Generic Attacks on Unbalanced Feistel Schemes with Contracting Functions

Jacques Patarin[1], Valérie Nachef[2], and Côme Berbain[3]

[1] Université de Versailles
45 avenue des Etats-Unis, 78035 Versailles Cedex, France
[2] Université de Cergy-Pontoise
2 avenue Adolphe Chauvin, 95011 Cergy-Pontoise Cedex, France
[3] France Telecom Research and Development
38-40 rue du Général Leclerc, 92794 Issy-les-Moulineaux, France
jacques.patarin@prism.uvsq.fr
valerie.nachef@u-cergy.fr
come.berbain@orange-ft.com

Abstract. In this paper, we describe generic attacks on unbalanced Feistel schemes with contracting functions. These schemes are used to construct pseudo-random permutations from kn bits to kn bits by using d pseudo-random functions from $(k-1)n$ bits to n bits. We describe known plaintext attacks (KPA) and non-adaptive chosen plaintext attacks (CPA-1) against these schemes with less than 2^{kn} plaintext/ciphertext pairs and complexity strictly less than $O(2^{kn})$ for a number of rounds $d \leq 2k-1$. Consequently at least $2k$ rounds are necessary to avoid generic attacks. For $k = 3$, we found attacks up to 6 rounds, so 7 rounds are required. When $d \geq 2k$, we also describe some attacks on schemes with generators, (i.e. schemes where the d pseudo-random functions are generated) and where more than one permutation is required.

Key words: unbalanced Feistel permutations, pseudo-random permutations, generic attacks, Luby-Rackoff theory, block ciphers.

1 Introduction

Feistel schemes are widely used in symmetric cryptography in order to construct pseudo-random permutations. In trying to design such scheme, one of the natural questions is: what is the the minimum number of rounds required to avoid all the "generic attacks". By generic attacks we mean all the attacks effective with high probability when the round functions are randomly chosen. We are mainly interested in generic attacks with a complexity that is much smaller than a search on all possible inputs of the permutation.

Many results are known on classical (balanced) Feistel schemes. In [7], Luby and Rackoff have shown their famous result: for more than 3 rounds all the generic chosen plaintext attacks on Feistel schemes require at least $O(2^{\frac{n}{2}})$ inputs. Moreover for more than 4 rounds all the generic attacks on adaptive chosen plaintext/ciphertext require at least $O(2^{\frac{n}{2}})$ inputs. These bounds are tight [1, 10]. It has also been proved that to avoid all attacks with less than 2^{2n} computations at least 6 rounds of balanced Feistel schemes are needed [2, 11, 12]. This result is still valid if the round functions are permutations [5, 6]. For more than 6 rounds, some attacks are still possible but with more than 2^{2n} computations [11]. All these results on classical Feistel schemes are summarized in Table 1:

Table 1. Results (from [12]) on G_2^d. For more than 6 rounds more that one permutation is needed or more than 2^{2n} computations are needed in the best known attacks to distinguish G_2^d from a random permutation with an even signature.

	KPA	CPA-1	CPCA-2
G_2^1	1	1	1
G_2^2	$2^{\frac{n}{2}}$	2	2
G_2^3	$2^{\frac{n}{2}}$	$2^{\frac{n}{2}}$	3
G_2^4	2^n	$2^{n/2}$	$2^{\frac{n}{2}}$
G_2^5	$2^{3n/2}$	2^n	2^n
G_2^6	2^{2n}	2^{2n}	2^{2n}
G_2^7	2^{3n}	2^{3n}	2^{3n}
G_2^8	2^{4n}	2^{4n}	2^{4n}
$G_2^d, d \geq 8$	$2^{(k-4)n}$	$2^{(k-4)n}$	$2^{(k-4)n}$

The aim of this paper is to look for similar results for the case of unbalanced Feistel schemes with contracting functions: we call such schemes "contracting Feistel Schemes". A precise definition of these schemes is given in Sect. 2. The case of unbalanced Feistel schemes with expanding functions instead of contracting functions is studied in [4, 14, 15]. Some results on contracting Feistel schemes or on small transformations of these schemes can be found in [8, 9]. In [9], Naor and Reingold studied the security of contracting Feistel schemes with pairwise independent permutations. They show lower bounds for the security of such schemes. Lucks [8] gives some security results on contracting Feistel schemes built with hash functions.

The paper is organized as follows. In Sect. 2 and 3, we introduce notations and present precise definitions of the considered schemes and an overview of our attacks. In Sect. 4, we study attacks for $k = 3$ and $d \leq 6$. Then in Sect. 5, we give attacks for any k and $d \leq 2k - 1$. Finally, Sect. 6 is devoted to what can be done with more than 2^{kn} computations. In particular, we describe attacks against permutation generators. All the results are summarized in the conclusion: these tables extend the above Table 1 to the case of unbalanced Feistel schemes with contracting functions.

2 Notation

Our notation is very similar to that used in [7] and [9]. We also follow the construction given in [9]. $[a, b]$ denotes the concatenation of strings a and b. An Unbalanced Feistel Scheme with Contracting Functions G_k^d is a Feistel scheme with d rounds. At round j, we denote by f_j the round function from $(k-1)n$ bits to n bits. On some input $[I^1, I^2, \ldots, I^k]$, G_k^d produces an output denoted by $[S^1, S^2, \ldots, S^k]$ by going through d rounds. At each round, the last $(k-1)n$ bits of the round entry are used as an input to the round function f_j, which produces n bits. Those bits are xored to the first n bits of the round entry. Finally before going to round $j+1$, the kn bit value is rotated by n bits.

We introduce the internal variable X^j: it is the only n-bit value which is modified at round j and which becomes the k coordinate of the internal state after j rounds. For example, we have:

$$
\begin{aligned}
X^1 &= I^1 \oplus f_1([I^2, \ldots, I^k]),\\
X^2 &= I^2 \oplus f_2([I^3, \ldots, I^k, X^1]),\\
X^3 &= I^3 \oplus f_3([I^4, \ldots, I^k, X^1, X^2]),\\
&\ldots
\end{aligned}
$$

The first round of G_k^d is represented in Fig. 1 below.

