k-1}$.
2 Let $p_j = i_j \mu \pmod{n}$, $j = 1, 2, \ldots, l$.
3 For $i = 1, 2, \ldots, l$: $S_i = \{p_j | p_i = p_j, j = 1, 2, \ldots, l\}$.
4 Let $i = 1$.
5 While ($i \leq l$ and $S_i \neq S_j (i \neq 1, j = 1, 2, \ldots, i-1)$)
6 begin
7 If ($p_i = 0$ and $|S_i| \equiv 1 \pmod 2$) or ($p_i \neq 0$ and $|S_i| \equiv 0 \pmod 2$), then $i = i + 1$.
8 Else Output 2μ, terminate.
9 end
10 Let $\mu = \mu/2$. Go to Step 2.

Proposition 2 *Let x be a binary vector of length n, and let the operation $CBM(x)$ be defined*

$$CBM(x) : x^T \leftarrow Mx^T,$$

where M is a circulant Boolean matrix. The $CBM(x)$ operation is equivalent to the $l - RX(x)$ operation, for some value l.

Proof. Since an arbitrary circulant Boolean matrix can be written as

$$M = P^i \oplus P^{i_2} \oplus \cdots \oplus P^{i_t}, \ \ (0 \leq i_1 < i_2 < \cdots < i_t < n),$$

we have that $CBM(x)$ can be written (in a matrix form) as $CBM(x) : x^T \leftarrow Mx^T$, i.e.

$$\begin{aligned} CBM(x): &\quad x^T \leftarrow \left(P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_t}\right) x^T, \\ \text{or, } CBM(x): &\quad x^T \leftarrow P^{i_1} x^T \oplus P^{i_2} x^T \oplus \cdots \oplus P^{i_t} x^T. \end{aligned}$$

Furthermore, as $x \lll i_j$ is equivalent to $P^{i_j} x^T$ $(j \in \{1, \ldots, t\})$, we have that

$$CBM(x) : x \leftarrow \Sigma(x) = (x \lll i_1) \oplus (x \lll i_2) \oplus \cdots \oplus (x \lll i_t),$$

which means that $CBM(x)$ is equivalent to $l - RX(x)$. Clearly, the value l depends on integers $i_1, \ldots, i_t$ in the representation of the matrix M. ∎

To illustrate Proposition 2, in the following example we analyze the linear layer of the cipher Pyjamask [GJK+19], which is of the Cir-Boolean matrix type.

Example 3 *The linear layer of Pyjamask-128 cipher is given as*

$$\begin{aligned} M_0 &= \mathrm{cir}([1,1,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,0,1,1,1,0,0,0,1,0]) \\ M_1 &= \mathrm{cir}([0,1,0,0,0,0,1,0,0,0,0,0,0,1,1,1,0,1,0,0,0,0,0,1,0,1,1,0,0,0,1,1]) \\ M_2 &= \mathrm{cir}([0,0,0,0,0,0,0,0,1,0,1,0,0,1,1,1,1,0,0,1,1,0,1,0,0,1,0,0,1,0,1,1]) \\ M_3 &= \mathrm{cir}([0,1,1,0,0,1,0,0,0,0,0,0,1,0,0,1,0,1,0,1,0,0,1,0,1,0,0,0,1,0,0,1]) \end{aligned}$$

Due to Proposition 2, a minimal cycle length of this liner layer is computed as for the operation $l - RX(x)$ *(utilizing Lemma 1 and Corollary 1). We only analyze the matrix* M_3*, since for* M_0, M_1, M_2 *the procedure is similar. By observing the positions of* $1's$ *in the first row of* M_3*, we deduce that* M_3 *can be written as*

$$M_3 = P^1 \oplus P^2 \oplus P^5 \oplus P^{12} \oplus P^{15} \oplus P^{17} \oplus P^{19} \oplus P^{22} \oplus P^{24} \oplus P^{28} \oplus P^{31},$$

and thus the mapping $x^T \leftarrow M_3 x^T$ *represents the operation* $11 - RX(x)$*, i.e.* $11 - RX(x) : x^T \leftarrow M_3 x^T$*. By Corollary 1, we have that* $(M_3)^{32} = E$*, where* 32 *is not necessarily the minimal cycle length. In order to find the minimal cycle length of* M_3*, one has to consider divisors* μ *of* 32 *which are powers of* 2 *(*$n = 128$ *is a power of* 2*), and check whether* $(M_3)^\mu = E$ *holds. We find that* $(M_3)^8 \neq E$ *whereas for* $\mu = 16$ *it holds*

