# Linear Frameworks for Block Ciphers

JOAN DAEMEN daemen.j@protonworld.com

*Proton World International, Zweefvliegtuigstraat 10*

*B-1130 Brussel, Belgium*

LARS R. KNUDSEN lars.knudsen@ii.uib.no

*University of Bergen, Department of Informatics*

*Hoyteknologisenteret, N-5020 Bergen, Norway*

VINCENT RIJMEN [*] vincent.rijmen@esat.kuleuven.ac.be

*Katholieke Universiteit Leuven , Dept. ESAT, SISTA/COSIC Lab*

*K. Mercierlaan 94, B-3001 Heverlee, Belgium*

**Abstract.** In this paper we generalize the structure of the ciphers SHARK, SQUARE and Rijndael. We show that the linear components play an essential role in the effect of the nonlinear S-boxes in providing resistance against differential and linear cryptanalysis and provide upper bounds for the probability of differential characteristics and the correlation of linear approximations for the general structure. We show how good linear components can be constructed efficiently from Maximum-Distance Separable codes. The presented block cipher structure can make optimal use of a wide range of processor word lengths and its parallelism allows very fast dedicated hardware implementations. Ciphers with variable block length can be constructed by varying certain parameters in the presented structure.

## 1. Introduction

Many papers have been published on the design of nonlinear functions and S-boxes, which are considered the most important components of ciphers and hash functions. In this paper we concentrate on the linear components in cipher structures, their interactions and their role in providing resistance against differential and linear

[*] F.W.O. Postdoctoral Researcher, sponsored by the Fund for Scientific Research - Flanders (Belgium)

cryptanalysis. The generation of S-boxes that are part of the presented structure and round key schedules are out of the scope of this paper. We realize that carelessly chosen S-boxes or key schedules allow efficient attacks.

The cipher structures presented in this paper are similar to the classical Substitution-Permutation structure [6]. Instead of permutations our constructions exhibit more general linear transformations. The main goal of this paper is to demonstrate that resistance against linear and differential cryptanalysis can be efficiently obtained by combining linear components with non-linear components, both selected according to very simple criteria.

There are several examples of ciphers (e.g, SAFER [12] and TWOPRIME [5]) that use linear transformations, however the transformations they use, produce a suboptimal diffusion. In the design of SAFER, the imperfections of the diffusion layer are neutralized by specific properties of the S-boxes. In our approach the linear and nonlinear components of the cipher can be selected in a more independent way.

This paper is organized as follows. In Section 2 we give some background definitions and discuss related work. We motivate our approach. In Section 3 we discuss the general cipher structure that we propose here. In Section 4 we explain how the linear components of the cipher influence the resistance of the ciphers against differential and linear cryptanalysis, the two most successful cryptanalytic techniques. The main contribution of this paper lies in Section 5: we develop criteria for linear transformations that optimize the resistance of the cipher against differential and linear cryptanalysis. We show how the resistance can be bounded accurately. We discuss some existing designs with suboptimal linear components in Section 6 and in Section 7 we present examples of ciphers with the proposed optimal structure. Section 8 discusses a dedicated attack that has to be taken into account. We conclude in Section 9.

## 2. Background, Related Work and Motivation

### *2.1. Block ciphers*

A *block cipher* is an algorithm that transforms plaintext blocks of a fixed size $n$ into ciphertext blocks of a fixed size $n'$ under the influence of a key $k$. Usually, this is done by a repeated application of an invertible transformation $\rho$, called the *round transformation.* In this paper we only consider block ciphers where $n = n'$. The first theory behind the design of encryption algorithms can be found in [16], where C.E. Shannon proposes to build strong ciphers by alternating simple substitutions with mixing transformations. The result of the combination is that *"any significant statistics from the encryption algorithm must be of a highly involved and very sensitive type — the redundancy has been both* diffused *and* confused *by the mixing transformation."*

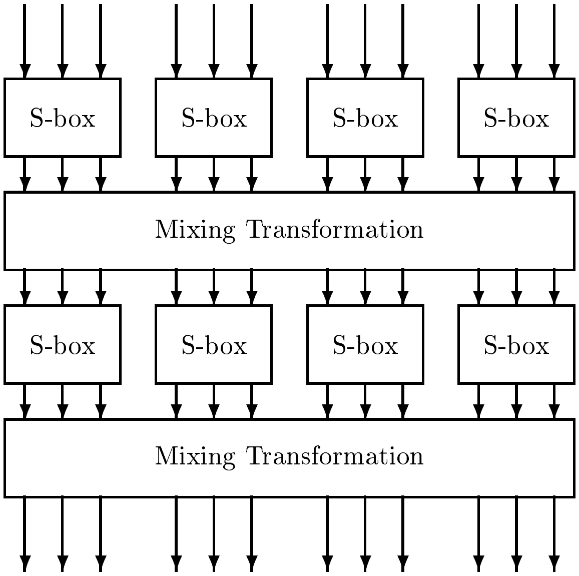

*Figure 1.* Example of a small block cipher, with a block length of 12 bits. The nonlinear S-boxes operate on four bits at a time. The mixing transformation mixes the outputs of the four S-boxes.

Figure 1 shows a small block cipher that consists of two *rounds.* Every round consists of the application of substitutions and the mixing transformation. The substitutions provide the nonlinearity of the cipher. The input block is divided into smaller blocks, which are all transformed by an *S-box.* An S-box $S$ is defined as a transformation in the set of $m$-bit vectors $\mathbb{Z}_2^m$.

$$S : \mathbb{Z}_2^m \to \mathbb{Z}_2^m : (x_1, x_2, \dots x_m) \mapsto (y_1, y_2, \dots y_m) = S((x_1, x_2, \dots x_m))$$

The mixing transformations connect the outputs of the S-boxes in one round with the inputs of the S-boxes in the next round. Their purpose is to make it impossible to recover meaningful statistics on the input/output of a small set of S-boxes from the statistics of the cipher output. The mixing transformations can be linear or nonlinear, but the introduction of nonlinearity is not their main purpose. An example of a linear mixing transformation is the *bit permutation.* In a bit permutation, the bits of the input are reordered but left otherwise unchanged, e.g. the following mapping: $(x_1, x_2, x_3, x_4, \ldots x_n) \mapsto (x_1, x_3, \ldots, x_{n-1}, x_2, x_4, \ldots x_n)$.

### *2.2. Design strategies*

In the cryptographic literature, a number of different approaches have been developed to design ciphers that resist differential and linear cryptanalysis. Some approaches concentrate on the design of S-boxes, other take the influence of the linear components into account.

In [7] H.M. Heys and S.E. Tavares study several examples of linear layers and the resulting resistance against differential and linear cryptanalysis. An important difference between their approach and our in this paper, is that we consider the diffusion effect of the linear transformation separately from the properties of the S-boxes, while they consider both at the same time.

The design of SHARK [15] exhibits a linear transformation that is based on Maximum-Distance Separable codes (MDS-codes). This approach was extended in [19] to produce mappings with optimal diffusion properties that are also involutions. Also in [18] linear mappings with optimal diffusion properties were studied.

When the block length of a cipher increases, the requirement for optimal diffusion results in increased memory requirements. For block lengths of 128 bits and more, the memory requirements prohibit fast implementations on currently in use processors. In this paper, we try to give a more general framework for linear transformations, allowing the requirement for optimal diffusion to be loosened in a controlled way in order to get more efficient implementations, as is done in the design of SQUARE [4] and Rijndael [3].

