New Algorithm for Exhausting Optimal Permutations for Generalized Feistel Networks

Stéphanie Delaune, Patrick Derbez, Arthur Gontier, Charles Prud'Homme

New Algorithm for Exhausting Optimal Permutations for Generalized Feistel Networks*

Stéphanie Delaune[1], Patrick Derbez[1], Arthur Gontier[1], and Charles Prud'homme[2]

[1] Univ Rennes, CNRS, IRISA, Rennes, France
{stephanie.delaune,patrick.derbez,arthur.gontier}@irisa.fr
[2] TASC, IMT-Atlantique, LS2N-CNRS, F-44307 Nantes
charles.prudhomme@imt-atlantique.fr

Abstract. The Feistel construction is one of the most studied ways of building block ciphers. Several generalizations were proposed in the literature, leading to the Generalized Feistel Network (GFN) construction, in which the round function operates on each pair of blocks in parallel until all branches are permuted. At FSE'10, Suzaki and Minematsu studied the diffusion of such construction, raising the question of how many rounds are required so that each block of the ciphertext depends on all blocks of the plaintext. Exhausting all possible permutations up to 16 blocks, they observed that there were always optimal permutations mapping even-number input blocks to odd-number output blocks and vice versa. Recently, both Cauchois *et al.* and Derbez *et al.* proposed new algorithms to build optimal even-odd permutations for up to 36 blocks. In this paper, we present a new algorithm based on iterative path building to search for optimal Feistel permutation. This algorithm is much faster in exhausting optimal non-even-odd permutations than all the previous approaches. Our first result is a computational proof that no non-even-odd permutation reaches a better diffusion round than optimal even-odd permutations up to 32 blocks. Furthermore, it is well known that permutations with an optimal diffusion round do not always lead to optimal permutations against differential cryptanalysis. We investigate several new criteria to build permutations leading to more secure GFN.

Keywords: Block Cipher · Feistel Network · Differential Analysis

1 Introduction

The Feistel Network is a classical design of modern block ciphers, used for many primitives as `DES` [6], `TWINE` [11] and `SIMON` [2]. The core idea of such a construction is to split the plaintext into two halves of equal length called blocks. At each round, the second block is duplicated and one side goes through a function F and is then xored to the first block. The two resulting blocks are then inverted.

* The work presented in this article was funded by the French National Research Agency as part of the DeCrypt project (ANR- 18-CE39-0007).

One big advantage of this scheme is that the function F has not to be invertible since the decryption function is the same as the encryption one in reverse order. Since its introduction, several improvements have been proposed to the original design. In particular, at ASIACRYPT'96, Nyberg defined the Generalized Feistel Network (GFN) which splits the message into $2k$ blocks and uses a round function of the form:

$$(x_0, x_1, \ldots, x_{2k-1}) \rightarrow \pi(x_0 \oplus F_{i,0}, x_1, \ldots, x_{2k-2} \oplus F_{i,k-1}, x_{2k-1})$$

where each $F_{i,j}$ is a pseudorandom function, and π is a permutation of the blocks [7]. This design was for instance used in both the block ciphers `TWINE` [11] and `PICCOLO` [9]. It is a generalization of the more classical *Type-2 Feistel* construction proposed by Zheng *et al.* at CRYPTO'89 [12], in which the permutation π is always the cyclic shift.

Cryptographic properties of GFN highly depend on the permutation used for blocks. For instance, if the identity function was chosen as the permutation, the resulting block cipher would be very weak as the parallel application of weak ciphers. Thus, selecting the *optimal* permutation is an interesting task for designers. At FSE'10, Suzaki and Minematsu focused on finding the permutations reaching the *lowest diffusion rounds* [10]. More precisely, they searched for the permutations minimizing the number of rounds required to achieve full block diffusion: each block of the ciphertext depends on all blocks of the plaintext and vice-versa. This criterion is tied to the resistance of the resulting cipher against impossible differential attacks, a powerful cryptanalysis technique. Along with a lower bound on the diffusion round of a GFN of $2k$ blocks, Suzaki and Minematsu gave optimal permutations (w.r.t. the diffusion round) for $2 \leq 2k \leq 16$. It is worthy to note that such an optimal permutation was then used to design block ciphers such as `TWINE` [11]. At FSE'19 Cauchois *et al.* identified new equivalence classes regarding the diffusion rounds and, together with new algorithms, were able to give optimal permutations up to $2k = 20$ [3]. Furthermore, restricting the search to even-odd permutations (i.e. permutations sending blocks of even index to blocks of odd ones and vice-versa), they were able to find the best even-odd permutations up to $2k = 24$. Finally, few months later, Derbez *et al.* proposed a new characterization of the problem restricted to even-odd permutations as well as a clever algorithm to exhaust the search space. As a result they found the best even-odd permutations up to $2k = 36$ [5]. In particular, they solved the problem opened by Suzaki and Minematsu regarding the case $2k = 32$.

It is also possible to optimize GFN for other criteria than the diffusion round. For instance in [8], Shi *et al.* searched for the permutations offering the best resistance against Demirci-Selçuk meet-in-the-middle attacks [4].

Our contribution. Since the original work of Suzaki and Minematsu [10], most of the new algorithms to find the permutations lowering the diffusion round were dedicated to the even-odd case. There are two main reasons for that. First, considering even-odd permutations only does highly reduce the search space, making it possible to exhaust it. Second, it was shown that up to $2k = 20$ at least one of the optimal permutations is an even-odd permutation.

In this paper, we focus on non-even-odd permutations and propose a new algorithm to solve the general case. In previous approaches, the core part of algorithms was somehow dedicated to answering the question: does block i diffuse into all blocks after R rounds? In our new algorithm, we answer the question: does block i diffuse to block j after R rounds? This more precise question allows us to cut the search earlier than previous algorithms while exhausting the permutations. Thus, our first result is a computational proof that, up to $2k = 32$, there is always at least one even-odd permutation which is optimal regarding the diffusion round. The best known diffusion rounds for even-odd and non-even-odd permutations are given in Table 1.

In the second part of the paper, we investigate more sophisticated criteria than the diffusion round and study whether the optimal permutations lead to optimal GFN regarding differential cryptanalysis.

Table 1. State of the art regarding optimal Diffusion Round. k is the number of Feistel pairs and the references are : Suzaki et al. [10], Cauchois et al. [3], Derbez et al. [5]

<table>
<tr><th rowspan="2">$2k$</th><th rowspan="2">Fibonacci bound</th><th colspan="2">even-odd</th><th colspan="2">non-even-odd</th></tr>
<tr><th>DR</th><th>Ref</th><th>DR</th><th>Ref</th></tr>
<tr><td>6</td><td>5</td><td>5</td><td rowspan="6">[10]</td><td>6</td><td rowspan="6">[3],[10]</td></tr>
<tr><td>8</td><td>6</td><td>6</td><td>6</td></tr>
<tr><td>10</td><td>6</td><td>7</td><td>7</td></tr>
<tr><td>12</td><td>7</td><td>8</td><td>8</td></tr>
<tr><td>14</td><td>7</td><td>8</td><td>8</td></tr>
<tr><td>16</td><td>7</td><td>8</td><td>8</td></tr>
<tr><td>18</td><td>8</td><td>8</td><td rowspan="4">[3]</td><td>9</td><td rowspan="2">[3]</td></tr>
<tr><td>20</td><td>8</td><td>9</td><td>9</td></tr>
<tr><td>22</td><td>8</td><td>8</td><td>9</td><td rowspan="5">new</td></tr>
<tr><td>24</td><td>8</td><td>9</td><td>≥ 9</td></tr>
<tr><td>26</td><td>8</td><td>9</td><td rowspan="4">[5]</td><td>≥ 9</td></tr>
<tr><td>28</td><td>9</td><td>9</td><td>≥ 9</td></tr>
<tr><td>30</td><td>9</td><td>9</td><td>≥ 9</td></tr>
<tr><td>32</td><td>9</td><td>9</td><td>≥ 9</td></tr>
</table>

2 Preliminaries

We recall in this section some notions and useful results that will be used throughout this paper.

