A New Criterion for Nonlinearity of Block Ciphers

Orr Dunkelman*[1] Nathan Keller[2]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
orrd@cs.technion.ac.il
[2]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91940, Israel
nkeller@math.huji.ac.il

Abstract. For years, the cryptographic community has searched for good nonlinear functions. Bent functions, almost perfect nonlinear functions, and similar constructions have been suggested as a good base for cryptographic applications due to their highly nonlinear nature. In the first part of this paper we study these functions as block ciphers, and present several distinguishers between almost perfect nonlinear permutations and random permutations. The data complexity of the best distinguisher is $O(2^{n/3})$ and its time complexity is $O(2^{2n/3})$ for an n-bit block size, independent of the key size.
In the second part of the paper we suggest a criterion to measure the effective linearity of a given block cipher. We devise a distinguisher for general block ciphers based on their effective linearity. Finally, we show that for several constructions, our distinguishing attack is better than previously known techniques.
Keywords: Almost perfect nonlinear permutations, highly nonlinear functions, effective linearity, differential cryptanalysis

1 Introduction

For years, highly nonlinear functions were extensively used in various cryptographic applications. Highly nonlinear functions were promoted since algorithms that are close to linear are susceptible to various approximation attacks. *Differential cryptanalysis* [5] and *linear cryptanalysis* [11] show that even partial approximations of the encryption algorithm by a linear function are sufficient to mount powerful distinguishing and key-recovery attacks.

In [12] the authors suggested to use (almost) perfect nonlinear functions (functions with a maximal distance from all linear structures) in block ciphers. The nonlinear functions can be used either as a building block in the structure of the block cipher (such as an S-box [15–17]) or as the entire block cipher (as

* The research presented in this paper was partially supported by the Clore scholarship programme.

discussed in [1]). Such (almost) perfect nonlinear constructions can be used to prove security against differential and linear cryptanalysis [18].

In [15] the following construction, later named in [18], was suggested: An almost perfect nonlinear permutation (APNP) is a permutation $f : GF(2^n) \rightarrow GF(2^n)$ such that for any $a \neq 0$, the function $g(x, a) = f(x) \oplus f(x \oplus a)$ assumes exactly 2^{n-1} different values. It means that for any two pairs of distinct input values with the same difference, the corresponding output pairs do not have the same difference. Due to this property, APNPs are considered secure against differential cryptanalysis [5] as well as against linear cryptanalysis [11].[1] A permutation that is very close to be an APNP is the S-box SubBytes of AES [7]. In the SubBytes permutation, for a non-zero input difference α, there are two pairs of inputs with difference α and the same output difference, while the other 126 pairs with input difference α have different output differences.

In the first part of this paper we analyze the concept of using highly nonlinear functions (such as APNPs) as the entire block cipher. We devise several distinguishers between an APNP and a random permutation based on the "too good" differential properties of APNPs. The data and memory complexities of the best distinguisher are $O(2^{n/3})$ with time complexity of $O(2^{2n/3})$ where n is the block size, independent of the key size. The distinguishers are then extended to ciphers with close to uniform difference distribution tables (that is, ciphers that are close to APNPs). This result leads to the conclusion that despite their favorable properties, highly nonlinear ciphers might possess inherent weakness.

In a way, the concept of this part of the paper is similar to the concept behind impossible differential cryptanalysis [4]. Usually, differential cryptanalysis uses differentials with high probability. The idea of impossible differential attacks is to exploit differentials with zero probability. The attacker utilizes the fact that the differential properties are "too strong". In the same way, our attacks recognize APNPs due to their "too high" level of nonlinearity.

In the second part of this paper we analyze the following encryption scheme: Let f be a permutation, f^{-1} be its inverse, and K be a randomly chosen non-zero key. We define $g(x) = f^{-1}(f(x) \oplus K)$, and show that g has several predictable differential properties. Those properties are related to the level of nonlinearity of the function f. We use those properties to define the *effective linearity* of the function f, and show that the effective linearity of almost perfect nonlinear permutations is 1, for random permutations it is 2, and is 2^n for linear permutations (where n is the block size). We note that the effective linearity of a block cipher can be approximately computed with complexity $O(2^{n/2})$. We present various scenarios in which this value can be used in distinguishing and key recovery attacks. For example, we show that a 2-round Feistel structure surrounded by two key dependent decorrelation modules can be easily distinguished from random permutations, regardless of the Feistel round function.

[1] Note that there are two colliding definitions to the term nonlinearity — nonlinearity as distance from linear functions, and nonlinearity in the sense of almost perfect nonlinear permutations. In this paper we use the latter definition.

This paper is organized as follows: In Section 2 we give the definition and theoretical background of highly nonlinear functions. In Section 3 we present two distinguishing attacks on ciphers with uniform difference distribution tables. Section 4 examines differential properties of the construction $g = f^{-1}(f(x) \oplus K)$, and defines the effective linearity of a function. In Section 5 we use the effective linearity to mount distinguishing and key recovery attacks. In the appendices we bring various computations and proofs. We summarize the paper in Section 6, and discuss the implications of our findings.

2 Definitions and Theoretical Background

There are several possible notions of high nonlinearity of a boolean function. In this paper we use the following definition, presented in [15]:

Definition 1. *A function $f : GF(2^n) \to GF(2^m)$ is perfectly nonlinear if for any non-zero $w \in GF(2^n)$ the difference $f(x + w) - f(x)$ obtains all the values $y \in GF(2^m)$ exactly 2^{n-m} times each.*

In [15] the properties of such functions were studied, and it was shown that perfectly nonlinear functions $f : GF(2^n) \to GF(2^m)$ exist if and only if $n \geq 2m$. However, in real life designs many designers prefer to use functions in which the size of the input is equal to the size of the output. Thus, perfectly nonlinear functions cannot be used. This led to the introduction of *almost perfect nonlinear functions*, defined as following:

Definition 2. *A function $f : GF(2^n) \to GF(2^n)$ is almost perfectly nonlinear if for any $a \neq 0$, the function $g(x, a) = f(x) \oplus f(x \oplus a)$ assumes exactly 2^{n-1} different values.*

If such f is also a permutation, f is called "almost perfectly nonlinear permutation" (in the sequel we abbreviate this notation to "APNP").