## 3. General Cipher Structure

In the presented approach a block cipher consists of the repeated application of the round transformation, which is itself composed of a number of invertible transformations. After the introduction of these transformations, we will show the general cipher structure In order to clarify the discussion, we illustrate the different steps by means of a simple example cipher. We conclude with a treatment of table-lookup implementation aspects and the inverse cipher.

### *3.1. The round transformation*

The cipher structure consists of the iterated application of a (keyed) round transformation. This round transformation, $\rho[k^t]$ is defined by

$$\rho[k^t] = \sigma[k^t] \circ \theta \circ \pi \circ \gamma, \tag{1}$$

where $\sigma, \theta, \pi$ and $\gamma$ are invertible transformations, defined in the next paragraphs. The input of the transformations is called the *intermediate state*, denoted $a$ and subdivided into $n_t$ $m$-bit tuples $a_i \in \mathbb{Z}_2^m$ with $i \in \mathcal{I}$. $\mathcal{I}$ is called the *index space*. The tuples can be grouped into a number of subsets, which correspond to elements of a partition $\Xi$ of the index space. We denote the number of partition element by $n_\Xi$, the partition element containing an index $i$ by $\xi(i)$ and the number of indices in $\xi$ by $n_\xi$. The block size of the cipher is given by $m \sum_{\xi \in \Xi} n_\xi = mn_t$.

For example, let $\mathcal{X}$ be a cipher with a block length of 48 bits. Let the input be divided into 6 8-bit tuples, arranged in a 2 × 3 array.

$$a = \begin{bmatrix} a_0 & a_2 & a_4 \\ a_1 & a_3 & a_5 \end{bmatrix}$$

The index space is $\mathcal{I} = \{0, 1, 2, 3, 4, 5\}$. We define the partition $\Xi$ as $\Xi = \{\{0, 1\}, \{2, 3, 4, 5, \}\}$, $n_\Xi = 2$, $n_{\xi(0)} = n_{\xi(1)} = 2$ and $n_{\xi(2)} = n_{\xi(3)} = n_{\xi(4)} = n_{\xi(5)} = 4$. The block size of the cipher is $8 \cdot (2 + 4) = 48$.

#### *3.1.1. The nonlinear substitution $\gamma$*

We have

$$\gamma : b = \gamma(a) \Leftrightarrow b_i = S_\gamma(a_i), \tag{2}$$

with $S_\gamma$ an invertible nonlinear $m$-bit substitution box. For the purpose of this paper, $S_\gamma$ needs not to be specified. Clearly, the inverse of $\gamma$ consists of applying the inverse substitution $S_\gamma^{-1}$ to all tuples. The results of this paper can also easily be generalized to include nonlinear mappings that use different S-boxes for different tuples. However, this does not result in a plausible improvement of the resistance against the considered attacks.

*3.1.2. The transpositions $\pi$ and $\mu$* The transpositions are defined as

$$\pi : b = \pi(a) \Leftrightarrow b_i = a_{p(i)}, \tag{3}$$

$$\mu : b = \mu(a) \Leftrightarrow b_i = a_{m(i)}, \tag{4}$$

with $p(i)$ and $m(i)$ permutations of the index space $\mathcal{I}$. The inverses of $\pi$ and $\mu$ are defined by $p^{-1}(i)$ and $m^{-1}(i)$ respectively. Since $\gamma$ operates on individual tuples in a uniform way, these transpositions commute with $\gamma$. In the example cipher $\mathcal{X}$, we define $\pi$ as the transformation that leaves the first row unchanged and shifts the second row one place to the right. $\mu$ exchanges the first and the third column.

$$\pi\left(\begin{bmatrix} a_0 & a_2 & a_4 \\ a_1 & a_3 & a_5 \end{bmatrix}\right) = \begin{bmatrix} a_0 & a_2 & a_4 \\ a_5 & a_1 & a_3 \end{bmatrix}$$

$$\mu\left(\begin{bmatrix} a_0 & a_2 & a_4 \\ a_1 & a_3 & a_5 \end{bmatrix}\right) = \begin{bmatrix} a_4 & a_2 & a_0 \\ a_5 & a_3 & a_1 \end{bmatrix}$$

*3.1.3. The linear transformation $\theta$* $\theta$ operates independently on every element of the partition $\Xi$. Within each partition element, all tuples are linearly combined. We have

$$\theta : b = \theta(a) \Leftrightarrow b_i = \sum_{j \in \xi(i)} C_{i,j} a_j \tag{5}$$

Although most of the properties to be derived are valid for any set of additive and multiplicative operations, for simplicity we limit ourselves to bitwise addition (EXOR, denoted by $+$) and multiplication in GF($2^m$). If the array of tuples with indices in $\xi$ is denoted by $a_\xi$, we have

$$\theta : b = \theta(a) \Leftrightarrow b_\xi = C_\xi a_\xi \tag{6}$$

with $C_\xi$ a $n_\xi \times n_\xi$ matrix. The $j$-th column of $C_\xi$ is denoted by $C_{\xi|j}$. The inverse of $\theta$ is specified by the partition $\Xi$ and the matrices $C_\xi^{-1}$.

In $\mathcal{X}$, the partition $\Xi$ has two elements. $\theta$ could be defined as

$$\theta\left(\begin{bmatrix} a_0 & a_2 & a_4 \\ a_1 & a_3 & a_5 \end{bmatrix}\right) = \begin{bmatrix} 2a_0 + a_1 & a_2 + a_3 + a_4 & a_2 + a_4 + a_5 \\ a_0 + a_1 & a_3 + a_4 + a_5 & a_2 + a_3 + a_5 \end{bmatrix}.$$

In this case there are are two $C_\xi$-matrices:

$$C_{\xi(0)} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, \qquad C_{\xi(2)} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}.$$

*3.1.4. The round key addition $\sigma$* We have

$$\sigma[k^t] : b = \sigma[k^t](a) \Leftrightarrow b_i = a_i + k_i^t, \tag{7}$$

$\sigma$ is an involution: $\sigma^{-1}[k^t] = \sigma[k^t]$. The order of $\sigma$ and $\theta$ can be changed if the key value in $\sigma$ is appropriately adapted. From $\theta(a) + k^t = \theta(a + \theta^{-1}(k^t))$ we can derive (with $\circ$ denoting "after", the composition of transformations)

$$\sigma[k^t] \circ \theta = \theta \circ \sigma[\theta^{-1}(k^t)]. \tag{8}$$

### 3.2. *Table-Lookup implementation*

The round transformation lends itself to table-lookup implementations. If $b = \rho[k^t](a)$, we have

$$b_i = k_i^t + \sum_{j \in \xi(i)} C_{i,j} S_\gamma(a_{p(j)}), \quad i = 0, \ldots, n_t - 1, \tag{9}$$

or,

$$b_\xi = k_\xi^t + \sum_{j \in \xi} T_{\xi|j}[a_{p(j)}], \tag{10}$$

with the tables $T_{\xi|j}$ defined by $T_{\xi|j}[x] = C_{\xi|j} S_\gamma(x)$. There are $n_\xi$ tables for every subset $\xi$. Each table contains $2^m$ entries (one entry for each possible value of $a_{p(j)}$).