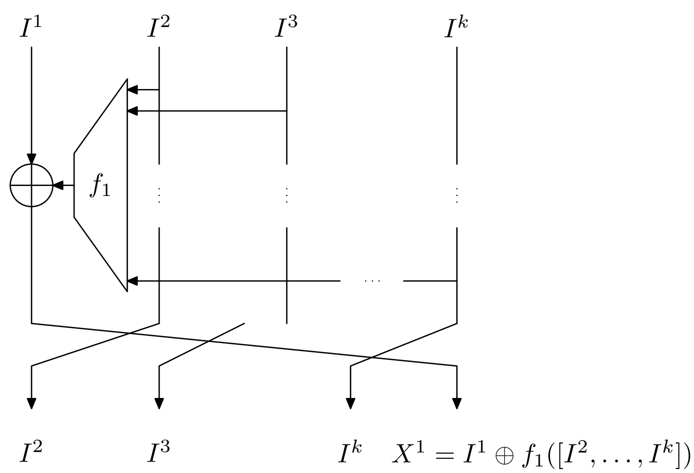

Fig. 1. First Round of G_k^d

3 Overview of the Attacks

We present several attacks that allow us to distinguish G_k^d from a random permutation. Depending on the number of rounds, it is possible to find some relations between the input variables and output variables. Those relations hold conditionally to equalities of some internal variables due to the structure of the Feistel scheme. Our attacks consist in using m plaintexts and ciphertexts tuples and in counting the number $\mathcal{N}_{G_k^d}$ of pairs of these tuples that satisfy the above relations. We then compare $\mathcal{N}_{G_k^d}$ with the equivalent number $\mathcal{N}_{perm}$ if a random permutation is used instead of G_k^d. Our attack is successful, i.e. it is able to distinguish G_k^d from a random permutation if the difference $|E(\mathcal{N}_{G_k^d}) - E(\mathcal{N}_{perm})|$ is much larger than the standard deviation σ_{perm} and than the standard deviation $\sigma_{G_k^d}$, where E denotes the expectancy function.More general cases of succes are also given in the extended version of this paper [13].

In order to compute these values, we need to take into account the fact that the m^2 pairs obtained from the m plaintext/ciphertext tuples are not independent. However their mutual dependence is very small. To compute σ_{perm} and $\sigma_{G_k^d}$, we will use this well-known formula that we will call the "Covariance Formula":

$$V(\sum x_i) = \sum_i V(x_i) + \sum_{i<j} \left[E(x_i, x_j) - E(x_i)E(x_j)\right]$$

where the x_i are random variables.

We can note that for a small number of rounds $d < k$, a distinguishing attack is very easy to find. The output of G_k^d is $[S^1, S^2, \ldots, S^k]$ which is equal to $[I^{d+1}, \ldots, I^k, X^1, \ldots, X^d]$. This shows that we can easily mount a KPA attack with one single message. We just have to test if the first coordinate of the output is equal to the coordinate of rank $d+1$ of the input. This leads us to start investigating attacks for scheme with at least k rounds.

4 Generic Attacks when $k = 3$ and $3 \leq d \leq 6$

We first study schemes with $k = 3$ since this case is slightly different from the general case $k \geq 4$ and since it gives simple examples of what we will do. We have $[S_i^1, S_i^2, S_i^3] = G_3^d([I_i^1, I_i^2, I_i^3])$.

4.1 Attacks on 3 Rounds: G_3^3

G_3^3: 3 rounds, CPA-1 with $m = 2$ messages. Let us choose $I_2^2 = I_1^2$, $I_2^3 = I_1^3$ and $I_2^1 \neq I_1^1$. Then the attack just tests if $S_1^1 \oplus S_2^1 = I_1^1 \oplus I_2^1$. This will occur with probability 1 if f is a G_3^3, and with probability $\simeq \frac{1}{2^n}$ if f is a random permutation. So with three rounds there is a generic attack with two non-adaptive chosen queries and $O(1)$ computations.

G_3^3: 3 rounds, KPA with $m \simeq 2^n$ messages. It is possible to transform this non-adaptive chosen plaintext attack into a known plaintext attack as follows. If we have $m \geq 2^n$ random inputs $[I_i^1, I_i^2, I_i^3]$, then (since $m^2 \geq 2^{2n}$) with a good probability we will have a collision $I_i^2 = I_j^2$ and $I_i^3 = I_j^3, i \neq j$. Then we test if $S_i^1 \oplus S_j^2 = I_i^1 \oplus I_j^1$. Now the attack requires $O(2^n)$ random queries and $O(2^n)$ computations.

4.2 Attacks on 4 Rounds: G_3^4

When the output $[I^1, I^2, I^3]$ is given, we have introduced the internal variable $X^1 = I^1 \oplus f_1([I^2, I^3])$ and the following conditions hold:

$$\begin{cases} I_i^2 = I_j^2 \text{ and } I_i^3 = I_j^3 \Rightarrow X_i^1 \oplus X_j^1 = I_i^1 \oplus I_j^1 \\ I_i^3 = I_j^3 \text{ and } X_i^1 = X_j^1 \Rightarrow S_i^1 \oplus S_j^1 = I_i^2 \oplus I_j^2 \\ X_i^1 = X_j^1 \text{ and } S_i^1 = S_j^1 \Rightarrow S_i^2 \oplus S_j^2 = I_i^3 \oplus I_j^3 \\ S_i^1 = S_j^1 \text{ and } S_i^2 = S_j^2 \Rightarrow S_i^3 \oplus S_j^3 = X_i^1 \oplus X_j^1 \end{cases}$$

The attack exploits the second condition. It proceeds as follows: we choose m messages such that $\forall i$, $I_i^3 = 0$ and $I_i^2 \neq I_j^2$ for all $i \neq j$. We then count $\mathcal{N}_{G_3^4}$ the number of pairs (i, j) with $i < j$ such that $I_i^2 \oplus I_j^2 = S_i^1 \oplus S_j^1$. For a random permutation, this condition appears only by chance. Thus we get:

$$\mathcal{N}_{perm} \simeq \frac{m^2}{2 \cdot 2^n} + O(\frac{m}{2^{\frac{n}{2}}}).$$

Here $0(\frac{m}{2^{\frac{n}{2}}})$ denotes the standard deviation. This can be easily proved using the Covariance Formula, see Appendix A or full version of this article [13].

For G_3^4, the equation $I_i^2 \oplus I_j^2 = S_i^1 \oplus S_j^1$ can occur at random with probability 2^{-n} or from the internal collision $X_i^1 = X_j^1$. Since I_i^3 is equal to zero for all i, we have $X_i^1 = I_i^1 \oplus f_1([I_i^2, 0])$. Sine f_1 is a random function and the I^2 are pairwise distinct, the values $f_1([I_i^2, 0])$ and consequently the X_i^1 are uniformly distributed random variables. Consequently the internal collision $X_i^1 = X_j^1$ appears with probability 2^{-n} and we have:

$$\mathcal{N}_{G_3^4} \simeq \frac{m^2}{2^n} + O(\frac{m}{2^{\frac{n}{2}}})$$

where $O(\frac{m}{2^{\frac{n}{2}}})$ denotes the standard deviation (proof is given below). We can distinguish the two permutations when the difference between the mean values is larger than the standard deviation i.e. when $\frac{m^2}{2^n} \geq \frac{m}{2^{\frac{n}{2}}}$, i.e. for $m \geq 2^{\frac{n}{2}}$. This generic attack requires $O(2^{\frac{n}{2}})$ random queries and $O(2^{\frac{n}{2}})$ computations

As explained previously, we can transform this attack in a known plaintext attack with $m \simeq 2^n$.