2.1 Generalized Feistel Networks

Generalized Feistel Networks have been introduced by [7] as a generalization of Type-2 Feistel construction [12]. Roughly, the cycle shift performed at each round in [12] is replaced by an arbitrary permutation leading to stronger schemes with better diffusion if the permutation is chosen wisely.

Definition 1. *A* Generalised Feistel Network *(GFN) is defined by a number k of Feistel pairs, a word size n, a number of rounds r, a permutation π over $2k$ elements (named* blocks*), and $r \cdot k$ cryptographic keyed functions F_j^i from $\mathbb{F}_2^n$ to $\mathbb{F}_2^n$ (with $1 \leq i \leq r$, and $1 \leq j \leq k$). The ciphertext of a message of size $2k \cdot n$ is given by $\mathcal{R}_r \circ \ldots \circ \mathcal{R}_1$, where $\mathcal{R}_i$ is the round function:*

$$\mathcal{R}_i : (X_0, \ldots, X_{2k-1}) \rightarrow \pi(X_0 \oplus F_1^i(X_1), X_1, \ldots, X_{2k-2} \oplus F_k^i(x_{2k-1}), X_{2k-1})$$

In this paper, neither the word size n, nor the exact definition of the keyed functions F_i^j are relevant. Hence, we simply use F hereafter, and we denote GFN_π^k a GFN with k Feistel pairs using permutation π.

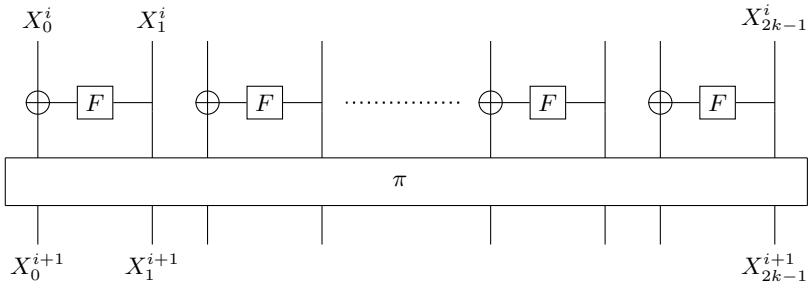


Fig. 1. Round function $\mathcal{R}_i$ of a GFN with k Feistel pairs

In the following, we denote by $X^i = (X_0^i, X_1^i, \ldots, X_{2k-1}^i)$ the input data of the $i+1^{\text{th}}$ round for $i \geq 0$. We say that X_j^i is an even block when j is even, and an odd one otherwise. An illustration for round $\mathcal{R}_i$ is given in Figure 1.

2.2 Diffusion Round

In [10], it has been observed that the *diffusion round* of a permutation π (denoted $DR(\pi)$) is closely related to the security of the corresponding GFN against some of the attacks mentioned above. Intuitively, the diffusion round is the round at which full diffusion is achieved. In other words, assuming good enough functions $F_{i,j}$, the diffusion round is the round from which every bit of the ciphertext depends on every bit of the plaintext. We now formally recall the definition of this notion.

Given $r > 0$ and $i, j \in \{0, \ldots, 2k-1\}$, if X_i^r is expressed by a formal expression containing a non-zero term in X_j^0, we say that X_j^0 *diffuses* to X_i^r, and we say that X_j^0 fully diffuses after r rounds when X_j^0 diffuses to X_i^r for all $i \in \{0, \ldots, 2k-1\}$. For instance, we have that X_0^0 diffuses to $X_{\pi(0)}^1$ whereas X_1^0 diffuses to both $X_{\pi(0)}^1$ and $X_{\pi(1)}^1$. In general, an even block X_i^r will only diffuse

to its successor $X^{r+1}_{\pi(i)}$, whereas an odd block X^r_i will diffuse to its successor $X^{r+1}_{\pi(i)}$ and the successor of its even neighbour $X^{r+1}_{\pi(i-1)}$.

Definition 2. *Given a permutation π over $2k$ elements, we denote $DR_i(\pi)$ as the minimum number of rounds r such that X^0_i* fully diffuses *after r rounds. Then, the* diffusion round *of a permutation π is given by* $DR(\pi) = \max_{0 \leq i < 2k}\{DR_i(\pi)\}$.

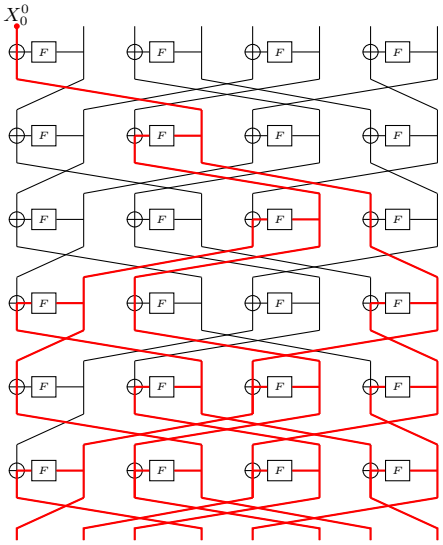


Fig. 2. Diffusion of X^0_0 after $r = 6$ successive rounds

Example 1. Let π = (3,0,5,6,1,2,7,4). This is an even-odd permutation. Figure 2 illustrates the diffusion of X^0_0 after successive rounds. For instance, we have that X^0_0 diffuses to X^2_5 and X^2_6, and full diffusion regarding X^0_0 is reached after 6 rounds, thus $DR_0(\pi) = 6$.

In GFN, decryption is made using π^{-1} and thus we want full diffusion to be effective for π and π^{-1}. We denote $DR^*(\pi) = \max(DR(\pi), DR(\pi^{-1}))$.

As recalled in introduction, finding permutations minimizing the diffusion round has deserved a lot of attention during the past few years. To ease the problem of finding optimal permutations, the focus has been made on even-odd permutations as they seem to achieve better diffusion [3,5]. The belief that even-odd permutations are better has only been formally established by exhausting all the optimal permutations up to $2k = 20$ [3]. In this paper, relying on a novel algorithm based on iterative path building, we will show that this is true up to $2k = 32$.

3 Path Building Algorithm

In this section, we first explain how to represent a permutation π over $2k$ elements as a graph before describing our algorithm. This representation fits well the understanding of our algorithm since its core idea is to build paths. This graph will also be of great help to propose a new characterization of the notion of diffusion round.