$$\begin{aligned}
(M_3)^{16} &= (P^1 \oplus P^2 \oplus P^5 \oplus P^{12} \oplus P^{15} \oplus P^{17} \oplus P^{19} \oplus P^{22} \oplus P^{24} \oplus P^{28} \oplus P^{31})^{16} \\
&= P^{1\times 16} \oplus P^{2\times 16} \oplus P^{5\times 16} \oplus \cdots \oplus P^{24\times 16} \oplus P^{28\times 16} \oplus P^{31\times 16} \\
&= P^{16} \oplus P^0 \oplus P^{16} \oplus P^0 \oplus P^{16} \oplus P^{16} \oplus P^{16} \oplus P^0 \oplus P^0 \oplus P^0 \oplus P^{16} \\
&= P^{16} \oplus E \oplus P^{16} \oplus E \oplus P^{16} \oplus P^{16} \oplus P^{16} \oplus E \oplus E \oplus E \oplus P^{16} = E.
\end{aligned}$$

Therefore, the minimum cycle length of M_3 *is actually* 16*, since Fact* 2) *given in Section 3.2.1 implies that no other divisors of* 32 *need to be considered. One applies the same procedure to matrices* M_0, M_1, M_2 *(see Table 6 in Section 4.3 for more details). As* 32 *is the least common multiple of these cycle lengths corresponding to* M_0, M_1, M_2*, the minimal cycle length of the linear layer of Pyjamask-128 is* $[32, 32, 32, 16] = 32$*.*

Due to the equivalence between Rotational-XOR and the Cir-Boolean operation for linear layers, the minimal cycle length of a linear layer employing the Cir-Boolean matrix operation can always be efficiently computed using Algorithm 2.

4 Experiments

In this section, we compute minimal cycle lengths of various well-known lightweight block ciphers whose linear layers employ ShiftRows, Rotational-XOR and Cir-Boolean matrix operation, respectively. Especially, for the Rotational-XOR type linear layers based on $l - RX(x)$ (l is odd) operation, we utilize Algorithm 2 for finding minimal cycle lengths. In Section 4.4, we also analyze several linear layers which are defined as the so-called bit permutations.

4.1 Encryption algorithms with the ShiftRows type linear layers

The computation of minimal cycle lengths for certain block ciphers that use the ShiftRows-type linear layers (based on Theorem 1) is given in Table 2. We use "RS-C" to denote the minimal cycle length of the individual rows of the $SR(x)$ operation, while "Cycle" denotes a minimal cycle length of the $SR(x)$ operation of the entire linear layer.

Although the minimal cycle lengths corresponding to different rows of ciphers in Table 2 are not equal, the minimal cycle length of entire linear layers for Knot, TANGRAM, Raindrop, ORANGE and PHOTON-Beetle algorithms are actually equal to the row length of the state matrix.

Table 2: Minimal cycle lengths of the $SR(x)$ operation for Knot, TANGRAM, Raindrop, ORANGE, PHOTON-Beetle.

Algorithm	Operation	RS-C	Cycle	Algorithm	Operation	RS-C	Cycle
Knot -256 [ZDY+19]	$x_0 \leftarrow x_0 \lll 0$	1	64	Raindrop -128 [WLL+19]	$x_0 \leftarrow x_0 \lll 0$	1	16
	$x_1 \leftarrow x_1 \lll 1$	64			$x_1 \leftarrow x_1 \lll 6$	8	
	$x_2 \leftarrow x_2 \lll 8$	8			$x_2 \leftarrow x_2 \lll 7$	16	
	$x_3 \leftarrow x_3 \lll 25$	64			$x_3 \leftarrow x_3 \lll 12$	4	
Knot -384 [ZDY+19]	$x_0 \leftarrow x_0 \lll 0$	1	96	Raindrop -256 [WLL+19]	$x_0 \leftarrow x_0 \lll 0$	1	32
	$x_1 \leftarrow x_1 \lll 1$	96			$x_1 \leftarrow x_1 \lll 12$	16	
	$x_2 \leftarrow x_2 \lll 8$	12			$x_2 \leftarrow x_2 \lll 14$	32	
	$x_3 \leftarrow x_3 \lll 55$	96			$x_3 \leftarrow x_3 \lll 24$	8	
Knot -512 [ZDY+19]	$x_0 \leftarrow x_0 \lll 0$	1	128	ORANGE [CN19b] PHOTON-Beetle [BCD+19]	$x_0 \leftarrow x_0 \lll 0$	1	8
	$x_1 \leftarrow x_1 \lll 1$	128			$x_1 \leftarrow x_1 \lll 1$	8	
	$x_2 \leftarrow x_2 \lll 16$	8			$x_2 \leftarrow x_2 \lll 2$	4	
	$x_3 \leftarrow x_3 \lll 25$	128			$x_3 \leftarrow x_3 \lll 3$	8	
TANGRAM -128/256 [ZJD+19]	$x_0 \leftarrow x_0 \lll 0$	1/1	32/64		$x_4 \leftarrow x_4 \lll 4$	2	
	$x_1 \leftarrow x_1 \lll 1$	32/64			$x_5 \leftarrow x_5 \lll 5$	8	
	$x_2 \leftarrow x_2 \lll 8$	4/8			$x_6 \leftarrow x_6 \lll 6$	4	
	$x_3 \leftarrow x_3 \lll 11/41$	32/64			$x_7 \leftarrow x_7 \lll 7$	8	