Proof of the standard deviation $\sigma_{G_3^4}$
We introduce the following random variables:

$$\begin{cases} \delta_{i,j} = 1 \text{ if } I_i^2 \oplus I_j^2 = S_i^1 \oplus S_j^1 \\ \delta_{i,j} = 0 \text{ otherwise.} \end{cases}$$

Since we have chosen all the I_i^3 equal to zero, we can say equivalently that $\delta_{i,j}$ is equal to one when $f_2([0, X_i^1]) = f_2([0, X_j^1])$. $\mathcal{N}_{G_3^4}$ is defined as $\sum_{i<j} \delta_{i,j}$ and it is easy to compute $E(\delta_{i,j}) = \frac{2}{2^n} - \frac{1}{2^{2n}}$. We now compute the variance

$V(\delta_{i,j}) = E(\delta_{i,j}^2) - E(\delta_{i,j})^2 = E(\delta_{i,j}) - E(\delta_{i,j})^2 = \frac{2}{2^n} - \frac{5}{2^{2n}} + \frac{4}{2^{3n}} - \frac{1}{2^{4n}}$. We recall the Covariance Formula:

$$V(\sum_{i<j} \delta_{i,j}) = \sum_{i<j} V(\delta_{i,j}) + \sum_{i<j,k<l,(i,j)\neq(k,l)} \left[E(\delta_{i,j}\,\delta_{k,l}) - E(\delta_{i,j})\,E(\delta_{k,l})\right].$$

We need to compute $Cov(i,j,k,l) = E(\delta_{i,j}\,\delta_{k,l}) - E(\delta_{i,j})\,E(\delta_{k,l})$ Let us first consider the case, where i,j,k,l are pairwise distinct We need to consider the influence of the equality $f_2([0, X_i^1]) = f_2([0, X_j^1])$ over the equality $f_2([0, X_k^1]) = f_2([0, X_l^1])$. It can only happen if $X_k^1 \neq X_l^1$ and if either $X_k^1 = X_i^1$ and $X_l^1 = X_j^1$ or $X_k^1 = X_j^1$ and $X_l^1 = X_i^1$. In that case we have also $X_i^1 \neq X_j^1$. This event happens with probability $\left(1 - \frac{1}{2^n}\right)\frac{2}{2^{2n}}$ and both equalities have a probability $\frac{1}{2^n}$ instead of $\frac{1}{2^{2n}}$. This gives a covariance equals to

$$\frac{2}{2^{3n}} - \frac{4}{2^{4n}} + \frac{2}{2^{5n}}.$$

The second case is if both equations are sharing an index, for example $i = k$ We need to consider the influence of the equality $f_2([0, X_i^1]) = f_2([0, X_j^1])$ over the equality $f_2([0, X_i^1]) = f_2([0, X_l^1])$. It can only happen if $X_i^1 \neq X_j^1$. This event happens with probability $\left(1 - \frac{1}{2^n}\right)\frac{1}{2^n}$ and both equalities have a probability $\frac{1}{2^n}$ instead of $\frac{1}{2^{2n}}$. This gives a covariance equals to

$$\frac{1}{2^{2n}} - \frac{2}{2^{3n}} + \frac{1}{2^{4n}}.$$

Consequently we have

$$V(\mathcal{N}_{G_3^4}) = \frac{m^2}{2^n} + O\left(\frac{m^3}{2^{2n}}\right) + O\left(\frac{m^4}{2^{3n}}\right)$$

Since m is smaller than 2^n, we get:

$$V(\mathcal{N}_{G_3^4}) \simeq \frac{m^2}{2^n} \text{ and } \sigma_{G_3^4} \simeq \frac{m}{2^{\frac{n}{2}}}.$$

4.3 Attacks on 5 Rounds: G_3^5

For 5 rounds, the internal variables are X^1 and $X^2 = I^2 \oplus f_2([I^3, X^1])$. We have the following conditions:

$$\begin{cases} I_i^2 = I_j^2 \text{ and } I_i^3 = I_j^3 \Rightarrow X_i^1 \oplus X_j^1 = I_i^1 \oplus I_j^1 \\ I_i^3 = I_j^3 \text{ and } X_i^1 = X_j^1 \Rightarrow X_i^2 \oplus X_j^2 = I_i^2 \oplus I_j^2 \\ X_i^1 = X_j^1 \text{ and } X_i^2 = X_j^2 \Rightarrow S_i^1 \oplus S_j^1 = I_i^3 \oplus I_j^3 \\ X_i^2 = X_j^2 \text{ and } S_i^1 = S_j^1 \Rightarrow S_i^2 \oplus S_j^2 = X_i^1 \oplus X_j^1 \\ S_i^1 = S_j^1 \text{ and } S_i^2 = S_j^2 \Rightarrow S_i^3 \oplus S_j^3 = X_i^2 \oplus X_j^2 \end{cases}$$

The attack proceeds as follows: we choose m messages such that $\forall i$, $I_i^2 = 0$, $I_i^3 = 0$ and the I_i^1 values are pairwise distinct. Notice that this directly implies $X_i^1 \oplus X_j^1 = I_i^1 \oplus I_j^1$, so the X_i^1 values are pairwise distinct. Let $\mathcal{N}$ be the number of pairs (i, j), $i < j$ such that $S_i^1 = S_j^1$ and $I_i^1 \oplus I_j^1 = S_i^2 \oplus S_j^2$. With a random permutation, these two conditions appear by chance and we have:

$$\mathcal{N}_{perm} \simeq \frac{m^2}{2 \cdot 2^{2n}} + O(\frac{m}{2^n}).$$

Here $O(\frac{m}{2^n})$ is the standard deviation. For a G_3^5, $S_i^1 = S_j^1$ and $I_i^1 \oplus I_j^1 = S_i^2 \oplus S_j^2$ appear at random or as a consequence of $X_i^2 = X_j^2$ and $S_i^1 = S_j^1$. This gives:

$$\mathcal{N}_{G_3^5} \simeq \frac{m^2}{2^{2n}}.$$

We can distinguish the two permutations when the difference between the mean values is larger than the standard deviation i.e. when $\frac{m^2}{2^{2n}} \geq \frac{m}{2^n}$, or $m \geq 2^n$. Remark: here $m \leq 2^n$ since $I_i^2 = 0$ and $I_i^3 = 0$; so the attack will succeed when $m \simeq 2^n$.

As before this attack leads to a KPA attack with 2^{2n} messages. But there is a better attack as we can see now.

