How to Backdoor a Cipher

Raluca Posteuca[1] and Tomer Ashur[1,2]

[1] imec-COSIC, KU Leuven, Leuven, Belgium
[2] TU Eindhoven, Eindhoven, The Netherlands
[tomer.ashur, raluca.posteuca]@esat.kuleuven.be

Abstract. Newly designed block ciphers are required to show resistance against known attacks, e.g., linear and differential cryptanalysis. Two widely used methods to do this are to employ an automated search tool (e.g., MILP, SAT/SMT, etc.) and/or provide a wide-trail argument. In both cases, the core of the argument consists of bounding the transition probability of the statistical property over an isolated non-linear operation, then multiply it by the number of such operations (e.g., number of active S-boxes).
In this paper we show that in the case of linear cryptanalysis such strategies can sometimes lead to a gap between the claimed security and the actual one, and that this gap can be exploited by a malicious designer. We introduce ЯooC, a block cipher with a carefully crafted backdoor. By using the means of the wide-trail strategy, we argue the resistance of the cipher against linear and differential cryptanalysis. However, the cipher has a key-dependent iterative linear approximation over 12 rounds, holding with probability 1. This property is based on the linear hull effect although any linear trail underlying the linear hull has probability smaller than 1.

Keywords: Block cipher Design, 0-correlation linear hulls, linear hull effect, Kleptography, backdoorgraphy

1 Introduction

The field of cryptography can roughly be split into two paradigms: symmetric-key cryptography and asymmetric-key cryptography. Although they can roughly achieve the same functionality in different ways, the work done in these two fields is fundamentally different. When designing a new asymmetric-key mechanism the work usually takes a more rigorous approach. The designer reduces the security of the mechanism to a problem believed to be computationally hard and shows that breaking the proposed system cannot be easier than solving this problem. In symmetric-key cryptography the field of *provable security* uses a similar approach, usually for building modes of operation. The difference is that instead of reducing the mode to a computationally hard problem, the security of the mode is reduced to the security of the underlying primitive.

However, as far as symmetric primitive design is concerned, a much less rigorous approach is taken. While some general-purpose approaches do exist

(e.g., the wide-trail strategy and other forms of counting probabilistic events), they are mostly used as a threshold (and sometimes not at all) for consideration. Real confidence in new primitives is built over time and through the attempts of cryptanalysts to break them. If significant weaknesses are not found after enough time has passed, the cipher is believed to be secure.

In this paper we present ЯooД[3], a block cipher containing a backdoor. We show that previously published methods for arguing the security of a symmetric-key primitives, for example the wide-trail strategy and automatic search tools such as MILP and SAT/SMT solvers, are insufficient against a malicious designer.

Our backdoor is based on the linear hull effect [Nyb94]. The idea behind this backdoor is to carefully construct the cipher such that after a certain number of rounds, 12 in our case, a certain linear hull occurs with probability 1, despite the fact that any subset of trails in this hull, for a smaller number of rounds, occurs with probability $0 \leq p < 1$. Once that the backdoor is instigated, the rest of the algorithm can further be strengthened to make sure that it is secure, but in a way that does not invalidate the backdoor property. If done properly, it would be nearly impossible for a cryptanalyst to detect this unique deterministic linear property among all 2^{2n} possible linear approximations using current trail based techniques (e.g., automatic tools such as SAT/SMT-solvers). Furthermore, making the propagation of this property key dependent, the designer can improve the hiding of the property by giving it a plausible distribution **and** transform the attack from distinguishing to key-recovery.

1.1 Our contributions

In this paper we discuss the confidence afforded by wide trail arguments (e.g., the wide-trail strategy and/or automated search tools such as MILP or SAT/SMT solvers) when applied by a kleptographic designer. We introduce the block cipher ЯooД and show that although we can argue its security against linear and differential cryptanalysis using standard methods, it contains a carefully crafted backdoor in the form of a key-dependent iterative linear approximation over 12 rounds with probability 1. Since the backdoor is key-dependent, it can be used as a subliminal channel to provide information about the key.

1.2 Organization of the paper

The rest of the paper is organized as follows: in Section 2 we mention some related work and present the notations used in the paper, together with some terminology regarding linear cryptanalysis. Our new cipher is presented in Section 3, together with a motivation regarding our design decisions. In Section 4 we argue the cipher's security using a wide-trail-like approach, while in Section 5 we expose the backdoor we baked into the cipher and show how it can be used

[3] The name of the cipher, ЯooД (pronounced [ru:d]), is a visual encoding of the word backDooR

for distinguishing and key-recovery attacks. Finally, Section 6 concludes this paper.

2 Preliminaries

In this section we present related work, we introduce notation used in the paper, and recall some terminology regarding linear cryptanalysis.

2.1 Related Work

The first approach for finding optimal statistical properties (e.g., differential characteristics and linear trails) is due to Matsui [Mat94]. Later, the wide trail strategy was introduced in [Dae95] as a means to guarantee that no such property has a sufficiently high probability to be exploited by an adversary. More recently, constraint solvers became common in arguing the absence of strong properties in the target cipher.

