Analysis of Impossible, Integral and Zero-Correlation Attacks on Type-II Generalized Feistel Networks using the Matrix Method

Céline Blondeau[1] and Marine Minier[2]

[1] Department of Computer Science, School of Science, Aalto University, Finland.
[2] Université de Lyon, INRIA, INSA-Lyon, CITI, France.

Abstract. While some recent publications have shown some strong relations between impossible differential and zero-correlation distinguishers as well as between zero-correlation and integral distinguishers, we analyze in this paper some relation between the underlying key-recovery attacks against Type-II Feistel networks. The results of this paper are build on the relation presented at ACNS 2013. In particular, using a matrix representation of the round function, we show that we can not only find impossible, integral and multidimensional zero-correlation distinguishers but also find the key-words involved in the underlined key-recovery attacks. Based on this representation, for matrix-method-derived strongly-related zero-correlation and impossible distinguishers, we show that the key-words involved in the zero-correlation attack is a subset of the key-words involved in the impossible differential attack. Other relations between the key-words involved in zero-correlation, impossible and integral attacks are also extracted. Also we show that in this context the data complexity of the multidimensional zero-correlation attack is larger than that of the other two attacks.

Keywords: block ciphers, Feistel like ciphers, impossible differential, zero-correlation, integral, key-recovery attacks, matrix method.

1 Introduction

Impossible differential (ID) [2,21], integral (INT) [22] and multidimensional zero-correlation (ZC) [10] attacks are efficient attacks for word-oriented block ciphers such as Feistel-like ciphers.

Classically, *ID distinguishers* take advantage of differentials which never occur for the studied permutations. The security of word-oriented block ciphers is evaluated with respect to this attack. As early as in 2003, based on a matrix representation of the round function, automated methods to find IDs have been proposed [20].

In *ZC cryptanalysis*, the attacker rather takes advantage of linear approximations that have probability 1/2 to hold. This relatively new attack, which can be seen as a multidimensional linear attack with capacity equal to zero [9], has also been applied to many word-oriented block ciphers [11,9,8,27,32,31]. The published attacks, which improve upon the state-of-the-art cryptanalysis, can either cover more rounds than the ID attacks, or perform in less time than the ID attacks on the same number of rounds.

In *INT attacks*, attackers look for particular subsets of chosen plaintexts where some parts of the input are equal to constant whereas the other parts take all possible values. The interesting subsets are the ones such that the sum taken on all the input values after a certain number of rounds is equal to a known value, to an other sum or to zero at some particular locations. This attack, also known as saturation or square attack was originally proposed by Knudsen as a dedicated attack against Square [17].

Recently, mathematical and structural relations between the underling distinguishers have been discussed in the literature. In 2012, Bogdanov et al. [9] showed that the existence of a particular type of integral distinguisher, called zero-correlation integral distinguisher, implies the existence of a zero-correlation distinguisher. Among other relations, in [6,7] it is shown that in some particular cases, ZC distinguishers and ID distinguishers are mathematically equivalent. While this condition is not often verified in practice, it is shown in [3] that for many Feistel-type ciphers, ID and ZC distinguishers which are build using a matrix method can be derived from each other. The results of [3] are derived from a matrix representation of the cipher [1] and from the fact that ID and ZC distinguishers can be derived from this representation using the so-called $\mathcal{U}$-method [20].

Motivation. In practice, the security regarding these three attacks is often analyzed independently and key-recovery attacks in the ID, ZC and INT contexts are part of different publications. As illustrated in Table 1 usually the number of attacked rounds in all these contexts is similar. When the attacks cover the same number of rounds, it often seems that the relation between these attacks can be seen as a kind of data/time/memory trade-off. While depending of the cipher, different tricks can be used to improve the time and memory complexity of the attacks, we observe that the number of key-words involved in the attack is usually independent of the method used to perform the key-recovery attack.

Our Contributions.

Relation between ZC and INT distinguishers. From the preliminary link between ZC and integral ZC distinguishers presented in [9], we derive a general relation between ZC and INT distinguishers. In particular we discuss cases where matrix-method-derived INT distinguishers cover less or more rounds than matrix-method-derived ZC distinguishers.

Relation between the data complexities of ID, ZC and INT attacks. When the distinguishers are matrix-method-related the same number of differential and linear approximations are involved in the attack. In such a case, we can compare the data complexity of ID and ZC. In particular we show that the data complexity of a ZC attack is in that case larger than that of an ID attack. A similar comparison with INT attacks is also studied.

Relation between the key-words involved in ID, ZC, and INT attacks. Illustrated by the ID, INT and ZC attacks on LBlock [34], we show that there exists a strong relation between the key-words involved in the different key-recovery attacks.