G_3^5: 5 rounds, KPA with $m = 2^{\frac{3n}{2}}$ messages
For this attack, let $\mathcal{N}$ be the number of pairs (i, j), $i < j$, such that $I_i^3 \oplus I_j^3 = S_i^1 \oplus S_j^1$. For a random permutation, we have:

$$\mathcal{N}_{perm} \simeq \frac{m^2}{2 \cdot 2^n} + O(\frac{m}{\sqrt{2^n}})$$

where $\frac{m}{\sqrt{2^n}}$ is the standard deviation, while for G_3^5 we obtain

$$\mathcal{N}_{G_3^5} \simeq \frac{m^2}{2 \cdot 2^n} + \frac{m^2}{2 \cdot 2^{2n}}.$$

We can distinguish the two permutations when the difference between the mean values is larger than the standard deviation i.e. when $\frac{m^2}{2^{2n}} \geq \frac{m}{\sqrt{2^n}}$, i.e. for $m \geq 2^{\frac{3}{2}n}$.

4.4 Attacks on 6 Rounds: G_3^6

For 6 rounds, the internal variables are X^1, X^2 and $X^3 = I^3 \oplus f_3([X^1, X^2])$. We have the following conditions:

$$\begin{cases} I_i^2 = I_j^2 \text{ and } I_i^3 = I_j^3 \Rightarrow X_i^1 \oplus X_j^1 = I_i^1 \oplus I_j^1 \\ I_i^3 = I_j^3 \text{ and } X_i^1 = X_j^1 \Rightarrow X_i^2 \oplus X_j^2 = I_i^2 \oplus I_j^2 \\ X_i^1 = X_j^1 \text{ and } X_i^2 = X_j^2 \Rightarrow X_i^3 \oplus X_j^3 = I_i^3 \oplus I_j^3 \\ X_i^2 = X_j^2 \text{ and } X_i^3 = X_j^3 \Rightarrow S_i^1 \oplus S_j^1 = X_i^1 \oplus X_j^1 \\ X_i^3 = X_j^3 \text{ and } S_i^1 = S_j^1 \Rightarrow S_i^2 \oplus S_j^2 = X_i^2 \oplus X_j^2 \\ S_i^1 = S_j^1 \text{ and } S_i^2 = S_j^2 \Rightarrow S_i^3 \oplus S_j^3 = X_i^3 \oplus X_j^3 \end{cases}$$

The attack proceeds as follows: we choose m messages such that $\forall i$, $I_i^3 = 0$. Let $\mathcal{N}$ be the number of pairs (i, j), $i < j$, such that $I_i^2 = I_j^2$ and $I_i^1 \oplus I_j^1 = S_i^1 \oplus S_j^1$. With a random permutation, we have:

$$\mathcal{N}_{perm} \simeq \frac{m^2}{2.2^{2n}} + O(\frac{m}{2^n})$$

where $O(\frac{m}{2^n})$ is the standard deviation. For a G_3^6, since all the I_i^3 values are equal, $I_i^2 = I_j^2$ and $X_i^2 = X_j^2$ and $X_i^3 = X_j^3$ imply $I_i^1 \oplus I_j^1 = S_i^1 \oplus S_j^1$. We get

$$\mathcal{N}_{G_3^6} \simeq \frac{m^2}{2.2^{2n}} + \frac{m^2}{2 \cdot 2^{3n}}.$$

We can distinguish the two permutations when the difference between the mean values is larger than the standard deviation i.e. when $\frac{m^2}{2^{3n}} \geq \frac{m}{2^n}$, i.e. for $m \geq 2^{2n}$.

We can obviously transform this CPA-1 attack into a KPA attack which will succeed as soon as we have $m \geq 2^{\frac{5n}{2}}$.

4.5 Experimental Results on G_3^6

We have implemented our CPA-1 and KPA attacks against G_3^6 for small values of n ($n = 6$ and $n = 8$). Our experimental values confirm the theoretical results. Our experiments were performed as follows:

- choose randomly an instance of G_3^6
- choose randomly a permutation: for this we use classical balanced Feistel scheme with a large number of rounds (more than 20)
- launch the attack in CPA-1 with $m = 2^{2n}$, in KPA with $m = 2^{3n}$ ($m = 2^{\frac{5n}{2}}$ also works).
- count the number of plaintext/ciphertext pairs satisfying the relations for the G_3^6 function and for the permutation
- iterate this procedure a large number of times (here 1000 times) to evaluate the mean values and the standard deviations
- compute the mean value and the standard deviation for both the G_3^6 function and the permutation

Table 2. Experimental results for KPA and CPA attacks on G_3^6

Attack	n	$\mathcal{N}_{G_3^6}$	$\mathcal{N}_{perm}$	$\mathcal{N}_{G_3^6} - \mathcal{N}_{perm}$	$\frac{m^2}{2 \cdot 2^{4n}}$	$\sigma_{G_3^6}$	σ_{perm}	$\frac{m}{\sqrt{2} \cdot 2^{\frac{3n}{2}}}$
KPA	6	131006	129011	1995	2048	159	372	362.038
KPA	8	8388308	8355787	32521	32768	2862	2833	2896.309
CPA	6	2058	2009	49	32	45	44	45.254
CPA	8	32781	32601	180	128	178	185	182.019

Conclusion. Our experimental values for $\mathcal{N}_{G_3^6} - \mathcal{N}_{perm}$ are very close to the theoretical expected values ($\frac{m^2}{2 \cdot 2^{4n}}$ in KPA and $\frac{m^2}{2 \cdot 2^{3n}}$ in CPA-1). Similarly, our experimental values for ϵ_{perm} are very close to the theoretical expected values ($\frac{m}{\sqrt{2} \cdot 2^{\frac{3n}{2}}}$ in KPA and $\frac{m}{\sqrt{2} \cdot 2^n}$ in CPA-1) . So these simulations confirm that we can distinguish G_3^6 from a random permutation with the complexity that we have given.

5 Generic Attacks when $k \geq 4$ and $k \leq d \leq 2k - 1$

5.1 Attacks for k Rounds

We first describe a CPA-1 attack with two messages. All the blocks of these two messages are equal to zero except the first one. We test if $I_1^1 \oplus I_2^1 = S_1^1 \oplus S_2^1$. Since $S^1 = X^1 = I^1 \oplus f_1([I^2, \ldots I^k])$, this will occur with probability 1 if f is a G_l^k, and with probability 2^{-n} if f is a random permutation. This gives the result.

As usual, we transform this attack into a KPA attack with $m = O(2^{\frac{n(k-1)}{2}})$. In that case with a high probability $I_i^2 = I_j^2$, $I_i^3 = I_j^3$, ..., $I_i^k = I_j^k$. We test again if $S_i^1 \oplus S_i^1 = I_i^1 \oplus I_j^1$.