The wide-trail strategy was used in the design of Rijndael [DR02], after which it become a common strategy to argue the security of newly designed SPN's (e.g., [AAB+20]). For ARX constructions, the long trail strategy is used in a similar manner [DPU+16]. Matsui's search algorithm was used directly by the NSA to argue the security of their ciphers Simon and Speck [BSS+17, Sec. 4]. Examples for the use of automatic search tools are numerous, e.g., [MWGP11].

All the methods above are explicit about dealing only with characteristics and trails. The leap of faith often made by algorithm designers is that the best differential characteristic provides a good estimate for the corresponding differential, and that the best linear trail does the same for the corresponding linear hull.

However, the above is not always the case. Recently, Dunkelman et al. showed in [DKLS20] that counting S-boxes is not enough in the case of differential cryptanalysis. They presented an example where the probability of the strongest differential characteristic is below what is required for a successful attack but the probability of the resulting differential is not.

The case for linear cryptanalysis is even more involved. Whereas the conclusion from [DKLS20] is that the number of differential characteristics contained in the differential is sufficient—at least in theory—for bounding the probability of the resulting differential, even this is not enough in the case of a linear hull. Since the linear trails in the same hull may have positive or negative contributions, the correlation of this hull may be higher or lower than that of the best linear trail. Hence, the number of trails is not enough and knowledge of the trails themselves is required to argue something meaningful. In the sequel, we use this exact property of linear hulls to build a backdoored block cipher.

2.2 Linear cryptanalysis

Linear cryptanalysis is one of the most important techniques used nowadays in the security evaluation of block ciphers. It was introduced in 1993 as a novel

attack against the DES cipher [Mat93]. The basic idea of this technique is to find probabilistic linear relations between bits of the plaintext and the ciphertext. In order to exploit such linear relations in practice, it is necessary that its probability be different from 0.5. The quality of a linear relation (also called linear approximation) is measured by its correlation or bias; in this paper we use the correlations.

2.3 Masks and Approximations

In order to describe a linear approximation, it is common in the literature to use masks associated to the plaintext and ciphertext. Applying a mask to a bit-string is akin to, in essence, a selection of bits of the latter.

Let x and a be binary strings of length n and let a_i and x_i be the i^{th} bit of a and x, respectively. Then,

$$a^t x = \bigoplus_{i=0}^{n-1} a_i x_i,$$

and we say that a is the mask of x.

The positions in which the mask a has the value 1 determine the *active bits* of x, while the remaining bits are said to be inactive.

Let $R_k(x) = y$ denote the round function of a block cipher, where x, y and k are the plaintext, the ciphertext and the key, respectively. A linear approximation for R_k is a tuple (α, β, κ), where α, β and κ are the input mask, the output mask and the key mask, respectively. Let p be the probability that the equation $\alpha^t x \oplus \beta^t y \oplus \kappa^t k = 0$ holds for a fixed k. Then the correlation of the linear approximation (α, β, κ) is defined as $corr(\alpha, \beta, \kappa) = 2p - 1$. In general, both p and $corr(\alpha, \beta, \kappa)$ are key-dependent (see, e.g., [AÅBL12]).

In practice, in order to analyse the correlation of any combination of input and output masks for a particular (non-linear) operation, the most common approach is to compute its LAT (linear approximation table). The LAT is a matrix in which the value in position (i, j) stands for the correlation associated to the input mask i and the output mask j.

Per [AP18], a pair of masks (α, β) is said to be connectable if and only if β can be obtained from α using the rules of propagation introduced in [Mat93, Bih94]. Otherwise, the pair (α, β) is said to be non-connectable. Note that a non-connectable pair is always associated with correlation zero; however the converse is not true and connectable pairs may also be associated with correlation zero.

2.4 Linear Hulls and Trails

An iterated block cipher with r rounds can be described as the following composition of round functions: $Enc_K = R_{k_{r-1}} \circ \ldots \circ R_{k_0}$, where K is the master key and k_i the round key at round i. A linear trail covering r rounds of a block cipher is a sequence of r linear approximations such that the output mask of the i^{th} linear

approximation is equal to the input mask of the $(i+1)^{\text{th}}$ round. Hence, a linear trail can be viewed as an $(r+1)$-dimension vector $(\lambda_1, \lambda_2, \ldots, \lambda_{r+1})$, where the pair $(\lambda_i, \lambda_{i+1})$ denotes the input and output masks at round i, respectively. The correlation of the linear trail is obtained by multiplying the correlation of all single-round linear approximations:

$$corr(\lambda_1, \ldots, \lambda_{r+1}) = \prod_{i=1}^{r} corr(\lambda_i, \lambda_{i+1})$$

In [Nyb94] it was first observed that in some cases there could be more than a single linear trail involving the same plaintext and ciphertext bits. This phenomenon is called the linear hull effect. A linear hull is defined as the set of all linear trails with the same input and output bits (i.e., the input and output masks are fixed, but intermediate round masks may vary). The correlation of a linear hull is obtained by summing the correlations of all linear trails in the set:

$$corr(\alpha, \beta) = \sum_{\lambda_1=\alpha, \lambda_{r+1}=\beta} corr(\lambda_1, \ldots, \lambda_{r+1})$$

The round function of a block cipher can also be viewed as a composition of its component operations. Therefore, the methods described above for computing the correlation of a linear trail can also be applied on a smaller scale to these atomic operations. In [AR16] and [AP18] the authors observed that the linear hull effect can manifest within a single round of a cipher, this being specifically true for the SIMON [BSS+15] and DES [DES] ciphers. In [APSD20], the authors used this property to design a new key-recovery attack against the full DES.