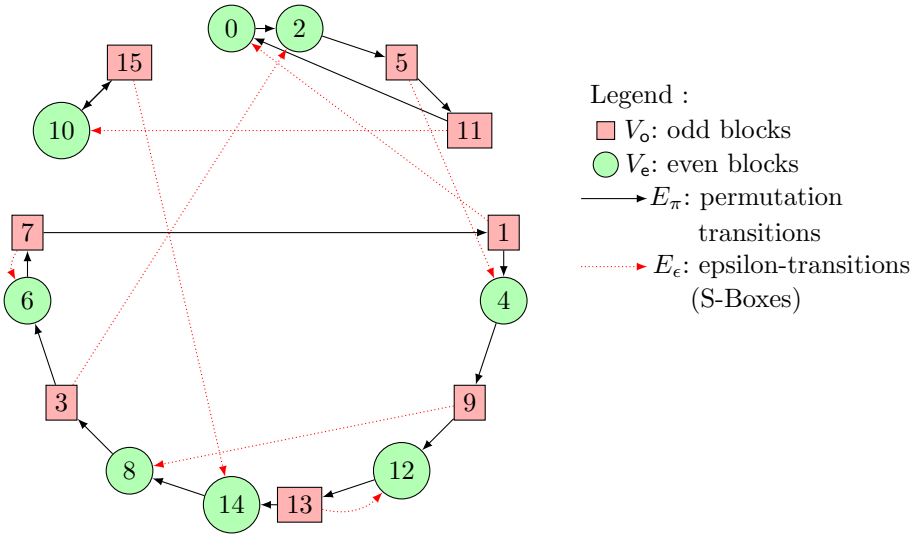
3.1 Graph Representation of a Feistel Permutation

Definition 3. *Given a permutation π over $2k$ elements, the* Feistel permutation graph *associated to π is the graph $G_\pi = (V, E)$ where:*

- $V = V_\mathsf{e} \cup V_\mathsf{o}$ *with* $V_\mathsf{e} = \{0, 2, \ldots, 2k-2\}$, *and* $V_\mathsf{o} = \{1, 3, \ldots, 2k-1\}$;
- $E = E_\epsilon \cup E_\pi$ *with* $E_\epsilon = \{(1,0), (3,2), (5,4), \ldots, (2k-1, 2k-2)\}$, *and* $E_\pi = \{(u, v) \mid u, v \in V \wedge \pi(u) = v\}$.

The set V is the set of all nodes which is divided into two halves, the set of even nodes V_e and the set of odd nodes V_o representing respectively the even blocks and the odd ones. The set E_π is the set of all the edges of the permutation π, whereas E_ϵ is the set of edges representing the S-Box passages from the odd to the even nodes (also called *epsilon-transitions*).

Example 2. Let $\pi = (2, 4, 5, 6, 9, 11, 7, 1, 3, 12, 15, 0, 13, 14, 8, 10)$. This is a non-even-odd permutation whose associated Feistel permutation graph is as follows:



In the following, we will often refer to the Feistel permutation graph G_π of a permutation π. The sets $V_\mathsf{e}, V_\mathsf{o}, E_\pi, E_\epsilon$ will be used to represent the even blocks, the odd blocks, the permutation transitions and the ϵ-transitions.

Definition 4. *A* path $p = (e_1, \ldots, e_n)$ *is a finite sequence of edges from* E *which joins two nodes from* V*. Moreover, when* $e_n \in E_\pi$*, such a path is called a* diffusable path *(or* d-path *for short).*

We say that a path p is of *length* ℓ if there are exactly ℓ edges from E_π in p. Note that there can be multiple occurrences of the same edge in a path. We sometimes need to consider d-paths since a Feistel round is composed of one edge in E_ϵ followed by one edge in E_π. Based on this graph representation, we propose a new characterization of $DR(\pi)$.

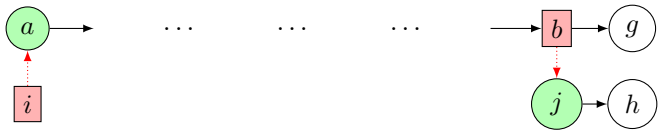
Corollary 1. *$DR(\pi)$ is the smallest integer R such that:*

$$\forall\ u, v \in V\text{, there exists a d-path of length } R \text{ from } u \text{ to } v \text{ in } G_\pi.$$

In order to compute the diffusion round of a permutation π, we can consider the d-paths of a certain length between all pairs of nodes in the graph G_π. As already noticed in [5], in the specific setting of even-odd permutations, it is actually sufficient to consider some specific sets of nodes, and only paths of length $R-1$ to establish that the diffusion round is equal to R. In the following, we formally define these specific paths for the general case (Proposition 1) and the even-odd case (Proposition 2).

Proposition 1. *Let π be a permutation, $DR(\pi)$ is the smallest integer R such that: $\forall a \in V_\mathsf{e}$, $\forall b \in V_\mathsf{o}$, there exists a path of length $R-1$ from a to b in G_π.*

Proof. Let $a \in V_\mathsf{e}$ and $b \in V_\mathsf{o}$ we have that $(a+1, a), (b, b-1) \in E_\epsilon$ with $a+1 \in V_\mathsf{o}$ and $b-1 \in V_\mathsf{e}$. Furthermore, we have $g, h \in V$ such that $(b, g), (b-1, h) \in E_\pi$ (see the graph below with $i = a+1$ and $j = b-1$).



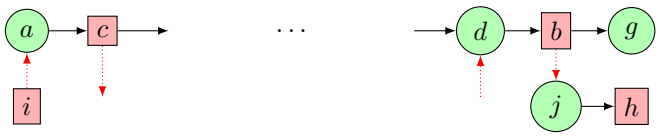
1) From Corollary 1, we know that there is a d-path of length R from a to g, thus there is a path of length $R-1$ from a to b.

2) Now, suppose that there is a $R' < DR(\pi)$ such that $\forall\ a \in V_\mathsf{e},\ b \in V_\mathsf{o}$ there is a path of length $R'-1$ from a to b. We then have a d-path of length R' from i to g, from i to h and from a to h. Since we have these d-paths for each pair $a \in V_\mathsf{e}, b \in V_\mathsf{o}$, we have full diffusion with R' leading to a contradiction. □

For any permutation π, the Proposition 1 reduces the number of paths we have to consider when studying diffusion. In the case of an even-odd permutation, the length of these paths can be further reduced.

Proposition 2. *Let π be an even-odd permutation, $DR(\pi)$ is the smallest integer R such that: $\forall c \in V_\mathsf{o}$, $\forall d \in V_\mathsf{e}$, there exists a path of length $R-3$ from c to d in G_π.*

Proof. Let $b, c \in V_{\mathsf{o}}$ and $a, d \in V_{\mathsf{e}}$ with $(a, c), (d, b) \in E_\pi$. We have that $(a+1, a), (b, b-1) \in E_\epsilon$ with $a+1 \in V_{\mathsf{o}}$ and $b-1 \in V_{\mathsf{e}}$. Furthermore, we have $g, h \in V$ such that $(b, g), (b-1, h) \in E_\pi$ (see the graph below with $i = a+1$ and $j = b-1$).



1) From Proposition 1, we know that there is a path of length $R-1$ from a to b, thus there is a path of length $R-3$ from c to d.