Remark 5 *The correctness of our theoretical analysis has been confirmed by choosing* 100 *random vectors and then iterating the liner layers for the ciphers listed in Table 2 for each of these vectors to obtain minimal cycle lengths.*

Similarly, in Table 3, we consider the minimal cycle length for the $SR(x)$ operation in linear layers of SKINNY-, AES- and sLiSCP-light-like algorithms.

Table 3: Minimal cycle lengths of the $SR(x)$ operation for SKINNY-, AES-, and sLiSCP-light-like ciphers.

Algorithm		Operation	RS-C	Cycle
Based on SKINNY[BJK+16]	ForkAE[ALP+19] SKINNY-AEAD/ SKINNY-HASH[BJK+19] Romulus[IKMP19]	$(s_0, s_1, s_2, s_3) \rightarrow (s_0, s_1, s_2, s_3)$	1	4
		$(s_4, s_5, s_6, s_7) \rightarrow (s_7, s_4, s_5, s_6)$	4	
		$(s_8, s_9, s_{10}, s_{11}) \rightarrow (s_{10}, s_{11}, s_8, s_9)$	2	
		$(s_{12}, s_{13}, s_{14}, s_{15}) \rightarrow (s_{13}, s_{14}, s_{15}, s_{12})$	4	
Based on AES[DR20]	mixFeed[CN19a] COMET[GJN19] ESTATE[CDJ+19] SAEAES[NMS+19]	$(s_0, s_1, s_2, s_3) \rightarrow (s_0, s_1, s_2, s_3)$	1	4
		$(s_4, s_5, s_6, s_7) \rightarrow (s_5, s_6, s_7, s_4)$	4	
		$(s_8, s_9, s_{10}, s_{11}) \rightarrow (s_{10}, s_{11}, s_8, s_9)$	2	
		$(s_{12}, s_{13}, s_{14}, s_{15}) \rightarrow (s_{15}, s_{12}, s_{13}, s_{14})$	4	
Based on sLiSCP-light[ARH+18]	Spoc[AGH+19a] Spix[AGH+19b]	$(s_0, s_1, s_2, s_3) \rightarrow (s_3, s_2, s_1, s_0)$	4	4

4.2 Encryption algorithms with the Rotational-XOR type linear layers

In Table 4, we use Algorithm 2 to compute minimal cycle lengths of linear layers that employ the $l - RX(x)$ operation, where l is odd. Table 4 indicates that in most of cases the minimal cycle length actually equals to the length of the input vector.

Table 4: Minimal cycle lengths of the $l - RX(x)$ (l is odd) operation for some block ciphers.

Algorithm	Domain	Operation	RS-C	Cycle
Ascon[DEMS19] ISAP[DEM$^+$19] (ISAP-A-128a/128)	$x_i \in \mathbb{F}_2^{64}$	$x_0 \leftarrow x_0 \oplus (x_0 \ggg 19) \oplus (x_0 \ggg 28)$	64	64
		$x_1 \leftarrow x_1 \oplus (x_1 \ggg 61) \oplus (x_1 \ggg 39)$	32	
		$x_2 \leftarrow x_2 \oplus (x_2 \ggg 1) \oplus (x_2 \ggg 6)$	64	
		$x_3 \leftarrow x_3 \oplus (x_3 \ggg 10) \oplus (x_3 \ggg 17)$	64	
		$x_4 \leftarrow x_4 \oplus (x_4 \ggg 7) \oplus (x_4 \ggg 41)$	32	
DBlock[WZY15]	$x \in \mathbb{F}_2^{32}$	$x \leftarrow x \oplus (x \lll 8) \oplus (x \lll 10) \oplus (x \lll 18) \oplus (x \lll 26)$	16	16
RoadRunneR[BS15]	$x \in \mathbb{F}_2^{8}$	$x \leftarrow x \oplus (x \lll 1) \oplus (x \lll 2)$	8	8
FBC[FZZ$^+$19]	$x \in \mathbb{F}_2^{32}$	$x \leftarrow x \oplus (x \lll 3) \oplus (x \lll 10)$	32	32
	$x \in \mathbb{F}_2^{64}$	$x \leftarrow x \oplus (x \lll 17) \oplus (x \lll 58)$	64	64