Note, that since the characteristic of the field $GF(2^n)$ is 2, it follows that for any function $f : GF(2^n) \to GF(2^n)$, every value of $g(x, a)$ is assumed an even number of times. Therefore, 2^{n-1} is the maximal possible output size of g for a non-zero α.

A permutation that is close to be an APNP is the S-box SubBytes of AES [7]. This S-box is a permutation $f : GF(2^8) \to GF(2^8)$ and for every $a \neq 0$, the function $g(x, a) = f(x) \oplus f(x \oplus a)$ assumes $2^{8-1} - 1 = 127$ values.

As stated before, this notion of nonlinearity is closely related to differential properties of the function f and the definition of an APNP can be restated in terms of the *difference distribution table* of the function used in differential cryptanalysis [5]. First we recall the definition of the difference distribution table of a function:

Definition 3. *Let $f : GF(2^n) \to GF(2^m)$ be a general function. The difference distribution table (DDT) of f is an $(2^n) \times (2^m)$ matrix whose (i, j) entry is defined as $\#\{x \in GF(2^n) | f(x) \oplus f(x \oplus i) = j\}$.*

A function f is considered optimally secure against differential cryptanalysis if the entries in the DDT of f are the lowest possible ones. This is the case when f is an almost perfect nonlinear function. We can now rephrase the definition of an almost perfect nonlinear function in terms of the DDT:

Definition 4. *A function $f : GF(2^n) \rightarrow GF(2^n)$ is almost perfectly nonlinear if the highest entry in the DDT of f (except for the entry $(0 \rightarrow 0)$ that equals 2^n) is 2.*

This definition is closely related to the following definition of δ-uniformity:

Definition 5. *For an $n \times s$ bits S-box $S(\cdot)$ (where $n \geq s$), we denote by δ the highest entry in the difference distribution table (except for (0,0) entry which is always 2^n), namely*

$$\delta = \max_{\alpha \in \{0,1\}^n, \alpha \neq 0, \beta \in \{0,1\}^s} \#\{x | S(x) \oplus S(x \oplus \alpha) = \beta\}$$

S is called differentially δ-uniform.

Hence, almost perfect nonlinear permutations are differentially 2-uniform.

We recall that differential cryptanalysis is mostly interested in differentials with high probability (or zero probability). This led various papers [17, 18, 21] to suggest using functions that are as differentially uniform as possible. If such functions are used, the cipher is expected to have fewer differentials with high probability, as well as less zero probability differentials.

We stress that usually APNPs are not used in real life ciphers. However, this is mostly due to implementation issues, as the common belief is that these functions are better than other constructions. For example, many ciphers use APNPs as building blocks, like the S-box used in the AES.

3 Distinguishing Highly Nonlinear Functions from Random Permutations

In this section we present two distinguishing attacks on highly nonlinear functions. Each is based on a different assumption and performs in a different model (known plaintext or chosen plaintext). These attacks are capable of identifying whether a given black box is a random permutation or an highly nonlinear function. Hence, if an APNP is used as the cipher, it can be distinguished from a random permutation. We discuss the possible applications of such an attack in Section 5.

We note that the Even-Mansour construction $E_{K_1,K_2}(P) = F(P \oplus K_1) \oplus K_2$ assumes that the underlying F is a pseudo random permutation [8]. Our attacks can distinguish the case where F is an APNP from the case that F is a random permutation, despite the commonly believed good security properties of APNPs.

3.1 A Chosen Plaintext Distinguisher

The first attack is a chosen plaintext attack based on the birthday paradox. Let $f : GF(2^n) \to GF(2^n)$ be a black box permutation for which we have to determine whether it is an APNP or a random permutation.

We perform the following algorithm with a parameter m (to be determined later):

1. Encrypt m distinct pairs of plaintexts (P_1, P_2), such that $P_1 \oplus P_2 = \alpha$ and $P_1 < P_2$ for some fixed non-zero value α by f to get the ciphertext pairs of the form $(C_1, C_2) = (f(P_1), f(P_2))$.
2. Store the XOR values of the ciphertexts, i.e., $C_1 \oplus C_2$ in a hash table.
3. If we obtain a collision in the hash table (two pairs with the same ciphertext difference), halt and conclude that f is not an APNP.
4. If no collisions are encountered, conclude that f is an APNP.

A collision is formed of two distinct pairs $(P_1, P_2 = P_1 \oplus \alpha)$ and $(P_3, P_4 = P_3 \oplus \alpha)$, whose corresponding ciphertexts (C_1, C_2) and (C_3, C_4), respectively, satisfy $C_1 \oplus C_2 = C_3 \oplus C_4$. Such a collision means that the equation $f(x) \oplus f(x \oplus \alpha) = C_1 \oplus C_2 = C_3 \oplus C_4$ has (at least) four solutions.

Recall that an APNP is a permutation for which the equation $f(x) \oplus f(x \oplus \alpha) = \beta$ for non-zero α and β has at most two solutions (x_0 and $x_0 \oplus \alpha$ for some x_0). Thus, for any value of m (even for $m = 2^{n-1}$), no such collision is expected.

For a random permutation, however, the algorithm is expected to find such an instance. And thus, once such an instance is found, the algorithm concludes that f is not an APNP.

3.2 Analysis of the Chosen Plaintext Attack

Recall, that the attack is based on encrypting pairs of plaintexts (P_1, P_2) that satisfy $P_1 \oplus P_2 = \alpha$ for some fixed arbitrary non-zero α. If there are two distinct plaintext pairs (P_1, P_2) and (P_3, P_4) whose corresponding ciphertexts (C_1, C_2) and (C_3, C_4), respectively, satisfy $C_1 \oplus C_2 = C_3 \oplus C_4$, then the black box permutation f is not an APNP for sure.

Let us examine the number of expected quartets for f. Consider the row corresponding to α in DDT^f, the difference distribution table of f. For every fixed output difference β, the value that corresponds to β in this row represents the number of pairs with input difference α and output difference β (recall that x and $x \oplus \alpha$ appear as two pairs $(x, x \oplus \alpha)$ and $(x \oplus \alpha, x)$). In other words, $DDT^f(\alpha, \beta)$ (entry (α, β) of DDT^f) is

$$DDT^f(\alpha, \beta) = |\{x \in GF(2^n) : f(x) \oplus f(x \oplus \alpha) = \beta\}|.$$

Note that a permutation is considered APNP if and only if its difference distribution table does not contain values greater than 2.