3 Cipher Description

We now describe our new design – the cipher ЯooC-κ where $\kappa \in \{64, 128\}$ stands for the key length. ЯooC is an SP-network based block cipher with a 64-bit state. The state is split into a 4x4 matrix of nibbles. In each round, the S-box layer described below operates on each column independently. Inside each column, four different 16-to-4 S-boxes are applied to the four nibbles such that their outputs form the new column. After the S-box layer is applied to all four columns, three permutations, namely P_1, P_2, and P_3, are applied to the full state. P_1 is used to permute nibbles between rows, P_2 is used to permute bits inside each nibble, and P_3 is used to permute nibbles between columns. Following the permutation layer, a round-dependant constant is injected into the state, followed by a key injection. For ЯooC-128, two more keys are XORed before and after the first and last rounds, respectively.

3.1 The State

ЯooC has a 64-bit state. Similar to AES and other recent SP-network based ciphers, the state is viewed as a 4x4 matrix. Unlike AES, each cell of the matrix

is a 4-bit nibble rather than a byte. The nibbles are indexed 1–16 as depicted in Figure 1.

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	16

Fig. 1. Internal state indices

3.2 The Substitution Layer

The substitution layer is composed of four different 16-bit to 4-bit S-boxes operating on each column independently. Each S-box consists of a mix of modular addition and XORs, operating on three or four nibbles. The description of the four S-boxes is:

$$S_1[a, b, c, d] = (b \boxplus a) \oplus c \tag{1}$$
$$S_2[a, b, c, d] = (b \oplus d) \boxplus c \tag{2}$$
$$S_3[a, b, c, d] = (b \boxplus a) \oplus c \oplus d \tag{3}$$
$$S_4[a, b, c, d] = (b \boxplus a) \oplus d \tag{4}$$

The S-boxes are applied in cyclic order as depicted in Figure 2.

S_1	S_2	S_3	S_4
S_2	S_3	S_4	S_1
S_3	S_4	S_1	S_2
S_4	S_1	S_2	S_3

Fig. 2. The order in which the S-boxes are applied to the state

3.3 The Permutation Layer

The permutation layer is composed of 3 permutations namely P_1, P_2, and P_3. The stated goal of this layer is to ensure proper mixing of all bits.

The column-permutation P_1. The goal of P_1 is to permute nibbles between rows inside the same column. It is similar to AES's ShiftRows, but rather than shifting the nibbles inside each row, it shifts them inside each column. This ensures that when P_2 and P_3 are applied, they would operate on different nibbles in each round. Figure 4 describes the column-permutation P_1.

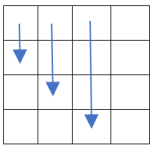

Fig. 3. The column-permutation P_1

The bit-permutation P_2. The first permutation, P_2, permutes bits within the same nibble. The purpose of this permutation is twofold: (i) it ensures that no difference propagates with probability 1 over more than 3 rounds; and (ii) it ensures that P_3, which only operates on the two most significant bits of the nibble, always gets different bits as input.

The bit-permutation P_2 consists of two operations: (i) a swap between the 2 least significant bits, denoted $\leftrightarrow$, and (ii) a cyclic left shift by 1, denoted $\lll$. The two operations used in this permutation are applied in a cyclic order on each row, and the bit-permutation P_2 is depicted in Figure 4.

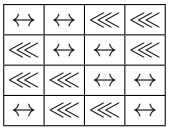

Fig. 4. The bit-permutation P_2

The row-permutation P_3. The last permutation, P_3, permutes bits between nibbles within the same row. The permutation is applied cyclically, inside each row. But instead of shifting the entire nibble, it only moves the two most significant. The row-permutation P_3 is depicted in Figure 5.

Composition. The permutation layer is obtained by composing the three permutations together, i.e., $P_3 \circ P_2 \circ P_1(S)$ where S is the state. Note that we did not seek to optimize diffusion using these permutations and instead argue the security of the cipher based on its large number of rounds.

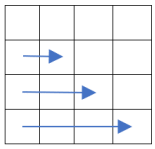

Fig. 5. The row-permutation P_3

3.4 The round constants injection

Following the permutation layer is the round constant injection. The round constant is an 8-bit value c representing the round number. It is viewed as two 4-bit values $c = c_1||c_2$ that are XORed to the third and fourth columns as depicted in Figure 6.

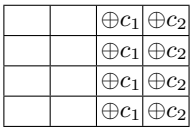

Fig. 6. The round constant injection

3.5 The Key Injection Layer

Finally, at the end of each round, a round key k is injected into the state. ЯooC comes in two versions offering 64-bit and 128-bit security. In the 64-bit version the master key K is used directly as the round key in all rounds, i.e., $k = K$. In the 128-bit version, the master key K is viewed as $K = k_1||k_2$. In order to avoid trivial related-key attacks, we use the PRINCE key schedule [BCG+12]. The 128-bit key $K = k_1||k_2$ is extended into 192 bits by the mapping $(k_1||k_2) \rightarrow (k_1||k||k_2) := (k_1||(k_1 \ll 1) \oplus (k_1 \ll 63)||k_2)$. The keys k_1 and k_2 are used for the initial and the final whitening, while k is used as a round key.