Each table entry corresponds to a column of $n_\xi$ $m$-bit tuples. The total number of memory bits taken by the tables is then

$$\sum_{i=1}^{n_\Xi} n_{\xi_i}^2 \cdot 2^m \cdot m. \tag{11}$$

The needed amount of table storage can be greatly reduced by restricting the number of different values for $C_{\xi|i,j}$.

By arranging the different components in the subsets $\xi$ in computer word units, the total number of table-lookups (and bitwise additions) for $\rho$ is given by $\sum_{i=1}^{n_\Xi} n_{\xi_i} = n_t$. If the computer word length is too small for the subsets $\xi$, the subsets can be further split up at the cost of additional table-lookups. All table-lookups can be done in parallel and even the bitwise addition can be partially parallelized.

### *3.3. The cipher structure*

An $r$-round cipher is defined by

$$\mu \circ \theta^{-1} \circ \rho[k^r] \circ \ldots \circ \rho[k^2] \circ \rho[k^1] \circ \sigma[k^0]. \tag{12}$$

The last round can be combined with $\theta^{-1}$ to give

$$\begin{aligned} \theta^{-1} \circ \rho[k^r] &= \theta^{-1} \circ \sigma[k^t] \circ \theta \circ \pi \circ \gamma \\ &= \sigma[\theta^{-1}(k^t)] \circ \pi \circ \gamma, \end{aligned}$$

i.e., the net effect is the removal of $\theta$ in the last round and a different definition of the last round key. The cancellation of $\theta$ in the last round ensures that the structure of the inverse cipher is the same as the structure of the cipher (cf. Section 3.4). This can be compared with the absence of the swap operation in the last round of Feistel ciphers.

### *3.4. The inverse cipher*

As we have shown, the cipher structure lends itself to efficient implementations. In this section we will show that the inverse cipher has the same structure if certain conditions are imposed on the transpositions $\pi$ and $\mu$.

The inverse of a two-round cipher, as defined above, is given by

$$\sigma[k^0] \circ \rho^{-1}[k^1] \circ \rho^{-1}[k^2] \circ \theta \circ \mu^{-1},$$

with

$$\rho^{-1}[k^t] = \gamma^{-1} \circ \pi^{-1} \circ \theta^{-1} \circ \sigma[k^t].$$

This can be expanded to

$$\sigma[k^0] \circ \gamma^{-1} \circ \pi^{-1} \circ \theta^{-1} \circ \sigma[k^1] \circ \gamma^{-1} \circ \pi^{-1} \circ \sigma[\theta^{-1}(k^2)] \circ \mu^{-1}$$

By using the commutativity between $\gamma$ and $\pi$ and applying (8) this is converted to

$$\sigma[k^0] \circ \pi^{-1} \circ \gamma^{-1} \circ \sigma[\theta^{-1}(k^1)] \circ \theta^{-1} \circ \pi^{-1} \circ \gamma^{-1} \circ \sigma[\theta^{-1}(k^2)] \circ \mu^{-1}$$

We now impose that

$$\pi^{-1} = \mu^{-1} \circ \pi \circ \mu. \tag{13}$$

In Section 5 we will impose conditions on $\pi$ to optimize the cipher's resistance against linear and differential cryptanalysis. Equation (13) should therefore be seen as a condition for $\mu$, given a certain $\pi$. It can be seen that for every $\pi$ there is at least one solution for $\mu$ (cf. Appendix A). We define $\theta'$ as

$$\theta' = \mu \circ \theta^{-1} \circ \mu^{-1}. \tag{14}$$

By using (13), (14) and the commutativity between $\gamma$ and $\mu$, the expression of the inverse cipher is converted into

$$\mu^{-1} \circ \sigma[\mu(k^0)] \circ \pi \circ \gamma^{-1} \circ \sigma[\mu(\theta^{-1}(k^1))] \circ \theta' \circ \pi \circ \gamma^{-1} \circ \sigma[\mu(\theta^{-1}(k^2))].$$

If the round transformation of the inverse cipher is defined as

$$\rho'[\kappa^t] = \sigma[\kappa^t] \circ \theta' \circ \pi \circ \gamma^{-1}, \tag{15}$$

the inverse cipher can be expressed as

$$\mu^{-1} \circ \theta'^{-1} \circ \rho'[\kappa^2] \circ \rho'[\kappa^1] \circ \sigma[\kappa^0]$$

with the inverse round keys given by

$$\kappa^t = \mu(\theta^{-1}(k^{r-t})).$$

The round transformation of the inverse cipher has the same structure as $\rho$ itself if the partition $\Xi'$ corresponding to $\theta'$ is equal to the partition $\Xi$ corresponding to $\theta$. In this case $\rho'$ can be implemented in the same ways as $\rho$ itself. In the case of the table-lookup implementations, the only difference is the contents of the tables.

This additional constraint boils down to the fact that $m(i)$ (which defines $\mu$) leaves the partition $\Xi$ invariant, i.e.,

$$\forall i, j \in \mathcal{I} : j \in \xi(i) \Rightarrow m(j) \in \xi(m(i)). \tag{16}$$

## 4. Differential and Linear Cryptanalysis

The feasibility and workload of a differential attack depends strongly on the probability of its underlying *characteristic* [1]. A *characteristic* consists of a difference propagation along the rounds of an iterated block cipher and is specified by a series of difference patterns. The *probability* of a characteristic is the average probability over all possible round keys and plaintexts that all intermediate difference patterns have the value specified in the above series. For a Markov cipher [10] with independent round keys the probability of a characteristic is equal to the product of the difference propagation probabilities between every pair of subsequent rounds (which can be easily calculated).

### *4.1. Propagation of difference patterns*

A difference pattern is defined as the bitwise difference, or exor, of two states: $a' = a \oplus a^*$. The analysis that is presented here, can easily be extended towards other definitions of a difference. Let $a'_i$ be the tuples of $a'$. We call the non-zero tuples $a'_i$ the *active tuples.*

*4.1.1. Linear and affine transformation* For a transformation $b = \lambda(a)$ linear over GF(2) the output difference corresponding to an input difference $a'$ is given by

$$b' = \lambda(a) \oplus \lambda(a^*) = \lambda(a \oplus a^*) = \lambda(a'). \tag{17}$$

This can be applied to $\pi$, $\theta$ and $\mu$. For $\sigma[k^t]$ it can easily be seen that the output difference pattern is equal to the input difference pattern (obviously with probability 1).

*4.1.2. The transformation $\gamma$* Since $\gamma$ operates on the individual tuples $a_i$

$$\mathrm{P}(b'|a') = \prod_i \mathrm{P}(b'_i|a'_i), \tag{18}$$

i.e., the probability of the difference propagation from $a'$ to $b'$ through $\gamma$ is given by the product of the probabilities for the individual tuples.

### *4.2. On multiple-round characteristics*

For a Markov cipher with independent round keys, the probability of a multiple-round characteristic is given by

$$\prod_t \mathrm{P}(b'^t|a'^t) = \prod_t \prod_i \mathrm{P}(b'^t_i|a'^t_i), \tag{19}$$

with $a'^t$ the intermediate difference patterns at the input of round $t$ and $a'^{t+1} = \theta(\pi(b'^t))$. (The key addition $\sigma$ doesn't change the difference pattern.)