Cipher	Attacked Rounds	Rounds Dist.	Type	Data	Time	Memory	Ref.
HIGHT	27	16	ID	2^{58} CP	$2^{126.6}$	2^{120}	[16]
HIGHT	27	16	ZC	$2^{62.79}$ DKP	$2^{120.78}$	2^{43}	[33]
Camellia-128	11	7	ID	$2^{118.4}$CP	$2^{118.43}$	$2^{92.4}$	[15]
Camellia-128	11	7	ZC	$2^{125.3}$KP	$2^{125.8}$	2^{112}	[12]
SIMON-32/64	19	11	ID	2^{32} CP	$2^{62.56}$	2^{44}	[15]
SIMON-32/64	20	11	ZC	2^{32} KP	$2^{56.96}$	$2^{41.42}$	[30]
SIMON-32/64	21	15	INT	2^{31} CP	2^{63}	2^{54}	[30]
LBlock	22	15	INT	2^{61}CP	$2^{70.00}$	2^{63}	[25]
LBlock	22	14	ID*	2^{58}CP	$2^{79.28}$	2^{68}	[19]
LBlock	22	14	ZC†	2^{64} DKP	$2^{70.54}$	2^{64}	[27]
LBlock	22	14	ID†	2^{60} CP	$2^{71.53}$	2^{59}	[14]
LBlock	23	14	ID†	2^{59} CP	$2^{75.36}$	2^{74}	[14]
LBlock	23	14	ZC	$2^{63.87}$ DKP	$\mathbf{2^{73.94}}$	$\mathbf{2^{60}}$	this paper

Table 1: Best attacks on some well known ciphers. *: In [27], it is mentioned that the attack applies to an old version of the cipher. †: parameters provided in the original article for the lowest time complexity. CP: Chosen plaintexts. KP: Known plaintexts. DKP: Distinct known plaintexts.

In particular, we show that when the matrix-method-derived distinguishers are strongly related then the key-words involved in a ZC attack on a Type-II GFN correspond to a subset of the key-words involved in an ID attack.

Attack on LBlock. For illustration purposes, we present a ZC attack on 23 rounds of LBlock. The time and memory complexities of this attack are smaller than those of the recent ID attack [14] on this cipher.

Outline. Some preliminary notations are defined in Section 2. In Section 3, we describe how the matrix method can be used to find ID, ZC and INT distinguishers on Feistel ciphers and recall the relations between these different distinguishers. In Section 4, we compare the data complexity of these attacks and illustrate on LBlock the strong relation between the key-words involved in the different attacks. In Section 5, we explain how we can use the matrix method to determine the key-words involved in the key-recovery part of the attack. In Section 6, we present an attack on LBlock. Section 7 concludes this paper.

2 Preliminaries

The structural link between ID and ZC distinguishers described in [3] is relevant for Feistel-like ciphers commonly referred as Generalized Feistel Network (GFN) [28]. More precisely, the round $i+1$ of a GFN inputs a block X_i of n bits divided in $b \geq 2$ blocks of c bits each and outputs a block X_{i+1}. We denote by $X_i[0], \cdots, X_i[b-1]$ the b input blocks of a GFN round and by $X_{i+1}[0], \cdots, X_{i+1}[b-1]$ the corresponding output blocks. A GFN can be separated into two successive layers, as done in [28,1]: a non-linear layer and a permutation layer, as shown in Figure 1. The non-linear layer is made of key-dependent functions F_i which input some blocks of size c bits and where the corresponding outputs are added (usually Xor-ed) to some other blocks. The permutation layer is a block-wise permutation of the b blocks denoted by π. For example, for the classical Type-II GFN the permutation is the circular shift.

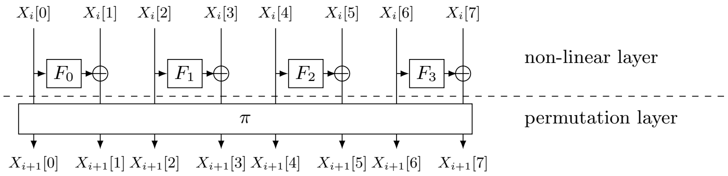


Fig. 1: One round of a Type-II GFN with $b = 8$ blocks.

A generic method that uses a matrix representation and captures the previous definition of GFNs is presented in [1]. This representation could be useful to find ID, ZC and INT distinguishers as well as to show some links between them as began in [3]. Definition 1 sums up this approach for the Type-II GFNs.

Definition 1. *Omitting key and constant addition, the round function of a Type-II GFN with b branches, b even, can be matricially represented as a combination of two $b \times b$ matrices $\mathcal{F}$, $\mathcal{P}$ with coefficients $\{0, 1, F_i\}$ where the $\{F_i\}_{i \leq b/2}$ denote the internal non-linear functions.*
• *Representing the non-linear layer (F-layer), the non-zero coefficients of the matrix $\mathcal{F}$ are equal to 1 in the diagonal and have coefficient F_i in row j and column ℓ if the input of the function F_i is given by the ℓ-th branch and the output is Xor-ed to the j-th branch. Meaning that, for a Type-II GFN, $\mathcal{F}$ have one F_i on each even row and even column.*
• *Representing the permutation of the branches (P-layer), the matrix $\mathcal{P}$ is a permutation matrix with only one non-zero coefficient per line and column. From these two matrices, a Type-II GFN round function can be represented by a $b \times b$ matrix $\mathcal{R}$ as $\mathcal{R} = \mathcal{P} \cdot \mathcal{F}$, the inverse of the round function is $\mathcal{R}^{-1} = \mathcal{F} \cdot \mathcal{P}^{-1}$.*

As observed in [28], optimal diffusion block permutations have the property that any input block with an even number is mapped to a block with an odd number, and vice versa. A Feistel cipher with this property is called *alternating Type-II GFN* (AGFN). Some of the results of this paper assume alternating Type-II GFNs.