3.6 The Round Function and the Overall Structure

So far, we presented the components used to construct ЯooC's round function. Let us now denote by X the state, α the S-box layer, β the permutation layer and k the round key; then, ЯooC's round function is obtained by a composition of these components:

$$R_i(X) = \beta(\alpha(X)) \oplus i \oplus k\,.$$

The full cipher is then a composition of round functions

$$C = (k_1 \times d) \oplus (R_{n-1} \circ R_{n-2} \circ \ldots R_0(P \oplus (k_2 \times d))),$$

with $d = 0$ for ЯooC-64 and $d = 1$ for ЯooC-128.

Part of the art of designing a new primitive is in choosing the right number of rounds. In the design of ЯooC we decided to take a conservative approach and set very high security margins. As the reader can see in Section 4, it can be shown that the probability of any differential characteristic (resp., squared correlation of any linear trail) dips below 2^{-64} after at most 97 rounds. We add a 100% security margin for ЯooC-64 and 200% security margin for ЯooC-128, resulting in 194 rounds for ЯooC-64 and 291 rounds for ЯooC-128.

4 The Security of ЯooC

In this section we discuss the security of ЯooC against linear cryptanalysis. We stress that the approach we take here is in line with common practices in symmetric-key design (see also Section 2.1). More precisely, we upper bound the probability of a linear trail over four rounds, and use this bound to establish the total number of rounds that should ensure the security of ЯooC against linear cryptanalysis. Since our cipher exhibits linear approximations with probability 1 over at most three rounds, we will consider four to be the reference number of rounds in our analysis.

Resistance Against Linear Cryptanalysis Let C^{S_ℓ} be the full Linear Approximation Table (LAT) for the substitution layer S_ℓ and $C^{S_\ell}_{i,j}$ the correlation of a linear approximation with input mask i and output mask j over S_ℓ. Observing C^{S_ℓ} we see that each $C^{S_\ell}_{i,j}$ can take one of 3 types of values:

- $C^{S_\ell}_{i,j} = 0$ - this is the trivial case normally attributed to non-connectable masks.
- $C^{S_\ell}_{i,j} = 1$ - this is a deterministic approximation with probability 1. In Table 1 we present all pairs of input and output masks with correlation 1. Note that this transition is only possible when the active nibbles of the first two S-boxes have input mask 1_x. The permutation P_2 was designed to guarantee that, while a linear approximation with correlation 1 is possible for a single round, the combination of the substitution and linear layers guarantees that it is not possible for 4 rounds or more.
- $0 < |C^{S_\ell}_{i,j}| < 1$ - this is the normal case. The standard approach when using wide-trail arguments to show the resistance of an algorithm against linear cryptanalysis is to find an upper bound for the absolute correlation of a single transition and multiply it by the minimal number of active S-boxes in a trail of length q as we will do in the sequel.
  For automated search tools the standard approach is to model the algorithm as a set of constraints (e.g., MILP or SAT/SMT) and use the model to

Table 1. Linear approximations with probability 1 for the S-box layer. Note that n_1 and n_2 are two 2-bit values, therefore they can have any value between 0 and 3.

Input mask	Output mask
$(0, 0, n_1, n_2)$	$(0, n_1 \oplus n_2, n_1, n_2)$
$(1, 0, n_1, n_2)$	$(1, n_1 \oplus n_2 \oplus 1, n_1, n_2)$
$(0, 1, n_1, n_2)$	$(1, n_1 \oplus n_2, n_1 \oplus 1, n_2 \oplus 1)$
$(1, 1, n_1, n_2)$	$(0, n_1 \oplus n_2 \oplus 1, n_1 \oplus 1, n_2 \oplus 1)$

count the number of active S-boxes and their transition probabilities. Setting the objective function appropriately allows to retrieve an optimal statistical property (e.g., a linear trail) and bounding this optimal property is commonly used to argue the resistance of the algorithm.

The above case distinction shows that any 3-round linear approximation has absolute correlation smaller than 1. Looking at the non-zero entries in the LAT, we can see that the largest absolute correlation is 2^{-1}. Taking this value as an upper bound for all 3-round linear approximations, and noting that a linear attack with correlation c requires at least c^{-2} data, we can find the number of rounds r after which a linear attack is no longer possible, by setting $q = r/4$ and solving $(2^{-1})^{-2q} = 2^{64}$ for q. We see that setting $q = 32 \longrightarrow r = 128$ ensures that the correlation of all linear trails dip below 2^{-32} for 129 rounds or more. We stress that this is a highly conservative estimate and that the number of rounds after which the correlation of any linear trail is smaller than 2^{-32} is likely to be much smaller than 129 rounds. Nevertheless, for our purpose, this analysis is sufficient.