For a random permutation f, we may assume that values in any single row of the difference distribution table behave almost as Poisson random variables.

That is, the values in the difference distribution table are distributed according to $2 \cdot Poi(1/2)$.[2] Thus, the value $2k$ is expected to appear in a given row about $2^n \cdot e^{-1/2} \cdot 2^{-k}/k!$ times.

Collisions in Step 3 can occur only for values of β whose corresponding entry of the difference distribution table is more than 2. Let us examine only values of β in the difference distribution table with 4 or more. Out of the 2^n possible β entries, $0.0902 \cdot 2^n$ such entries exist. Due to the birthday paradox, when we want to have success rate of p, we require $p > 1 - e^{-m \cdot (m-1)/(2 \cdot 0.0902 \cdot 2^n)}$. Therefore, to ensure success probability of 0.8 we need $m > 0.1618 \cdot 2^{n/2}$ pairs of this kind.

However, the algorithm encrypts also pairs whose output difference β has 2 in the difference distribution table. Hence, the real number of pairs we need to examine is about 4 times larger, as only one out of 4 pairs (more precisely, about 23%) has an output difference meeting our requirement. Therefore, the data complexity of the algorithm is $N = 2m = 1.4070 \cdot 2^{n/2}$ plaintexts (or queries to the black box).

The time complexity of the algorithm is $N = 1.4070 \cdot 2^{n/2}$ encryptions and $m = 0.7035 \cdot 2^{n/2}$ memory accesses in the worst case. The memory requirements are $m = 0.7035 \cdot 2^{n/2}$ memory cells in the worst case.

Changing the attack scenario into a known plaintext attack does not change the attack significantly. The data complexity is $m = 1.3459 \cdot 2^{n/2}$ queries, and the time complexity is $O(2^{2n/3})$ using Wagner's algorithm for the generalized birthday paradox problem [24].

3.3 An Improvement to the Chosen Plaintext Attack

An improvement to the algorithm uses the fact that the above is true for any non-zero α. The attack requires m distinct plaintexts, such that the XOR value of any two of them is among a list of m values (for example, setting some of the bits of all plaintexts to be zero). In this case for each pair of plaintexts (P_1, P_2) we compute $(P_1 \oplus P_2, f(P_1) \oplus f(P_2))$, and insert it into a hash table. A collision in the hash table suggests a quartet of values (P_1, P_2) and (P_3, P_4) such that $P_1 \oplus P_2 = P_3 \oplus P_4$ and $f(P_1) \oplus f(P_2) = f(P_3) \oplus f(P_4)$. This cannot be achieved for an almost perfect nonlinear permutation, and thus, can be also used for distinguishing.

For m plaintexts chosen in this way, we have $m^2/2$ pairs, each producing a string of $2n$ bits. Not all 2^{2n} possible $2n$-bit strings are produced in this process. More precisely, the number of possible values for this string is $m \cdot 2^n$. If we choose m such that $(m^2/2)^2 > 1.17m2^n$, we have a chance of 50% to find a collision (in case of a random permutation) according to the birthday paradox.

Setting $m = 1.794 \cdot 2^{n/3}$ we expect to find such a collision with probability 0.8. Thus, the data complexity of this attack is $m = 1.794 \cdot 2^{n/3}$ chosen plaintexts, and the time complexity of the attack is $m^2/2 = 1.609 \cdot 2^{2n/3}$ memory accesses.

[2] Recall that in an XOR difference distribution table all values are even.

3.4 Other Kinds of Permutations

Permutations that are very close to APNPs are widely used in block ciphers. For this kind of permutations, the above attacks still succeed with almost the same success rate.

Another question which arises, is what happens when the permutation we wish to distinguish is not so close to be differentially 2-uniform. That is, what if there are many entries with value of 4 in the difference distribution table of the permutation. Formally, out of the 2^{n-1} pairs, assume that at most a ratio p of the pairs are in entries with value of 4 in the difference distribution table, while the other non-zero entries are 2 (up to the $0 \rightarrow 0$ entry).

For these functions, the above algorithms fail, as the probability to have two pairs whose output difference is the same, is no longer negligible. This can be solved when p is far from 0.23,[3] e.g., $p < 0.2$ or $p > 0.3$ (in case $p > 0.3$, we require that at least p of the pairs are in entries with value of 4).

The transformation of the above algorithms to deal with such permutations is changing the identification from "find such an instance, halt and output ..." to "count how many instances there are, and compare this number to how many should be". The analysis of the exact number of plaintexts m needed is quite straightforward given p and the requested success rate.

4 Differential Properties of $f^{-1}(f(x) \oplus K)$ and their Applications

In this section we consider the differential properties of some special structure derived from a permutation f and show how to utilize these properties in order to study the structure of f itself. Let $g(x) = f^{-1}(f(x) \oplus K)$ be a permutation where K is some fixed key. First, we show that using the properties of g we can determine whether f is an APNP or a random permutation. Then we show how to generalize this result in order to classify functions according to their level of nonlinearity. We formalize this classification by defining the *effective linearity* coefficient (EL) of a permutation which corresponds to the level of linearity determined by our method.

4.1 Theoretical Background

Let $f : GF(2^n) \rightarrow GF(2^n)$ be a black box permutation for which we have to determine whether it is an APNP or a random permutation. Choose an arbitrary non-zero $K \in GF(2^n)$, and define the permutation $g(x) = f^{-1}(f(x) \oplus K)$.

[3] The expected ratio of 4 or more in the difference distribution table of a random permutation is about 0.09 of the entries. However, entries with 6,8, or even more, contribute more quartets. The total number of quartets counted by the above algorithms for a random permutation is equal to the case where 0.23 of the entries of the table that are 4 (while the remaining are 2's and 0's).

Let A, B be a pair of plaintexts with a non-zero input difference α (e.g., $A \oplus B = \alpha$) and consider $\gamma = g(A) \oplus g(B)$. We shall compute the probability of the event $\gamma = \alpha$, and show that this probability can be used to distinguish APNPs from random permutations.