In [27,3], another matrix representation is used to find ZC distinguishers on Feistel-like ciphers. The $\mathcal{U}$-method used to find ZC distinguishers relies on the mirror representation of the round function.

Definition 2. *For a Feistel-like cipher given the matrix representation of the round function $\mathcal{R} = \mathcal{P} \cdot \mathcal{F}$, we call mirror function the round function described by the matrix $\mathcal{M} = \mathcal{P} \cdot \mathcal{F}^T$, where $\mathcal{F}^T$ denotes the transposition of the matrix $\mathcal{F}$.*

3 Distinguishers

The matrix method, which is defined in [20,35,27,24] is used to find respectively ID, INT and ZC distinguishers. For these three attacks, different quantities are involved. For instance, ID distinguishers are derived from an inconsistency between partial differences and ZC distinguishers are derived from an inconsistency between partial linear masks. To show the similarities between ID, ZC and INT attacks, the same notations will be used in the different contexts. These notations are summed up in Table 2. For instance, for a given a value, the quantity A_a denotes respectively a non-zero difference in the ID context, a non-zero mask in the ZC context and symbolizes a permutation of the elements of $\mathbb{F}_2^c$ in the INT context. While for type-1 distinguishers (see Definition 3 in Section 3.2) $\tilde{A}_a$ does not have to be differentiated from A_a, such distinction is necessary to find type-2 and type-3 distinguishers. Using these notations, the state of a b-branches GFN can be represented as a vector of b elements of the form given in Table 2. When necessary, this b-word vector will be denoted by V_i^{ID}, V_i^{ZC} or V_i^{INT}. For example, in Figure 2 we have : $V_{i-1}^{ID} = (0, \tilde{A}_0, A_1, 0, A_2, A_3, U_0, U_1)$. The vector V_i^{ID} is computed using the rules given in Table 3.

	ID	ZC	INT
0	Zero difference	Zero mask	Constant value*
$\tilde{A}_a$	Known non-zero difference	Known non-zero mask	Known permutation
A_a	Non-zero difference	Non-zero mask	Permutation
U_u	Unknown difference	Unknown mask	Unknown
$A_a \oplus A_{a'}$	$= U_u$	$= U_u$	Known sum

Table 2: General notations representing a partial-state. a and u are some counters. *: Usually denoted by C in the INT context.

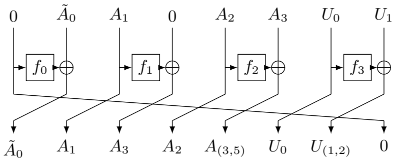


Fig. 2: An example of passing one round of a classical Type-II GFN in the ID context using the previous notations and the rules of Table 3.

3.1 The Matrix Method on GFNs

ID Distinguisher. In [20,24], the matrix method which allows to efficiently find ID distinguishers for Feistel-type ciphers is described. The algorithm mainly implements the rules given in Table 3 to propagate a partial difference through the rounds of a cipher. The values a and u are arbitrary values starting at 0. As explained in [35], these indexes allow simplifications of type $A_a \oplus A_a = 0$.

Using the matrix representation of the round function, we can derive an ID distinguisher on a GFN.

Proposition 1. *Given V_0^{ID} and W_0^{ID} a representation of the input and output differences, we have an ID distinguisher (V_0^{ID}, W_0^{ID}) on $s_0 + s_1$ rounds if we have an inconsistency between $X = \mathcal{R}^{s_0} \cdot V_0^{ID}$ and $Y = \mathcal{R}^{-s_1} \cdot W_0^{ID}$.*

Rules to pass an F function

F	0	$\tilde{A}_i$	A_i	U_j or $A_i \oplus A_j$
	0	$A_{(\max_a)+1}$	$A_{(\max_a)+1}$	$U_{(\max_u)+1}$

Rules to pass a $\oplus$

$\oplus$	0	$\tilde{A}_i$	A_i	U_i
0	0	$\tilde{A}_i$	A_i	U_i
A_j	A_j	$\tilde{A}_i \oplus A_j$	$A_i \oplus A_j$	$A_j \oplus U_i$
$\tilde{A}_j$	$\tilde{A}_j$	$\tilde{A}_i \oplus \tilde{A}_j$	$A_i \oplus A_j$	$A_j \oplus U_i$
U_j	U_j	$\tilde{A}_i \oplus U_j$	$A_i \oplus U_j$	$U_j \oplus U_i$

Table 3: Propagation rules in the ID, ZC and INT context when using the matrix method.

More details on inconsistencies in the ID and ZC contexts are provided later in this section.

ZC Distinguisher. In [27], the matrix method to find ZC distinguishers is presented. The main difference with the method used in the ID context comes from the fact that the mirror matrix $\mathcal{M}$ (see Definition 2) must be used instead of the round function matrix $\mathcal{R}$.

Proposition 2. *Given V_0^{ZC} and W_0^{ZC} a representation of the input and output linear masks, we have a ZC distinguisher (V_0^{ZC}, W_0^{ZC}) on $s_0 + s_1$ rounds if we have an inconsistency between $\mathcal{M}^{s_0} \cdot V_0^{ZC}$ and $\mathcal{M}^{-s_1} \cdot W_0^{ZC}$.*

INT Distinguisher. In [35], the authors proposed an algorithm to automatically find INT distinguishers. The algorithm uses the same rules and the same matrix representation $\mathcal{R}$ as in the ID context, only the termination rules differ.For Type-II GFNs, these rules can be expressed as follows.