Equation (19) is the product of the difference propagation probabilities of all the S-boxes in the rounds. An S-box in the computational graph is called active in a characteristic if the corresponding input tuple is active. Since $\mathrm{P}(b'_i|a'_i) = 1$ if $a'_i = b'_i = 0$, (19) can be interpreted as the *product of the probabilities of the active S-boxes only.*

### *4.3. Preventing high-probability characteristics*

Assume that all difference propagation probabilities of $S_\gamma$, are below some value $p$. Methods for constructing the $m$-bit substitution $S_\gamma$ are out of the scope of this paper but can be found in cryptographic literature [9, 14]. In [14] constructions are shown with $p = 2^{2-m}$. Now an upper bound for the probability of a characteristic is given by $p^w$ where $w$ is the number of active S-boxes in the characteristic. It follows that the characteristics with the highest bounds on their probability are the ones with the least number of active S-boxes.

In our study, we only distinguish between active and non-active S-boxes, without specifying the exact difference patterns. The *tuple weight* $w_t(a')$ of a difference pattern $a'$ is equal to the number of active tuples in $a'$. The number of active S-boxes in a round is equal to the tuple weight of its input difference. This means that the sum of the tuple weights of the input differences over the rounds determines the number of active S-boxes.

Figure 2 gives a schematic representation of the different transformations of a round. As $\gamma$ and $\sigma[k]$ operate on each tuple individually, they do not affect the propagation of the difference patterns and can be left out in the future discussions.

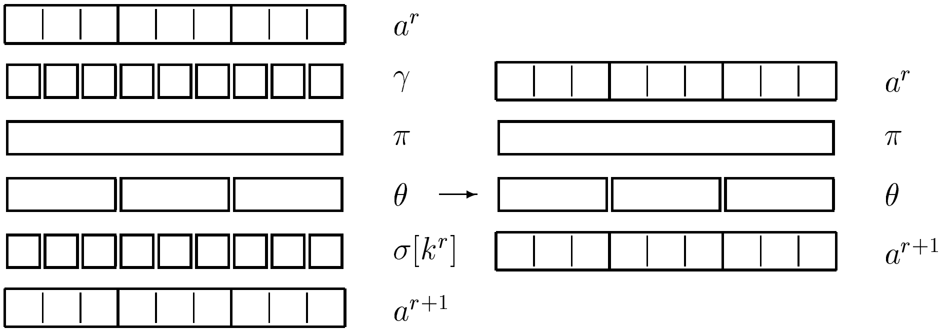

*Figure 2.* Schematic representation of the different transformations of a round. Since $\gamma$ and $\sigma$ have no influence on the propagation of a difference pattern, they can be ignored in the discussion.

## *4.4. Linear cryptanalysis*

The feasibility and workload of linear attacks depends strongly on the probability of the underlying *linear approximation* [13], or the correlation between input bits and output bits of the cipher.

A *linear approximation* is specified by a series of selection vectors. For a given cipher key, the correlation (positive or negative) corresponding to a linear approximation can be approximated by the product of the correlations between the linear combinations of bits of every pair of subsequent rounds.

The propagation of selection patterns through the different transformations of the ciphers can be treated in a completely analogous way as the propagation of difference patterns [2, 13].

Let a linear combination of state bits be specified by a selection vector $v$. If a state $a$ is considered as a binary vector with the bits as components, the linear combination corresponding with $v$ is expressed as $v^T a$, where the '$^T$' suffix denotes vector transposition.

A transformation $\lambda$, linear over GF(2), can be described by $\lambda(a) = Ma$ with $M$ a binary matrix. We have $v^T \lambda(a) = v^T M^T a = (M^T v)^T a$. Hence, defining $\lambda^T$ as $\lambda^T(a) = M^T a$, the relation between an input selection vector $u$ and output selection vector $v$ is

$$u = \lambda^T(v). \tag{20}$$

This reasoning applies to the linear transformations $\pi, \mu$ and $\theta$ and it can be extended to the affine transformation $\sigma[k]$. Equation (20) is analogous to (17), except for the vector transposition.

For the nonlinear transformation $\gamma$ we have that the input-output correlation is given by the product of the input-output correlations the individual tuples and we get an expression that is analogous to (18). Also, the correlation between input and output of a multiple-round linear approximation can be expressed with a formula like (19) (under similar assumptions of independent round keys). We prevent high-correlation linear approximations in a similar way as we prevent high-probability differential characteristics: by upper bounding the input-output correlation of an active S-box and lower bounding the number of active S-boxes in an arbitrary linear approximation.

In the next section we will develop conditions to guarantee a minimum number of active S-boxes in multiple-round characteristics. The results can easily be translated to the case of linear approximations, because the selection vectors of a linear approximation propagate in a way analogous to the propagation of differences: for the nonlinear transformations both situations are exactly alike, for the linear transformations the difference lies in the presence or absence of a vector transposition.

## 5. Imposing Conditions on $\theta$ and $\pi$

In this section we will discuss conditions on $\theta$ and $\pi$ that impose strict lower bounds on the number of active S-boxes in multiple-round characteristics and linear approximations.

A subset $\xi$ (for a definition of subsets, cf. Section 3) in a difference pattern is called active if any of the tuples in $\xi$ is active. The number of active subsets in a difference pattern, its so-called *subset weight*, is denoted $w_s(a')$. Since $\theta$ operates on each subset separately, the subset weight is invariant under $\theta$. Since $\pi$ is a transposition of the tuples, it leaves the tuple weight invariant. Let us illustrate this with a few examples for the cipher $\mathcal{X}$, defined in Section 3. If only $a_0$ and $a_2$ are active, $w_t(a') = 2$, and since $\xi(0) \neq \xi(2)$, $w_s(a') = 2$. If only $a_2$ and $a_3$ are active, $w_s(a') = 1$, because $\xi(2) = \xi(3)$; no tuple of the subset $\xi(0)$ is active.

### *5.1. Bounding the worst-case diffusion of $\boldsymbol{\theta}$*

$\theta$ is the single transformation in $\rho$ that provides inter-tuple *diffusion*. Intuitively, good diffusion means that $\theta$ (and also $\theta^{-1}$) converts difference patterns with low tuple weight to difference patterns with a high tuple weight. A good measure for the worst-case diffusion of $\theta$ is given by its *differential branch number* $\mathcal{B}(\theta)$.

*Definition 1.* The differential branch number of a linear transformation $\theta$ is

$$\mathcal{B}(\theta) = \min_{a \neq 0}(w_t(a) + w_t(\theta(a))) \,. \tag{21}$$

We will write $\mathcal{B}$ instead of $\mathcal{B}(\theta)$ when it is clear from the context which $\theta$ is involved.

Since $\theta$ operates on the tuples in separate subsets $\xi$, the differential branch number of $\theta$ is given by the minimum differential branch number of the transformations given by $C_\xi$. An upper bound for the differential branch number of $C_\xi$ is given by $n_\xi + 1$, since the output difference corresponding to an input difference with a single non-zero tuple in $\xi$ can have at most tuple weight $n_\xi$. Therefore, the differential branch number of $\theta$ is upper bounded by

$$\mathcal{B}(\theta) \leq \min_\xi n_\xi + 1. \tag{22}$$

We now have

Theorem 1 *For any two consecutive rounds the sum of the number of active tuples over the two rounds is lower bounded by $N\mathcal{B}$, where $N$ is the number of active subsets at the input of the second round.*

**Proof:** Figure 3 depicts two rounds. It is clear that $w_t(a^1) + w_t(a^2)$ is only bounded by the properties of the linear transformations $\pi$ and $\theta$ of the first round.

