Constructing Lightweight Optimal Diffusion Primitives with Feistel Structure

Zhiyuan Guo[1,2,3](✉), Wenling Wu[1,2,3], and Si Gao[1,2,3]

[1] TCA Laboratory, SKLCS, Institute of Software,
Chinese Academy of Sciences, Beijing, China
{guozhiyuan,wwl}@tca.iscas.ac.cn
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] University of Chinese Academy of Sciences, Beijing, China

Abstract. As one of the core components in any SPN block cipher and hash function, diffusion layers are mainly introduced by matrices with maximal branch number. Surprisingly, the research on optimal binary matrices is rather limited compared with that on MDS matrices. Especially, not many general constructions for binary matrices are known that give the best possible branch number and guarantee the efficient software/hardware implementations as well. In this paper, we propose a new class of binary matrices constructed by Feistel structure with bit permutation as round functions. Through investigating bounds on the branch number our structure can achieve, we construct optimal binary matrices for a series of parameters with the lowest hardware cost up to now. Compared to the best known results, our optimal solutions for size 16×16 and 32×32 can save about 20 % and 33.3 % gate equivalents respectively. Without loss of hardware efficiency, a list of software-friendly optimal binary matrices can be constructed by Feistel structure with cyclic shift as round functions. The characteristics of this class of matrices are summarized and involutory optimal instances with commonly used dimensions are also provided. In the case of 8×8, we prove that optimal matrices from our structure can not be involutory. Finally, we extend the strategy to Generalized Feistel Structure and present some typical experimental results.

Keywords: Lightweight cryptography · Diffusion layer · Optimal binary matrix · Feistel structure · Multiple platforms

1 Introduction

As a central part of Substitution-Permutation Networks, diffusion layers are very important for the overall security and efficiency of cryptographic schemes. On the one hand, they play a role in spreading internal dependencies, which contributes to enhancing the resistance of statistical cryptanalysis. On the other hand, with the rapid development of lightweight cryptography, designing hardware-efficient diffusion layers has already been a hot research topic due to the increasing importance of ubiquitous computing.

O. Dunkelman and L. Keliher (Eds.): SAC 2015, LNCS 9566, pp. 352–372, 2016.
DOI: 10.1007/978-3-319-31301-6_21

The quality of a diffusion layer is connected to its branch number, whose cryptographic significance corresponds to the minimal number of active S-boxes in any two consecutive rounds. Obviously, the larger the branch number is, the better the diffusion effect will be, and simultaneously the cipher will not be vulnerable to unexpected attacks. Therefore, most designers chose to focus on diffusion layers with the best possible branch number to ensure a relatively strongest security.

From a coding theory perspective, Maximum Distance Separable (MDS) codes are quite good choices for the construction of diffusion layers since their branch numbers are maximum (known as the Singleton bound [1]). Not only are MDS matrices used in many block ciphers [2–4], but they promote generations of various related design strategies [5–7]. However, a problem with using MDS matrices is that they usually come at the price of a less efficient implementation. Due to Galois field multiplications, hardware implementations will often suffer from an important area requirement, with the result that MDS matrices are not suitable for the resource-constrained environments, such as RFID systems and sensor networks. Although this unfavorable situation is greatly improved with the advent of recursive MDS matrices [8–10], the temporary memory required (and hence hardware area) for the computation of matrices is still not reduced to a degree of satisfaction sometimes.

Another attractive type of diffusion layers is derived from Maximum Distance Binary Linear (MDBL) codes. The corresponding binary matrices are optimal in the sense that they achieve the largest possible branch number. Though the diffusion speed of optimal binary matrices can not keep pace with the one of MDS matrices, it is an overwhelming advantage that they involve no finite field multiplication, which is more propitious to a low-cost implementation. Typical examples are block ciphers E2 [11], Camellia [12] and ARIA [13], who get an excellent hardware efficiency and remarkable software performance on various platforms as well. It is accordingly our belief that, in many cases, it is easier to obtain an overall construction through using optimal binary matrix (or in general a matrix with branch number not meeting the Singleton bound), despite sacrificing the diffusion speed to a certain extent.

Compared with the study on constructions of MDS matrices, the research on designs of MDBL matrices is rather limited [14,15]. Early strategy from [16] (partially) guided the design of diffusion layers in E2 and Camellia, and unified method presented in [17] was conducive to summarizing the characteristics of 8×8 optimal binary matrices. For constructions of large dimensions (e.g. 16×16 and 32×32), designers in [18,19] considered combining small matrices into bigger ones, where each block matrix corresponds a finite field element. Indeed, in our opinion, the generalities of most previous constructions (focusing only on a few dimensions) are very weak, not to mention making them have efficient implementation. Here, one exception is the proposal of Dehnavi et al. [20], who recently investigated a special kind of binary linear layers for commonly used sizes with efficient implementation.

Feistel structure is one of the most prominently used structures in cryptography and accounts for substantial portion of data encrypted today. This was facilitated by the introduction of DES [21], which indicated the generation of modern block cipher. Not only is this classical structure used in large quantities of symmetric-key algorithms, but it inspires plenty of designs of cryptographic primitives. For example, S-boxes in [22–24] are constructed by 3-round Feistel structure, and linear layers in E2 and Camellia are implicitly implemented with 4-round Feistel structure.

Our Contributions. In this paper, we propose constructing diffusion layers over $\mathbb{F}_2$ with maximal branch number and efficient hardware implementation by use of Feistel structure with bit permutation as round functions. After introducing necessary notations and concepts in Sect. 2, we investigate the bounds on the branch number this construction can achieve, which will later help us judge whether hardware efficiency (hereby focus mainly on the area and latency) of the resulting matrix is optimal. Meanwhile, taking account of the improvement of software performance, we restrict the round function to cyclic shift and give the overall search strategy for "optimal solutions" in Sect. 3.

In order to demonstrate the generality of our construction, we provide typical optimal solutions for a series of feasible parameters (up to 32) in Sect. 4. To the best of our knowledge, the hardware cost of most proposals is the lowest compared with previous results. For cryptographic applications, our focus is further placed on involutory optimal solutions with commonly used dimensions in Sect. 5, and we prove that it is impossible to obtain an involutory 8×8 optimal diffusion layer from this structure.

Along similar lines, we present diffusion layers constructed by Generalized Feistel Structure in Sect. 6, improving their applicabilities on other platforms without loss of hardware efficiency. According to figures listed in Sect. 7, we afterwards show that optimal solutions for size 16×16 and 32×32 can save about 20 % and 33.3 % gate equivalents respectively, compared to the best known results. Finally, we conclude the paper in Sect. 8.