Let $\beta = f(A) \oplus f(B)$. If f is an APNP, (A, B) is the only pair of plaintexts with input difference α and output difference β. Now, consider the pair $(f(A) \oplus K, f(B) \oplus K)$. There are two cases:

1. $f(A) \oplus K = f(B)$. In this case, we have $g(A) = B$ and $g(B) = A$ and thus $\gamma = g(A) \oplus g(B) = B \oplus A = \alpha$. This case occurs when $\beta = f(A) \oplus f(B) = K$ which happens with probability of 2^{-n}.
2. $f(A) \oplus K \neq f(B)$. In this case, the pairs $(f(A), f(B))$ and $(f(A) \oplus K, f(B) \oplus K)$ differ, but still have the same XOR difference β. Thus, if f is an APNP, the difference $\gamma = g(A) \oplus g(B) = (f^{-1}(f(A) \oplus K)) \oplus (f^{-1}(f(B) \oplus K))$ cannot be equal to α. Therefore, in this case $\gamma \neq \alpha$ always.

Combining the two cases together we obtain

$$\Pr_{A,B,K \in GF(2^n), A \neq B, K \neq 0}[\gamma = \alpha] = 2^{-n}. \tag{1}$$

The analysis presented in Appendix A shows that for a random permutation this probability ($\Pr[\gamma = \alpha]$) equals to $2 \cdot 2^{-n}$. The difference between the probabilities can be used in order to distinguish between an APNP and a random permutation.

We have experimentally verified that the value $2 \cdot 2^{-n}$ is the correct value for a random permutation and that 2^{-n} is the correct value for an APNP. This was done by generating sets of random permutations of 8,10,12,14 and 16 bits, and counting all possible quartets $(A, B, g(A), g(B))$ (for a large set of K values, for all α values).

4.2 An Adaptive Chosen Plaintext and Ciphertext Distinguisher

The algorithm of the distinguisher is as follows: Let $f : GF(2^n) \rightarrow GF(2^n)$ be a black box permutation for which we have to determine whether it is an APNP or a random permutation and $m, threshold$ be integers specified later.

1. Encrypt m distinct plaintexts P_i, for $i = 1, \ldots, m$.
2. Choose an arbitrary $K \in GF(2^n)$.
3. Decrypt the values $f(P_i) \oplus K$ to get $g(P_i) = f^{-1}(f(P_i) \oplus K)$.
4. Store the m values of the form $P_i \oplus g(P_i)$ into a hash table.
5. Count the number of collisions in the hash table. If the number of collisions is greater than *threshold* output "random permutation". Otherwise, output "APNP".

We note that if $P_1 \oplus g(P_1) = P_2 \oplus g(P_2)$ then we have a right quartet $((P_1, P_2), (g(P_1), g(P_2)))$. Starting with m plaintexts, we get m values of $P \oplus$

$g(P)$. Once there is a collision in the hash table, the colliding values suggest a quartet. If there are three values in the same entry of the hash table, then we get three quartets, or generally, if there are k values in the same entry, we get $k(k-1)/2$ quartets.

For an APNP, about 2^{-n} of the all possible quartets satisfy our conditions, while for a random permutation, about $2 \cdot 2^{-n}$ satisfy our conditions. For $m = 4 \cdot 2^{n/2}$ and $threshold = 10$ the success rate is 0.816. For $m = 2^{n/2}$ the success rate of this attack is 0.594.

This attack may seem less desirable, given the attacks of the previous section, as it has a similar data complexity but a more stern attack model. However, this attack can be easily extended to other cases, as we present in Section 4.3.

We remark that one can make a slight change in the attack such that only quartets of the form $(A, B, g(A), g(B))$ where $A \neq B \neq g(A) \neq g(B)$ are counted. Using this variant of the attack the number of expected collisions is 0 for APNP, and if we get even one collision the permutation is certainly not an APNP. The number of expected collisions for a random permutation is $\frac{1}{4} \cdot 2^{-n}$ of the total number of possible quartets.

This can be used to increase the success probability of the attack by setting $threshold = 1$. More accurately, for the same data complexity as before, the success rate of the attack is about 0.98. We can also reduce the data complexity by a factor of $\sqrt{2}$ and still have a success rate of 0.86.

4.3 The Effective Linearity of a Permutation

Following the previous attack, we define the effective linearity of a permutation.

Definition 6. *Let $f : GF(2^n) \rightarrow GF(2^n)$ be a permutation. The effective linearity of f is:*

$$EL(f) = 2^n \left(\frac{1}{2^n - 1}\right)^2 \cdot \sum_{K \in GF(2^n)\setminus\{0\}} \sum_{\alpha \in GF(2^n)\setminus\{0\}} \Pr\left[\alpha \xrightarrow{g(x)=f^{-1}(f(x)\oplus K)} \alpha\right]$$

Actually, $EL(f)$ is the average of the probabilities $\Pr[\alpha = \gamma]$ over all non-zero K's and α's multiplied by 2^n.

For a random permutation this value is expected to be close to 2 (as shown in Appendix B). If this value is not close to 2, then our attacks can be applied to the permutation and distinguish it from a random permutation.

We can either calculate $EL(f)$ analytically when the difference distribution table of f is known (like in the analysis for a random permutation), or by experimentally measuring it. Taking several sets of $O(2^{n/2})$ messages and using several K values, we can use statistical methods to evaluate $EL(f)$. Note that the $O(2^{n/2})$ complexity is achieved by using many (if not all) values of α simultaneously.

The effective linearity of f is not lower than 1 (as when $K = f(A) \oplus f(B)$ we get that $g(A) = B$ and $g(B) = A$) and cannot be higher than 2^n (which is

the value for linear permutations). As the value for a random permutation is 2, we suggest designing ciphers with effective linearity close to 2.

It is possible to show (see Appendix B) that for a two round Feistel construction whose round functions are both APNP, the effective linearity is 3, while if the used functions are random permutations it is expected to be at least 8. An interesting observation regarding the effective linearity of Feistel constructions, is that after three Feistel rounds using random permutations as round functions, the effective linearity is 2. This might be viewed as another realization of the Luby-Rackoff result about Feistel constructions [10].