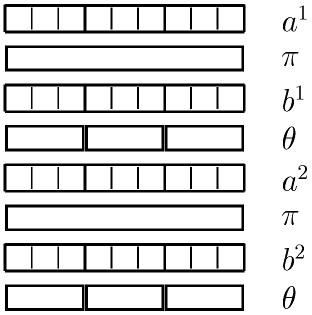

*Figure 3.* Transformations relevant in the proof of Theorem 1.

Consider $a^2$, the difference pattern at the input of the second round. Definition 1 implies that the sum of the active tuples before and after $\theta$ of the first round is lower bounded by $\mathcal{B}$. Since $\theta$ operates on the tuples in separate subsets the number of active subsets is not changed by $\theta$ and the lower bound holds for each active subset separately. If there are $N$ active subsets at the beginning of the second round, we have that $w_t(b^1) + w_t(a^2) \geq N\mathcal{B}$. Since $\pi$ leaves the number of active tuples invariant, $w_t(b^1) = w_t(a^1)$. ■

For the case of linear approximations, we obtain similar results using the *linear branch number* of the mapping $\theta$, which corresponds to $\min_{a\neq 0}(w_t(a) + w_t(\theta^T(a)))$.

*5.1.1. Branch Numbers and Linear Codes* For a detailed discussion of linear codes, we refer the reader to [11].

*Definition 2.* A *linear* $[n, k, d]$*-code* $\mathcal{C}$ over $\mathrm{GF}(2^m)$ is a $k$-dimensional subspace of the vector space $(\mathrm{GF}(2^m))^n$, where any two different vectors of the subspace have a Hamming distance of at least $d$ ($d$ is the largest number with this property).

A linear code $\mathcal{C}$ can be described by a *generator matrix* $G$, which is a $k \times n$ matrix whose rows form a vector space basis for the code. Since the choice of a basis in a vector space is not unique, a code has many different generator matrices that can be reduced to one another by elementary row operations and column permutations. The *echelon form* of the generator matrix is the following:

$$G_e = \begin{bmatrix} I_{k\times k} & C_{k\times(n-k)} \end{bmatrix} . \tag{23}$$

The *dual code* $\mathcal{C}^\perp$ of a code $\mathcal{C}$ is defined as the set of vectors that are orthogonal to all the vectors of $\mathcal{C}$:

$$\mathcal{C}^\perp = \{x \mid < x, y >= 0, \forall y \in \mathcal{C}\}.$$

The differential branch number of a linear mapping can be related to the minimal distance of the associated linear code.

*Definition 3.* Let $\theta$ be a linear mapping from $(\mathrm{GF}(2^m))^{n_t}$ to $(\mathrm{GF}(2^m))^{n_t}$. The *associated code* of $\theta$, $\mathcal{C}_\theta$, is the linear code that has codewords given by the vectors $(x \| \theta(x))$. The code $\mathcal{C}_\theta$ has $2^{n_t}$ codewords and has length $2n_t$.

It follows from Definition 1 that the differential branch number of a mapping equals the minimal distance between two different codewords of its associated code. The upper bound on the differential branch number of a mapping, given in (22), corresponds to the well-known Singleton bound on the minimal distance of a (linear) code.

This relation between linear transformations and linear codes allows us to construct efficiently mappings with a high differential branch number. Given a linear code $\mathcal{C}_\theta$, the associated mapping $\theta$ is given by

$$\theta : x \mapsto \theta(x) = x \cdot C, \tag{24}$$

where $C$ is found from the echelon form of the generator matrix of $\mathcal{C}$.

It can be proven that the linear branch number of a mapping $\theta$ is equal to the minimal distance of the dual code of $\mathcal{C}_\theta$.

### *5.2. A maximum-diffusion condition on* $\boldsymbol{\pi}$

$\pi$ is the single transformation in $\rho$ that provides inter-subset *diffusion.* In this case, good diffusion means that $\pi$ distributes the different tuples of a subset to as many different subsets as possible.

*Definition 4.* We say $\pi$ is *diffusion-optimal* if the different tuples in a subset are distributed over $n_\xi$ different subsets. More formally, we have

$$\forall i, j \in \mathcal{I}, i \neq j : (\xi(i) = \xi(j)) \Rightarrow (\xi(p(i)) \neq \xi(p(j))). \tag{25}$$

It is easy to see that this implies the same condition for $\pi^{-1}$. A diffusion-optimal transformation $\pi$ implies $w_s(\pi(a)) \geq \max_\xi(w_t(a_\xi))$. Therefore a diffusion-optimal transformation can only exist if $n_\Xi \geq \max_i(n_{\xi_i})$. In words, $\pi$ can only be diffusion-optimal if there are at least as many subsets as there are tuples in the largest subset. We can now prove the following theorem:

Theorem 2 *If $\pi$ is diffusion-optimal, in any four consecutive rounds the total number of active S-boxes is lower bounded by $\mathcal{B}^2$.*

**Proof:** Figure 4 depicts three complete rounds. By applying Theorem 1 on the first two rounds (1 and 2) and on the last two rounds (3 and 4), the number of active tuples over the four rounds is lower bounded by $(w_s(a^2) + w_s(a^4))\mathcal{B}$. We will now prove that

$$w_s(a^2) + w_s(a^4) = w_s(a^2) + w_s(b^3) \geq \mathcal{B}.$$

Since $\pi$ is diffusion-optimal, for each subset of $b^2$ it holds that all its tuples come from different subsets of $a^2$. A fortiori, the number of active tuples in each subset of $b^2$ is upper bounded by the number of active subsets in $a^2$, i.e. $w_s(a^2)$. At least one

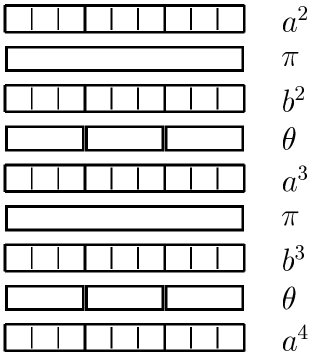

*Figure 4.* Relevant transformations for the proof of Theorem 2.

subset $\xi$ of $b^2$ is active. We denote by $\delta$ the number of active tuples in $\xi$, $\delta \leq w_s(a^2)$. From Definition 1 it follows that the corresponding subset in $a^3$ has at least $\mathcal{B} - \delta$ active tuples. Therefore $w_t(a^3) \geq \mathcal{B} - \delta \geq \mathcal{B} - w_s(a^2)$. Since $\pi$ is diffusion-optimal, $w_s(b^3) \geq w_t(a^3)$. Combining these inequalities proves the theorem. ■

The total number of active S-boxes over four rounds is upper bounded by $4n_t$. One can wonder how the lower bound of Theorem 2 relates to this upper bound. From (22) we have that $\mathcal{B}^2 \leq \min(n_\xi + 1)^2 = \min n_\xi^2 + 2 \min n_\xi + 1$. Diffusion-optimality implies that $\min(n_\xi + 1)^2 \leq \min n_\xi n_\Xi + 2 \min n_\xi + 1 \leq n_t + 2n_t + n_t = 4n_t$.