2 Preliminaries

In this section, we fix the basic notions and further more introduce several judgement methods of branch number. Since diffusion layers investigated in the present paper are linear transformations on the n-dimensional vector space over $\mathbb{F}_2$, we directly use an $n \times n$ binary matrix to represent a linear layer in the subsequent discussions.

2.1 Branch Number

Assume $\mathbf{v} = (v_1, v_2, \ldots, v_n)^T$ is a vector such that $v_i \in \mathbb{F}_2$, $1 \leq i \leq n$. Then the Hamming weight of $\mathbf{v}$, denoted by $w_b(\mathbf{v})$, is equal to the number of non-zero elements in $\mathbf{v}$.

Definition 1. *[3] The differential branch number of a diffusion layer D is given by*

$$\mathcal{B}_d(D) = \min_{\mathbf{v} \neq 0} \{w_b(\mathbf{v}) + w_b(D(\mathbf{v}))\}. \tag{1}$$

Analogously, we can define the linear branch number.

Definition 2. *[3] The linear branch number of a diffusion layer D is given by*

$$\mathcal{B}_l(D) = \min_{\mathbf{v} \neq 0} \{w_b(\mathbf{v}) + w_b(D^T(\mathbf{v}))\}, \tag{2}$$

where D^T is the transposition of D.

As the differential branch number of an $n \times n$ linear transformation is equal to the minimum distance of its associated $[2n, n]$ linear code, the maximal $\mathcal{B}_d$ of a binary matrix is known for small dimension according to [25]. A binary matrix is optimal if it achieves the maximal $\mathcal{B}_d$ and $\mathcal{B}_l$. Since each $n \times n$ (with the exception of $n = 32$) diffusion layer over $\mathbb{F}_2$ constructed in this article satisfies $\mathcal{B}_d = \mathcal{B}_l$, we omit linear branch number in the sequel.

Definition 3. *[14] Two matrices A, B are permutation homomorphic to each other if there exists a row permutation ρ and a column permutation γ satisfying*

$$\rho(\gamma(A)) = \gamma(\rho(A)) = B. \tag{3}$$

Proposition 1. *[14] If two matrices A, B are permutation homomorphic to each other, then A, B are of the same branch number.*

2.2 Judgement Methods

There is a one-to-one correspondence between an $n \times n$ linear transformation $\theta(x) = M \cdot x$ and a linear code $\mathcal{C}_\theta$ with the generator matrix $G_\theta = [I_{n \times n} | M]$, so we can use the following property to determine $\mathcal{B}_d(M)$.

Proposition 2. *[3] A linear code has minimum distance d if and only if every $d - 1$ columns of its parity check matrix are linearly independent and there exists some set of d columns that are linearly dependent.*

To deal with an $n \times n$ binary matrix with branch number s, it costs approximately $\sum_{i=1}^{s} \binom{2n}{i}$ Gaussian eliminations according to Proposition 2, while it needs to exhaust all possible non-zero input vectors based on Definition 1. After analyzing the characteristics of minimum-weight codewords among $\mathcal{C}_\theta$, we give Algorithm 2 (cf. Appendix A for more details) as main detection method, reducing the time complexity to $2 \sum_{i=1}^{\lfloor s/2 \rfloor} \binom{n}{i}$.

3 On Properties of Proposed Diffusion Layers

3.1 The General Construction

Throughout this paper, we consider diffusion layers over $\mathbb{F}_2$ constructed by Feistel structure with bit permutation as round functions. Let $x = (x_L, x_R)$ and $y = (y_L, y_R)$ be the n-bit input and output respectively, then a diffusion layer shown in Fig. 1 can be characterized as

$$M = \begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \begin{pmatrix} P_r & I \\ I & 0 \end{pmatrix} \cdots \begin{pmatrix} P_1 & I \\ I & 0 \end{pmatrix}, \qquad (4)$$

where the size of each block matrix is $\frac{n}{2} \times \frac{n}{2}$. In the rest of this paper, we only extract the sequence of permutation matrices, namely, $[P_1, P_2, \ldots, P_r]$ to represent M for simplicity. According to [21], it holds that

$$M^{-1} = [P_r, P_{r-1}, \ldots, P_1], \qquad (5)$$

which means the inverse matrix could be implemented with the same structure by simply reversing the order of round functions. This decided advantage guarantees the diffusion layer and its inverse require equal XOR's in terms of hardware implementation. In particular, the encryption and decryption can even use the exact same circuit in the case of involutory instances, i.e. round transformations appearing symmetrically (e.g. $[P_1, P_2, P_1]$ and $[P_1, P_2, P_2, P_1]$).

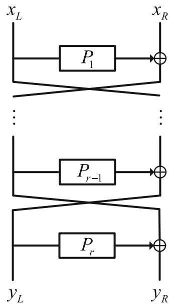

Fig. 1. A diffusion layer over $\mathbb{F}_2$ constructed with Feistel structure

As hardware efficiency can have very different meanings depending on the utilization scenario targeted by the designer, we hereby chose to focus on two classical metrics: silicon area and latency. Clearly, to make the perfect diffusion layer hardware-optimal under this construction, the number of iterations should be as small as possible on the premise of maximal branch number. Therefore it is necessary to investigate the bound on the branch number of resulting diffusion layer.

3.2 Upper Bound of the Branch Number

Before elaborating our main theorem, we need to introduce the following novel observation. Remember that in technical terms, the Fibonacci sequence $F(n)$ is defined by the recurrence relation $F(n+2) = F(n+1) + F(n)$, with seed values $F(0) = 1$, $F(1) = 1$.[1]

Lemma 1. *Assume* $(0, \alpha)$ *is the input of the structure shown in Fig. 1 such that* $w_b(\alpha) = 1$. *Then the Hamming weight of the output of round* i, $w_b(y_i)$, *is upper bounded by the* i*-th number of the Fibonacci sequence:*

$$w_b(y_i) \leq F(i). \tag{6}$$

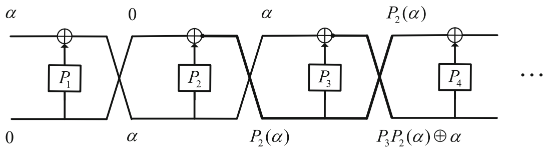

Fig. 2. Propagation of the Hamming weight on the input $(0, \alpha)$

Proof. Let us illustrate it by mathematical induction. According to the output of the first three rounds (see Fig. 2), we know that $w_b(y_1) = F(1)$, $w_b(y_2) = F(2)$ and $w_b(y_3) \leq F(3)$.