2) Now suppose that there is $R' < DR(\pi)$ such that $\forall\ c \in V_{\mathsf{o}},\ d \in V_{\mathsf{e}}$ there is a path of length $R'-3$ from c to d. We then have a d-path of length R' from i to g, from i to h and from a to h. Since we have these d-paths for all pairs $a \in V_{\mathsf{e}}, b \in V_{\mathsf{o}}$ then we have full diffusion with R' leading to a contradiction. □

3.2 The MakePath Algorithm

We present a new algorithm to search for permutations with optimal diffusion round. Our algorithm is based on path building to efficiently enumerate permutations with full diffusion or any other path-based property. Thanks to Propositions 1 and 2, we will only consider paths of length $R-3$ from odd to even nodes in the even-odd case and paths of length $R-1$ from even to odd nodes in the general case. To obtain effective procedures, we enumerate the paths while building a Feistel permutation graph. With this method, the more paths we add to the graph, the fewer possibilities remain for the following ones. Thanks to this, we can also define a strategy to cut the search as soon as possible by trying the paths with the least possibilities first. Our algorithm is composed of the three following functions:

- MakePath builds all the possible paths from a node a to a node b and is described in Algorithm 1. Starting from node a, the function calls itself on each possible next node for the path until all paths reach b with the length R. More precisely, on a node x, there is only three possibilities. If x is odd, there is one call to the even node $x-1$. In this call, the length l does not decrease because ϵ-transitions are not counted in the path length (line 2-3). If $\pi[x]$ has already been fixed, we have no choice, and thus we follow it (line 4-5). If $\pi[x]$ is free, we have to try all the remaining free nodes (line 7-9). On each valid path, the function calls NextPath that will choose the next path to build (line 13).
- HasProperty checks whether the property of interest is satisfied between two nodes. For example, when considering the full diffusion property, we have to check whether a path of length R exists between 2 nodes (more details in Section 4).
- NextPath chooses two nodes a and b that does not have the property described in HasProperty. If such a pair of nodes exists, it calls MakePath

on it to link them with the next path. It is described in Algorithm 2. For the choice of a and b, the strategy consists of starting by the paths with the least possibilities. To do so, we can either count the remaining possible paths during the search, or we can set a static path priority (more details in Section 4.1).

Algorithm 1: MakePath(x, π, b, l)

```
Data: x: current node, π: partial permutation, b: target node, l: remaining
      length to reach R
if l > 0 then
    if x is odd then
        MakePath(x − 1, π, b, l);
    if π[x] is fixed then
        MakePath(π[x], π, b, l − 1);
    else
        for all y not used in π do
            π[x] ← y;
            MakePath(y, π, b, l − 1);
        end
        free π[x];
    end
else if x = b then NextPath(π) ;
```

Algorithm 2: NextPath(π)

```
Data: π: partial permutation
for all (a, b) given by Strategy() do
    if ¬HasProperty(a, π, b, R) then
        MakePath(a, π, b, R);
        return;
end
Add π to solution pool
```

Our algorithm starts by a call to NextPath with an undefined permutation and a given global parameter R. It stops when one of the following conditions holds:

1. There is no possible path from a to b, and thus there is no solution.
2. The permutation is complete, i.e. fully defined: it is a solution if HasProperty is true for each pair of nodes.
3. The algorithm ends without fixing the whole permutation. In this case, any completion of the permutation lead to a valid solution.

Once all the recursive branches of our algorithm have been explored, all the paths of length R have been exhausted. Thus, at the end of the algorithm, we find all the permutations achieving full diffusion at round R if any. The algorithm can build these permutations from scratch, but it will find a lot of similar solutions. Indeed, starting by a graph like the one given in Example 2, a similar graph can be obtained by simply relabelling the Feistel pairs. To avoid these redundancies, we need to break some symmetries before running the search. To do so, we will rely on the notion of *skeleton* defined in the following section.

3.3 Skeletons

As explained in [3], in the even-odd case, the permutation can be split in two parts, the odd to even edges and the even to odd edges. This makes the search easier $(k!)^2$. Moreover, half of the permutation can be further reduced to all its possible cycle decompositions to break some symmetries. This reduces the search to $\mathcal{N}_k k!$ where $\mathcal{N}_k$ is the number of partitions of k. In the following, we propose a generalization of the cycle decompositions to consider non-even-odd permutations as well, and we rely for that on our graph representation.

Definition 5 (ϵ-cycle). *An ϵ-cycle is a path $c = (e_1, \ldots, e_{2l})$ in which the first and last nodes are equal and edges alternate between E_π and E_ϵ one by one.*

We note a l-ϵ-cycle an ϵ-cycle of size l i.e. with l ϵ-transitions. Moreover, we will only use one representative of $c = (e_1, \ldots, e_{2l})$ and we will not consider all the equivalent ϵ-cycles like $(e_{2l}, e_1, \ldots, e_{2l-1})$ or $(e_1, \ldots, e_{2l}, e_1, \ldots, e_{2l})$. Some examples are given in Figure 3.

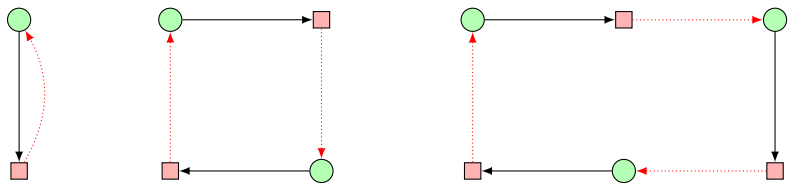


Fig. 3. 1-ϵ-cycle, 2-ϵ-cycle, and 3-ϵ-cycle

Let P be a partition of the integer k. For each $i \in P$, we fix one representative ϵ-cycle of the corresponding size. For example, there are three possible decompositions in ϵ-cycle for $k = 3$, i.e. $\{3\}$, $\{2, 1\}$, and $\{1, 1, 1\}$. This corresponds to one 3-ϵ-cycle, or one 2-ϵ-cycle with one 1-ϵ-cycle, or three 1-ϵ-cycles. This holds only for the even-odd case. To have a similar method in the general case, we rely on ϵ-chains to handle the non-even-odd parts of the permutation.

Definition 6 (ϵ-chain). *An ϵ-chain is a path $ch = (e_1, \ldots, e_{2l+1})$ in which the two first nodes are in V_o and the two last nodes are in V_e. The edges alternate between E_π and E_ϵ one by one.*

We note an l-ϵ-chain an ϵ-chain of size l, i.e. with l ϵ-transitions. Except for the first and the last node, all the nodes in an ϵ-chain are pairwise distinct. Indeed, if a node appears two times in an ϵ-chain, then it is not an ϵ-chain but an ϵ-cycle. However, the first and last node can be in an other structure, like an other ϵ-cycle or an other ϵ-chain, or in the same ϵ-chain, making the ϵ-chain loops on itself. This loop may occur at the beginning of the ϵ-chain, at its end, or on both sides. Some examples of free and looping chains are given in respectively Figures 4 and 5.

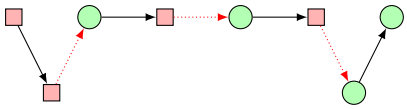


Fig. 4. A 3-ϵ-chain.

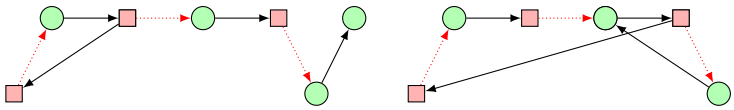


Fig. 5. Two 3-ϵ-chains looping on themselves.