## 6. Existing Designs

According to the designers of SAFER [12], the Pseudo-Hadamard Transform (PHT) “provides guaranteed complete diffusion.” Every output byte depends on every input byte. However, there is one output bit that depends on only one input bit [18]. This means that the branch number of the PHT is two. The choice of exponentiation S-boxes ensures that this imperfection of the diffusion cannot be used in a linear attack.

The stream cipher TWOPRIME [5] uses two linear layers. The first linear layer has the form

$$Y_j = \sum_{i=0}^{7} X_i - X_j, j = 0, 1, \ldots, 7$$

(here $\sum$ denotes byte addition). The authors observe that a change of one input byte leads to a change of seven output bytes. However, it is easy to see that the change of two input bytes may lead to a change of only two output bytes. The branch number is four.

In a general paper about avalanche characteristics in S-P-Networks [17], it is suggested to use a linear transformation described by

$$z(i) = \bigoplus_{l=1, l \neq i}^{M} w(l), 1 \leq i \leq M,$$

where $w(i)$ represents an input word and $z(i)$ represents an output word of the linear transformation. Again it can be seen that while the change of one input word affects $M-1$ output words, equal changes in two input words will cancel each other in all but two output words.

In all three cases the described diffusion is suboptimal in the sense that a much higher branch number could be achieved. The additional cost that is caused by the more complex diffusion of our constructions can often be reduced by incorporating the diffusion layer into the table lookup operation of the nonlinear substitution, as was shown in Section 3.2.

## 7. Concrete Constructions

In the remaining part of the paper we will present several concrete constructions for the transformations $\theta$ and $\pi$ and its implications with respect to characteristics and linear approximations. Also a dedicated attack is presented.

### *7.1. Constructions for $\boldsymbol{\theta}$*

For memory efficient implementations all subsets $\xi$ of the partition $\Xi$ preferably have the same size. The fact that the branch number is upper bounded by the smallest subset (equation (22)) points in the same direction. Hence we will consider in the following only the case where all subsets have the same size.

Additionally we can impose the requirement that $\theta$ acts in the same way on each partition element. In this case all $C_\xi$ are equal and the sum (11) reduces to one term. As already mentioned, it is possible to reduce memory requirements even further by restricting the number of different values for $C_{\xi|i,j}$.

Using (23) and (24) we can construct mappings with a maximal branch number from a Maximum-Distance Separable code (MDS-code). Alternatively, we can search directly for the matrix $C_\xi$. The following theorem translates a property of MDS-codes given in [11] into a condition for which $\mathcal{B}(\theta)$ reaches the upper bound given in (22).

Theorem 3 *The differential branch number $\mathcal{B}(\theta)$ equals $n_\xi + 1$ if and only if all square submatrices of $C_\xi$ are nonsingular.*

It can be proven that also the linear branch number of $\theta$ equals $n_\xi + 1$ under this condition. Thus, for mappings with maximal branch number, we don't need to distinguish between the differential and linear branch number. A mapping with maximal branch number is called an $(n_\xi, n_\xi)$-multipermutation in [18].

### *7.2. Choices for the structure of $\mathcal{I}$ and $\pi$*

We present two general structures for $\mathcal{I}$ and $\pi$. In the first structure the different tuples of a state can be seen as arranged in a multidimensional regular array or hypercube of dimension $d$ and side $n_\xi$. Ciphers constructed in this way have a block size of $mn_\xi^d$. In the second structure the tuples of a state are arranged in a rectangle with one side equal to $n_\xi$. This gives more freedom for the choice of the block size of the cipher.

*7.2.1. The hypercube structure* In this construction the subsets $\xi$ correspond to parallel one-dimensional columns in the array. The transformation $\pi$ corresponds to a rotation of the hypercube around a diagonal axis (called the $p$-axis). The transformation $\mu$ corresponds to a reflection with respect to a 2-dimensional hyperplane through the $p$ axis and parallel to the $\xi$ sets.

The indices $i \in \mathcal{I}$ are represented by a vector of length $d$ and elements $i_j$ between 0 and $n_\xi - 1$. We have

$$i = (i_0, i_1, \ldots, i_{d-1}).$$

The subsets $\xi$ are given by

$$j \in \xi(i) \text{ if } j_1 = i_1, j_2 = i_2, \ldots \text{ and } j_{d-1} = i_{d-1}.$$

$p(i)$, defining $\pi$, is given by

$$p : j = p(i) \Leftrightarrow (j_0, j_1, \ldots, j_{d-2}, j_{d-1}) = (i_1, i_2, \ldots, i_{d-1}, i_0).$$

$m(i)$, defining $\mu$, is given by

$$m : j = m(i) \Leftrightarrow (j_0, j_1, \ldots, j_{d-2}, j_{d-1}) = (i_0, i_{d-1}, \ldots, i_2, i_1).$$

It can easily be seen that this construction fulfills (13), that $\mu$ preserves $\Xi$ and that $\pi$ is diffusion-optimal (if $d > 1$). We will briefly illustrate this for $d$ equal to 1, 2 and 3.

**Dimension 1:**

Dimension 1 is a degenerate case because the partition counts only one subset, and $\pi$ cannot be diffusion-optimal. SHARK [15] is an example where $n_t = n_\xi = 8$ and $m = 8$, resulting in a block size of 64 bits.

**Dimension 2:**

Figure 5 shows the two-dimensional array, the transposition $\pi$, and the partition $\Xi$. The transposition $\mu$ is reduced to the identical transformation for $d = 2$.

The two-dimensional structure is adopted in SQUARE [4], with $m = 8$ and $n_\xi = 4$, resulting in a block cipher with a block size of 128 bits in which every four-round differential characteristic or linear approximation has at least $\mathcal{B}^2 = 25$ active S-boxes. Using S-boxes with maximal $p$- and $c$-values as given in Section 4.3, this results in an upper bound of $2^{-150}$ for the probability of a four-round differential characteristic, (and an upper bound of $2^{-75}$ for the correlation of a four-round relation).

**Dimension 3:**

For dimension three, with $n_\xi = 2$ and $m = 8$, we get a 64-bit cipher that has some similarity to SAFER [12], however the round function of SAFER actually looks more like a triple application of $\theta \circ \pi$ for every application of $\gamma$. Therefore

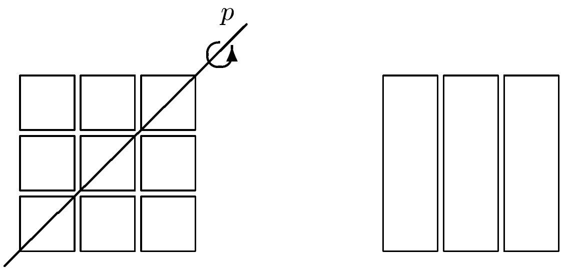

*Figure 5.* The arrangement of the tuples (left) and subsets (right) for the hypercube structure with $d = 2$ and $n_\xi = 3$. The $p$-axis is indicated at the left.

SAFER also can (almost) be seen as an example of a cipher with a diffusion layer of dimension 1.

Theorem 2 guarantees for our constructions a lower bound on the number of active S-boxes per four rounds of 9. For differentials or linear approximations of more than four rounds, the minimum number of active S-boxes per round rises significantly: it can be shown that after six rounds for instance there are already minimum 18 active S-boxes.