Now suppose the induction hypothesis is true for round i, $3 \leq i < r$. Then we only need to prove $w_b(y_{i+1}) \leq F(i+1)$. Notice that the transformation of round $i+1$ can be represented as

$$\begin{cases} y_{i+1,L} = P_{i+1}(y_{i,L}) \oplus y_{i,R} \\ y_{i+1,R} = y_{i,L} \end{cases}$$

and we always have $w_b(y_{i,L}) = w_b(P_{i+1}(y_{i,L}))$, which implies $w_b(y_{i+1,L}) \leq w_b(y_{i,L}) + w_b(y_{i,R})$.
Likewise, we obtain

$$w_b(y_{i+1,R}) = w_b(y_{i,L}) \leq w_b(y_{i-1,L}) + w_b(y_{i-1,R}).$$

Thus it holds

$$\begin{aligned} w_b(y_{i+1}) &= w_b(y_{i+1,L}) + w_b(y_{i+1,R}) \\ &\leq w_b(y_{i,L}) + w_b(y_{i,R}) + w_b(y_{i-1,L}) + w_b(y_{i-1,R}) \\ &\leq F(i) + F(i-1) \\ &= F(i+1), \end{aligned}$$

and we complete the proof. □

[1] Alternatively, the chosen starting points are fixed to $F(0) = 0$, $F(1) = 1$, which has no substantial impact on the global sequence.

Theorem 1. *The branch number of the diffusion layer constructed as in Fig. 1 satisfies*

$$\mathcal{B}_d^{(r)} \leq \begin{cases} 2F\left(\frac{r+1}{2}\right) & r \text{ is odd} \\ F\left(\frac{r}{2}\right) + F\left(\frac{r}{2}+1\right) & r \text{ is even}, \end{cases} \tag{7}$$

where the superscript is used to emphasize the number of rounds in the proposed construction.

Proof. Our strategy is similar to the start-from-the-middle technique [26]: start with a particular state value at the middle round and then propagate forward and backward to the output and input of the Feistel structure respectively. Note that middle round means different positions depending on whether the number of rounds is odd or even.

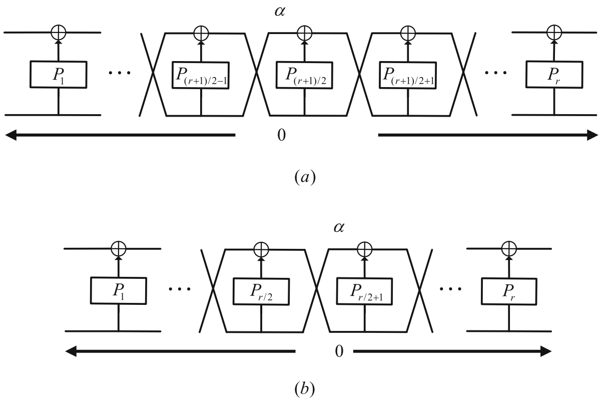

Fig. 3. Upper bound of the branch number of proposed diffusion layers

When r is odd, let $(0, \alpha)$ be the input of round $(r+1)/2$ such that $w_b(\alpha) = 1$. For $r = 1$, it is easy to see that $w_b(y_0) + w_b(y_1) = 2F(1)$. For $r \geq 3$, both the forward and backward propagations begin with the same initial value[2] $(\alpha, 0)$ and contain $(r+1)/2 - 1$ rounds (see Fig. 3.(a)). According to Lemma 1, we obtain one input/output pair whose Hamming weight satisfies

$$w_b(y_0) \leq F\left(\frac{r+1}{2}\right), \quad w_b(y_r) \leq F\left(\frac{r+1}{2}\right),$$

which implies the branch number of the resulting diffusion layer is at most $2F((r+1)/2)$.

When r is even, we change the target position to round $r/2+1$, with the result that each direction consists of $r/2$ rounds (see Fig. 3.(b)). Likewise it holds

$$w_b(y_0) + w_b(y_r) \leq F\left(\frac{r}{2}+1\right) + F\left(\frac{r}{2}\right),$$

[2] Notice that we consider the input of round $(r+1)/2+1$ as the forward starting point.

since the inputs of the backward and forward direction are $(\alpha, 0)$ and $(0, \alpha)$ respectively. Hence we complete the proof. □

Remark 1. For an expected branch number, Theorem 1 gives insights on the lower bound on the number of rounds our construction should have, efficiently reducing a lot of unnecessary search works. As an illustration, we need at least 8 iterations in the Feistel structure to get a diffusion layer with $\mathcal{B}_d = 12$ due to $\mathcal{B}_d^{(7)} \leq 2F(4) = 10$.

3.3 Search Strategy for Software-Friendly Diffusion Layers

In this section, we will explain how to improve software performances of the proposed diffusion layers without loss of hardware efficiency. Compared with the bit permutation, cyclic shift is undoubtedly much more attractive as suitable rotation can be implemented as a single instruction on the corresponding processor. For example, while constructing a 16×16 binary matrix with cyclic shift as round transformations, all operations of each round are based on 32-bit words on condition that 4-bit S-boxes are used. As a result, instead of bit permutation, cyclic shift is our first choice and the round function is afterwards restricted to $P_i(x) = x <<< t_i$, $0 \leq t_i < n/2$.

The pseudo-code of our basic search procedure is shown in Algorithm 1. The function BasicSearch (n, r, T, $\mathbb{G}$) returns all $n \times n$ binary matrices with $\mathcal{B}_d \geq T$ constructed by r-round Feistel structure. Here $\mathbb{G}$ denotes the set of transformation matrices that can be selected as round functions. On the basis of the above strategy, we initialize it to the set of matrices representing cyclic shift (which implies $|\mathbb{G}| = n/2$) and begin the first attempt with minimum possible r according to Theorem 1. If no optimal solution is found (i.e. $\mathbb{E} = \emptyset$), choose to increase r or relax restrictions on some round functions to continue searching, until suboptimal solutions are returned.

Algorithm 1. Search for optimal diffusion layers over $\mathbb{F}_2$

```
function BasicSearch($n$, $r$, $T$, $\mathbb{G}$)
    $\mathbb{E} \leftarrow \emptyset$
    for all $M \in \{ [P_1, P_2, \ldots, P_r] | P_i \in \mathbb{G}, 1 \leq i \leq r\}$ do
        if $\mathcal{B}_d(M) \geq T$ then
            $\mathbb{E} \leftarrow \mathbb{E} \cup \{M\}$
        end if
    end for
    return $\mathbb{E}$
end function
```

Remark 2. "Optimal solutions" here refer to binary matrices with maximal $\mathcal{B}_d$ constructed by the least possible number of cyclic shift operations. For instance, an 8×8 matrix with $\mathcal{B}_d = 5$ constructed by 4 cyclic shifts is optimal solution owing to $\mathcal{B}_d^{(3)} \leq 4$. Moreover, "suboptimal solutions" have different forms since bigger r results in higher cost in hardware implementation, while enlarged $\mathbb{G}$ (from the set of cyclic shift matrices to the one of permutation matrices) leads to the loss of advantages in software performance. Consequently, there are various trade-offs when we search for suboptimal solutions[3].