5.2 Attacks for $k + t$ Rounds, with $1 \leq t < k - 1$

In the CPA-1 attack, we choose $\forall i$, $I_i^{t+2} = \ldots = I_i^k = 0$ and pairwise distinct $[I_i^1, \ldots I_i^t]$. This choice limits the maximal number of plaintext/ciphertext tuples to $m \leq 2^{(t+1)n}$. We then count the number $\mathcal{N}$ of pairs (i, j), $i < j$, such that $I_i^{t+1} \oplus I_j^{t+1} = S_i^1 \oplus S_j^1$. For a random permutation, we have:

$$\mathcal{N}_{perm} \simeq \frac{m(m-1)}{2 \cdot 2^n} + O(\frac{m}{2^{\frac{n}{2}}}).$$

Here $0(\frac{m}{2^{\frac{n}{2}}})$ denotes the standard deviation. This can be easily proved using the Covariance Formula, see Appendix A or full version of this article [13].

For an unbalanced Feistel scheme, the preceding condition appears at random, but we also have the following property:

$$X_i^1 = X_j^1, \ldots, X_i^t = X_j^t \Rightarrow S_i^1 \oplus S_j^1 = I_i^{t+1} \oplus I_j^{t+1}$$

since $S_i^1 = X^{t+1} = I^{t+1} \oplus f_{t+1}([I^{t+2}, \ldots I^k, X^1, \ldots, X^t])$. This gives

$$\mathcal{N}_{G_k^{k+t}} \simeq \frac{m(m-1)}{2 \cdot 2^n} + \frac{m(m-1)}{2 \cdot 2^{tn}} \text{ ,so } |E(\mathcal{N}_{G_k^{k+t}}) - E(\mathcal{N}_{perm})| \simeq \frac{m(m-1)}{2 \cdot 2^{tn}}.$$

Here again for $\mathcal{N}_{G_k^d}$, the standard deviation can be computed by using the Covariance Formula, as we have shown for G_3^4 (see full version of this article for the details [13]). Thus we distinguish when $\frac{m^2}{2^{tn}} \geq \frac{m}{2^{\frac{n}{2}}}$ i.e. when $m \geq 2^{(t-\frac{1}{2})n}$, which is compatible with the bound given above.

As usual, we are able transform this attack into a KPA attack which succeeds if $m \geq 2^{(\frac{k+t-2}{2})n}$.

5.3 Attacks for $2k-1$ Rounds

In that case we can only mount a KPA attack. We consider the following KPA attack: let $\mathcal{N}$ be the number of pairs (i,j), $i<j$, such that $I_i^k \oplus I_j^k = S_i^1 \oplus S_j^1$. For a random permutation, we have $\mathcal{N}_{perm} \simeq \frac{m(m-1)}{2\cdot 2^n} + O(\frac{m}{\sqrt{2^n}})$ and for an unbalanced Feistel scheme, $\mathcal{N}_{G_k^{2k-1}} \simeq \frac{m(m-1)}{2\cdot 2^n} + \frac{m(m-1)}{2\cdot 2^{(k-1)n}}$, since $I_i^k \oplus I_j^k = S_i^1 \oplus S_j^1$ is also implied by the following equations: $X_i^1 = X_j^1, X_i^2 = X_j^2, \cdots, X_i^{k-1} = X_j^{k-1}$. This is because $S^1 = X^k = I^k \oplus f_{2k-1}([X^1,\ldots,X^{k-1}])$. Thus we can distinguish when $\frac{m^2}{2\cdot 2^{(k-1)n}} \geq \frac{m}{\sqrt{2^n}}$. This gives $m \geq 2^{(k-\frac{3}{2})n}$.

We can remark that for more than $2k$ rounds we will have to proceed with different attacks, since $X_i^1 = X_j^1, \ldots, X_i^k = X_j^k$ implies $i=j$ because we have a permutation.

6 Attacks with more than 2^{kn} Computations

Until now we have studied Unbalanced Feistel schemes with random functions. In practice, for example in designing block ciphers we need to consider generators of pseudo-random permutations. In this section, we will describe attacks against a generator of permutations (and not only against a single permutation randomly generated by a generator of permutations), i.e. we will be able to study several permutations generated by the generator. This allows more than 2^{kn} computations.

Let G be a "G_k^d generator", i.e. from a binary string K, G generates a d round unbalanced Feistel permutation G_k^d. Let G' be a truly random permutation generator, i.e. from a string K, G' generates a truly random permutation G'_K of B_{kn}. Let G'' be a truly random even permutation generator, i.e. from a string K, G'' generates a truly random permutation G''_K of A_{kn}, with A_{kn} being the group of all the permutations of $\{0,1\}^{kn} \to \{0,1\}^{kn}$ with even signature. We are looking for attacks that distinguish G from G', and also for attacks that will distinguish G from G''.

Adversarial model: an attacker can choose some strings $K_1, \ldots K_f$, can ask for some inputs $[I^1,\ldots,I^k]$, and can ask for some $G_{K_\alpha}[I^1,\ldots,I^k]$ (with K_α being one of the K_i). Here the attack is more general than in the previous sections, since the attacker can have access to many different permutations generated by the same generator.

Adversarial goal: the aim of the attacker is to distinguish G from G' (or from G'') with a high probability and with a complexity as small as possible.

6.1 Brute Force Attacks

A possible attack is an exhaustive search for the d round functions $f_1,\ldots,f_d$ from $\{0,1\}^{(k-1)n}$ to $\{0,1\}^n$ that have been used in the unbalanced Feistel construction. This attack always exists, but since we have $2^{d\cdot n\cdot 2^{(k-1)n}}$ possibilities for $f_1, \ldots, f_d$, this attack requires about $2^{d \cdot n \cdot 2^{(k-1)n}}$ computations and about $\frac{d}{k} \cdot 2^{(k-1)n}$ random queries but only for one permutation of the generator. This attacks means that an adversary with infinite computing power will be able to distinguish G_k^d from a random permutation (or from a truly random permutation with even signature) when $m \geq \frac{d}{k} \cdot 2^{(k-1)n}$.

6.2 Attack by the Signature

Theorem 1. *Let Ψ be an unbalanced Feistel permutation on $\{0,1\}^{\alpha+\beta} \to \{0,1\}^{\alpha+\beta}$ with round functions of $\{0,1\}^{\beta} \to \{0,1\}^{\alpha}$. Then if $\alpha \geq 2$ and $\beta \geq 1$, Ψ has an even signature.*

The proof of this theorem is quite similar to the proof in the case of a symmetric Feistel scheme [11, 3]. However the fact that $\alpha \geq 2$ changes a few things. Consequently a complete proof is included in the full version [13], available from the authors.

Let f be a permutation from kn bits to kn bits. Then using $O(2^{kn})$ computations on the 2^{kn} input/output values of f, we can compute the signature of f. To achieve this we just compute all the cycles c_i of f, $f = \prod_{i=1}^{\alpha} c_i$ and use the formula:

$$\text{signature}(f) = \prod_{i=1}^{\alpha} (-1)^{length(c_i)+1}.$$

The consequence is that it is possible to distinguish G a generator of G_k^d from a generator of truly random permutations from kn bits to kn bits after $O(2^{kn})$ computations on $O(2^{kn})$ input/output values.