Unlike the correlation of a single trail, the correlation of the full hull is harder to estimate and overcoming this hurdle is usually an educated guess. Continuing with our conservative approach, we set a safety margin of 100% for ЯооД-64 and double the number of rounds from 129 to 258. For ЯооД-128, we triple the number of rounds and set it to 387. As we will see in the sequel, our backdoor is designed to cover a number of rounds that is a multiple of 12, hence we set the number of rounds for ЯооД-64 to 264, and the number of rounds for ЯооД-128 to 396.

Resistance Against Differential Cryptanalysis By following a similar approach it can be shown that the number of rounds resulting from the previous analysis is sufficient also for arguing the resistance of the cipher against differential cryptanalysis.

5 The Backdoor

In the previous sections we presented ЯооД and argued its security against common attacks. Yet, despite this "proof", ЯооД contains a carefully crafted backdoor exploiting the linear hull effect.

5.1 The rationale behind the design decisions

This sections is meant to guide the reader through the reasoning and techniques that lead to the design of ЯooC. Our strategy starts from an observation on the behaviour of linear trails with respect to modular addition, this being the basis for the design of our S-box layer. The permutation layer has two goals: on the one hand, it plausibly provides enough diffusion to appear benign, while on the other hand it preserves the backdoor property over subsequent S-box layers.

Linear behaviour of the modular addition. Modular addition is one of the main functions used in the design of various symmetric ciphers. As presented in [Wal03], the Algebraic Normal Form of the addition modulo 2^n is obtained using the following recurrence relation:

Lemma 1 Let $z = x + y \pmod{2^n}$. Then,

$$z_i = x_i \oplus y_i \oplus c_i, \text{ where}$$
$$c_0 = 0 \text{ and } c_i = x_{i-1}y_{i-1} \oplus c_{i-1}(x_{i-1} \oplus y_{i-1})$$

By analysing the behaviour of modular addition with respect to linear cryptanalysis, the following observation arises.

Let us define the maximal non-zero index of a mask α by the value

$$index_\alpha = \max_{0 \le i \le n-1} \{\alpha_i = 1\}$$

where α_i represents the i^{th} bit of the binary decomposition of the mask α.

Observation 1 *Let $z = x + y \ mod\ 2^n$. Let α_x, α_y, and α_z be the masks for x, y, and z respectively. Then the tuple $(\alpha_x, \alpha_y, \alpha_z)$ is a connectable tuple of masks only if*

$$index_{\alpha_x} = index_{\alpha_y} = index_{\alpha_z}.$$

For example, if $\alpha_x = 1$, then the only connectable tuple is (1,1,1), yet if $\alpha_x \in \{2, 3\}$, then the connectable tuples are the ones for which $\alpha_y, \alpha_z \in \{2, 3\}$.

Therefore, the tuple masks $(\alpha_x, \alpha_y, \alpha_z)$ define independent classes of connectable masks, depending on the value of the maximal non-zero index.

Linear behaviour of XOR. Let $a = b \oplus c$ and let α_a, α_b, and α_c the masks for a, b, and c, respectively. The rule of propagation of linear trails through the XOR operation implies the following constraint:

$$\alpha_a = \alpha_b = \alpha_c.$$

The S-box Layer. In the design of the S-box layer we used modular addition and XORs, with the goal of designing an invertible function that preserves the modular addition property, possibly with deterministic modifications. More precisely, the S-box layer satisfies the following observation:

Observation 2 *Let (m_0, m_1, m_2, m_3) be the input mask of the S-box layer. Then, if the input mask satisfies the constraint that $m_i \in \{0, 1, 2, 3\}, \forall i \in \{0, 1, 2, 3\}$, then so will any connectable output mask.*

Therefore, if the active bits of the input mask are restricted to the two least significant bits of each nibble, then the active bits of any connectable output mask will also be restricted to these two least significant bits.

For example, the input mask $(1, 2, 3, 0)$ has only the following connectable output masks: $(2, 2, 1, 2), (2, 3, 0, 2), (3, 2, 1, 2)$ and $(3, 3, 0, 2)$ over the S-box layer.

The permutations P_1 and P_3. Since the permutation P_1 permutes the nibbles between rows inside the same column, the property presented in Observation 2 is preserved by this permutation.

The permutation P_3 permutes the two most significant bits of each nibble within the same row. Therefore, the two least significant bits of each nibble are not affected by this permutation, thus the property is preserved.

The permutation P_2. The bit-permutation P_2 consists of two operations: (i) a swap between the 2 least significant bits, and (ii) a cyclic left shift by 1. The former is visibly designed to preserve the backdoor property. In the latter, the property is preserved if besides the constraints described above, the input masks of the nibbles undergoing through the rotation are either 0 or 1.

From the previous observations we can see that by choosing input masks that satisfy Observation 2, all the connectable output masks, for any number of rounds, will satisfy the same constraints.

5.2 The description of the backdoor

Our backdoor is based on the existence of the linear hull effect, more precisely on the presence of an iterative key-dependent linear hull with probability 1 for input mask $(0, 0, 2, 2)^t$ in the second column, and zero for all other columns.

In the remaining of this section we will only refer to the linear approximations of the second column, denoting it by $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)^t$ its corresponding input mask.