Remark 6 *For DBlock cipher the situation is different since the minimal cycle length is actually half of the input size. This, in general, may be viewed as a disadvantage since shorter cycles impose less conditions on the subset of weak keys which potentially increases its cardinality . Nevertheless, DBlock uses a quite complicated key scheme and reasonable number of rounds which probably efficiently counteract this potential weakness.*

On the other hand, when l is even, we recall that the operation $l - RX(x)$ does not possess a cycle of minimal length and is never a permutation. For several (hypothetical) such linear layers (shown in Table 5), one can devise a method similar to Algorithm 2 for finding minimal number of iterations (denoted by Min-Iter) needed to obtain the zero vector O at the output. More details, regarding the existence of minimal number of iterations for reaching the zero vector O, are given in the appendix. These results have been practically confirmed by implementing the linear layers described in Table 5.

Table 5: The minimal number of iterations of the $l - RX(x)$ (l is even) operation.

Domain	Operation	Min-Iter
$x \in F_2^{64}$	$x \leftarrow (x \lll 9) \oplus (x \lll 21) \oplus (x \lll 41) \oplus (x \lll 57)$	32
	$x \leftarrow x \oplus (x \lll 17) \oplus (x \lll 19) \oplus (x \lll 23)$	64
	$x \leftarrow (x \lll 16) \oplus (x \lll 24) \oplus (x \lll 32) \oplus (x \lll 40)$	4
	$x \leftarrow (x \lll 11) \oplus (x \lll 23) \oplus (x \lll 43) \oplus (x \lll 47)$	16
$x \in F_2^{32}$	$x \leftarrow (x \lll 11) \oplus (x \lll 31) \oplus (x \lll 43) \oplus (x \lll 47)$	2
	$x \leftarrow x \oplus (x \lll 17) \oplus (x \lll 22) \oplus (x \lll 31)$	16
	$x \leftarrow (x \lll 2) \oplus (x \lll 5) \oplus (x \lll 13) \oplus (x \lll 34)$	4
	$x \leftarrow x \oplus (x \lll 4) \oplus (x \lll 9) \oplus (x \lll 17)$	8

4.3 Encryption algorithms using linear layers of the Cir-Boolean matrix type

As an extension of Example 3, we further specify minimal cycle lengths for Pyjamask-96/128 [GJK$^+$19] and Midori [TLS16] in Table 6. It should be noted that Pyjamask employs a linear layer based on circulant matrix, for which it was shown to be equivalent to the $l - RX(x)$

operation (for some odd integer l). For this cipher, the state length of 96 bits uses the linear layer with the matrices M_0, M_1, M_2 whereas the length 128 corresponds to the case when all four matrices $M_0, \ldots, M_3$ are used to define the linear layer, see also Example 3. In both cases, the input vectors are of the length 32. In difference to this algorithm, when analyzing the i-th row of Midori [TLS16] denoted by S_i, the state is represented as a matrix whose entries correspond to 4-bit nibbles. Unlike Pyjamask which operates on the bit level, the shift operations of S_i are performed on the state cells.

Table 6: Minimal cycle lengths of the $CBM(x)$ operation for Pyjamask-96/128 and Midori64.