Corollary 1. *Given a representative vector Z^{INT}, for a GFN as given in Definition 1, we have an INT distinguisher on $s_0 + s_1$ rounds if the following rules are fulfilled:*
- *Termination at the end of an INT distinguisher: If $W_0^{INT} = \mathcal{R}^{s_1} \cdot Z^{INT}$ is such that there exists $i < b$ and a set J with $W_0^{INT}[i] = \oplus_{j \in J} A_j$ and for $\tilde{W} = \mathcal{R}^{s_1+1} \cdot Z^{INT}$, for all i and J, we have $\tilde{W}[i] \neq \oplus_{j \in J} A_j$.*
- *Termination at the beginning of an INT distinguisher: If $V_0^{INT} = \mathcal{R}^{-s_0} \cdot Z^{INT}$ is such that $\exists\, i \in \{0, \cdots, (b-1)\}$ with $V_0^{INT}[i] = 0$ (with classical notation $= C$).*

Throughout this paper, we assume that the ID, ZC and INT distinguishers are derived from a matrix method.

3.2 Equivalence between Matrix-Method-Derived ID and ZC Distinguishers

In [3], a condition of equivalence between matrix-method-derived ID and ZC distinguishers is given.

Theorem 1 ([3]). *Let $\mathcal{R}$ be the matrix representation of the round function of a GFN and $\mathcal{M}$ be the matrix representation of its mirror function as given in Section 2. If there exists a $b \times b$ permutation matrix $\mathcal{Q}$ such that*

$$\mathcal{R} = \mathcal{Q} \cdot \mathcal{M} \cdot \mathcal{Q}^{-1} \text{ or } \mathcal{R} = \mathcal{Q} \cdot \mathcal{M}^{-1} \cdot \mathcal{Q}^{-1}, \tag{1}$$

we deduce that: an impossible differential distinguisher on r rounds involving a number of differentials equal to M exists if and only if a zero-correlation linear distinguisher on r rounds involving M linear masks exists.

As shown in [3], most of the Feistel networks verify this condition. In the remainder of this paper, we say that a Feistel-like cipher has matrix-method-derived *related ID and ZC distinguishers* when the relation is derived from the previous property.

Lemma 1. *Given the matrices $\mathcal{R} = \mathcal{P} \cdot \mathcal{F}$ and $\mathcal{M} = \mathcal{P} \cdot \mathcal{F}^T$,*
• If the condition $\mathcal{R} = \mathcal{Q} \cdot \mathcal{M} \cdot \mathcal{Q}^{-1}$ of (1) is fulfilled and $\mathcal{P} \cdot \mathcal{Q} = \mathcal{Q} \cdot \mathcal{P}$ we have $\mathcal{F} = \mathcal{Q} \cdot \mathcal{F}^T \cdot \mathcal{Q}^{-1}$.
• If the condition $\mathcal{R} = \mathcal{Q} \cdot \mathcal{M}^{-1} \cdot \mathcal{Q}^{-1}$ of (1) is fulfilled we have $\mathcal{F} = \mathcal{Q}_1 \cdot \mathcal{F}^T \cdot \mathcal{Q}_2$ with $\mathcal{Q}_1 = \mathcal{P}^{-1} \cdot \mathcal{Q}$ and $\mathcal{Q}_2 = \mathcal{P}^{-1} \cdot \mathcal{Q}^{-1}$.

Remark 1. For a Type-II GFN as given in Definition 1, there always exists a permutation matrix $\mathcal{Q}$ such that $\mathcal{F}^T = \mathcal{Q} \cdot \mathcal{F} \cdot \mathcal{Q}^{-1}$. This matrix $\mathcal{Q}$ corresponds to the permutation $\sigma : (0, \cdots, (b-1)) \to (1, 0, 3, 2, \cdots, (b-1), (b-2))$.

While in [13] improved ID distinguishers are presented, using the matrix method for Type-II GFNs we can observe three types of ID and ZC distinguishers [24].

Definition 3. *Given $X = \mathcal{R}^{s_0} \cdot V_0^{ID}$ and $Y = \mathcal{R}^{-s_1} \cdot W_0^{ID}$ or $X = \mathcal{M}^{s_0} \cdot V_0^{ZC}$ and $Y = \mathcal{M}^{-s_1} \cdot W_0^{ZC}$. Following the work of [13,24], for a Type-II GFN, we define three types of distinguishers. Let $p \in \{0, \cdots, (b-1)\}$ denote the index of the studied state Y and $Y[p]$ the status of this p-th word.*
•An ID (resp. ZC) distinguisher of type-1 denoted by ID1 (resp. ZC1) is a distinguisher with independent input and output differences (resp. masks). For these distinguishers, the inconsistency is usually:

$$\exists\, p \in \{0, \cdots, (b-1)\} \mid (Y[p] = 0 \text{ and } X[p] = A_a) \text{ or } (Y[p] = A_a \text{ and } X[p] = 0).$$