Next, we introduce the following statement to relate certain matrices that lead to the same branch number.

Theorem 2. *For any diffusion layer $M = [P_1, P_2, \ldots, P_r]$ constructed by r-round Feistel structure, there always exists a corresponding $M' = [I, P_2{}', \ldots, P_r{}']$ such that $\mathcal{B}_d(M') = \mathcal{B}_d(M)$.*

Proof. First of all, it is not difficult to see that we can place P_1 after the XOR operation in round 1 as shown in Fig. 4.(b), since

$$y_{1,L} = P_1(x_L) \oplus x_R = P_1(x_L \oplus P_1^{-1}(x_R)).$$

By using similar equivalent transforms, P_1 can be moved to the end of the structure (see Fig. 4.(c) and (d)), with each round function redefined as $P_i{}' = P_i \cdot P_1$ (i is even) or $P_i{}' = P_1^{-1} \cdot P_i$ (i is odd). Then depending on whether the number of rounds is even or odd, it holds

$$M = \begin{pmatrix} P_1 & 0 \\ 0 & I \end{pmatrix} \cdot M' \cdot \begin{pmatrix} I & 0 \\ 0 & P_1^{-1} \end{pmatrix} \text{ or } M = \begin{pmatrix} I & 0 \\ 0 & P_1 \end{pmatrix} \cdot M' \cdot \begin{pmatrix} I & 0 \\ 0 & P_1^{-1} \end{pmatrix},$$

respectively, which means M and M' are permutation homomorphic to each other. Thus their branch numbers are equal according to Proposition 1 and we complete the proof. □

Note that all matrices constructed by r-round Feistel structure can be classified according to any P_i, $i \in \{1, \ldots r\}$, although we simply choose $P_1 = I$ in Theorem 2. In other words, each diffusion layer constructed with $P_1 = I$ is a representative of an equivalence class, from which one can obtain all diffusion layers in the same equivalence class through analogous transforms mentioned above. We will make use of this property to reduce the search space by one round in the subsequent experiment.

4 Constructing Optimal Diffusion Layers with Feistel Structure

In this section, we will provide the results on constructing diffusion layers for various parameters. According to the search strategy, cyclic shift is preferred choice and for convenience, we abuse the symbol R_i, $i = 0, \ldots, n/2 - 1$, to

[3] As we take maximal branch number as primary premise, the branch number of suboptimal solutions is equal to that of optimal solutions.

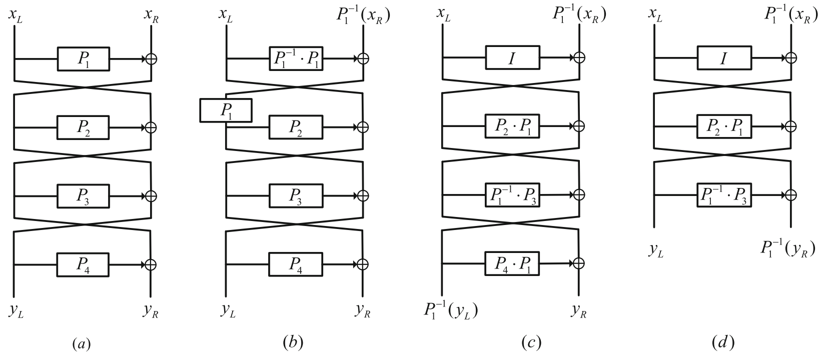

Fig. 4. Equivalence partitioning of the proposed diffusion layers

represent the matrix which corresponds to the transformation $L(x) = x <<< i$, where the size of x is $n/2$. As an example, an 8×8 matrix $M = [R_0, R_3, R_2, R_1]$ denotes the diffusion layer constructed by the following round functions:

$$P_1 = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}, \; P_2 = \begin{pmatrix} 0\,0\,0\,1 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{pmatrix}, \; P_3 = \begin{pmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}, \; P_4 = \begin{pmatrix} 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 1\,0\,0\,0 \end{pmatrix}.$$

Clearly, R_0 is the representation of identity matrix and more R_0's implies more efficient software implementation.

4.1 Diffusion Layers for $n = 4$, 8, 16 and 32

First, we ran Algorithm 1 to search for optimal diffusion layers with commonly used dimensions in cryptography, that is, $n = 2^k$, where $k = 2, 3, 4, 5$. The total number of optimal solutions and typical instance of M we obtained are summarized in Table 1, accompanied by the cost in hardware implementation for each parameter. Notice that the branch number of 32×32 binary matrices we can find is 12, which is consistent with the best known value.

Table 1. Experimental results for optimal diffusion layers with $n = 4$, 8, 16, and 32

n	$\mathcal{B}_d$	Optimal solutions		XOR gates
		Total number	Example of M	
4	4	2	$[R_0, R_1, R_0]$	6
8	5	32	$[R_0, R_1, R_2, R_0]$	16
16	8	9760	$[R_0, R_1, R_1, R_2, R_2, R_0]$	48
32	$12^{\dagger}$	6272	$[R_0, R_1, R_1, R_{13}, R_{13}, R_0, R_8, R_6]$	128

The time complexity of the exhaustive search for optimal solutions with given length n is $(n/2)^r$, where r is the least possible number of rounds to achieve the maximal $\mathcal{B}_d$. Actually it took us less than 20 h to find all the best 32×32 binary matrices through parallel search, and with the help of Theorem 2, we can further reduce the search time by a factor of 2^4. Additionally, the examples in Table 1 are perfect in the sense that they are constructed with the most possible number of R_0's, cutting down on as many rotation instructions as possible in software implementation. Yet it is noteworthy that the technique of equivalence partitioning is not applicable in this case, as matrices in the same equivalence class are constructed by different sequences of round functions.

Moreover, we need to point out for any $n \times n$ diffusion layer constructed by r-round Feistel structure, the number of XOR gates required for hardware implementation is $nr/2$, which enjoys an overwhelming advantage even if we have not done any other optimization. For instance, although the total number of ones in any 32×32 binary matrix with $\mathcal{B}_d = 12$ is lower bounded by $11 \times 32 = 352$, each of our optimal solutions can be implemented just with 128 XOR's, which is the best result up to our knowledge.