Another interesting remark about Feistel constructions, is that if the permutation of the first round p_1 has effective linearity $EL(p_1)$, and the second round's permutation p_2 has effective linearity $EL(p_2)$, then the effective linearity $EL(f)$ of the two round Feistel satisfies $EL(f) \geq EL(p_1) \cdot EL(p_2)$. The exact proof is given in Appendix B.

When the round functions are not bijective, the difference distribution table of the 2-round Feistel construction is expected to have more zero entries than usual. As the sum of every line in the difference distribution table is constant, it follows that the remaining entries are expected to be higher, leading to an higher effective linearity. Thus, the more entries having a zero value in the difference distribution table, the higher the effective linearity is expected to be. For example, we show in Appendix B that the effective linearity of 2-round DES is at least 220 (independent of the key).

5 Various Attacks Based on the Effective Linearity of Permutations

In this section we present several possible scenarios in which measuring the effective linearity of various permutations can be used in order to mount distinguishing and key recovery attacks.

5.1 Treating Decorrelation Modules

Let us consider a cipher of the form $E = DM_2 \circ F_2 \circ F_1 \circ DM_1$, where DM_i is a decorrelation module (with some key) [22], and F_i is a Feistel round with a random permutation as the round function (along with some key).

We recall that once the key is set the decorrelation module is linear, but when the key is random, the probability of any non-trivial differential going through the decorrelation module equals $1/(2^n - 1)$ on average. A similar condition can be proved with respect to linear cryptanalysis as well.

Due to the nature of the decorrelation module, any differential (even a truncated one) cannot have probability higher than the trivial one through the first decorrelation module. The same is true for linear approximations as well. Moreover, it is impossible to devise a SQUARE-like property for this cipher as the decorrelation module prevents the attacker from setting a good input set.

While all these methods fail, we can efficiently distinguish the above E from a random permutation. As the decorrelation modules are linear, and as we have two rounds of a Feistel structure, we can easily determine that the effective linearity of E is 8, while the effective linearity of random permutations is only 2.

We note that the minimal value of the effective linearity of 2-round Feistel construction is 3 (achieved by applying APNPs as the round function in the two Feistel rounds). Thus, even if the Feistel round functions are replaced, our attack still works.

Our technique is able to pass the decorrelation module as if it does not exist. This is due to the fact that we count on many possible differentials, and we do not restrict ourselves to differentials of some structure, or even to sets of plaintexts of a given structure.

5.2 Distinguishing Known Ciphers and Identifying Black Box Permutations

It is possible to precompute the effective linearity of ciphers in advance (also for reduced round variants). Then, given a black box the attacker computes its EL and if the black box is one of the previously known encryption schemes, he can detect it.

However, we still do not know whether this attack is applicable against encryption schemes that are actually used today. It may occur that the measured effective linearity values are too close to 2 and the distinguishing will become infeasible. Moreover, as we noted in Section 4.1, the number of detected quartets in the distinguisher slightly depends on K and thus in some cases the difference between two encryption schemes can be less than the difference between applications of the same scheme with different values of K. In this case the attack might fail.

The effective linearity of a permutation depends on its difference distribution table. When computing the difference distribution table of a permutation one usually computes the average probabilities over all the possible keys and assumes that the probability for any single key is close to the average (see [5]). However, in some ciphers there are classes of keys for that some differential properties of the cipher differ from the average case, like for IDEA [6]. Such classes are called "differentially weak key classes". Usually such classes can be detected only if some explicit differential characteristic is known for the whole cipher.

We can use our distinguisher in order to detect such classes when the entire differential structure differs for different keys, even if any concrete characteristic is unknown. For example, there is a differential weak key class of IDEA. In that weak key class there exists some differential with probability 1 of the form $\alpha \rightarrow \beta$. For this weak key class, the effective linearity is higher by 1 than the effective linearity of IDEA with key not in the weak key class. We note that using the differential is easier for the purpose of distinguishing whether the key is in the weak key class. However, if decorrelation modules are added before IDEA and

after it, then the differential distinguisher is not applicable anymore, while our distinguisher still succeeds.

Our technique can be used also for key recovery attacks, by measuring the effective linearity of reduced round versions of the cipher. Then, for a given n-round construction, we can try all possible subkeys of the last round, and try to peel it off. If the peeling succeeds, the effective linearity of the obtained cipher equals to the one of $(n-1)$ rounds of the cipher (instead of the expected $(n+1)$ rounds in case of a wrong guess).

5.3 Attacks against Encryption Schemes of the Form $f^{-1}(f(x) \oplus K)$

The distinguisher can be applied directly against encryption schemes of the form $h(x) = f^{-1}(f(x) \oplus K)$ for an arbitrary permutation f, when K is a secret key. In this scenario, the data requirements are even less than in the original distinguisher: the attacker can use known plaintexts instead of adaptively chosen plaintexts.

The attack is performed essentially in the same way as the original distinguisher. The difference is that we already have the values of $h(x)$ which correspond to $g(x)$ in the original distinguisher. The number of found quartets supplies the attacker with the probability $\Pr[h(x) \oplus h(y) = \alpha | x \oplus y = \alpha]$. Thus, we know the average of the probability of the differential $\alpha \rightarrow \alpha$ through $h(\cdot)$, even without constructing the difference distribution table of h itself.

If f is a random permutation, the expected result for h is $2 \cdot 2^{-n}$. This, in contrast to a random permutation $h'(\cdot)$ with the respective probability of 2^{-n}. Therefore, even if f is a perfectly random permutation, the attacker can distinguish between h and a random permutation.

We remark that constructions of the form $h(x)$ are not so rare. For example, two rounds of any involution cipher are of the form $h(x)$. KHAZAD [2] is one example of such a cipher.

Note that there is an adaptive chosen plaintext attack that requires only 2 plaintexts that distinguishes $h(\cdot)$ from a random permutation. Its transformation into a known plaintext attack requires $O(2^{n/2})$ known plaintexts which is equivalent to the data complexity of our attack. However, in the chosen plaintext model our attack has a lower data complexity of $O(2^{n/3})$ (instead of the $O(2^{n/2})$ required for the transformation of the basic attack into a chosen plaintext attack). This last statement is true whenever f is not an APNP.