• An ID (resp. ZC) distinguisher of type-2 denoted by ID2 (resp. ZC2) is a distinguisher where a non-zero output-difference word (resp. output-mask word) should be different from an input one. Given $\tilde{A}_{a_0}$ a word of V_0 and W_0, the inconsistency is:

$$\exists\, p \in \{0, \cdots, (b-1)\} \mid Y[p] = X[p] = \tilde{A}_a.$$

• An ID (resp. ZC) distinguisher of type-3 is a distinguisher where non-zero input- and output-difference words (resp. output-mask words) should be equal. Given $\tilde{A}_{a_0}$ a word of V_0 and W_0, we have an ID distinguisher of type-3 (ID3) on $s_0 + s_1 + 1$ rounds if:

$$\exists\, p \in \{0, \cdots, (b-1)\} \mid (\mathcal{F} \cdot X)[p] = \tilde{A}_{a_0} \oplus A_{a_1} \text{ and } (\mathcal{P}^{-1} \cdot Y)[p] = \tilde{A}_{a_0}. \quad (2)$$

Replacing $\mathcal{F}$ by $\mathcal{F}^T$ in (2) we obtain a ZC distinguisher of type-3 (ZC3).

Clearly, our matrix method captures the distinguishers found using the matrix method described in [24] as in the rules given in Table 3 used to define the transitions of our matrix method, we have: $F(\tilde{A}_i) = A_{(\max_a)+1}$ and $\tilde{A}_i \oplus A_j$ is kept as $\tilde{A}_i \oplus A_j$ when crossing a xor operation. The link between ZC and ID distinguishers thus depends of Theorem 1. However, our matrix method does not capture the distinguishers of [13] which depend on differential transition properties of the involved S-boxes.

3.3 Comparison with INT Distinguishers

A zero-correlation integral distinguisher is defined in [9] as an integral distinguisher with balanced output words. In Section 3 of [9] a direct relation between zero-correlation integral distinguisher and ZC distinguisher of type-1 is extracted.

As a zero-correlation integral distinguisher is stronger than a general integral distinguisher with sum over an output word equal to zero, we can see that in general, one more round could be added to the zero-correlation integral distinguisher to transform it into an INT distinguisher because if the partial outputs of the zero-correlation integral distinguisher occur equally often, the last linear transformation maps it into an integral distinguisher. More precisely for Type-II GFNs, we deduce that the number of rounds on which an INT distinguisher applies, is greater than or equal to the number of rounds on which a ZC distinguisher of type-1 applies. More precisely for an alternating Type-II GFN, we can show the following results which are proved in the extended version [5].

Lemma 2. *For an alternating GFN of Type-II, given $s^{ZC1}_{\max}$ (resp. $s^{INT}_{\max}$) the maximum number of rounds on which a matrix-method-derived ZC distinguisher of type-1 (resp. an INT distinguisher) applied, we have $s^{ZC1}_{\max} = s^{INT}_{\max}$ or $s^{ZC1}_{\max} = s^{INT}_{\max} - 1$.*

Remark 2. In practice, for the Type-II GFNs of [28], we observed (see for instance Table 4) and proved[3], that the number of rounds on which matrix-method-derived INT and ID distinguishers apply is $s^{INT} = s^{ID}$, $s^{INT} = s^{ID} - 1$ or $s^{INT} = s^{ID} + 1$. This relation follows by the relation between the ZC and ID distinguishers.

Name	π	s^{ID}	s^{INT}
Type-II	{7,0,1,2,3,4,5,6}	17	16
Nyberg	{2,0,4,1,6,3,7,5}	14	15
No.1	{3,0,1,4,7,2,5,6}	11	11
No.2	{3,0,7,4,5,6,1,2}	10	11

Table 4: GFNs of [28] with $b = 8$. For these ciphers based on the results of [3], we can show that $s^{ID} = s^{ZC}$. π defines the permutation of the branches.

Remark 3. According to Prop. 1 and Cor. 1, the same matrix representation is used to derive ID and INT distinguishers. Given Z^{INT} as in Cor. 1, we denote by $\tilde{X} = \mathcal{R}^{s_0} \cdot Z^{INT}$ and $\tilde{Y} = \mathcal{R}^{-s_1} \cdot Z^{INT}$. Now for a Type-II alternating GFN, we can find the different types of impossible differential distinguishers by analyzing the different inconsistency cases between $\tilde{X}$ and $\tilde{Y}$ or a permutation of $\tilde{Y}$.

[3] These cases can be derived from the proofs given page 15-16 of [28].

4 The Key-Recovery: Notations and Examples

The time and memory complexities of ID, ZC and INT key-recovery attacks are dependent on the number of key-words involved in the attacks. In ID attacks, a distinction is classically made between the key-words which lead to a reduction of the number of pairs and the key-words which are just helping in the partial encryption/decryption without reducing the number of involved pairs ([14], Figure 2). While the first ones are usually called *sieving key-words* leading to a so called *sieving step*, we called *guessed key-words* the latter ones. The *involved key-words* corresponds to the sieving key-words and guessed key-words.

Similar concepts can be observed in ZC and INT attacks, where most of the involved key-words conduct to a reduction of the size of the table storing the partial distribution. To simplify the description and the comparison, the same terminology will be used in these three contexts. In the ZC and INT contexts, a sieving key-word corresponds to a key-word leading to a reduction of the size of the stored distributions. The number of key-words involved in a key-recovery attack on $s + r$ rounds is denoted by $|\mathcal{K}|$. The number of key-words leading to a sieving step is denoted by $|\mathcal{S}|$ and the number of guessed key-words is denoted by $|\mathcal{G}|$. To compute the complexity of a key-recovery attack we denote by $HW(V)$, the weight of a b-word vector V as $HW(V) = b - |\{\ell \in \{0 \cdots (b-1)\} | V[\ell] = 0\}|$.