Algorithm	$CBM(x)$ Operation	Equivalent $l - RX(x)$ Operation	RS-C
Midori64[TLS16]	$S_i^T \leftarrow MS_i^T$, $i = 0, 1, 2, 3$	$S_i \leftarrow (S_i \lll 1) \oplus (S_i \lll 2) \oplus (S_i \lll 3)$	4
Pyjamask -96/128[GJK+19]	$x_0^T \leftarrow M_0 x_0^T$	$x_0 \leftarrow x_0 \oplus (x_0 \lll 1) \oplus (x_0 \lll 3) \oplus (x_0 \lll 8)$ $\oplus (x_0 \lll 13) \oplus (x_0 \lll 18) \oplus (x_0 \lll 19) \oplus (x_0 \lll 24)$ $\oplus (x_0 \lll 25) \oplus (x_0 \lll 26) \oplus (x_0 \lll 30)$	32
	$x_1^T \leftarrow M_1 x_1^T$	$x_1 \leftarrow (x_1 \lll 1) \oplus (x_1 \lll 6) \oplus (x_1 \lll 13) \oplus (x_1 \lll 14)$ $\oplus (x_1 \lll 15) \oplus (x_1 \lll 17) \oplus (x_1 \lll 23) \oplus (x_1 \lll 25)$ $\oplus (x_1 \lll 26) \oplus (x_1 \lll 30) \oplus (x_1 \lll 31)$	32
	$x_2^T \leftarrow M_2 x_2^T$	$x_2 \leftarrow (x_2 \lll 8) \oplus (x_2 \lll 10) \oplus (x_2 \lll 13) \oplus (x_2 \lll 14)$ $\oplus (x_2 \lll 15) \oplus (x_2 \lll 16) \oplus (x_2 \lll 19) \oplus (x_2 \lll 22)$ $\oplus (x_2 \lll 24) \oplus (x_2 \lll 28) \oplus (x_2 \lll 31)$	32
	$x_3^T \leftarrow M_3 x_3^T$	$x_3 \leftarrow (x_3 \lll 1) \oplus (x_3 \lll 2) \oplus (x_3 \lll 5) \oplus (x_3 \lll 12)$ $\oplus (x_3 \lll 15) \oplus (x_3 \lll 17) \oplus (x_3 \lll 19) \oplus (x_3 \lll 22)$ $\oplus (x_3 \lll 24) \oplus (x_3 \lll 28) \oplus (x_3 \lll 31)$	16

4.4 Algorithms with bit permutations

Apart from linear layers analyzed in this work, there exist various block ciphers which employ bit-permutations as linear layers, which act on the whole state at once (where the state is considered as a vector, eg. PRESENT block cipher [BKL+07]). By running simulations specifically designed for ciphers listed in Table 7, we have successfully determined the corresponding minimal cycle lengths.

In fact, the cycle lengths of individual bits are not exactly the same after performing permutation operations. For example, in GIFT-128, the cycle lengths of different bit positions take values in the set $\{1, 2, 5, 10, 31\}$. Therefore, the cycle length of the linear layer of GIFT-128 is 310, corresponding to the least common multiple of the individual bit cycle lengths. The same is true for other algorithms.

5 Conclusions

In this article, we have studied the problem of finding a minimal cycle length of linear layers based on cyclic shifts and XOR operations. These operations are quite typical whenever the

Table 7: Minimal cycle lengths of linear layers defined as bit-permutation.

Algorithm	Operation	Cycle
GIFT-64[BPP+17]	$P(i) = 16((3\lfloor i \bmod 16/4 \rfloor + i \bmod 4)) \bmod 4 + 4\lfloor i/16 \rfloor + (i \bmod 4), \quad i = 0, \ldots, 63.$	4
GIFT-128[BPP+17] HYENA[CDJN19] GIFT-COFB[BCI+19]	$P(i) = 32((3\lfloor i \bmod 16/4 \rfloor + i \bmod 4)) \bmod 4 + 4\lfloor i/16 \rfloor + (i \bmod 4), \quad i = 0, \ldots, 127.$	310
PRINT-48[KLPR10] PRINT-96[KLPR10]	$P_b(j) = \begin{cases} 3j \bmod b-1 & \text{for } 0 \le j \le b-2, \\ b-1 & \text{for } j = b-1, \quad b = 48/96. \end{cases}$	23/36
Elephant-Spongent-160[BCDM19]	$P_{160}(j) = \begin{cases} 40j \bmod 159 & \text{for } 0 \le j \le 158, \\ 159 & \text{for } j = 159. \end{cases}$	26
Elephant-Spongent-176[BCDM19]	$P_{175}(j) = \begin{cases} 44j \bmod 175 & \text{for } 0 \le j \le 174, \\ 175 & \text{for } j = 175. \end{cases}$	30
PRESENT[BKL+07]	$P_{64}(j) = \begin{cases} 16j \bmod 63 & \text{for } 0 \le j \le 62, \\ 63 & \text{for } j = 63. \end{cases}$	3

state of a block cipher has a matrix representation and additionally assuming that the linear layer acts on each row of the state matrix. The main motivation for this theoretical analysis comes from Proposition 1 which allows for an efficient specification of invariants of a whole encryption round and therefore it may potentially give rise to various distinguishing attacks. It is an interesting research topic to find invariants of the entire encryption rounds of suitable block ciphers (through Proposition 1) by specifying a set of cycles which lie in the intersection of the linear and S-box layer.