The linear hull effect As stated in Section 2, the linear hull effect occurs if there are several linear trails that have fixed input and output masks. In order to fully describe a linear hull, the intuitive approach is to perform an exhaustive evaluation of all the connectable input and output masks for each round. For most of the ciphers in the literature, this evaluation is heavy, both memory and time-wise. But the backdoor crafted in ЯooD allows for a light evaluation of the

connectable linear masks over 12 rounds of the cipher, provided that the input mask is well chosen.

Figure 7 presents the exhaustive evaluation of the key-less linear approximations generated using the input mask $(0,0,2,2)^t$, over 12 rounds of ЯooC. We underline that the input masks of round i are equal to the output masks of round $i-1$, therefore these are omitted from the table. Note that the table also overlooks the linear hulls that have correlation 0. The convergence of the linear hull is most visible in the transition from round 9 to round 10, where the number of connectable output masks decreases from 10 to only 4. But in fact, the linear hull effect first appears already in the transition from round 4 to round 5, since, for example, the input masks $(2,3,0,2)^t$ and $(3,2,0,2)^t$ are connected to the same set of output masks, more precisely $(1,0,0,0)^t$, $(3,2,0,2)^t$, $(3,2,2,0)^t$, and $(1,0,2,2)^t$.

In addition, the alert reader will notice that the masks $(1,0,0,0)^t$ and $(3,2,0,2)^t$ do not appear as possible output masks for round 5. The reason for this is that the 5-round linear hull with input mask $(0,0,2,2)^t$ and either of these output masks, has correlation 0.

Round Number	Output masks	Cumulative correlation
Round 1	$(1,1,0,0)^t$	$corr = 1$
Round 2	$(2,2,0,2)^t$	$corr = 1$
Round 3	$(1,0,0,0)^t$	$corr = 0.5$
	$(1,0,2,2)^t$	$corr = 0.5$
	$(3,2,0,2)^t$	$corr = -0.5$
	$(3,2,2,0)^t$	$corr = 0.5$
Round 4	$(0,0,2,2)^t$	$corr = 0.5$
	$(0,1,0,0)^t$	$corr = 0.25$
	$(0,1,2,2)^t$	$corr = 0.25$
	$(1,0,0,0)^t$	$corr = -0.25$
	$(1,0,2,2)^t$	$corr = -0.25$
	$(1,1,2,2)^t$	$corr = 0.5$
	$(2,3,0,2)^t$	$corr = 0.25$
	$(2,3,2,0)^t$	$corr = -0.25$
	$(3,2,0,2)^t$	$corr = -0.25$
	$(3,2,2,0)^t$	$corr = 0.25$
Round 5	$(0,0,2,2)^t$	$corr = -0.25$
	$(0,1,2,2)^t$	$corr = 0.25$
	$(1,0,2,2)^t$	$corr = -0.25$
	$(1,1,0,0)^t$	$corr = 0.5$
	$(1,1,2,2)^t$	$corr = -0.25$
	$(2,2,2,0)^t$	$corr = 0.25$
	$(2,3,2,0)^t$	$corr = -0.25$
	$(3,2,2,0)^t$	$corr = 0.25$
	$(3,3,0,2)^t$	$corr = 0.5$
	$(3,3,2,0)^t$	$corr = 0.25$
Round 6	$(0,1,2,2)^t$	$corr = 0.5$
	$(1,0,0,0)^t$	$corr = -0.25$
	$(1,0,2,2)^t$	$corr = 0.25$
	$(1,1,0,0)^t$	$corr = -0.25$
	$(1,1,2,2)^t$	$corr = -0.25$
	$(2,2,0,2)^t$	$corr = 0.5$
	$(3,2,0,2)^t$	$corr = 0.25$
	$(3,2,2,0)^t$	$corr = 0.25$
	$(3,3,0,2)^t$	$corr = -0.25$
	$(3,3,2,0)^t$	$corr = 0.25$

Round Number	Output masks	Cumulative correlation
Round 13	$(1,1,0,0)^t$	$corr = 1$
Round 12	$(0,0,2,2)^t$	$corr = 1$
Round 11	$(1,0,0,0)^t$	$corr = 1$
Round 10	$(2,2,0,2)^t$	$corr = 0.5$
	$(2,3,0,2)^t$	$corr = 0.5$
	$(3,2,0,2)^t$	$corr = 0.5$
	$(3,3,0,2)^t$	$corr = -0.5$
Round 9	$(1,1,0,0)^t$	$corr = 0.5$
	$(1,1,2,2)^t$	$corr = -0.5$
	$(2,2,0,2)^t$	$corr = -0.25$
	$(2,2,2,0)^t$	$corr = -0.25$
	$(2,3,0,2)^t$	$corr = 0.25$
	$(2,3,2,0)^t$	$corr = 0.25$
	$(3,2,0,2)^t$	$corr = 0.25$
	$(3,2,2,0)^t$	$corr = 0.25$
	$(3,3,0,2)^t$	$corr = 0.25$
	$(3,3,2,0)^t$	$corr = 0.25$
Round 8	$(0,0,2,2)^t$	$corr = 0.5$
	$(0,1,0,0)^t$	$corr = -0.25$
	$(0,1,2,2)^t$	$corr = 0.25$
	$(1,0,2,2)^t$	$corr = -0.5$
	$(1,1,0,0)^t$	$corr = -0.25$
	$(1,1,2,2)^t$	$corr = 0.25$
	$(2,3,0,2)^t$	$corr = 0.25$
	$(2,3,2,0)^t$	$corr = 0.25$
	$(3,3,0,2)^t$	$corr = 0.25$
	$(3,3,2,0)^t$	$corr = 0.25$
Round 7	$(0,0,2,2)^t$	$corr = -0.25$
	$(0,1,2,2)^t$	$corr = 0.25$
	$(1,0,0,0)^t$	$corr = 0.5$
	$(1,0,2,2)^t$	$corr = 0.25$
	$(1,1,2,2)^t$	$corr = 0.25$
	$(2,2,0,2)^t$	$corr = -0.25$
	$(2,3,0,2)^t$	$corr = 0.25$
	$(3,2,0,2)^t$	$corr = -0.25$
	$(3,3,0,2)^t$	$corr = -0.25$
	$(3,3,2,0)^t$	$corr = 0.5$