Definition 7. *A* skeleton *of size k is a set of ϵ-cycles and ϵ-chains whose sum of sizes is k.*

Example 3. The skeleton of the graph given in Example 2 is depicted below (see Figure 6). It is composed of three ϵ-cycles of size 3, 1, and 1, as well as two ϵ-chains of size 2 and 1.

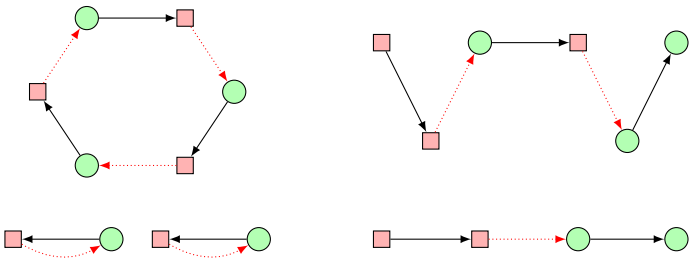


Fig. 6. Skeleton of Example 2

The skeleton of Figure 6 is also valid for graphs similar to Example 2 but with different node numbers. In fact we can permute two pairs of nodes to find a different permutation having the same skeleton. This is why we will only use one representative of each skeleton. The number of skeletons is given by the formula $\sum_{i=0}^{k} \mathcal{N}_i \times \mathcal{N}_{k-i}$ with $\mathcal{N}_i$ the number of partitions of the integer i. The formula has two parts, one for the ϵ-cycles with $\mathcal{N}_i$ and one for the ϵ-chains with $\mathcal{N}_{k-i}$. The formula then sums the skeletons with each possible division into ϵ-cycles and ϵ-chains. For $2k = 16$ there are 22 even-odd skeletons and 163 skeletons with at least one ϵ-chain. For $2k = 32$ there are 231 even-odd skeletons and 5591 non-even-odd ones. Starting from a skeleton, we can complete it with edges to make a Feistel permutation graph. These edges are $\{(a, b) \mid a \in V_{\mathsf{o}}, b \in V_{\mathsf{e}}\}$. Furthermore, if there is one or more ϵ-chain in the skeleton, we also need to fix the first and last node of the ϵ-chains. To do this, we use the MakePath algorithm on each partial solution (skeleton). It is much faster than building the permutation from scratch because some symmetries are broken. The algorithm can be run on each skeleton independently to facilitate parallelization. Nevertheless, there are some symmetries left in our algorithm. Indeed, a l-ϵ-cycle will produce l similar solutions. Moreover, if there are m times the same ϵ-cycle or ϵ-chain, there will be $m!$ similar solutions. Breaking these symmetries in our algorithm increases its running time, and it is left to future work to take them into account effectively.

4 Non-even-odd Case: Search for Optimal Permutations

The search for optimal permutation has been focused on even-odd permutation because in practice, the non-even-odd ones where never better up to $2k = 20$. In this section, we first use our algorithm to show that this is true for up to 32 blocks. We then give a useful example that we found while looking for a general proof.

4.1 Up to $2k = 32$

To test whether a non-even-odd permutation can have a better diffusion round than the even-odd ones, we used Algorithm 1 on all the skeletons having at least one ϵ-chain. We fixed R to be one round less than the diffusion round known for the best even-odd permutation, and ran our algorithm with the property HasPath (described in Algorithm 3).

The running time of our algorithm is highly related to the strategy implemented into the NextPath function (Algorithm 2). The best strategy we found was to first build the paths that start and end on the smallest ϵ-chains. This is because the paths starting by consecutive even nodes and ending by consecutive odd nodes have the least possibilities and therefore are most likely to be impossible to build. The case $2k = 22$ is quite small so we increased R to find the optimal non-even-odd permutations. They are given in Table 2. These optimal permutations have a diffusion round of 9 which is one round more than the optimal even-odd permutations.

Algorithm 3: HasPath(x, π, b, l)

Data: x: current node, π: partial permutation, b: target node, l: remaining length

```
if l > 0 then
    return (x is odd ∧ HasPath(x − 1, π, b, l))
                ∨ (π[x] is fixed ∧ HasPath(π[x], π, b, l − 1));
else return x = b ;
```

Table 2. Optimal Non-even-odd permutations for 2k=22

π =(3, 18, 5, 16, 7, 12, 9, 10, 1, 14, 13, 2, 15, 8, 11, 21, 17, 4, 19, 6, 0, 20)
π =(3, 6, 5, 12, 7, 10, 9, 18, 1, 2, 13, 4, 15, 16, 17, 8, 11, 21, 19, 14, 0, 20)
π =(3, 12, 5, 0, 7, 10, 9, 18, 1, 2, 13, 4, 15, 16, 17, 21, 11, 8, 19, 14, 6, 20)
π =(3, 8, 5, 16, 7, 21, 9, 14, 1, 2, 13, 18, 15, 0, 17, 6, 11, 12, 19, 4, 10, 20)
π =(3, 21, 5, 10, 7, 0, 9, 14, 1, 2, 13, 18, 15, 8, 17, 6, 11, 12, 19, 4, 16, 20)
π =(3, 8, 5, 6, 7, 4, 1, 12, 11, 2, 9, 21, 15, 19, 13, 17, 10, 16, 14, 20, 0, 18)
π =(3, 4, 5, 14, 7, 0, 9, 16, 11, 2, 1, 12, 15, 21, 13, 6, 19, 10, 17, 8, 18, 20)
π =(3, 6, 5, 10, 7, 16, 9, 18, 11, 14, 1, 2, 15, 4, 13, 0, 19, 8, 17, 21, 12, 20)

For $2k = 24$ to $2k = 32$, our algorithm ended without finding any non-even-odd permutations with a better diffusion round than the optimal even-odd ones. As a result, we establish that the non-even-odd permutations do not achieve a better diffusion round than the even-odd permutations up to $2k = 32$. All results are summarized in Table 1 and have been obtained on a 128 core CPU. The hardest instance with 32 blocks and $R = 8$ takes around 8 hours of computing time. In Cauchois et al [3], it is mentioned that "$2^{46.4}$ tests of diffusion rounds" are needed when considering 20 blocks. Actually, our algorithm is faster and tackles this instance in around 8 seconds on our supercomputer. The source code is publicly available at `https://gitlab.inria.fr/agontier/ANewAlgoForGFN`.

4.2 Towards an impossibility result

Intuitively, a non-even-odd permutation should not reach a better diffusion round than the optimal even-odd one. Indeed, every time there are two consecutive odd nodes $u, v \in V_{\mathsf{o}}$ such that $(u, v) \in E_\pi$, there are also somewhere in the graph G_π two consecutive even nodes $x, y \in V_{\mathsf{e}}$ such that $(x, y) \in E_\pi$. We recall that each odd node has two outgoing edges (one in E_π and one in E_ϵ) whereas each even node has only one. Therefore, all the paths starting from the node x have one edge less to achieve full diffusion and any path that passes through (u, v) will gain one edge. Since the number of even to even edges is the same as the number of odd to odd edges, one could think that they compensate.

One of our objective during this work was to provide a formal proof that the diffusion round of the non-even-odd permutations are also bounded by the Fibonacci bound as for even-odd permutations. Thus we made the conjecture that the total number of paths in a permutation graph and its inverse permutation

graph could not exceed the sum of the Fibonacci bounds. However, we found a non-even-odd permutation for which the number of odd nodes reached from the even nodes was in total, and with redundancy, greater than the even-odd Fibonacci bound, which suggests that an improvement of the diffusion round is possible by considering non-even-odd permutations.