Acknowledgements

Guoqiang Deng is supported by Guangxi Science and Technology Project (Guike AD18281024). Yongzhuang Wei is supported in part by the National Natural Science Foundation of China (61872103), in part by Guangxi Science and Technology Foundation (Guike AB18281019), in part by Guangxi Natural Science Foundation (2019GXNSFGA245004). Xuefeng Duan is supported in part by the National Natural Science Foundation of China(11561015, 11761024), in part by the Natural Science Foundation of Guangxi Province (2016GXNSFFA380009, 2017GXNSFBA198082, 2016GXNSFAA380074). Enes Pasalic is partly supported by the Slovenian Research Agency (research program P1-0404 and research projects J1-1694, J1-9108 and N1-0159). Samir Hodžić is supported by a grant from the Independent Research Fund Denmark for Technology and Production, grant no. 8022-00348A.

References

[AGH+19a] R. AlTawy, G. Gong, M. He, A. Jha, K. Mandal, M. Nandi, and R. Rohit. SpoC-Submission to the NIST LWC Competition. information available at https://uwaterloo.ca/communications-security-lab/lwc/spoc. 2019.

[AGH+19b] R. AlTawy, G. Gong, M. He, K. Mandal, and R. Rohit. Spix-Submission to the NIST LWC Competition. information available athttps://uwaterloo.ca/communications-security-lab/lwc/spix. 2019.

[ALP+19] E. Andreeva, V. Lallemand, A. Purnal, R. Reyhanitabar, A. Roy, and D. Vizár. ForkAE-Submission to the NIST LWC Competition. information available at https://www.esat.kuleuven.be/cosic/forkae. 2019.

[ARH+18] R. AlTawy, R. Rohit, M. He, K. Mandal, G. Yang, and G. Gong. Sliscp-light: Towards hardware optimized sponge-specific cryptographic permutations. *ACM Trans. Embedded Comput. Syst.*, 17(4):81:1–81:26, 2018.

[BBI+15] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni. Midori: A block cipher for low energy. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015*, volume 9453 of *Lecture Notes in Computer Science*, pages 411–436. Springer, 2015.

[BCD+19] Z. Bao, A. Chakraborti, N. Datta, J. Guo, M. Nandi, T. Peyrin, and K. Yasuda. PHOTON-Beetle-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/lightweight/beetle. 2019.

[BCDM19] T. Beyne, Y. Chen, C. Dobraunig, and B. Mennink. Elephant-Submission to the NIST LWC Competition. information available at https://www.esat.kuleuven.be/cosic/elephant/. 2019.

[BCI+19] S. Banik, A. Chakraborti, T. Iwata, K. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S.M. Sim, and Y. Todo. GIFT-COFB-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/lightweight/COFB/. 2019.

[BCLR17] C. Beierle, A. Canteaut, G. Leander, and Y. Rotella. Proving resistance against invariant attacks: How to choose the round constants. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology - CRYPTO 2017*, volume 10402 of *Lecture Notes in Computer Science*, pages 647–678. Springer, 2017.

[BJK+16] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S.M. Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology - CRYPTO 2016*, volume 9815 of *Lecture Notes in Computer Science*, pages 123–153. Springer, 2016.

[BJK+19] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S.M. Sim. SKINNY-AEAD/SKINNY-HASH-Submission to the NIST LWC Competition. information available at https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates. 2019.

[BKL+07] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe. PRESENT: an ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007.

[BPP+17] S. Banik, S.K. Pandey, T. Peyrin, Y. Sasaki, S.M. Sim, and Y. Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017*, volume 10529 of *Lecture Notes in Computer Science*, pages 321–345. Springer, 2017.

[BS90] E. Biham and A. Shamir. Differential cryptanalysis of des-like cryptosystems. In Alfred Menezes and Scott A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer, 1990.

[BS15] A. Baysal and S. Sahin. Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *Lightweight Cryptography for Security and Privacy*, volume 9542 of *Lecture Notes in Computer Science*, pages 58–76. Springer, 2015.

[CDJ+19] A. Chakraborti, N. Datta, A. Jha, C. M. Lopez, M. Nandi, and Y. Sasaki. ESTATE-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/ lightweight/estate. 2019.

[CDJN19] A. Chakraborti, N. Datta, A. Jha, and M. Nandi. HYENA-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/lightweight/hyena/. 2019.

[CN19a] B. Chakraborty and M. Nandi. MixFeed-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/lightweight/Mixfeed. 2019.

[CN19b] B. Chakraborty and M. Nandi. ORANGE-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/ lightweight/Orange/. 2019.