Fig. 7. Correlations of the linear hull with input mask $(0,0,2,2)^t$ for any possible output mask with non-zero correlation.

5.3 Key-dependence

As mentioned above, Figure 7 lists all linear approximations spanning from the input mask $(0, 0, 2, 2)^t$ over 12 rounds or less. In this section we discuss the consistency of the linear hull effect in conjunction with different choices for the round key.

As presented in Section 2, the correlation of a linear hull is computed by summing the correlations of all the linear trails in the set. To compute the correlation of a linear trail over one round of the cipher, two steps are in order. The absolute value of the correlation is solely determined by the Boolean functions describing the round's operations, while the sign of the correlation depends on the round key. Therefore, depending on the value of the round key, the sign of a trail's correlation is a random variable, while the absolute value is a constant.

For simplicity, we assume a linear hull consisting of only two linear trails with the same correlation over ρ rounds. For some choices of the round key, the signs of the correlations will be different, therefore the two trails will cancel each other, leading to a linear hull with correlation 0. Otherwise, if the signs are the same, the correlation will be non-zero.

As an instructive example, we recall the transition from round 4 to round 5 mentioned above. The intermediate masks $(2, 3, 0, 2)^t$ and $(3, 2, 0, 2)^t$ are connected to the same set of output masks after a single round, more precisely $(1, 0, 0, 0)^t$, $(3, 2, 0, 2)^t$, $(3, 2, 2, 0)^t$, and $(1, 0, 2, 2)^t$. For the zero key, the correlation of the 5-round linear hull with input mask $(0, 0, 2, 2)^t$ and either of the output masks $(1, 0, 0, 0)^t$, or $(3, 2, 0, 2)^t$ is zero due to the cancellation effect. For the other two output masks of this linear hull, the correlation is non-zero. On the other hand, there exist certain key values for which it is the correlation of the output masks $(1, 0, 0, 0)^t$ and $(3, 2, 0, 2)^t$ that is non-zero, while the correlation for the other two output masks is zero.

The round key is added to the internal state using the XOR operation, hence, due to the rules of propagation for linear trails, the key masks are equal to the output masks of each round. Since all the round keys are equal, there are at most 6 key bits involved in the computation of one linear hull's correlation. More precisely,

- the two least significant bits of the first two nibbles;
- the second least significant bit of the last two nibbles.

For 16 of the 64 values the above 6 bits can take, the backdoor property asserts that the 12-round iterative linear hull holds with probability 1. Table 2 presents these key values.

5.4 Key-recovery

For a 2^{-2} portion of the key space (the 16 cases presented in Table 2) the 12-round iterative linear hull $(0, 0, 2, 2)^t \to (0, 0, 2, 2)^t$ has correlation 1. Observing this correlation, the adversary learns that the key was chosen from this smaller

Table 2. The key bits that ensure the existence of an iterative 12-round linear hull with probaility 1 when using the input mask $(0, 0, 2, 2)^t$

Key bits			
$(0, 0, 0, 0)$	$(3, 1, 0, 0)$	$(2, 2, 0, 0)$	$(1, 3, 0, 0)$
$(3, 0, 2, 0)$	$(0, 1, 2, 0)$	$(1, 2, 2, 0)$	$(2, 3, 2, 0)$
$(1, 0, 0, 2)$	$(2, 1, 0, 2)$	$(3, 2, 0, 2)$	$(0, 3, 0, 2)$
$(2, 0, 2, 2)$	$(1, 1, 2, 2)$	$(0, 2, 2, 2)$	$(3, 3, 2, 2)$

space, and the exhaustive search complexity is reduced by a factor of 2^2. Observing that the correlation is different than 1, the adversary learns that the key was chosen from the complement space, and the exhaustive search complexity is reduced accordingly. Overall, the adversary learns $0.25 \cdot 2 + 0.75 \cdot \log_2(2^{64} - 2^{62}) = 0.8113$ key bits.