Remark: to compute the signature of a permutation g we need however to know all the input/outputs of g (or all of them minus one, since the last one can be found from the others if g is a permutation).

6.3 Attacks of G_k^d Generators when $d = 2k$

Let μ be the number of permutations that we will use. After $2k$ rounds, the output is given by $[S^1, S^2, \ldots, S^k] = [X^{k+1}, X^{k+2}, \ldots, X^{2k}]$ where we have $X^{k+1} = X^1 \oplus f_{k+1}([X^2, \ldots, X^k])$. Remember that $X^1 = I^1 \oplus f_1([I^2, \ldots, I^k])$. Let us describe the KPA attack which concentrates on $S^1 = X^{k+1}$. Let $\mathcal{N}$ be the number of pairs (i, j), $i < j$, such that

$$I_i^2 = I_j^2, \ldots, I_i^k = I_j^k, \quad X_i^{k+1} \oplus X_j^{k+1} = I_i^1 \oplus I_j^1. \tag{1}$$

There we have necessary $I_i^1 \neq I_j^1$ and $X_i^1 \neq X_j^1$. When we are testing random permutations, $\mathcal{N}_{perm} \simeq \mu \cdot \frac{m^2}{2 \cdot 2^{kn}} + O(\sqrt{\mu} \cdot \frac{m}{2^{\frac{kn}{2}}})$. For G_k^k, since $I_i^2 = I_j^2, \ldots, I_i^k = I_j^k, X_i^2 = X_j^2, \ldots, X_i^k = X_j^k$ imply (1) we have:

$$\mathcal{N}_{G_k^d} = \mu \cdot \frac{m^2}{2 \cdot 2^{kn}} + \mu \cdot \frac{m^2}{2 \cdot 2^{(2k-2)n}}.$$

Thus we can distinguish the two generators when: $\mu \cdot \frac{m^2}{2^{(2k-2)n}} \geq \sqrt{\mu} \cdot \frac{m}{2^{\frac{kn}{2}}}$, or when $\mu \cdot m \geq 2^{(3k-4)n}$. When $m = 2^{kn}$, we find $\mu = 2^{(k-4)n}$ and $\mu \cdot m = 2^{(2k-4)n}$.

6.4 Attacks G_k^d Generators for d Rounds with $d \geq 2k$

It is possible to generalize the attack given above for any $d \geq 2k$. We give here only the main ideas. We concentrate the attack on X^{d-k+1}. In the constraints, there are d conditions and $d-k$ internal variables X^i. We choose conditions number k, $2k$, ... , until we get $\xi = \left\lfloor \frac{d}{k} \right\rfloor$ conditions. This gives ξ (internal or external) $\cdot(k-1)$-multiple equations. When they are satisfied, we have:

1. One equation between the input and output variables.
2. φ equations between the output variables where

$$\varphi = (k-1) - \left(d - \left\lfloor \frac{d}{k} \right\rfloor k\right) = (k-1) - (d \bmod k)$$

We have μ permutations and the attack proceeds as follows: let $\mathcal{N}$ be the number of pairs (i,j), $i<j$, such that these $\varphi+1$ equations are satisfied. When we are testing a permutation generator, we have

$$\mathcal{N}_{perm} = \mu \cdot \frac{m(m-1)}{2 \cdot 2^{(\varphi+1)n}} + O(\sqrt{\mu} \cdot \frac{m}{2^{(\frac{\varphi+1}{2})n}}).$$

With a G_k^d, the $\xi(k-1)$-multiples equations imply the $\varphi+1$ equations described above. This shows that

$$\mathcal{N}_{G_k^d} = \mu \cdot \frac{m(m-1)}{2 \cdot 2^{(\varphi+1)n}} + \mu \cdot \frac{m(m-1)}{2 \cdot 2^{(k-1)n}}.$$

We get the condition:

$$\mu \cdot \frac{m^2}{2^{(k-1)n}} \geq \sqrt{\mu} \cdot \frac{m}{2^{(\frac{\varphi+1}{2})n}},$$

$$\mu \cdot m^2 \geq 2^{(2(k-1)\xi-\varphi-1)n}.$$

For the maximal value $m = 2^{kn}$, we find $\mu = 2^{(2(k-1)\xi-\varphi-2k-1)n}$ and the complexity is $\lambda = \mu \cdot m = 2^{(2(k-1)\xi-\varphi k-1)n}$. Thus we can write

$$\lambda = 2^{(2(k-1)\left\lfloor \frac{d}{k} \right\rfloor + (d \bmod k) - 2k)n} = 2^{\left(d+(k-2)\left\lfloor \frac{d}{k} \right\rfloor - 2k\right)n}.$$

7 Conclusion

Until now, attacks and proofs of security on contracting unbalanced Feistel Schemes have not received much attention. There are much more papers on classical Feistel schemes and even attacks on expanding unbalanced Feistel schemes have been more studied than attacks on contracting unbalanced Feistel schemes.

This may be not justified since contracting Feistel schemes seem to have very good security properties. For example, to avoid all known generic attacks with the number of messages less than 2^{kn} (where kn is the number of bits of the input and the output) with these schemes, we need only $2k$ rounds (if $k \geq 4$) or 7 rounds (if $k = 3$). So each bit will be changed only 2 times (if $k \geq 4$) unlike with balanced Feistel schemes where 3 changes (i.e. 6 rounds) are necessary and unlike expanding unbalanced Feistel schemes where much more changes are needed [4, 11, 14].

Table 3. Results on G_3^d. For more than 7 rounds more that one permutation is needed or more than 2^{3n} computations are needed in the best known attacks to distinguish from a random permutation with an even signature.

	KPA	CPA-1 [a]
G_3^1	1	1
G_3^2	1	1
G_3^3	2^n	2
G_3^4	2^n	$2^{n/2}$
G_3^5	$2^{3n/2}$	2^n
G_3^6	$2^{5n/2}$	2^{2n}
G_3^7	2^{3n}	2^{3n}
G_3^8	2^{4n}	2^{4n}
G_3^9	2^{6n}	2^{6n}
G_3^{10}	2^{7n}	2^{7n}
G_3^{11}	2^{8n}	2^{8n}
G_3^{12}	2^{10n}	2^{10n}
$G_3^d, d \geq 12$	$2^{(d+\lfloor \frac{d}{3} \rfloor - 6)}$	$2^{(d+\lfloor \frac{d}{3} \rfloor - 6)}$

[a] Here we do not show CPA-2, CPCA-1 and CPCA-2 since for G_3^d, no better attacks are found compared with CPA-1.

Storing a random function of $(k-1)n$ bits to n bits requires a large memory and this may be a practical disadvantage of G_k^d compared with balanced Feistel schemes or Feistel schemes with expanding functions. However if a function generator is used to generate pseudo-random functions, this may not be a problem.