[DEM+19] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas, and T. Unterluggauer. Isap-Submission to the NIST LWC Competition. information available at https://isap.iaik.tugraz.at/. 2019.

[DEMS19] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. Ascon-Submit the national cryptographic algorithm design competition to enter the second round of grouping algorithm, information available at https://ascon.iaik.tugraz.at. 2019.

[DKR97] J. Daemen, L.R. Knudsen, and V. Rijmen. The block cipher square. In Eli Biham, editor, *Fast Software Encryption, FSE'97*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer, 1997.

[DR20] J. Daemen and V. Rijmen. *The Design of Rijndael - The Advanced Encryption Standard (AES), Second Edition.* Information Security and Cryptography. Springer, 2020.

[FZZ+19] X. Feng, X. Zeng, G. Zeng, D. Tang, F. Zhang, and G. Gan. FBC-Submit the national cryptographic algorithm design competition to enter the second round of grouping algorithm, information available at https://sfjs.cacrnet.org.cn/site/term/list-76-1.html. 2019.

[GJK+19] D. Goudarzi, J. Jean, S. Kölbl, T. Peyrin, M. Rivain, Y. Sasaki, and S.M. Sim. Pyjamask-Submission to the NIST call for lightweight cryptography. information available at https://pyjamask-cipher.github.io/index.html. 2019.

[GJN19] S. Gueron, A. Jha, and M. Nandi. COMET-Submission to the NIST LWC Competition. information available at https://www.isical.ac.in/lightweight/comet. 2019.

[GLSV14] V. Grosso, G. Leurent, F.X. Standaert, and K. Varici. SCREAM v1. Submission to CAESAR competition. 2014.

[GLSV15] V. Grosso, G. Leurent, F.X. Standaert, and K. Varici. SCREAM v3. Submission to CAESAR competition. 2015.

[Guo17] Z. Guo. *Structural analysis and component design of symmetric cipher*. PhD thesis, Chinese Academy of Sciences, 2017.

[IKMP19] T. Iwata, M. Khairallah, K. Minematsu, and T. Peyrin. Romulus-Submission to the NIST LWC Competition. information available at https://romulusae.github.io/romulus. 2019.

[KLPR10] L.R. Knudsen, G. Leander, A. Poschmann, and M.J.B. Robshaw. Printcipher: A block cipher for ic-printing. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010.

[LAAZ11] G. Leander, M.A. Abdelraheem, H. AlKhzaimi, and E. Zenner. A cryptanalysis of printcipher: The invariant subspace attack. In Phillip Rogaway, editor, *Advances in Cryptology-CRYPTO 2011*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221. Springer, 2011.

[LMR15] G. Leander, B. Minaud, and S. Rønjom. A generic approach to invariant subspace attacks: Cryptanalysis of robin, iscream and zorro. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Computer Science*, pages 254–283. Springer, 2015.

[Mat93] M. Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

[NMS+19] Y. Naito, M. Matsui, Y. Sakai, D. Suzuki, K. Sakiyama, and T. Sugawara. SAEAES-Submission to the NIST LWC Competition. information available at https://www.saeaes.net. 2019.

[Sch74] S. Schwarz. Circulant boolean relatton matrices. *Czechoslovak Mathematical Journal*, 24(2):252–253, 1974.

[SM412] Bulletin of the International Cryptography Authority (No. 23): SM4 block cipher algorithm, GMT 0002-2012, information available at http://www.oscca.gov.cn. 2012.

[TLS16] Y. Todo, G. Leander, and Y. Sasaki. Nonlinear invariant attack - practical attack on full scream, iscream, and midori64. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016*, volume 10032 of *Lecture Notes in Computer Science*, pages 3–33, 2016.

[WLL+19] M. Wang, Y. Li, M. Li, Y. Fu, Y. Fan, and L. Huang. Raindrop-Submit the national cryptographic algorithm design competition to enter the second round of grouping algorithm, information available at https://sfjs.cacrnet.org.cn/site/term/list-76-1.html. 2019.

[WRP+20] Y. Wei, R. Rodiguez, and E. Pasalic. Cycle structure of generalized and closed loop invariants. Available at https://eprint.iacr.org/2020/1095.pdf, 2020.

[WS08] J. Wang and Y. Shao. Determination of two properties of cyclic matrix. *Pure mathematics and applied mathematics*, 24(4):762–767, 2008.

[WYWP18] Y. Wei, T. Ye, W. Wu, and E. Pasalic. Generalized nonlinear invariant attack and a new design criterion for round constants. *IACR Trans. Symmetric Cryptol.*, 2018(4):62–79, 2018.