We also note that this construction covers all block ciphers with cycle 2. This follows from the fact that all ciphers with cycle 2 can be described as $h(x) = f^{-1}(f(x) \oplus K)$ for some permutation f and a non-zero constant K.

6 Summary

In the first part of this paper we presented several distinguishers for highly nonlinear permutations and random permutations. We conclude that while using

Construction	Round Function	Effective Linearty
APNP	APNP	1
Random Permutation	Random Permutation	2
Affine Permutation	Affine Permutation	2^n
2-round Feistel	APNP	3
	Random Permutation	≥ 8
	DES	≥ 220
3-round Feistel	Random Permutation	2

Table 1. Various Constructions and their Effective Linearity

APNPs as part of the encryption scheme seems desirable, using APNPs as the entire cipher can possess inherent weakness.

In the second part of the paper we have examined the structure $f^{-1}(f(x) \oplus K)$ for various permutations f. We have shown how to use the differential properties of this construction in order to study the differential structure of f. We also proved that this construction can be used to effectively determine the average probability of a differential of f. Finally, we have defined the *effective linearity* of a permutation that measures this probability. Table 1 contains various constructions and their effective linearity.

The effective linearity can be used to distinguish between an f that is an almost perfect nonlinear permutation and an f that is a random permutation. On the other hand, it can be used to distinguish ciphers with a relatively close to linear structure from random permutations, even if no concrete differential is known. Our attacks have better performance compared to previously known attacks for several structures. For example, we can distinguish between a random permutation and a permutation formed by two Feistel rounds surrounded by two key-dependent decorrelation modules, regardless of the round functions of the 2-round Feistel construction.

7 Acknowledgments

The authors would like to thank Osnat Ordan and Dana Cohen for their help in conducting the experiments, which verified our claims. It is also a pleasure to acknowledge the references and ideas expressed by Serge Vaudenay, Jennifer Seberry, and Eli Biham. We would also like to thank the anonymous referees for their valuable comments and insightful suggestions.

References

1. Kazumaro Aoki, Serge Vaudenay, *On the Use of GF-Inversion as a Cryptographic Primitive*, proceedings of Selected Areas in Cryptography 2003, Lecture Notes in Computer Science 3006, pp. 234–247, Springer-Verlag, 2004.
2. Paulo S.L.M. Baretto, Vincent Rijmen, *The KHAZAD Block Cipher*, Submitted to NESSIE, available online at *http://www.nessie.eu.org*.

3. Thomas Beth, Cunsheng Ding, *On Almost Perfect Nonlinear Permutations*, Advances in Cryptography, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 65–76, Springer-Verlag, 1994.
4. Eli Biham, Alex Biryukov, Adi Shamir, *Cryptanalysis of Skipjack reduced to 31 rounds*, Advances in Cryptology, proceedings of EUROCRYPT '99, Lecture Notes in Computer Science 1592, pp. 12–23, Springer-Verlag, 1999.
5. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
6. Joan Daemen, Rene Govaerts, Joos Vandewalle, *Weak Keys for IDEA*, Advances in Cryptology, proceedings of CRYPTO '93, Lecture Notes in Computer Science 773, pp. 224–231, Springer-Verlag, 1994.
7. Joan Daemen, Vincent Rijmen *The design of Rijndael: AES — the Advanced Encryption Standard*, Springer-Verlag, 2002.
8. Shimon Even, Yishay Mansour, *A Construction of a Cipher from a Single Pseudorandom Permutation*, Journal of Cryptology, Vol. 10, Number 4, pp. 151–162, Springer-Verlag, 1997.
9. Philip Hawkes, Gregory G. Rose, *Primitive Specification for SOBER-t16 Submission to NESSIE* and *Primitive Specification for SOBER-t32 Submission to NESSIE*, Submitted to NESSIE, available online at *http://www.nessie.eu.org*.
10. Michael Luby, Charles Rackoff, *How to Construct Pseudorandom Permutations from Pseudorandom Functions*, SIAM journal of Computing, Volume 17. No. 2, pp. 373–386, 1988.
11. Mitsuru Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 386–397, Springer-Verlag, 1994.
12. Willi Meier, Othmar Staffelbach, *Nonlinearity Criteria for Cryptographic Functions*, Advances in Cryptology, proceedings of EUROCRYPT '89, Lecture Notes in Computer Science 434, pp. 549–562, Springer-Verlag, 1990.
13. Willi Meier, Othmar Staffelbach, *Fast Correlation Attacks on Stream Ciphers (Extended Abstract)*, Advances in Cryptology, proceedings of EUROCRYPT '88, Lecture Notes in Computer Science 330, pp. 300–315, Springer-Verlag, 1988.
14. US National Bureau of Standards, *Data Encryption Standard*, Federal Information Processing Standards Publications No. 46, 1977.
15. Kaisa Nyberg, *Perfect nonlinear S-boxes*, Advances in Cryptology, proceedings of EUROCRYPT '91, Lecture Notes in Computer Science 547, pp. 378–386, Springer-Verlag, 1991.
16. Kaisa Nyberg, *On the construction of highly nonlinear permutations*, Advances in Cryptology, proceedings of EUROCRYPT '92, Lecture Notes in Computer Science 658, pp. 92–98, Springer-Verlag, 1993.
17. Kaisa Nyberg, *Differentially uniform mappings for cryptography*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 55–64, Springer-Verlag, 1994.
18. Kaisa Nyberg, Lars R. Knudsen, *Provable Security Against Differential Cryptanalysis*, Advances in Cryptology, proceedings of CRYPTO '92, Lecture Notes in Computer Science 740, pp. 566–578, Springer-Verlag, 1993.
19. Oscar S. Rothaus, *On Bent Functions*, Journal of Combinatorial Theory, Series A, Vol. 20 (1976), pp. 305–310, 1976.
20. Jennifer Seberry, Xian-Mo Zhang, Yuliang Zheng, *Relationships Among Nonlinearity Criteria (Extended Abstract)*, Advances in Cryptology, proceedings of EUROCRYPT '94, Lecture Notes in Computer Science 950, pp. 376–388, Springer-Verlag, 1995.