There are still many open problems on contracting unbalanced Feistel schemes. Naor and Reingold have shown a very nice security result [9]: we have security until the birthday bound when we use pairwise independent functions for the first and the last rounds. However, if we do not use such first and last rounds, the exact security is still an open problem and even the birthday security bound is not proved yet.

In conclusion, contracting unbalanced Feistel schemes seem to be one of the best designs for permutation generators. In this paper, we have presented attacks on these schemes with fewer than $2k$ rounds.

Table 4. Results on G_k^d for any $k \geq 4$. For more than $2k$ rounds more that one permutation is needed or more than $2^{(2k-4)n}$ computations are needed in the best known attacks to distinguish from a random permutation with an even signature.

	KPA	CPA-1[a]
G_k^d, $1 \leq d \leq k-1$	1	1
G_k^k	$2^{\frac{n(k-1)}{2}}$	2
G_k^{k+1}	$2^{\frac{n(k-1)}{2}}$	$2^{\frac{n}{2}}$
G_k^{k+2}	$2^{\frac{k}{2}n}$	$2^{\frac{3}{2}n}$
G_k^{k+3}	$2^{(\frac{k+1}{2})n}$	$2^{\frac{5}{2}n}$
G_k^{k+i}, $1 \leq i < k$	$2^{(\frac{k+i-2}{2})n}$	$2^{(\frac{2i-1}{2})n}$
G_k^{2k}	$2^{(2k-4)n}$	$2^{(2k-4)n}$
G_k^d, $d \geq 2k$	$2^{(d+(k-2)\lfloor\frac{d}{k}\rfloor-2k)n}$	$2^{(d+(k-2)\lfloor\frac{d}{k}\rfloor-2k)n}$

[a] Here we do not show CPA-2, CPCA-1 and CPCA-2 since for G_k^d, no better attacks are found compared with CPA-1.

References

1. William Aiello and Ramarathnam Venkatesan. Foiling Birthday Attacks in Length-Doubling Transformations - Benes: A Non-Reversible Alternative to Feistel. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 307–320. Springer-Verlag, 1996.
2. Don Coppersmith. Luby-Rackoff: Four rounds is not enough. Technical Report RC20674, IBM Research Report, december 1996.
3. Shimon Even and Oded Goldreich. Des-like functions can generate the alternating group. *IEEE Transactions on Information Theory*, 29(6):863–865, 1983.
4. Charanjit S. Jutla. Generalized Birthday Arracks on Unbalanced Feistel Networks. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO '98*, volume 1462 of *Lecture Notes in Computer Science*, pages 186–199. Springer-Verlag, 1998.
5. Lars R. Knudsen. DEAL - A 128-bit Block Cipher. Technical Report 151, University of Bergen, Department of Informatics, Norway, february 1998.
6. Lars R. Knudsen and Vincent Rijmen. On the Decorrelated Fast Cipher (DFC) and Its Theory. In Lars R. Knudsen, editor, *Fast Software Encrytion – FSE '99*, volume 1636 of *Lecture Notes in Computer Science*, pages 81–94. Springer-Verlag, 1999.
7. Michael Luby and Charles Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM J. Comput.*, 17(2):373–386, 1988.
8. Stefan Lucks. Faster Luby-Rackoff Ciphers. In Dieter Gollman, editor, *Fast Software Encryption – FSE '96*, volume 1039 of *Lecture Notes in Computer Science*, pages 189–203. Springer-Verlag, 1996.
9. Moni Naor and Omer Reingold. On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited. *J. Cryptology*, 12(1):29–66, 1999.
10. Jacques Patarin. New Results on Pseudorandom Permutation Generators Based on the DES Scheme. In Joan Feigenbaum, editor, *Advances in Cryptology – CRYPTO '91*, volume 576 of *Lecture Notes in Computer Science*, pages 301–312. Springer-Verlag, 1991.

11. Jacques Patarin. Generic Attacks on Feistel Schemes. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 222–238. Springer-Verlag, 2001.
12. Jacques Patarin. Security of Random Feistel Schemes with 5 or More Rounds. In Matthew K. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 106–122. Springer-Verlag, 2004.
13. Jacques Patarin, Valérie Nachef, and Côme Berbain. Generic Attacks on Unbalanced Feistel Schemes with Contracting Functions, Extended Version. *available from the authors*, 2006.
14. Jacques Patarin, Valérie Nachef, and Côme Berbain. Generic Attacks on Unbalanced Feistel Schemes with Expanding Functions. *available from the authors*, 2006.
15. Bruce Schneier and John Kelsey. Unbalanced Feistel Networks and Block Cipher Design. In Dieter Gollmann, editor, *Fast Software Encrytion – FSE '96*, volume 1039 of *Lecture Notes in Computer Science*, pages 121–144. Springer-Verlag, 1996.

A Computation of the Variance for Random Permutations

In this section, we compute the value of the variance when we are testing a random permutation and we want to distinguish it from a G_k^{k+t}, $1 \leq t \leq k-1$. The input is $[I^1, \ldots, I^k]$ and the output is $[S^1, \ldots, I^K]$. We want to compute $\mathcal{N}_{perm}$ which is the number of (i,j), $i<j$ satisfying the relation $I_i^{t+1} \oplus S_i^1 = I_j^{t+1} \oplus S_j^1$. We have the condition $\forall i$, $I_i^{t+2} = I_i^{t+3} = \ldots = I_i^k = 0$. This implies that $m \leq 2^{(t+1)n}$. We introduce the following random variables:

$$\begin{cases} \delta_{i,j} = 1 \text{ if } I_i^{t+1} \oplus S_i^1 = I_j^{t+1} \oplus S_j^1 \\ \delta_{i,j} = 0 \text{ otherwise} \end{cases}$$

Then $\mathcal{N}_{perm} = \sum_{i<j} \delta_{i,j}$ and $E(\delta_{i,j}) = Pr_{f \in_R B_{kn}}[I_i^{t+1} \oplus S_i^1 = I_j^{t+1} \oplus S_j^1]$

Notice that if $m \ll 2^n$, we may assume that the I^{t+1} values are pairwise distinct (or are all equal) and if $m \geq 2^n$, we may assume that each element of $\{0,1\}^n$ is reached by about $\frac{m}{2^n}$ values of I_i^{t+1} (in CPA-1, we can choose m to be a multiple of 2^n and each element of $\{0,1\}^n$ is reached by exactly $\frac{m}{2^n}$ values of I_i^{t+1}. It is also possible to choose that I_i^{t+1} are random values). If $I_i^{t+1} = I_j^{t+1}$, $E(\delta_{i,j}) = Pr_{f \in_R B_{kn}}[S_i^1 = S_j^1] = \frac{2^{(k-1)n}-1}{2^{kn}-1} \simeq \frac{1}{2^n}$. and if $I_i^{t+1} \neq I_j^{t+1}$, $E(\delta_{i,j}) = \frac{2^{(k-1)n}}{2^{kn}-1} \simeq \frac{1}{2^n}$. This gives us the average value:

$$E(\mathcal{N}_{perm}) \simeq \frac{m(m-1)}{2 \cdot 2^n} + o(\frac{m}{2^{(k+\frac{1}{2})n}}).$$