[WZY15] W. Wu, L. Zhang, and X. Yu. The dblock family of block ciphers. *Science China Infomation Sciences*, 58(3):1–14, 2015.

[ZDY+19] W. Zhang, T. Ding, B. Yang, Z. Bao, Z. Xiang, F. Ji, and X. Zhao. KNOT-Submission to the NIST LWC Competition. information available at https://csrc.nist.gov/Projects/lightweight-cryptography/round-2-candidates. 2019.

[ZJD+19] W. Zhang, F. Ji, T. Ding, B. Yang, X. Zhao, and Z. Xiang. TANGRAM-Submit the national cryptographic algorithm design competition to enter the second round of grouping algorithm 19. information available at https://sfjs.cacrnet.org.cn/site/term/html. 2019.

Appendix

A The operation $l - RX(x)$ with l being even

Here, we derive some basic properties of the operation $l - RX(x)$ when l is even.

Theorem 3 *Let x be a binary vector of length n. Suppose the operation $l - RX(x)$ is given as*

$$l - RX(x) : x \leftarrow \Sigma(x) = (x \lll i_1) \oplus (x \lll i_2) \oplus \cdots \oplus (x \lll i_l), \tag{6}$$

where $0 \leq i_1 < i_2 < \cdots < i_l < n$. If n is of the form $n = 2^k$, then:

i) The operation $l - RX^{\mu}(x)$ is not a permutation.

ii) For $\mu \geq n$, it holds that $l - RX^{\mu}(x) = O$, for any $x \in \mathbb{F}_2^n$.

iii) For $\mu \geq 1$, $l - RX^{\mu}(x) \neq x$ holds, whenever we have that $x \neq O$. That is, these is no cycle of minimal length in this case.

Proof.

i) The operation $l - RX(x)$ is not a (linear) permutation since taking $x = \mathbf{e} = (1, 1, \ldots, 1)$ or $x = O = (0, 0, \ldots, 0)$ we have $l - RX(\mathbf{e}) = l - RX(O)$, for any even l.

ii) Clearly, $l - RX(O) = O$, which is also a fixed point. Recall that $l - RX(x)$ is equivalently defined as $l - RX(x) : x^T \leftarrow Qx^T$, where the matrix Q is given by

$$Q = P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l}.$$

Assuming that $\mu = n = 2^k$, by Lemma 1 we have that

$$Q^n = \ (P^{i_1} \oplus P^{i_2} \oplus \cdots \oplus P^{i_l})^n = P^{n \cdot i_1} \oplus P^{n \cdot i_2} \oplus \cdots \oplus P^{n \cdot i_l}.$$

Since $P^n = E$, we have that $Q^n = E^{i_1} \oplus E^{i_2} \oplus \cdots \oplus E^{i_l} = O$ holds when l is even. Consequently, we have $Q^n x^T = O x^T = O$, that is $l - RX^n(x) = O$ for any $x \in \mathbb{F}_2^n$. Then, obviously $Q^{\mu} x^T = Q^t O = O$ for $\mu > n$.

iii) Since for any $x \in \mathbb{F}_2^n$, it holds that $l - RX^{\mu}(x) = O$ when $\mu \geq n$, we know if the minimum cycle length μ exists for $l - RX(x)$ operation, then μ must be less than n. Let us assume

that $\mu < n$. We distinguish the following cases:

1) If $\mu \,|\, n$, then there exists a positive integer k such that $n = k\mu$. Since μ denotes the minimal cycle length, i.e. $Q^{\mu} = E$, we have that $Q^n = (Q^{\mu})^k = E$, which contradicts the fact that $l - RX^n(x) = O$ for any $x \in \mathbb{F}_2^n$.

2) If $\mu \nmid n$, then there exist integers k and m such that $n = k\mu + m$, where $m < \mu$. In this case we have that

$$Q^n = O \Leftrightarrow Q^{k\mu+m} = O \Leftrightarrow (Q^{\mu})^k Q^m = O \Leftrightarrow (E)^k Q^m = O \Leftrightarrow Q^m = O.$$

Since $m < \mu$, this shows that $Q^{\mu} = Q^{\mu-m}Q^m = Q^{\mu-m}O = O$, which contradicts the fact that $Q^{\mu} = E$.

By summarizing the cases 1) and 2), we deduce that $\mu < n$ can not be the minimal cycle length for the $l - RX(x)$ operation. Therefore, for even l, the operation $l - RX(x)$ does not possess a minimal cycle length μ with $\mu \geq 1$. □