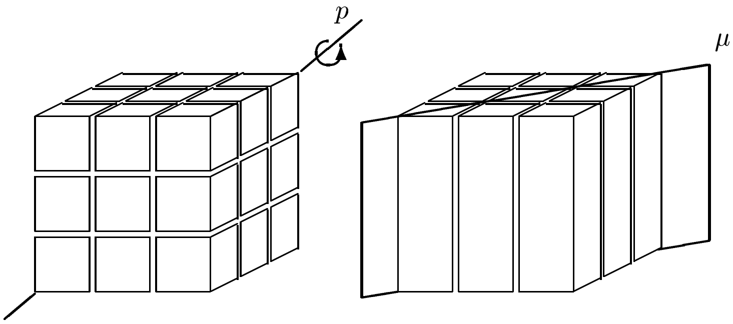

*Figure 6.* The arrangement of the tuples (left) and subsets (right) for the hypercube structure with $d = 3$ and $n_\xi = 3$. The $p$-axis is indicated at the left, the reflection plane of $\mu$ at the right.

*7.2.2. The rectangular structure* In this construction the subsets $\xi$ still correspond to parallel one-dimensional columns in the array. The other dimensions of the array are determined from the required block size of the cipher (cf. Section 3.1). Figure 7 shows an example.

The transformation $\pi$ corresponds to a cyclic shift of the elements of the rows. If every row is shifted by a different number of tuples, the diffusion of $\pi$ is optimal. (Note that this is only possible if $n_\Xi \geq n_\xi$, i.e., if the number of rows is at most the number of columns) The transformation $\mu$ inverts the order of the columns.

If a tuple has $m = 8$ bits and every subset contains $n_\xi = 4$ tuples, then setting the number of subsets $n_\Xi$ to 4, 6 or 8 gives a block size of 128, 192 or 256 bits respectively. It is possible to extend this structure to constructions in more dimensions, e.g., if $d = 3$, the result will be a *cuboid.*

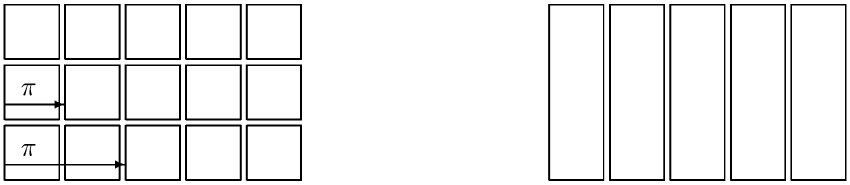

*Figure 7.* The arrangement of the tuples (left) and subsets (right) for the rectangular structure (in two dimensions) with $n_\xi = 3$ and $n_\Xi = 5$. The transformation $\pi$ leaves the first row invariant, shifts the second row by one position and the third row by two positions.

## 8. A Dedicated Attack

In this section we generalize the attack as described in [4] to the general cipher structure and the concrete constructions as presented in this paper. The attack is a chosen plaintext attack and is independent of the specific choices of $S_\gamma$, $\theta$ and the key schedule. After describing the basic attack, we will show how it can be extended at the beginning and the end.

### *8.1. Preliminaries*

Let a $\Lambda$-set be a set of $2^m$ states that are all different in the state tuples in some positions (the *active*) and all equal in the state tuples in other positions (the *passive*).

Applying the transformations $\gamma$ and $\sigma[k^t]$ on (the elements of) a $\Lambda$-set results in a (generally different) $\Lambda$-set with the same active positions. Applying $\pi$ results in a $\Lambda$-set with the active positions transposed by $p()$.

Applying $\theta$ to a $\Lambda$-set does not necessarily result in a $\Lambda$-set.

If the input $\Lambda$-set has only a single active position per subset $\xi$, the output is a $\Lambda$-set. Assume that position $k$ of the input $\Lambda$-set is active and all other positions in $\xi$ are passive. We have:

$$b_i = C_{i,k} a_k + \sum_{j \neq k} C_{i,j} a_j ,$$

The rightmost term in this expression has the same value for all elements of the $\Lambda$-set. If $C_{i,k} = 0$ it follows that $b_i$ is the same for all output vectors, i.e., $i$ is a passive position. If not, there will be no two output vectors with the same value for $b_i$ since

$$b_i = b'_i \Leftrightarrow C_{i,k} a_k = C_{i,k} a'_k \Leftrightarrow C_{i,k}^{-1} C_{i,k} a_k = C_{i,k}^{-1} C_{i,k} a'_k \Leftrightarrow a_k = a'_k ,$$

and so $i$ is an active position. It follows that if a $\Lambda$-set with not more than one active position per $\xi$ is input to $\theta$, the output is a $\Lambda$-set.

Moreover, in all cases, the output set corresponding to a $\Lambda$-set is balanced for every position. We have:

$$\sum_{a \in \Lambda} b_i = \sum_{a \in \Lambda} \sum_j C_{i,j} a_j = \sum_j C_{i,j} \sum_{a \in \Lambda} a_j = \sum_j C_{i,j} 0 = 0.$$

### *8.2. The generic propagation structure*

Consider a $\Lambda$-set with a single active position. We will trace the evolution of this set through three rounds. The transformations $\sigma$ and $\gamma$ of the first round leave the position of the active position invariant. Then $\pi$ simply transposes the active position. After the application of $\theta$, the active positions are confined to a single $\xi$-set. This is still the case at the input of $\pi$ of the second round. Because of the diffusion-optimality of $\pi$, after its application there is no subset $\xi$ with more than 1 active position. Hence, after application of $\theta$ it is still a $\Lambda$-set. This is still the case at the input of $\theta$ of the third round. Since in general the subsets $\xi$ may contain more than one active position, the output will no longer be a $\Lambda$-set. However, as shown above, the tuples of the output of $\theta$ will be balanced over the $\Lambda$-set. This will still be the case after applying $\sigma$ at the end of the third round.

*8.2.1. Extension for the hypercube and rectangular structure* In the hypercube structure, the active position gives rise to a line of active positions after one round, a plane of active positions after two rounds, a hyperplane of dimension 3 after three rounds, and so on. Hence if the dimension of the hypercube is $d$, the generic propagation structure can be stretched to $d + 1$ rounds instead of just 3 for $d > 2$.

If the number of columns in the rectangular structure is at most twice the number of rows and the shift constants of $\pi$ have been chosen in a careful way, the generic propagation structure cannot be stretched to a higher number of rounds.

### *8.3. The basic attack*

In the basic attack, we assume a cipher that has a number of rounds equal to the range of the propagation structure plus 1. This is 4 in the general case, $d + 2$ in the case of the hypercube structure and 5 in the case of the rectangular structure.

As $\theta$ is not present in the last round, an output tuple $b_i$ of the cipher depends on only one input tuple $a_{p(i)}$ of the last round:

$$b_i = S_\gamma[a_{p(i)}] \oplus \kappa'_i,$$

with $\kappa'_i$ a key tuple. By assuming a value for $\kappa'_i$, the value of $a_{p(i)}$ for all elements of the $\Lambda$-set can be calculated from the ciphertexts. If the values of this tuple are not balanced over $\Lambda$, the assumed value for the key tuple was wrong. This is expected to eliminate all but approximately 1 key value. This can be repeated for every balanced tuple at the end of the last-but-one round. For the general case and the hypercube case, this allows the reconstruction of the full key of the last round.