We now compute the variance $V(\delta_{i,j}) = E(\delta_{i,j}^2) - E(\delta_{i,j})^2 = E(\delta_{i,j}) - E(\delta_{i,j})^2$. If $I_i^{t+1} = I_j^{t+1}$, $V(\delta_{i,j}) = \frac{1}{2^n} \cdot \frac{1}{1-\frac{1}{2^{kn}}} - \frac{1}{2^{kn}-1} - \left(\frac{1}{2^n} \cdot \frac{1}{1-\frac{1}{2^{kn}}} - \frac{1}{2^{kn}-1}\right)^2$. And if $I_i^{t+1} \neq I_j^{t+1}$, $V(\delta_{i,j}) = \frac{1}{2^n} \cdot \frac{1}{1-\frac{1}{2^{kn}}} - \left(\frac{1}{2^n} \cdot \frac{1}{1-\frac{1}{2^{kn}}}\right)^2$. Finally $V(\delta_{i,j}) \simeq \frac{1}{2^n}(1-\frac{1}{2^n})$ and

$$\sum_{i<j} V(\delta_{i,j}) \simeq \frac{m(m-1)}{2} \cdot \frac{1}{2^n}(1-\frac{1}{2^n}).$$

We recall the formula:

$$V(\mathcal{N}_{perm}) = V(\sum_{i<j} \delta_{i,j}) = \sum_{i<j} V(\delta_{i,j}) + \sum_{i<j,p<l,(i,j)\neq(p,l)} \left[E(\delta_{i,j}\,\delta_{p,l}) - E(\delta_{i,j})\,E(\delta_{p,l})\right]$$

The second term is the covariance term. We will see that

$$V(\mathcal{N}_{perm}) = \frac{m(m-1)}{2\cdot 2^n} + O\left(\frac{m^2}{2^{2n}}\right) + O\left(\frac{m^4}{2^{2n}\cdot 2^{(2k-1)n}}\right) + O\left(\frac{m^3}{2^{2n}\cdot 2^{(k-1)n}}\right)$$

where the two first terms correspond to the sum of the variance of $\delta_{i,j}$, the third term corresponds to the covariance of four distinct indexes (i,j,k,l), and the last term corresponds to the covariance of 4-tuples of indexes with one in common, like for example (i,j,i,l). Therefore, for m larger than 2^n but smaller than 2^{kn}, we have as claimed

$$V(\mathcal{N}_{perm}) = \frac{m(m-1)}{2\cdot 2^n} + o\left(\frac{m^2}{2^n}\right) \simeq \frac{m^2}{2\cdot 2^n}.$$

In order to exactly compute the covariance term, we can separate the computation into several cases. Here we only study the main case, i.e. we suppose that i,j,p,l are pairwise distinct and that $I_i^{t+1} \neq I_j^{t+1}$, $I_p^{t+1} \neq I_l^{t+1}$ and $I_i^{t+1} \oplus I_j^{t+1} \oplus I_p^{t+1} \oplus I_l^{t+1} \neq 0$. For all other cases, computation is similar and is included in the full version of this paper [13].

To compute this probability we need to count the total number A of possibilities for the outputs $[S_i^1,\ldots,S_i^k]$, $[S_j^1,\ldots,S_j^k]$, $[S_p^1,\ldots,S_p^k]$ and $[S_l^1,\ldots,S_l^k]$. Since we are using a permutation, we have $A = 2^{kn}\cdot(2^{kn}-1)\cdot(2^{kn}-2)\cdot(2^{kn}-3)$.

We also have to compute B the number of outputs $[S_i^1,\ldots,S_i^k]$, $[S_j^1,\ldots,S_j^k]$, $[S_p^1,\ldots,S_p^k]$ and $[S_l^1,\ldots,S_l^k]$ satisfying the above relations in the case we consider. For $[S_i^1,\ldots,S_i^k]$, there are 2^{kn} possibilities. When this output is fixed, $S_j^1 = S_i^1 \oplus I_i^{t+1} \oplus I_j^{t+1}$. Thus there are $2^{(k-1)n}$ possibilities for $[S_j^1,\ldots,S_j^k]$. Now we have to fix $[S_i^1,\ldots,S_i^k]$ and $[S_j^1,\ldots,S_j^k]$. There are 5 cases that we are going to study now. If $S_p^1 = S_i^1 \oplus I_p^{t+1} \oplus I_l^{t+1}$, then $S_p^1 \neq S_i^1$, $S_p^1 \neq S_l^1$ and $S_l^1 = S_i^1$. Thus we have $2^{(k-1)n}\cdot(2^{(k-1)n}-1)$ possibilities for $[S_p^1,\ldots,S_p^k]$ and $[S_l^1,\ldots,S_l^k]$. Then we consider the case where $S_p^1 = S_j^1 \oplus I_p^{t+1} \oplus I_l^{t+1}$. This case is different from the previous one since $S_i^1 \neq S_j^1$. We get again $2^{(k-1)n}\cdot(2^{(k-1)n}-1)$ possibilities for $[S_p^1,\ldots,S_p^k]$ and $[S_l^1,\ldots,S_l^k]$. If $S_p^1 = S_i^1$ or if $S_p^1 = S_j^1$, there are $(2^{(k-1)n}-1)\cdot 2^{(k-1)n}$ possibilities for $[S_p^1,\ldots,S_p^k]$ and $[S_l^1,\ldots,S_l^k]$. The last case is when we have eliminated the previous cases. This gives $(2^n-4)\cdot 2^{(k-1)n}\cdot 2^{(k-1)n}$ possibilities for $[S_p^1,\ldots,S_p^k]$ and $[S_l^1,\ldots,S_l^k]$. Finally $B = 2^{(4k-2)n}\cdot(1-\frac{4}{2^{kn}})$. Consequently, since $E(\delta_{i,j}\,\delta_{p,l}) = \frac{B}{A}$ we get:

$$E(\delta_{i,j}\,\delta_{p,l}) - E(\delta_{i,j})E(\delta_{p,l}) = \frac{1}{2^{2n}}\left(-\frac{2}{2^{2kn}} + O\left(\frac{1}{2^{3kn}}\right)\right).$$

Finally these terms of covariance are equal to $\frac{-2m^4}{4\cdot 2^{2n}\cdot 2^{2kn}} \leq O\left(\frac{m^4}{2^{2n}\cdot 2^{(2k-1)n}}\right)$ as claimed.