4.2 Results for Other Parameters

To illustrate the generality of our proposal, we also search for diffusion layers with other sizes (n is even and $n < 20$), despite the fact that they are probably not often used. As shown in Table 2, we obtain optimal solutions for each given length, with the exception of $n = 12$. In other words, no 12×12 optimal binary matrix can be constructed by 6-round Feistel structure with cyclic shift as round functions. This means we need to consider searching for suboptimal solutions using the methods described in Sect. 3.3.

Laying particular stress on hardware efficiency, we hereby adopt the second strategy (i.e. enlarging $\mathbb{G}$ for only one round function) and find 120 suboptimal solutions. One of the best results is

$$M_{12\times 12} = [R_5, P_1, R_4, R_1, R_1, R_0],$$

where P_1 is a permutation matrix. Due to the lack of space, we will give the concrete form of P_1 and $M_{12\times 12}$ in Appendix B.

Table 2. Experimental results for diffusion layers with other interesting sizes

n	$\mathcal{B}_d$	Optimal solutions		XOR gates
		Total number	Example of M	
6	4	12	$[R_0, R_1, R_0]$	9
10	6	80	$[R_0, R_1, R_2, R_0, R_4]$	25
14	8	42	$[R_0, R_1, R_3, R_6, R_5, R_3]$	42
18	8	36720	$[R_0, R_1, R_1, R_2, R_2, R_0]$	54

4.3 Other Information from the Proposed Structure

Below we elaborate how to acquire a prior knowledge of the resulting matrix. First, it is clear that any matrix constructed by r-round Feistel structure can be characterized as

$$M^{(r)} = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}, \tag{8}$$

where each block matrix is an expression that consists of P_i, $1 \leq i \leq r$. For example, based on

$$M^{(3)} = \begin{pmatrix} P_2P_1 \oplus I & P_2 \\ P_3P_2P_1 \oplus P_3 \oplus P_1 & P_3P_2 \oplus I \end{pmatrix},$$

we get $A_4 = P_3P_2 \oplus I$ in this case. Let $T(A_i)$ be the number of terms in A_i and through exploring the regularity on changes of $T(A_i)$, we have

Theorem 3. *The four block matrices constituting $M^{(r)}$ as shown in (8) satisfy*

$$T(A_1) = F(r-1), T(A_2) = F(r-2), T(A_3) = F(r), T(A_4) = F(r-1).$$

This observation is straightforward and we omit the proof here. It seems that Theorem 3 places major focus only on the expanded form, nevertheless, we will later see that it contributes to understanding the generic picture of optimal matrices.

As a matter of fact, each $\frac{n}{2} \times \frac{n}{2}$ block matrix in the resulting matrix $M_{n\times n}$ is a circulant matrix [27] for the optimal solution. To explain conveniently, we denote a $t \times t$ circulant matrix with i ones in the first row by $U_i^{(t)}$. Then each optimal solution in the case of $n = 4$ satisfies[4]

$$M_{4\times 4} = \begin{pmatrix} U_2^{(2)} & U_1^{(2)} \\ U_1^{(2)} & U_2^{(2)} \end{pmatrix},$$

since the terms in A_i are always eliminated pairwise during calculating. Taking $n = 8$ as anther illustration, in the light of $T(A_1) = 3, T(A_2) = 2, T(A_3) = 5, T(A_4) = 3$ and the fact that each row in an optimal solution has at least 4 ones, we conclude

$$M_{8\times 8} = \begin{pmatrix} U_3^{(4)} & U_2^{(4)} \\ U_3^{(4)} & U_3^{(4)} \end{pmatrix}, \quad or \quad \begin{pmatrix} U_3^{(4)} & U_2^{(4)} \\ U_1^{(4)} & U_3^{(4)} \end{pmatrix},$$

which is in accord with the experimental results. Specifically, there are only 8 instances with the latter form (among 32 optimal solutions) and one example will be introduced in Appendix C.

[4] The necessary condition implies the number of optimal solutions of $M_{4\times 4}$ is at most 4 since $U_2^{(2)}$ is determined and $U_1^{(2)}$ has only two forms. As can be seen in our search result, solutions achieve the maximal branch number only if "$U_1^{(2)} = U_1^{(2)}$".

5 Searching for Involutory Optimal Diffusion Layers

In this section, we consider constructing involutory optimal matrix by Feistel structure with cyclic shift as round function, which enjoys an attractive advantage as it requires only one procedure to be implemented for the encryption and decryption. Notice that the search strategy is derived from Algorithm 1 and most instances come from the optimal solutions. For sizes that are often used in block ciphers, we list some examples in Table 3. Below are some explications of our experimental results:

(1) For parameter $n = 4$, the two optimal solutions we find themselves are involutory, however, the proportion ($= 24/9760$) is very low for $n = 16$.
(2) In the case of $n = 32$, the largest branch number of the involutory matrices constructed by 8-round Feistel structure is 11. We do not search further (for involutory matrices with $\mathcal{B}_d = 12$) since the results are lightweight enough to have promising applications.
(3) As for $n = 8$, we do not obtain even one involutory instance despite of many attempts on the number of rounds. Before jumping to the full explanation of this situation, we need the following lemma.

Table 3. Experimental results for involutory diffusion layers with $n = 4$, 16, and 32

n	$\mathcal{B}_d$	Total number	Example of M
4	4	2	$[R_1, R_0, R_1]$
16	8	24	$[R_0, R_1, R_2, R_2, R_1, R_0]$
32	11	640	$[R_0, R_1, R_6, R_{14}, R_{14}, R_6, R_1, R_0]$

Lemma 2. *Assume M is an 8×8 optimal matrix as shown in (8) where each A_i, $1 \le i \le 4$, is a circulant matrix. Then no A_i can be "0" or $U_4^{(4)}$.*

Proof. Suppose not, then two cases should be discussed:

(a) Without loss of generality, we let $A_1 = 0$, then $A_3 = U_4^{(4)}$ since every column in M contains at least 4 ones, which implies M is singular and hence is a contradiction.
(b) Similarly let $A_1 = U_4^{(4)}$. Then it needs $A_3 = U_3^{(4)}$ or $A_3 = U_1^{(4)}$ to make M invertible. Note that in these cases, we have $\mathcal{B}_d \le 4$ as there exists a vector $\mathbf{v} = (1, 1, 0, \ldots, 0)^T$ with $w_b(\mathbf{v}) = 2$ such that $w_b(\mathbf{v}) + w_b(M \cdot \mathbf{v}) = 4$. This contradicts the optimality condition.