Example 4. We consider the permutation π =(3,2,1,5,0,6,7,4) depicted in the leftmost graph of Figure 7. The rightmost one represents π^{-1}.

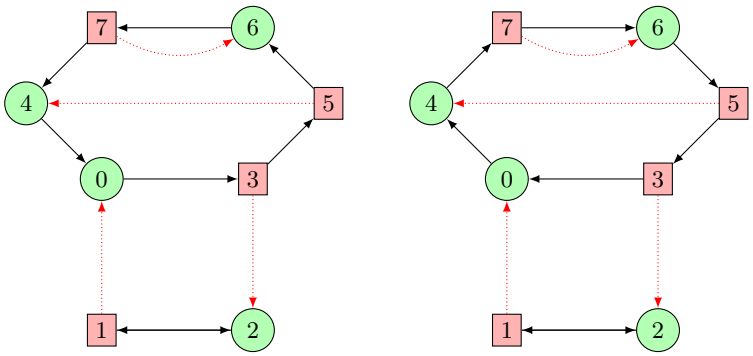


Fig. 7. Permutation graph of π and π^{-1}

On these two graphs, we give in Table 3 the number of paths of length $R = 5$ that ends on an odd node from each even node. There are 22 paths for π, and 21 paths for π^{-1}.

Table 3. Number of paths in π and π^{-1}

start node	0	2	4	6
number of paths	5	8	5	4

start node	0	2	4	6
number of paths	4	5	5	7

When considering only the even-odd permutations, the maximum number of paths given by the Fibonacci suite is 5 for each node and thus $4 \times 5 = 20$ in total. This example shows that the diffusion round in the general case (i.e. considering both even-odd and non-even-odd permutations) cannot be bounded by the Fibonacci suite if we consider the sum of all paths on π and π^{-1}. However, we may note that there is one node (e.g. node 6 for π) having less paths than the Fibonacci suite. We always observe this phenomenon on the permutations we considered. We think that to establish an impossibility result (a non-even-odd permutation can not be better than the optimal even-odd one), we should focus on these nodes.

5 Even-odd Case: Search for New Properties

As studied in the literature, the diffusion round is a property that can be used to find *good* Feistel permutations. This criteria is tied to the resistance of the resulting ciphertext against e.g. impossible differentials, saturation attacks and pseudorandomness analysis [10]. However, permutations with optimal diffusion round can also be weak against other cryptanalysis techniques. For instance, the designers of `WARP` [1] selected a permutation achieving full diffusion in 10 rounds while permutations with a diffusion round of 9 actually exist. The main reason is that all optimal permutations for the diffusion round are much weaker regarding truncated differential cryptanalysis than the one they selected. These permutations require at least 32 rounds to reach 64 active S-Boxes, while the permutation used in `WARP` (which is non optional w.r.t. the diffusion round) only requires 19 rounds to reach the same resistance.

Therefore, it would be interesting to look for other properties which might lead to stronger ciphers. With our algorithm it is quite simple to change the property we are looking for as we only need to provide a new HasProperty function. In this section, we thus propose several properties derived from the diffusion round and study the quality of their solutions against truncated differential cryptanalysis. We consider two properties, the first one is a generalization of the diffusion round where we consider not one but X paths between each pair of blocks. The second one consists of counting the S-Boxes on each path instead of the paths themselves.

5.1 Number of Paths

The diffusion round property ensures that each solution has at least one d-path of length R between each pair of blocks. We propose a new property parameterized by an integer X, namely X-DR, which extends the diffusion round to at least X d-paths of length R between each pair of blocks.

Definition 8. *X-$DR(\pi)$ is the smallest integer R such that:*

$$\forall u, v \in V, \text{ there are } X \text{ d-paths of length } R \text{ from } u \text{ to } v \text{ in } G_\pi.$$

This new property introduces the parameter X denoting the minimum number of paths we want between each pair of nodes. When $X = 1$, this corresponds to the full diffusion property. To use this new property in our algorithm, the call to HasProperty line 2 of Algorithm 2 is replaced by a call to NumberOfPaths with the slight modification that this number of paths must be greater or equal to the parameter X. This function counts the number of paths between two nodes, it is given in Algorithm 4.

Since we want more than one path between two nodes, the function MakePath may need to create multiple paths. Due to these multiple paths, we must set an order between paths to prevent introducing new symmetries. For example, we should not build a path p after a path q if we already tried to build them in the other order. Proposition 2, stated and proved for the diffusion round, is still

Algorithm 4: NumberOfPath(x, π, b, l)

```
Data: x: current node, π: partial permutation, b: target node, l: remaining
      length
if l > 0 then
    if π[x] is fixed then
        if x is odd then
            return
              NumberOfPath(x − 1, π, b, l) + NumberOfPath(π[x], π, b, l − 1);
        else return NumberOfPath(π[x], π, b, l − 1) ;
    else return 0 ;
else
    if x = b then return 1 ;
    else return 0 ;
```

valid when considering X-DR. It is stated in Proposition 3, and for sake of completeness the proof is given in Appendix.

Proposition 3. *Let π be an even-odd permutation, X-$DR(\pi)$ is the smallest integer R such that: $\forall c \in V_{\mathsf{o}}, d \in V_{\mathsf{e}}$, there are X paths of length $R-3$ from c to d in G_π.*

To compare this criterion w.r.t. truncated differential analysis, we computed the minimal number of active S-Boxes for each possible permutation for $k = 6$, $k = 7$, and $k = 8$. We give in Table 4 the best number (i.e. the minimum one) we obtained from round 1 to round 16 :

Table 4. Best minimal number of active S-Boxes for each round

k \ Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
6	0	1	2	3	4	6	8	11	14	16	19	22	24	26	28	29
7	0	1	2	3	4	6	8	11	14	19	23	26	28	30	33	35
8	0	1	2	3	4	6	8	11	14	19	22	26	29	31	34	37

Then, we took the 500 first solutions given by our algorithm for the criterion. We computed the minimal number of active S-Boxes for each of these solutions, and we counted the number of solutions that reached the optimal value for each round from 10 to 16. The results are given in Table 5. Note that to get 500 solutions, we sometimes needed to consider the criterion to a higher round than the optimal one. For example the diffusion round for $k = 6$ is $R = 8$. However, there are only 245 solutions with these parameters. Thus, we had to increase R until we reached 500 solutions. This is summarized in the range column of Table 5.

For $k = 8$, we do not see a trend and we have similar results for $k = 7$ and $k = 6$. In fact, the property seems uncorrelated to the optimal number of active

Table 5. Number of solutions with an optimal number of active S-Boxes from round 10 to round 16 in the 500 first solutions considering $k = 8$

Round / X-DR	10	11	12	13	14	15	16	Range
1 path	9	16	0	0	0	0	0	8
2 paths	24	37	0	0	0	0	0	9
3 paths	0	4	0	0	0	0	0	10
4 paths	15	15	0	0	0	0	0	10-11
5 paths	0	1	0	0	0	0	0	11
6 paths	9	9	0	0	0	0	0	11-12
7 paths	0	0	0	0	0	0	0	12
8 paths	0	0	0	0	0	0	0	12

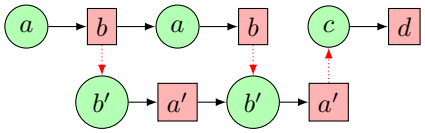
S-Boxes. We can see that increasing the parameter X increases the round R we need to go to find 500 solutions. Indeed when we search for two paths instead of one, the property is so strict that there are no solutions for $R = 8$. We also see that few to none of the 500 solutions are optimal in general.