### *8.4. Extension by Rounds at the End*

If an additional round is added, the above value of $a_{p(i)}$ corresponds to the output of the last-but-two rounds instead of the last-but-one round. This can be done by additionally assuming a value for a set of $n_\xi$ tuples of the last round key. As in the case of the basic attack, wrong key assumptions are eliminated by verifying that $a_{p(i)}$ is not balanced. More rounds may be added at the cost of extra key tuples that must be guessed.

If the extension is by $r$ rounds, a value for $\omega = \sum_{0 \leq i \leq r} n_\xi^i$ key tuples ($m$ bits each) must be assumed at the same time. In most concrete designs, an extension by more than one round requires too much key bits to be guessed.

Since checking a single $\Lambda$-set leaves only $2^{-m}$ of the wrong key assumptions as possible candidates, finding the correct value takes about $\omega$ $\Lambda$-sets.

### *8.5. Extension by Rounds at the Beginning*

The basic idea is to choose a set of plaintexts that results in a $\Lambda$-set at the input of some intermediate round with a single active tuple. If the extension is by $r$ rounds, this requires the additional assumption of values for $\omega' = \sum_{1 \leq i \leq r} n_\xi^i$ key tuples.

If the extension is only by a single round, the ciphertexts corresponding to a chosen set of $2^{mn_\xi}$ plaintexts must be known.

By making an assumption for the relevant key tuples, from this set a subset can be composed that gives rise to a $\Lambda$-set with a single active tuple at the input of the second round. Extensions at the beginning and at the end may be combined in the same attack at the cost of augmenting the number of key tuples to $m(1 + \omega + \omega')$ and the number of chosen plaintexts to $2^{mn_\xi^r}$ with $r$ the number of rounds of the extension at the beginning.

## 9. Conclusions

In this paper we have presented a construction for a general block cipher where the diffusion of the linear components serves as a lever to enhance the effectiveness of the nonlinear substitution boxes. With appropriate choices of the nonlinear substitution boxes, our constructions lead to efficient block ciphers with resistance against statistical attacks, such as differential and linear attacks. The constructions are very flexible make it possible to construct block ciphers with many different block and key lengths by varying certain parameters. Moreover, the presented block cipher construction can make optimally use of a wide range of processor word lengths and its parallelism allows very fast implementations.

We have also presented a dedicated attack. For ciphers constructed according to the generic structure, this attack will typically break 6 rounds. For the hypercube structure with dimension $d$, the attack breaks $3d$ rounds. In the rectangular structure, the attack will not break more than 6 rounds, if the row shifts have been carefully chosen and the number of columns is at most twice the number of rows.

## References

1. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard,* Springer-Verlag, 1993.
2. J. Daemen, R. Govaerts and J. Vandewalle, “Correlation matrices,” *Fast Software Encryption, LNCS 1008*, B. Preneel, Ed., Springer-Verlag, 1995, pp. 275–285.
3. J. Daemen, V. Rijmen, “The block cipher Rijndael,” available from NIST’s AES homepage, `http://www.nist.gov/aes/`.
4. J. Daemen, L. Knudsen, V. Rijmen, “The block cipher Square,” *Fast Software Encryption, LNCS 1267*, E. Biham, Ed., Springer-Verlag, 1997, pp. 149–165.
5. C. Ding, V. Niemi, A. Renvall and A. Salomaa, “TWOPRIME: a fast stream ciphering algorithm,” *Fast Software Encryption, LNCS 1267*, E. Biham, Ed., Springer-Verlag, 1997, pp. 88–102.
6. H. Feistel, “Cryptography and computer privacy,” *Scientific American*, Vol. 228, No. 5, May 1973, pp. 15–23.
7. H.M. Heys and S.E. Tavares, “Substitution-permutation networks resistant to differential and linear cryptanalysis,” *Journal of Cryptology*, Vol. 9, No. 1, 1996, pp. 1–20.
8. T. Jakobsen and L.R. Knudsen, “The interpolation attack on block ciphers,” *Fast Software Encryption, LNCS 1267*, E. Biham, Ed., Springer-Verlag, 1997, pp. 28–40.

9. K. Kim, T. Matsumoto and H. Imai, "A recursive construction method of S-boxes satisfying strict avalanche criterion," *Advances in Cryptology, Proc. Crypto'90, LNCS 537,* S. Vanstone, Ed., Springer-Verlag, 1991, pp. 564–575.

10. X. Lai, J.L. Massey and S. Murphy, "Markov ciphers and differential cryptanalysis," *Advances in Cryptology, Proceedings Eurocrypt'91, LNCS 547*, D.W. Davies, Ed., Springer-Verlag, 1991, pp. 17–38.

11. F.J. MacWilliams and N.J.A. Sloane, *"The Theory of Error-Correcting Codes,"*, North-Holland, Amsterdam, 1977.

12. J. Massey, "SAFER-K64, a byte-oriented block-ciphering algorithm," Fast Software Encryption, LNCS 809, R. Anderson, Ed., Springer-Verlag, 1994, pp. 1–17.

13. M. Matsui, "Linear cryptanalysis method for DES cipher," *Advances in Cryptology, Proceedings Eurocrypt'93, LNCS 765*, T. Helleseth, Ed., Springer-Verlag, 1994, pp. 386–397.

14. K. Nyberg, "Differentially uniform mappings for cryptography," *Advances in Cryptology, Proc. Eurocrypt'93, LNCS 765,* T. Helleseth, Ed., Springer-Verlag, 1994, pp. 55-64.

15. V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, E. De Win, "The cipher SHARK," *Fast Software Encryption, LNCS 1039*, D. Gollmann, Ed., Springer-Verlag, 1996, pp. 99–111.

16. C.E. Shannon, "Communication theory of secrecy systems," *Bell System Technical Journal*, No. 30, 1949, pp. 50–64.

17. S. Tavares and A.M. Youssef, "On the avalanche characteristics of substitution permutation networks," *Proceedings of Pragocrypt '96,* J. Přibyl, Ed., CTU Publishing House (Prague), 1996, pp. 18–29.

18. S. Vaudenay, "On the need for multipermutations: cryptanalysis of MD4 and SAFER," *Fast Software Encryption, LNCS 1008*, B. Preneel, Ed., Springer-Verlag, 1995, pp. 286–297.

19. A.M. Youssef, S. Mister and S.E. Tavares, "On the design of linear transformations for substitution permutation encryption networks," *Records of the Workshop on Selected Areas in Cryptography (SAC '97)*, Ottawa, August 11–12, 1997, pp. 40–48..

## Appendix A

## Construction of $\mu$

The transposition $\mu$ is defined by (13):

$$\pi^{-1} = \mu^{-1} \circ \pi \circ \mu.$$

A solution for $\mu$ can be constructed as follows. Decompose the transposition $\pi$ into its cycles.

$$\pi = (i_{j_0} i_{j_1} \ldots i_{j_k}) \circ (i_{j_{k+1}} i_{j_{k+2}} \ldots i_{j_l}) \circ \ldots \circ (i_{j_p} i_{j_{p+1}} \ldots i_{j_{n_t - 1}})$$

The inverse of $\pi$ is then composed from the same cycles, but read from right to left instead of from left right. Now, let $\mu$ be the transposition that 'mirrors' the elements from every cycle ($\mu = \mu^{-1}$). It is easy to see that the succession of steps 'mirror', 'shift right', 'mirror' corresponds to 'shift left', or $\mu^{-1} \circ \pi \circ \mu = \pi^{-1}$. Depending on the cycle decomposition of $\pi$, other solutions for $\mu$ might be possible.