To illustrate this concept, we discuss the examples of ID, ZC and INT attacks on a 4 branches Feistel as well as on LBlock [34].

4.1 Example of Attacks on a 4 Branches Feistel Network

In Figure 3, we illustrate ID, ZC and INT key-recovery attacks on the last three rounds of a classical type-II GFN with 4 branches. While some of the notations will be defined later, we illustrate the meaning of guessed and sieving key-words in the ID and ZC context by providing some steps of the key-recovery algorithms. For instance some steps of the generic ZC attack on this structure could be:

- Store the distribution of $[X^0[2], X^{12}[0,1,2,3]]$ (5 words)
- Try the sieving key K_1^{12} and compute the distribution of $[X^0[2], X^{11}[1], X^{12}[1,2]]$ (4 words)
- Try the guessed key K_2^{12} and compute the distribution of $[X^0[2], X^{11}[1,2,3]]$ (4 words)

The last step consists at studying the distribution of $[X^0[2] \oplus X^9[1]]$.

4.2 Relation between the Involved Key-Words on LBlock

LBlock is a new lightweight block cipher designed by Wu and Zhang in 2011 [34]. It uses 80-bit keys and 64-bit blocks seen at nibble level and is based on a modified 32-round Feistel structure. We denote by $P = L_0||R_0$ the 64-bit plaintext, where L_0 and R_0 are 32-bit vectors. The encryption process is as follows: $R_i = L_{i-1}$ and $L_i = F(L_{i-1}, K_i) \oplus (R_{i-1} \lll 8)$ where the F function could be divided into three steps (see Figure 4). First, the 32-bit subkey K_i is added to L_{i-1} by a simple XOR. Then, a nonlinear layer applies to the result. This nonlinear layer consists of the application nibble by nibble of eight different 4-bit Sboxes $S_0, \ldots, S_7$ (see

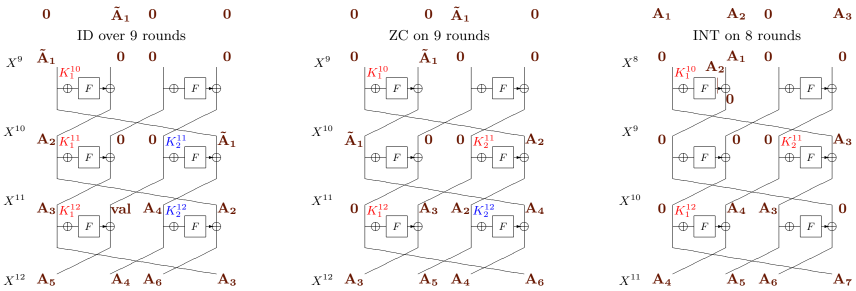


Fig. 3: Key-recovery in the ID and ZC context and the first partial sum in the INT context. The key-words in red are sieving key-words, the blue ones are guessed key-words.

[34] for a complete description of the S-boxes). Finally, the resulting nibbles are permuted as shown on Figure 4.

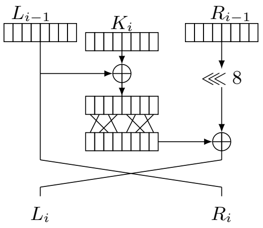


Fig. 4: A round of LBlock.

The key-schedule takes as input a master key K seen as a key register denoted at bit level as $K = K_{79}K_{78} \cdots K_0$ and outputs round-subkeys k_i of 32 bits. It repeats the following steps for $i = 1$ to 31 knowing that k_1 is initialized with the 32 leftmost bits of the key register K:

1. $K \lll 29$
2. $[K_{79}K_{78}K_{77}K_{76}] = S_9[K_{79}K_{78}K_{77}K_{76}]$ where S_9 is the tenth S-box.
3. $[K_{75}K_{74}K_{73}K_{72}] = S_8[K_{75}K_{74}K_{73}K_{72}]$ where S_8 is the ninth S-box.
4. $[K_{50}K_{49}K_{48}K_{47}] = [K_{50}K_{49}K_{48}K_{47}] \oplus [i]_2$
5. k_{i+1} is selected as the leftmost 32 bits of the key register K.

For this cipher which can be seen, as described in [27], as a Type-II GFN, ID and ZC distinguishers can be derived from each other [3] and can be applied on 14 rounds [34]. INT distinguishers reach 15 rounds [34].