Consequently, there is no "0" or $U_4^{(4)}$ among the four block matrices and we complete the proof. □

In addition, the following statement, deduced from [17], is very useful for our illustration.

Proposition 3. *For any 8×8 binary matrix with $\mathcal{B}_d = 5$, if the rows have only two different Hamming weights and each contains half number of rows, then it must belong to one of the following cases:*

(1) the rows are of Hamming weight 4 and 5.
(2) the rows are of Hamming weight 5 and 6.

Theorem 4. *Any 8×8 optimal diffusion layer constructed by r-round Feistel structure with cyclic shift as round functions can not be involutory.*

Proof[5]. First, the conditions in Proposition 3 are always satisfied for every 8×8 optimal matrix constructed by the structure as shown in Fig. 1. Furthermore, the four block matrices are all circulant matrices as explained before and according to Lemma 2 and Proposition 3, it is easy to see that the form of any resulting matrix belongs to one of the following eight cases:[6]

$$\begin{pmatrix} U_3 & U_3 \\ U_3 & U_2 \end{pmatrix}, \begin{pmatrix} U_2 & U_3 \\ U_3 & U_3 \end{pmatrix}, \begin{pmatrix} U_1 & U_3 \\ U_3 & U_2 \end{pmatrix}, \begin{pmatrix} U_2 & U_3 \\ U_3 & U_1 \end{pmatrix}, \tag{9}$$

$$\begin{pmatrix} U_3 & U_3 \\ U_2 & U_3 \end{pmatrix}, \begin{pmatrix} U_3 & U_2 \\ U_3 & U_3 \end{pmatrix}, \begin{pmatrix} U_3 & U_2 \\ U_1 & U_3 \end{pmatrix}, \begin{pmatrix} U_3 & U_1 \\ U_2 & U_3 \end{pmatrix}. \tag{10}$$

For the sake of clarity, we simply denote the resulting matrix by

$$\tilde{M} = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

and now suppose $\tilde{M}$ is involutory. Then bases on $\tilde{M}^2 = I$ and the property that multiplications here are commutative, we have

$$\begin{cases} A^2 \oplus BC = I & (11) \\ (A \oplus D)B = 0 & (12) \\ (A \oplus D)C = 0 & (13) \\ A^2 = D^2 & (14) \end{cases}$$

Next, we claim neither A nor D is equal to U_2. Otherwise, one is singular and the other is invertible, since there is only one U_2 among each of the eight matrices. This contradicts (14) and we thereby exclude all cases in (9).

[5] Throughout this proof, we omit the superscript in $U_i^{(4)}$ for simplicity.

[6] With the view of permutation homomorphic, these forms can be considered as two types on condition that the (row and column) permutation in Definition 2 is blockwise.

For cases in (10), we suppose $C = U_2$ without loss of generality. Then B is a nonsingular matrix and we have $B^{-1} = U_1,\ or\ B^{-1} = U_3$.

Due to $BC \neq 0$, we have $A^2 \neq I$ from (11). Furthermore, as $A = U_3$ and $A^2 = (U_4 \oplus U_1)(U_4 \oplus U_1) = U_1^2$, it holds

$$A^2 = \begin{pmatrix} 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 1\,0\,0\,0 \\ 0\,1\,0\,0 \end{pmatrix}.$$

Then according to (11), we can easily obtain

$$C = \begin{cases} (I \oplus A^2)U_1 & B^{-1} = U_1 \\ (I \oplus A^2)(U_4 \oplus U_1) = (I \oplus A^2)U_1 & B^{-1} = U_3, \end{cases}$$

which implies the first and third columns in C are the same (the remaining two columns are also the same). Therefore, there always exists a vector $\mathbf{v} = (1, 0, 1, 0, \ldots, 0)^T$ with $w_b(\mathbf{v}) = 2$ such that $w_b(\mathbf{v}) + w_b(C \cdot \mathbf{v}) = 4$. This is a contradiction and all cases in (10) are thus excluded.

In summary, no involutory 8 × 8 optimal binary matrix can be constructed by Feistel structure with cyclic shift as round functions and we complete the proof. □

6 Diffusion Layers Constructed with Generalized Feistel Structure

As explained in Sect. 3.3, we restrict the round function to cyclic shift with the purpose of improving software performance. However, an unfavourable situation we are likely to face is that the length of $n/2$ words is longer than the word size of the processor. For example, in the case of 8-bit S-box, the software efficiency of our 16 × 16 optimal diffusion layer on 32-bit processor is weaken since the rotation on a 64-bit word becomes complicated.

To make up for the above shortcomings, an instinctive idea is to construct diffusion layers using r-round Type-II Generalized Feistel Structure (GFS, [29]), which be characterized as

$$M_{gfs} = \begin{pmatrix} 0\,0\,0\,I \\ I\,0\,0\,0 \\ 0\,I\,0\,0 \\ 0\,0\,I\,0 \end{pmatrix} \begin{pmatrix} P_{2r-1} & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & P_{2r} & I \\ I & 0 & 0 & 0 \end{pmatrix} \cdots \begin{pmatrix} P_1 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & P_2 & I \\ I & 0 & 0 & 0 \end{pmatrix} \tag{15}$$

where round functions in round i, P_{2i-1} and P_{2i}, are cyclic left shifts. Also, we use $[P_1, P_2, \ldots, P_{2r}]$ to represent M_{gfs} for simplicity.

Owing to the slow diffusion property of Type-II GFS, one may take it for granted that the number of rounds will be increased compared with the Feistel structure on the premise of the same $\mathcal{B}_d$. However, that is not the case according

Table 4. Experimental results for diffusion layers constructed by Type-II GFS

n	$\mathcal{B}_d$	example of M_{gfs}
8	5	$[R_0, R_0, R_0, R_1,$ $R_1, R_1, R_0, R_1]$
16	8	$[R_0, R_0, R_0, R_0, R_1, R_1,$ $R_1, R_2, R_0, R_0, R_0, R_0]$
32	11	$[R_0, R_0, R_0, R_0, R_4, R_5, R_1, R_1,$ $R_1, R_0, R_0, R_0, R_1, R_3, R_3, R_4]$

to our search results listed in Table 4. Actually, the conclusion of Theorem 1 also holds for Type-II GFS and hence the solutions we find are optimal in terms of the number of rounds.

Compared to the Feistel structure, the cost of hardware implementation of each round in Type-II GFS remains unchanged, which means we can almost perfectly solve the problem introduced at the beginning of this section. Yet, it is to be noticed that the time complexity of r-round search becomes $(n/4)^{2r}$, far greater than $(n/2)^r$ when $n > 8$. For example, while searching for 32×32 binary matrix with $\mathcal{B}_d = 11$ constructed by Type-II GFS, the total number of matrices to be detected is 2^{48}. Despite the help of equivalence partitioning technique, the search space (i.e. 2^{42}) is still so huge that we need highly parallel computations to obtain all solutions.