6 Conclusion

In this paper we showed how widely used methods for arguing the resistance of symmetric-key primitives are insufficient to prevent a carefully crafted vulnerability. To that effect, we presented ЯooC, a backdoored block cipher whose resistance can be argued using wide-trail arguments, but which contains a vulnerability in the form of a 12-round linear hull with correlation 1. Since the building blocks of the cipher are chosen to preserve this vulnerability, we consider it to be a subliminal channel, i.e., a backdoor. This is a troubling outlook for symmetric-key design as the methods we employed in this paper are highly common, and we hope that future work can address this caveat in arguing resistance against linear attacks.

Acknowledgments We would like to thank Vincent Rijmen for his always useful comments and support on the road to writing this paper, Tanja Lange who encouraged us to complete it, and Thomas H. Ptacek whose comment on Hacker News [Hac] motivated us to start working on this subject. Tomer Ashur is an FWO post-doctoral fellow under Grant Number 12ZH420N.

References

AAB+20. Abdelrahaman Aly, Tomer Ashur, Eli Ben-Sasson, Siemen Dhooghe, and Alan Szepieniec. Design of symmetric-key primitives for advanced cryptographic protocols. *IACR Trans. Symmetric Cryptol.*, 2020(3):1–45, 2020.

AÅBL12. Mohamed Ahmed Abdelraheem, Martin Ågren, Peter Beelen, and Gregor Leander. On the distribution of linear biases: Three instructive examples. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, volume 7417 of *Lecture Notes in Computer Science*, pages 50–67. Springer, 2012.

AP18. Tomer Ashur and Raluca Posteuca. On linear hulls in one round of DES. *IACR Cryptology ePrint Archive*, 2018:635, 2018.

APSD20. Tomer Ashur, Raluca Posteuca, Danilo Sijacic, and Stef D'haeseleer. Generalized matsui algorithm 1 with application for the full DES. In Clemente Galdi and Vladimir Kolesnikov, editors, *Security and Cryptography for Networks - 12th International Conference, SCN 2020, Amalfi, Italy, September 14-16, 2020, Proceedings*, volume 12238 of *Lecture Notes in Computer Science*, pages 448–467. Springer, 2020.

AR16. Tomer Ashur and Vincent Rijmen. On linear hulls and trails. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *Progress in Cryptology - INDOCRYPT 2016 - 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings*, volume 10095 of *Lecture Notes in Computer Science*, pages 269–286, 2016.

BCG+12. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçın. Prince – a low-latency block cipher for pervasive computing applications. In Xiaoyun Wang and Kazue Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 208–225, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

Bih94. Eli Biham. On matsui's linear cryptanalysis. In Santis [San95], pages 341–355.

BSS+15. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The SIMON and SPECK lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference, San Francisco, CA, USA, June 7-11, 2015*, pages 175:1–175:6. ACM, 2015.

BSS+17. Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. Notes on the design and analysis of SIMON and SPECK. *IACR Cryptol. ePrint Arch.*, 2017:560, 2017.

Dae95. J. Daemen. Cipher and hash function design strategies based on linear and differential cryptanalysis. 1995.

DES. FIPS publication 46-3, Data Encryption Standard (DES).

DKLS20. Orr Dunkelman, Abhishek Kumar, Eran Lambooij, and Somitra Kumar Sanadhya. Counting active s-boxes is not enough. In Karthikeyan Bhargavan, Elisabeth Oswald, and Manoj Prabhakaran, editors, *Progress in Cryptology - INDOCRYPT 2020 - 21st International Conference on Cryptology in India, Bangalore, India, December 13-16, 2020, Proceedings*, volume 12578 of *Lecture Notes in Computer Science*, pages 332–344. Springer, 2020.

DPU+16. Daniel Dinu, Léo Perrin, Aleksei Udovenko, Vesselin Velichkov, Johann Großschädl, and Alex Biryukov. Design strategies for ARX with provable bounds: Sparx and LAX. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 484–513, 2016.

DR02. Joan Daemen and Vincent Rijmen. *The Design of Rijndael*. Springer-Verlag, Berlin, Heidelberg, 2002.

Hac. Linux cryptography: Speck's real standing with the academic community. `https://news.ycombinator.com/item?id=17215626`. Accessed: 2021-03-31.

Mat93. Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer, 1993.

Mat94. Mitsuru Matsui. On correlation between the order of s-boxes and the strength of DES. In Alfredo De Santis, editor, *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 366–375. Springer, 1994.

MWGP11. Nicky Mouha, Qingju Wang, Dawu Gu, and Bart Preneel. Differential and linear cryptanalysis using mixed-integer linear programming. In Chuankun Wu, Moti Yung, and Dongdai Lin, editors, *Information Security and Cryptology - 7th International Conference, Inscrypt 2011, Beijing, China, November 30 - December 3, 2011. Revised Selected Papers*, volume 7537 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2011.

Nyb94. Kaisa Nyberg. Linear approximation of block ciphers. In Santis [San95], pages 439–444.

San95. Alfredo De Santis, editor. *Advances in Cryptology - EUROCRYPT '94, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*. Springer, 1995.

Wal03. Johan Wallén. Linear approximations of addition modulo 2n. In Thomas Johansson, editor, *Fast Software Encryption*, pages 261–273, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.