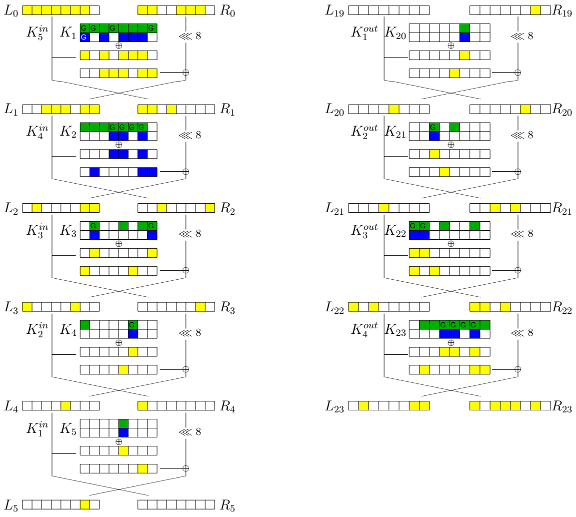
Derived from these distinguishers, ID and ZC attacks on 22 rounds [19,14,27] as well as ID attacks on 23 rounds [14,15] have been performed. An INT attack on 22 rounds has also been presented [25]. For the illustration purpose of this section, we assume that the round-keys are independent. As the attacks of [19,14,27,25] depend on the key-schedule more details on these attacks will only be presented in Section 6. In this example represented in Figure 5 and in Figure 6, we observe a strong relation between the key-words involved in the different attacks, when the distinguishers are, what we will call, *strongly related* (the same key-words are involved in a key-recovery attack on $1+s+1$ rounds). For this illustration we use as reference the ID attack of [14] derived from the ID distinguisher: $[((0,0,0,0,0,0,0,0),(0,0,0,0,*,0,0,0)) \not\rightarrow ((0,0,0,0,0,*,0,0),(0,0,0,0,0,0,0,0))]$.

For the ZC distinguisher we use: $[((0,0,0,0,0,0,*,0),(0,0,0,0,0,0,0,0)) \not\rightarrow ((0,0,0,0,0,0,0,0),(0,0,0,0,0,0,*,0))]$.

The keys $K_5[3]$ and $K_{20}[2]$ are involved in both attacks, meaning that the chosen ID and ZC distinguishers are strongly related. We also observe that in that case, as for the classical type-II GFN given in Figure 3, the key-words involved in the ZC attack corresponds to a subset of the key-words involved in the ID attack. We also compare this attack, with an INT attack derived from the following INT distinguisher on 15 rounds:

$[((A,A,A,A,A,A,C,A),(A,A,A,A,A,A,A,A)) \rightarrow ((*,*,*,*,*,*,*,*),(*,*,*,*,*,*,A\oplus A,*))]$.

Before providing in Section 5 a proof regarding the relations between the different involved key-words, we analyze in the next section, relations between the data complexities of these different attacks.

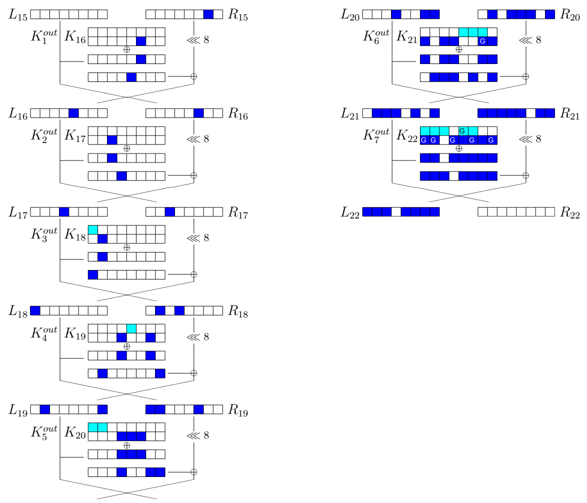


In blue (bottom): the key-words involved in the ZC attack
In green (top): the key-words involved in the ID attack
G: a guessed key-word (non-sieving key-word)
In yellow: the state-words involved in the ZC attack

Fig. 5: The key-recovery part of ID, ZC attacks on LBlock.

4.3 Data Complexity of a Key-Recovery Attack

ZC Attack. We denote by M the number of linear approximations which have correlation zero and which are evaluated in the attack. Improving the estimate of [9], it has been proven in [8] that the data complexity N^{ZC} of a distinct-known-



In dark blue (bottom): the key-words involved in the first partial sum
In light blue (top): the key-words involved in the second partial sum
G: a guessed key-word (non-sieving key-word)
In dark blue: the state-words involved in the first partial sum

Fig. 6: The key-recovery part of an INT attack on LBlock.

plaintext ZC key-recovery attack is

$$N^{ZC} \approx \frac{(2^n - 1)\left[z_{1-\alpha} + z_{1-\beta}\right]}{\sqrt{M/2} + z_{1-\alpha}} + 1, \tag{3}$$

where z_x is obtained from the inverse of the cumulative distribution function Φ of the normal distribution: $z_x = \Phi^{-1}(x)$, $1-\alpha$ corresponds to the success probability of the attack and $-\log_2(\beta)$ to the advantage of the attack. As the manipulation of the normal distribution makes the comparison between attacks difficult, in the remainder of this paper we use an approximation of it. In particular, we can show (see for instance [29]) that for $\beta < 2^{-5}$, $z_{1-\beta} = \Phi^{-1}(1-\beta) \approx \sqrt{-2\log(\beta)}$. While an approximation of $z_{1-\alpha}$ can also be used, in many ZC attacks, the success probability is fixed to $1 - \alpha = 1 - 2^{-2.7} \approx 0.85$ such that $z_{1-\alpha} \approx 1.02$. Using this approximation, the data complexity of a ZC key-recovery attack becomes

$$N^{ZC} \approx \frac{2^n \left[z_{1-\alpha} + \sqrt{2 \cdot \log(1/\beta)}\right]}{\sqrt{(M-1)/2} + z_{1-\alpha}} + 1. \tag{4}$$