5.2 Number of S-Boxes

Having X paths between each pair of blocks does not ensure that these paths are "good" from the differential analysis point of view. Instead of constraining the number of paths, we propose to ensure that a minimum number of S-Boxes are present in the d-paths between each pair of blocks.

Definition 9. *X-$SB(\pi)$ is the smallest integer R such that: $\forall u, v \in V$, there are X S-Boxes traversed by d-paths of length R from u to v in G_π. A S-Box reached by two paths of the same length will be counted only once.*

For example, in the two paths of length 5 from a to d depicted below, the S-Box corresponding to the red edge (b, b') will be counted twice (as it occurs at two different lengths), whereas the S-Box corresponding to the red edges (a', c) will be counted only once (even if it occurs on both paths).



To use this new property in our algorithm, the call to HasProperty line 2 of Algorithm 2 is replaced by a call to DetectSBoxes with the slight modification that the sum of detected S-Boxes must be greater or equal to the parameter X. DetectSBoxes is described in Algorithm 5. Unlike paths, we cannot simply count the S-Boxes because of the redundancy described in the previous example. We have to use a Boolean matrix of dimension 2 or an equivalent structure to remember at which path length l we encounter each S-Box.

Algorithm 5: DetectSBoxes(x, π, b, l, M)

```
Data: x: current node, π: partial permutation, b: target node, l: remaining
      length, M: Boolean matrix of dimension 2
M0 ← Matrix filled with false values;
if l > 0 then
    if π[x] is fixed then
        if x is odd then
            M2 ← copy(M);
            M2[x, l] ← true;
            M3 ← DetectSBoxes(π[x], π, b, l − 1, M);
            M4 ← DetectSBoxes(x − 1, π, b, l, M2);
            return Bit-wise OR(M3,M4);
        else return DetectSBoxes(π[x], π, b, l − 1, M) ;
    else return M0 ;
else
    if x = b then return M ;
    else return M0 ;
```

Proposition 2, stated and proved for the diffusion round, is also valid when considering X-SB. It is stated in Proposition 4, and for sake of completeness, the proof is given in Appendix.

Proposition 4. *Let π be an even-odd permutation, X-$SB(\pi)$ is the smallest integer R such that: $\forall c \in V_o, d \in V_e$, there are X S-Boxes traversed by paths of length $R-3$ from c to d in G_π. A S-Box reached by two paths of the same length will be counted only once.*

As for the X-DR criteria, we looked at the quality of optimal permutations for the X-SB criteria regarding truncated differential cryptanalysis for $k = 6$, $k = 7$, and $k = 8$. The results are summarized in Table 6 for $k = 8$ and are similar for lower k.

Overall, these two new properties did not bring better solutions for the truncated differential analysis. For each criterion, the number of optimal solution in the 500 first solutions is very low.

5.3 TWINE

Finally, we studied our criteria on the permutation used in `TWINE` [11]. The values of our criteria for `TWINE` are given in Table 7. To see if these are good values, we used our algorithm to enumerate permutations with strictly greater values for our criteria. The algorithm concluded that there is no permutation with a better X-SB than `TWINE` up to $X = 22$. The experimentation was not done beyond due to its computational cost. However, `TWINE` is not optimal for 4-DR and 6-DR. There is only one permutation that is optimal on 4-DR and 6-DR at the same time. This permutation is π =(3, 4, 5, 8, 1, 12, 9, 10, 11, 2, 7, 14, 13, 6, 15,

Table 6. Number of solutions with an optimal number of active S-Boxes from round 10 to round 16 in the 500 first solutions considering $k = 8$

Round / X-SB	10	11	12	13	14	15	16	Range
1 S-Box	25	44	0	0	0	0	0	8
2 S-Boxes	25	44	0	0	0	0	0	8
3 S-Boxes	0	1	0	0	0	0	0	9
4 S-Boxes	18	30	0	0	0	0	0	9
5 S-Boxes	4	12	0	0	0	0	0	9-10
6 S-Boxes	4	9	0	2	2	2	0	9-10
7 S-Boxes	0	6	0	0	0	0	0	10
8 S-Boxes	0	9	0	0	0	0	0	10-11
9 S-Boxes	0	1	0	1	15	1	0	11
10 S-Boxes	0	4	0	0	0	0	0	11
11 S-Boxes	0	6	0	0	0	0	0	11-12
12 S-Boxes	0	0	0	0	0	0	0	11-12

Table 7. X-DR and X-SB values for `TWINE`

1 to 2-SB	3 to 6-SB	7 to 8-SB	9 to 14-SB	15 to 22-SB
8	9	10	11	12
1-DR	2-DR	3-DR	4 to 5-DR	6 to 9-DR
8	9	10	11	12

0). To compare it with `TWINE`, we computed the truncated differentials on both permutations in Table 8.

Table 8. Truncated Differentials for `TWINE` and π

Round	8	9	10	11	12	13	14	15	16
TWINE	**11**	**14**	18	**22**	24	27	30	32	35
π	**11**	**14**	**19**	**22**	24	26	28	30	32

This new permutation π is better than `TWINE` and optimal at round 10. However, it is worse for rounds 13 to 16. In fact, in all the $k = 8$ permutations, none can reach the optimal number of active S-Boxes at every round.

6 Conclusion

In this paper, we proposed a new generic algorithm based on path building to enumerate permutations regarding a chosen property for Generalized Feistel Networks. The main advantage of our algorithm is that it is not restricted to the even-odd permutations nor the diffusion round property. Furthermore, it was fast enough to prove that no non-even-odd permutation reaches a strictly better

diffusion round than optimal even-odd permutations up to 32 blocks. Thus we fully solved the problem opened by Suzaki and Minematsu in [10] and partially solved by Derbez *et al.* in [5].

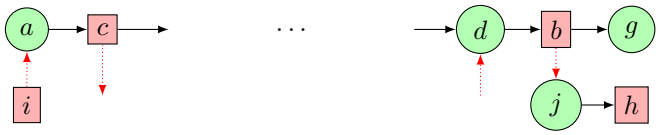
However, in both [5] and [1], it was highlighted that optimal permutations regarding the diffusion round might still lead to ciphers far from offering an optimal resistance against differential cryptanalysis. We thus tried two more complex properties derived from the diffusion round and studied the quality of the solutions they provide against truncated differential cryptanalysis.

Future work. We believe that providing a formal proof that there is always at least one even-odd permutation optimal with respect to the diffusion round would be a great result which should lead to a better understanding of GFN. We are confident that obtaining such a proof is possible and the particular example described Section 4.2 seems to be a good starting point. Another interesting problem concerns properties that would ensure some level of resistance against differential cryptanalysis. Indeed, our work clearly shows that permutations ensuring fast and strong diffusion are rarely optimal regarding this type of distinguishers.