7 Comparison with Known Results

In this section, we compare our Feistel-structure-based proposals with previous known results on hardware implementation. As can be seen in Table 5, solutions we found for $n = 32$ can save approximately 33.3 % gate equivalents compared to the best known result. Furthermore, while considering the constructions with $\mathcal{B}_d = 11$, this improvement shots up to a staggering 64.7 %.

For the size $n = 16$, a noteworthy comparison comes between the diffusion layer used in ARIA and ours. The area can be reduced by 36.8 % provided that the original linear layer is replaced by our optimal instances. Moreover, in the case of $n = 8$, the hardware cost of the design in [16] is equal to ours. The reason, which we have mentioned in the introduction, is that the examples given in [16] can be implicitly implemented by 4-round Feistel structure (while the last swap is not removed).

Here we omit comparisons on the software performance for two reasons. One is some of the previous constructions place the major focus on maximizing the branch number using algebraic methods, ignoring the estimate of implementation efficiency, and thus there is no need to make comparisons. The other is diffusion layers in cryptographic algorithms are often implemented together with S-boxes, which makes the comparisons complicated. Nevertheless, as explained

Table 5. Comparison of our diffusion layers with the known results

n	$\mathcal{B}_d$	XOR gates	Involutory	Reference
4	4	**6**	Yes	This paper
8	5	34	Yes	[18]
	5	16	No	[16]
	5	16	No	This paper
16	8	95	Yes	[18]
	8	76	Yes	[13]
	8	64	No	[20]
	8	60	Yes	[14]
	8	**48**	Yes	This paper
32	10	286	No	[15]
	10	**112**	Yes	This paper
	11	363	No	[19]
	11	**128**	Yes	This paper
	12	328	Yes	[19]
	12	192	No	[20]
	12	**128**	No	This paper

in Sects. 3.3 and 6, our proposals still have excellent software performance even without any optimization.

8 Conclusion

In this paper, we propose a new class of optimal diffusion layers over $\mathbb{F}_2$ by use of Feistel structure with bit permutation as round functions. Through investigating bounds on the branch number our structure can achieve, we construct optimal binary diffusion layers for a series of parameters (up to 32×32) with excellent software/hardware performances. As far as we know, the hardware cost of most proposals is the lowest compared to the previous results. Involutory optimal instances for the commonly used dimensions are also presented, with the exception of 8×8. Finally, we investigate optimal diffusion layers constructed by Type-II GFS and provide some typical solutions. Since the hardware cost of our results are extremely low, we expect our strategy will be useful for future construction of lightweight ciphers based on (involutory) binary diffusion components.

Acknowledgments. The authors would like to thank the anonymous referees for their valuable comments. We are also grateful to Shengbao Wu and Renzhang Liu for providing useful suggestions. This work is supported by the National Basic Research Program of China (No. 2013CB338002) and National Natural Science Foundation of China (No. 61272476, No. 61232009, No. 61202422).

A Determination of Branch Number

For a $[2n, n, d]$ linear code $\mathcal{C}_\theta$, there exists at least one codeword $\mathbf{v} = (v_1, \ldots, v_{2n})^T$ such that $w_b(\mathbf{v}) = d$. Let

$$\mathbf{v}_{\mathbf{left}} = (v_1, \ldots, v_n)^T, \quad \mathbf{v}_{\mathbf{right}} = (v_{n+1}, \ldots, v_{2n})^T,$$

and then it must hold

$$w_b(\mathbf{v}_{\mathbf{left}}) \le d/2 \quad \textit{or} \quad w_b(\mathbf{v}_{\mathbf{right}}) \le d/2,$$

which implies the number of input vectors to be computed according to (1) is at most $2^{d/2}$.

Specifically, as both $G_\theta = [I|M]$ and $G'_\theta = [M^{-1}|I]$ are the generator matrices of $\mathcal{C}_\theta$, we can determine $\mathcal{B}_d(M)$ by searching for the minimum value between $w_b(\mathbf{x}) + w_b(M \cdot \mathbf{x})$ and $w_b(\mathbf{x}) + w_b(M^{-1} \cdot \mathbf{x})$ among at most $2^{d/2}$ input vectors of low Hamming weights.

Algorithm 2. Determining the branch number of a binary matrix

```
function BinaryBranchNumber($M$, $dimension$)
    $InvM \leftarrow$ inverse matrix of $M$
    $\mathbf{x} \leftarrow (1, 0, \ldots, 0)^T$
    $miniweight \leftarrow 2 \cdot dimension$
    while $w_b(\mathbf{x}) < miniweight/2$ do
        $weight \leftarrow w_b(\mathbf{x}) + w_b(M \cdot \mathbf{x})$
        if $weight \le miniweight$ then
            $miniweight \leftarrow weight$
        end if
        $weight \leftarrow w_b(\mathbf{x}) + w_b(InvM \cdot \mathbf{x})$
        if $weight \le miniweight$ then
            $miniweight \leftarrow weight$
        end if
        update $\mathbf{x}$ in the increasing order by Hamming weight
    end while
    return $miniweight$
end function
```

B $\boldsymbol{M_{12\times 12}}$ Constructed by $[\boldsymbol{R_5}, \boldsymbol{P_1}, \boldsymbol{R_4}, \boldsymbol{R_1}, \boldsymbol{R_1}, \boldsymbol{R_0}]$

$$P_1 = \begin{pmatrix} 0&0&0&0&0&1 \\ 0&0&0&0&1&0 \\ 1&0&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&0&1&0&0 \end{pmatrix} \qquad M_{12\times 12} = \begin{pmatrix} 0&0&0&1&0&1&1&1&1&0&1&1 \\ 1&1&1&1&0&0&1&1&0&0&1&0 \\ 1&0&1&0&1&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&1&1&1&1 \\ 1&0&0&1&1&1&0&1&0&1&1&0 \\ 0&1&1&1&0&1&0&1&1&1&0&0 \\ 0&0&1&1&1&0&1&0&1&1&1&0 \\ 0&1&1&0&1&0&1&1&1&0&0&1 \\ 1&1&1&1&1&0&0&0&0&1&0&1 \\ 1&1&0&0&0&1&1&1&0&1&0&1 \\ 0&1&0&0&1&1&0&0&1&1&1&1 \\ 1&1&0&1&1&1&1&0&1&0&0&0 \end{pmatrix}$$