21. Jennifer Seberry, Xian-Mo Zhang, Yuliang Zheng, *Pitfalls in Designing Substitution Boxes (Extended Abstract)*, Advances in Cryptology, proceedings of CRYPTO '94, Lecture Notes in Computer Science 839, pp. 383–396, Springer-Verlag, 1995.
22. Serge Vaudenay, *Provable Security for Block Ciphers by Decorrelation*, Journal of Cryptology, Vol. 16, Number 4, pp. 249–286, Springer-Verlag, 2003.
23. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, Springer-Verlag, 1999.
24. David Wagner, *A Generalized Birthday Problem (Extended Abstract)*, Advances in Cryptology, proceedings of CRYPTO '02, Lecture Notes in Computer Science 2442, pp. 288–304, Springer-Verlag, 2002.

A Theoretical Analysis of the Effective Linearity of a Random Permutation

In this section we analyze the probability of a pair A, B and their respective $g(A), g(B)$ to satisfy $A \oplus B = g(A) \oplus g(B)$ for a permutation f where $g(x) = f^{-1}(f(x) \oplus K)$. Examine the amount of quartets of the form $(A, B, g(A), g(B))$ such that $\gamma = g(A) \oplus g(B) = A \oplus B = \alpha$. As in the analysis for APNP, we also have two cases:

When $g(A) = B$ the expression $\gamma = \alpha$ holds if and only if $g(B) = A$. As in the analysis of APNPs, this event occurs with probability 2^{-n}. Note, that this case also contains the case where $g(B) = A$.

When $g(A) \neq B$ all the elements of the quartet $(A, B, g(A), g(B))$ are distinct. We observe that any quartet of this kind (which we denote by a right quartet) can be uniquely represented by the two ciphertext pairs $(f(A), f(B))$, $(f(A) \oplus K, f(B) \oplus K)$. These two pairs satisfy the following system of equations:

$$\begin{cases} C \oplus D = \beta \\ f^{-1}(C) \oplus f^{-1}(D) = \alpha \end{cases} \tag{2}$$

for some fixed β.

On the other hand, for any fixed value β, consider pairs of inputs to the function f^{-1} that solve System 2. Every pair of such pairs (C, D), (C_1, D_1) can be used if and only if one of the following holds:

$$\begin{cases} C \oplus C_1 = D \oplus D_1 = K \\ C \oplus D_1 = D \oplus C_1 = K \end{cases} \tag{3}$$

The probability of this event is $2 \cdot 2^{-n}$.

For sake of simplicity we assume that this probability is independent of K. Actually there is a measure of dependence of this value on K. Moreover, the same computation can be rewritten in another way such that the dependence on α is neglected and the dependence on K is computed explicitly. The resulting formulae is similar to Equation 5 below when K is substituted instead of α. Actually, this is the case in our distinguisher since we look for the average of the results for different α values, that allows us to reduce the dependence on α.

Since the expected result considers the average for all possible keys, changing the formulae to be dependent on K does not affect the expected result.

Denote by t the number of pairs of solutions of System 2 for a specific value of β, summed over all possible β. Then the expected number of right quartets is $2 \cdot 2^{-n} \cdot t$.

In order to compute the value of t, we have to consider the function f^{-1}. We consider some fixed value β_0 and the element of the difference distribution table of f^{-1}, $DDT^{f^{-1}}$, corresponding to the pair (β_0, α). If $DDT^{f^{-1}}(\beta_0, \alpha) = 2k$, there are k solutions of the system (2) and thus there are $k(k-1)/2$ pairs of solutions. Summing over all the possible values of β, we get the equation

$$t = \sum_{\beta \in GF(2)^n} \frac{DDT^{f^{-1}}(\beta, \alpha)/2 \cdot (DDT^{f^{-1}}(\beta, \alpha)/2 - 1)}{2}, \tag{4}$$

where $DDT^{f^{-1}}$ is the difference distribution table of f^{-1}.

Recall that the difference distribution table of the function f^{-1} is actually the transpose of the difference distribution table of f. Denoting the difference distribution table of f by DDT^f, we have $(DDT^f)^T = DDT^{f^{-1}}$. Thus, we are able to rewrite Equation 4 in terms of the difference distribution table of f as:

$$t = \sum_{\beta \in GF(2)^n} \frac{DDT^f(\alpha, \beta)/2 \cdot (DDT^f(\alpha, \beta)/2 - 1)}{2} \tag{5}$$

Note that the analysis which was performed for the case where f is an APNP is a partial case of the analysis of the general case presented here. Indeed, if f is an APNP then the elements of DDT^f are all equal to $0, 2$ and then we get $t = 0$ since all the elements in the sum equal to zero.

Now, assume that f is a random permutation. As was stated earlier, the elements of DDT^f are distributed according to $2 \cdot Poi(1/2)$. Thus, the value $2k$ is expected to appear in a given row (and in particular, in the row corresponding to α) about $2^n \cdot e^{-1/2} \cdot 2^{-k}/k!$ times. Substituting these figures to Equation 5, we get

$$t = \sum_{k=1}^{2^{n-1}} (2^n \cdot e^{-1/2} \cdot 2^{-k}/k!) \cdot (k \cdot (k-1)/2) = 1/2 \cdot 2^n \left[\underbrace{\sum_{k=1}^{2^{n-1}} k^2 e^{-1/2} 2^{-k}/k!}_{A} - \underbrace{\sum_{k=1}^{2^{n-1}} k e^{-1/2} 2^{-k}/k!}_{B} \right] \tag{6}$$

Let X be a random variable distributed according to $Poisson(\frac{1}{2})$, then we have $A = E[X^2]$ and $B = E[X]$. For such X it is known that $E[X] = \frac{1}{2}$ and $Var[X] = E[X^2] - E[X]^2 = \frac{1}{2}$. Thus, $E[X^2] = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$. Hence we get

$$t = 1/2 \cdot 2^n \left[\sum_{k=1}^{2^{n-1}} k^2 e^{-1/2} 2^{-k}/k! - \sum_{k=1}^{2^{n-1}} k e^{-1/2} 2^{-k}/k! \right] = 1/2 \cdot 2^n \left[\frac{3}{4} - \frac{1}{2} \right] = 1/8 \cdot 2^n \tag{7}$$

Therefore, the expected value of t is $\frac{1}{8} \cdot 2^n$, and the expected number of quartets for a fixed value of α is $\frac{1}{8} \cdot 2^n \cdot 2 \cdot 2^{-n} = \frac{1}{4}$. To compute the probability of a quartet to be a right one, we have to compute the total amount of quartets. Each quartet is constructed by a pair (A, B) such that $A \oplus B = \alpha$. The total number of such pairs is 2^{n-1}. However, each quartet is suggested by the two pairs (A, B) and $(g(A), g(B))$ and thus the ratio should be doubled.