Appendix A Proofs of Proposition 3 and 4

Proposition 3 *Let π be an even-odd permutation π, X-$DR(\pi)$ is the smallest integer R such that: $\forall c \in V_\mathsf{o}, d \in V_\mathsf{e}$, there are X paths of length $R-3$ from c to d in G_π.*

Proof. Let $b, c \in V_\mathsf{o}$ and $a, d \in V_\mathsf{e}$ with $(a, c), (d, b) \in E_\pi$. We have that $(a+1, a), (b, b-1) \in E_\epsilon$ with $a+1 \in V_\mathsf{o}$ and $b-1 \in V_\mathsf{e}$. Furthermore, we have $g, h \in V$ such that $(b, g), (b-1, h) \in E_\pi$ (see the graph below with $i = a+1$ and $j = b-1$).

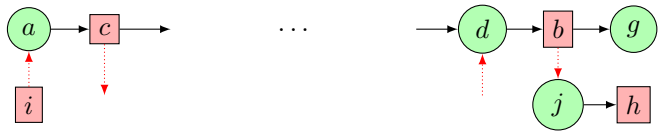


1) From Definition 8, we know that there is X d-paths of length R from a to g, thus there is X paths of length $R-3$ from c to d.

2) Now suppose that there is $R' < X$-$DR(\pi)$ such that $\forall\ c \in V_\mathsf{o},\ d \in V_\mathsf{e}$ there is X paths of length $R'-3$ from c to d. We then have X d-paths of length R' from i to g, from i to h and from a to h. Since we have these d-paths for all pairs $a \in V_\mathsf{e}, b \in V_\mathsf{o}$ then we have full diffusion with X-$DR(\pi) = R'$ and thus the contradiction X-$DR(\pi) < X$-$DR(\pi)$. □

Proposition 4 *Let π be an even-odd permutation π, X-$SB(\pi)$ is the smallest integer R such that: $\forall c \in V_\mathsf{o}, d \in V_\mathsf{e}$, there are X S-Boxes traversed by paths of length $R-3$ from c to d in G_π. A S-Box reached by two paths at the same time will be counted only once.*

Proof. Let $b, c \in V_{\mathsf{o}}$ and $a, d \in V_{\mathsf{e}}$ with $(a, c), (d, b) \in E_\pi$. We have that $(a+1, a), (b, b-1) \in E_\epsilon$ with $a+1 \in V_{\mathsf{o}}$ and $b-1 \in V_{\mathsf{e}}$. Furthermore, we have $g, h \in V$ such that $(b, g), (b-1, h) \in E_\pi$ (see the graph below with $i = a+1$ and $j = b-1$).



1) From Definition 9, we know that there is X S-Boxes in all the d-paths of length R from a to g, thus there is X S-Boxes in all paths of length $R-3$ from c to d.

2) Now suppose that there is $R' < X\text{-}SB(\pi)$ such that $\forall\ c \in V_{\mathsf{o}},\ d \in V_{\mathsf{e}}$ there is X S-Boxes in all the paths of length $R'-3$ from c to d. We then have X S-Boxes in all the d-paths of length R' from i to g, from i to h and from a to h. Since we have these d-paths for all pairs $a \in V_{\mathsf{e}}, b \in V_{\mathsf{o}}$ then we have full diffusion with $X\text{-}SB(\pi) = R'$ and thus the contradiction $X\text{-}SB(\pi) < X\text{-}SB(\pi)$. □

References

1. Banik, S., Bao, Z., Isobe, T., Kubo, H., Liu, F., Minematsu, K., Sakamoto, K., Shibata, N., Shigeri, M.: WARP : Revisiting GFN for lightweight 128-bit block cipher. In: Dunkelman, O., Jr., M.J.J., O'Flynn, C. (eds.) Selected Areas in Cryptography - SAC 2020 - 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. Lecture Notes in Computer Science, vol. 12804, pp. 535–564. Springer (2020). https://doi.org/10.1007/978-3-030-81652-0_21, `https://doi.org/10.1007/978-3-030-81652-0_21`
2. Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., Wingers, L.: The SIMON and SPECK Families of Lightweight Block Ciphers. IACR Cryptology ePrint Archive **2013**, 404 (2013), `http://eprint.iacr.org/2013/404`
3. Cauchois, V., Gomez, C., Thomas, G.: General diffusion analysis: How to find optimal permutations for generalized type-ii feistel schemes. IACR Trans. Symmetric Cryptol. **2019**(1), 264–301 (2019). https://doi.org/10.13154/tosc.v2019.i1.264-301, `https://doi.org/10.13154/tosc.v2019.i1.264-301`
4. Derbez, P., Fouque, P.: Automatic search of meet-in-the-middle and impossible differential attacks. In: Robshaw, M., Katz, J. (eds.) Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9815, pp. 157–184. Springer (2016). https://doi.org/10.1007/978-3-662-53008-5_6, `https://doi.org/10.1007/978-3-662-53008-5_6`
5. Derbez, P., Fouque, P., Lambin, B., Mollimard, V.: Efficient search for optimal diffusion layers of generalized feistel networks. IACR Trans. Symmetric Cryptol. **2019**(2), 218–240 (2019). https://doi.org/10.13154/tosc.v2019.i2.218-240, `https://doi.org/10.13154/tosc.v2019.i2.218-240`
6. DES: Data Encryption Standard. FIPS PUB 46, Federal information processing standards publication 46 (1977)

7. Nyberg, K.: Generalized feistel networks. In: Kim, K., Matsumoto, T. (eds.) Advances in Cryptology - ASIACRYPT '96, International Conference on the Theory and Applications of Cryptology and Information Security, Kyongju, Korea, November 3-7, 1996, Proceedings. Lecture Notes in Computer Science, vol. 1163, pp. 91–104. Springer (1996). https://doi.org/10.1007/BFb0034838, https://doi.org/10.1007/BFb0034838
8. Shi, D., Sun, S., Derbez, P., Todo, Y., Sun, B., Hu, L.: Programming the demirciselçuk meet-in-the-middle attack with constraints. In: Peyrin, T., Galbraith, S.D. (eds.) Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part II. Lecture Notes in Computer Science, vol. 11273, pp. 3–34. Springer (2018). https://doi.org/10.1007/978-3-030-03329-3_1, https://doi.org/10.1007/978-3-030-03329-3_1
9. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., Shirai, T.: Piccolo: An Ultra-Lightweight Blockcipher. In: Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings. pp. 342–357 (2011). https://doi.org/10.1007/978-3-642-23951-9_23, https://doi.org/10.1007/978-3-642-23951-9_23
10. Suzaki, T., Minematsu, K.: Improving the generalized feistel. In: Hong, S., Iwata, T. (eds.) Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers. Lecture Notes in Computer Science, vol. 6147, pp. 19–39. Springer (2010). https://doi.org/10.1007/978-3-642-13858-4_2, https://doi.org/10.1007/978-3-642-13858-4_2
11. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: Twine : A lightweight block cipher for multiple platforms. In: Knudsen, L.R., Wu, H. (eds.) Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7707, pp. 339–354. Springer (2012). https://doi.org/10.1007/978-3-642-35999-6_22, https://doi.org/10.1007/978-3-642-35999-6_22
12. 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.) Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings. Lecture Notes in Computer Science, vol. 435, pp. 461–480. Springer (1989). https://doi.org/10.1007/0-387-34805-0_42, https://doi.org/10.1007/0-387-34805-0_42