C $\boldsymbol{M_{8\times 8}}$ Constructed by $[\boldsymbol{R_0}, \boldsymbol{R_2}, \boldsymbol{R_1}, \boldsymbol{R_1}]$

$$M_{8\times 8} = \begin{pmatrix} 1&1&0&1&1&0&0&1 \\ 1&1&1&0&1&1&0&0 \\ 0&1&1&1&0&1&1&0 \\ 1&0&1&1&0&0&1&1 \\ 0&1&0&0&1&1&1&0 \\ 0&0&1&0&0&1&1&1 \\ 0&0&0&1&1&0&1&1 \\ 1&0&0&0&1&1&0&1 \end{pmatrix}$$

References

1. Tilborg, H.C.A.: Coding Theory - A First Course. Lecture Notes on Error-Correcting Codes. Springer (1993)
2. Daemen, J., Knudsen, L.R., Rijmen, V.: The block cipher SQUARE. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149–165. Springer, Heidelberg (1997)
3. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Heidelberg (2002)
4. Shirai, T., Shibutani, K., Akishita, T., Moriai, S., Iwata, T.: The 128-bit blockcipher CLEFIA (extended abstract). In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 181–195. Springer, Heidelberg (2007)
5. Sajadieh, M., Dakhilalian, M., Mala, H., Omoomi, B.: On construction of involutory MDS matrices from Vandermonde Matrices in $GF(2^q)$. Des. Codes Crypt. **64**(3), 287–308 (2012)
6. Chand Gupta, K., Ghosh Ray, I.: On constructions of circulant MDS matrices for lightweight cryptography. In: Huang, X., Zhou, J. (eds.) ISPEC 2014. LNCS, vol. 8434, pp. 564–576. Springer, Heidelberg (2014)

7. Sim, S.M., Khoo, K., Oggier, F., Peyrin, T.: Lightweight MDS involution matrices. In: Leander, G. (ed.) FSE 2015. LNCS, vol. 9054, pp. 471–493. Springer, Heidelberg (2015)
8. Sajadieh, M., Dakhilalian, M., Mala, H., Sepehrdad, P.: Recursive diffusion layers for block ciphers and hash functions. In: Canteaut, A. (ed.) FSE 2012. LNCS, vol. 7549, pp. 385–401. Springer, Heidelberg (2012)
9. Wu, S., Wang, M., Wu, W.: Recursive diffusion layers for (lightweight) block ciphers and hash functions. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 355–371. Springer, Heidelberg (2013)
10. Augot, D., Finiasz, M.: Direct construction of recursive MDS diffusion layers using shortened BCH codes. In: Cid, C., Rechberger, C. (eds.) FSE 2014. LNCS, vol. 8540, pp. 3–17. Springer, Heidelberg (2015)
11. Kanda, M., Moriai, S., Aoki, K., Ueda, H., Takashima, Y., Ohta, K., Matsumoto, T.: E2 - a new 128-bit block cipher. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **E83**(A1), 48–59 (2000)
12. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: *Camellia*: a 128-bit block cipher suitable for multiple platforms - design and analysis. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39–56. Springer, Heidelberg (2001)
13. Kwon, D., Kim, J., Park, S., Sung, S., Sohn, Y., Song, J., Yeom, Y., Yoon, E., Lee, S., Lee, J., Chee, S., Han, D., Hong, J.: New block cipher: ARIA. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 432–445. Springer, Heidelberg (2004)
14. Koo, B.-W., Jang, H.S., Song, J.H.: Constructing and cryptanalysis of a 16 × 16 binary matrix as a diffusion layer. In: Chae, K.-J., Yung, M. (eds.) WISA 2003. LNCS, vol. 2908, pp. 489–503. Springer, Heidelberg (2004)
15. Koo, B.-W., Jang, H.S., Song, J.H.: On constructing of a 32 × 32 binary matrix as a diffusion layer for a 256-bit block cipher. In: Rhee, M.S., Lee, B. (eds.) ICISC 2006. LNCS, vol. 4296, pp. 51–64. Springer, Heidelberg (2006)
16. Kanda, M., Takashima, Y., Matsumoto, T., Aoki, K., Ohta, K.: A strategy for constructing fast round functions with practical security against differential and linear cryptanalysis. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 264–279. Springer, Heidelberg (1999)
17. Gao, Y., Guo, G.: Unified approach to construct 8×8 binary matrices with branch number 5. In: First ACIS International Symposium on Cryptography, and Network Security, Data Mining and Knowledge Discovery, E-Commerce and Its Applications, and Embedded Systems, pp. 413–416 (2010)
18. Aslan, B., Sakalli, M.: Algebraic construction of cryptographically good binary linear transformations. Secur. Commun. Netw. **7**(1), 53–63 (2014)
19. Sakalli, M., Aslan, B.: On the algebraic construction of cryptographically good 32×32 binary linear transformations. J. Comput. Appl. Math. **259**, 485–494 (2014)
20. Dehnavi, S., Rishakani, A., Shamsabad, M.: Bitwise linear mappings with good cryptographic properties and efficient implementation. IACR Cryptology ePrint Archive, 225 (2015)
21. DAta Encryption Standard (1999). http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf
22. Lim, C.: CRYPTON: a new 128-bit block cipher. In: The First AES Candidate Conference. National Institute for Standards and Technology (1998)
23. Specification of the 3Gpp. Confidentiality, Integrity Algorithms 128-EEA3, 128-EIA3. Document 4: Design and Evaluation Report, version 1.3 (2011)

24. Li, Y., Wang, M.: Constructing S-boxes for lightweight cryptography with Feistel structure. In: Batina, L., Robshaw, M. (eds.) CHES 2014. LNCS, vol. 8731, pp. 127–146. Springer, Heidelberg (2014)
25. Code Tables: Bounds on the parameters of various types of codes. http://codetables.de
26. Gilbert, H., Peyrin, T.: Super-Sbox cryptanalysis: improved attacks for AES-like permutations. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 365–383. Springer, Heidelberg (2010)
27. Davis, P.: Circulant Matrices, 2nd edn. American Mathematical Society, Providence (2012)
28. Sakalli, M., Akleylek, S., Aslan, B., Bulus, E., Sakalli, F.: On the construction of 20 × 20 and 24 × 24 binary matrices with good implementation properties for lightweight block ciphers and hash functions. Math. Prob. Eng. **2014**, 1–12 (2014). Article ID 540253
29. Zheng, Y., Matsumoto, T., Imai, H.: On the construction of block ciphers provably secure and not relying on any unproved hypotheses. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 461–480. Springer, Heidelberg (1990)