Taking this into consideration, we get that the probability of a quartet to be right is $(\frac{1}{4}/2^{n-1}) \cdot 2 = 2^{-n}$. Summing this result with the result of the first case, we get

$$\Pr[\gamma = \alpha] = 2 \cdot 2^{-n} \tag{8}$$

Summarizing this result, the probability $\Pr[g(A) \oplus g(B) = A \oplus B]$ equals 2^{-n} for APNPs and $2 \cdot 2^{-n}$ for random permutations. This fact can be used in order to distinguish between an APNP and a random permutation.

B Effective Linearity of Feistel Constructions

Let us examine a 2-round Feistel construction $f : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ with a permutation p_1 as the first round function, and a permutation p_2 as the second round function. Both permutations are defined over the space $\{0,1\}^n$.

An input difference (α_L, α_R) that enters this encryption scheme becomes after the first round (without the swap, which has no effect on our results) into $(\alpha_L \oplus \beta_1, \alpha_R)$, where $\alpha_R \xrightarrow{p_1} \beta_1$. After the second round the output difference is $(\alpha_L \oplus \beta_1, \alpha_R \oplus \beta_2)$ where $\alpha_L \oplus \beta_1 \xrightarrow{p_2} \beta_2$.

If the two permutations p_1 and p_2 are independent then the probability of the event $(\alpha_L, \alpha_R) \rightarrow (\alpha_L \oplus \beta_1, \alpha_R \oplus \beta_2)$ is $\Pr[\alpha_R \xrightarrow{p_1} \beta_1] \cdot \Pr[\alpha_L \oplus \beta_1 \xrightarrow{p_2} \beta_2]$. Thus, the difference distribution table of the 2-round Feistel construction contains in any entry the multiplication of the two related entries from p_1's and p_2's difference distribution tables.

The first observation, is that if p_1 and p_2 are both APNPs, we get that all entries in the difference distribution table of the construction are either zero, 4, or 2^{n+1} (in 2^{n+1} out of the 2^{2n} entries in each row/column), up to the $0 \rightarrow 0$ entry. Thus, we can compute the effective linearity of two round Feistel construction with independent APNP round functions is 3.

Our second observation is a more general one in nature. When we inspect the difference distribution table of the 2-round construction (from $2n$ bits to $2n$ bits) in order to compute the effective linearity of the construction, we find that:

$$\begin{aligned} t_f &= \sum_{\beta \in \{0,1\}^{2n}} \binom{DDT^f(\alpha, \beta)/2}{2} = \\ &= \sum_{\beta_1, \beta_2 \in \{0,1\}^n} \sum \binom{DDT^{p_1}(\alpha_R, \beta_1) DDT^{p_2}(\alpha_L \oplus \beta_1, \beta_2)}{2} \geq \\ &\geq 2 \cdot \sum_{\beta_1 \in \{0,1\}^n} \binom{DDT^{p_1}(\alpha_R, \beta_1)}{2} \cdot \sum_{\beta_2 \in \{0,1\}^n} \binom{DDT^{p_2}(\alpha_L, \beta_2)}{2} = \end{aligned}$$

$$= 2 \cdot t_{p_1} \cdot t_{p_2}$$

We note that this is done under the assumption that the probability $\Pr[\gamma = \alpha]$ is quite independent with α (this is required in order to omit the β_1 difference from the sum).

Thus, the value of t_f of the construction is at least twice the multiplied values t_{p_1}, t_{p_2} of the permutations p_1 and p_2. As the effective linearity is related to twice the value of t, then the effective linearity of the construction is the multiplication of the two effective linearities. Or formally:

$$EL(f) \geq EL(p_1) \cdot EL(p_2)$$

Our third observation is that not only $EL(f) \geq EL(p_1) \cdot EL(p_2)$, in many cases $EL(f) \geq EL(p_1) \cdot EL(p_2) + EL(p_2)$. The proof will be given in the final version of this paper. The definition of the effective linearity requires that the entire value K is non-zero. In the Feistel construction, it is possible that the last round will be canceled even if the constant is non-zero (if the left half is 0 and the right half is non-zero). In that case, the left half remains constant during the computation of g (i.e., the left half of $g(x)$ is the same as the left half of x). This case occurs with probability 2^{-n}.

The last observation is valid only if the second permutation is not a linear permutation. If it is a linear permutation, then the second round has no effect on the linearity of the construction.

The same reasoning can be used when the round functions are non-bijective. In that case, as the round functions are not bijective, the number of zero entries in the difference distribution table is greater, and there is a possibility that $p_1(x) = p_1(y)$ even if $x \neq y$. Obviously, this implies that the average probability of $\alpha = \gamma$ is higher.

For example, when considering 2-round DES, we get from [5] that the difference distribution table of a DES round contains about 80% zero entries. This means, that given that an entry is non-zero, its expected value is 6. As the difference distribution table of 2-round DES is related to the multiplication of two 1-round difference distribution tables, the expected entry in non-zero entries is 36. After considering the number of non-zero entries, we get that the effective linearity of such a permutation is about 220. This is done under the assumption that the difference distribution table is uniform (all non-zero entries but the $0 \rightarrow 0$ one are 36). In case it is not uniform (which is more likely) the effective linearity is higher (as the effective linearity is proportional to the sum of squares of the entries, and by Jensen's inequalities is expected to be higher).

Our last result regarding Feistel constructions refers to a 3-round Feistel construction. If the round functions are random permutations, then the left half of the output difference is expected to behave randomly and uniformly. Thus, the difference distribution table, which is a multiplication of the difference distribution table of the left half and the right half, should have the same behavior as of the right side — of a random permutation. Thus, its effective linearity is predicted to be 2.