ID Attack. For most of the attacks in the differential context, the data complexity of an ID key-recovery attack is computed as a function of the structure of the input and output differences but also of the different sieves performed during the attack. In an ID attack, if at least one key remains at the end of the key-recovery part we are certain that the correct key is among them. In statistical terms [4], this particularity means that the success probability $1-\alpha$ of the attack is equal

to 1. Given structures of size 2^t and given 2^m structures used in an ID attack, the data complexity is $N^{ID} = 2^{m+t}$.
In the ID cryptanalysis context [2,15], it is commonly assumed that the false alarm probability, denoted β in this paper is determined by $\beta = (1 - 2^{-q})^{2^{m+2t-1-p}} \approx e^{2^{m+2t-p-q-1}}$, where 2^{-p} corresponds to the product probabilities discarding plaintext-pairs and 2^q corresponds to the probability of discarding a key. Thus, we have

$$N^{ID} \approx 2^{p+q-t} \cdot 2 \cdot \log(1/\beta). \tag{5}$$

When the data complexity is smaller than the maximal size of a structure or if the ID attack can be done in the known plaintext model, we have $N^{ID} = \sqrt{2^{p+q-1}\log(1/\beta)}$. If we have M differentials involved in our ID distinguisher, we can show that $2^{p+q+n-t} = 2^{2n}/M$, where n is the size of the encrypted blocks. In the case where more than one structure is used, we have

$$N^{ID} \approx \frac{2^n}{M} \cdot 2 \cdot \log(1/\beta). \tag{6}$$

From a direct comparison between (4) and (6) we deduce that:

Lemma 3. *Given two related ID and ZC distinguishers, involving M differentials or linear approximations, if the success probability of the ZC attack is 50% ($z_{1-\alpha} = 0$), we have $N^{ID} \leq N^{ZC}$ as long as $\log(1/\beta) \leq M$. This relation remains true, for larger success probabilities since the data complexity of the ZC attack will, in that case, be larger too.*

INT Attack. Given V_0^{INT} as defined in Cor. 1, the data complexity of an INT attack is proportional to the number v of active words in V_0^{INT}: $N^{INT} = \mathcal{O}(2^{c \cdot v})$. For this attack, false alarm occurs when the sum on a word is randomly equal to 0. As this event occurs with probability 2^{-c}, we can derive in a time/memory trade-off way, the success of an INT attack with $\beta = (1 - 2^{-c})^{2^{c|\mathcal{K}|}}$ where $|\mathcal{K}|$ is the number of key-words involved in the key-recovery part of the attack. The overall data complexity of an INT attack is:

$$N^{INT} \approx 2^{c \cdot v} \cdot 2 \cdot \log(1/\beta). \tag{7}$$

Remark 4. For many Feistel-like ciphers, the number M of differences involved in an ID distinguisher of type-1, is $M = 2^{2 \cdot c}$ (see examples in [24]) and the weight v of V_0^{INT} is $b - 1$. In this case, a comparison between (7) and (5) gives $N^{ID1} = N^{INT}/2^c$.
This relation stays about to be true in the case of an ID distinguisher of type-2 (in this case, $M = (2^c - 1)^2$). When comparing an ID distinguisher of type-3 with $M = 2^c$ to an INT distinguisher with $v = b - 1$ we obtain that $N^{INT} = N^{ID3}$.

Remark 5. Assuming as in many attacks that $HW(V_0^{ZC1}) = HW(W_0^{ZC1})$ and $HW(V_0^{INT}) = b - HW(V_0^{ZC1})$ and given a success probability $1 - \alpha$ of 50% in the ZC context, we obtain that $N^{INT}/N^{ZC1} \approx \sqrt{2\log(1/\beta)}$. Thus, the data complexity of an INT attack is greater than the one of a ZC1 attack. For larger success probability, this is no more true and in general $N^{INT} \leq N^{ZC}$.

After the analysis of the data complexities, some questions arise. In particular we could wonder what is the benefit of a ZC attack if its data complexity is larger than the one of an INT or ID attack. In the next section, using a matrix representation of the round function, we study the relation between the key-words involved in the different attacks and show that for matrix-method-derived related distinguishers the number of key-words involved in a ZC key-recovery attack is smaller than the number of key-words involved in a ID attack on the same number of rounds.

5 Key-Words Involved in a Key-Recovery Attack

From a matrix representation of Type-II GFN ciphers, we analyze in this section the key-recovery part of these attacks. This analysis is done for *related* and *strongly related* matrix-method-derived ID, ZC and INT distinguishers.
We say that the *ZC and INT distinguishers are strongly related* if the same key-words are involved in a key-recovery attack over one round of partial decryption. In this section, strongly related INT and ZC distinguishers do not necessary apply on the same number of rounds.
We say that the *ZC and ID distinguishers are strongly related* if the same key-words are involved in a key-recovery attack over one round of partial encryption and one round of partial decryption.

For instance, for the Nyberg constructions, the matrix-method derived ID and ZC distinguishers are related but are not strongly related. This can be proven by observing that for this non-alternating cipher, $\mathcal{P} \cdot \mathcal{Q} \neq \mathcal{Q} \cdot \mathcal{P}$ for $\mathcal{Q}$ defined as in Remark 1. But, for this cipher, the distinguishers are related in the sense of Theorem 1.

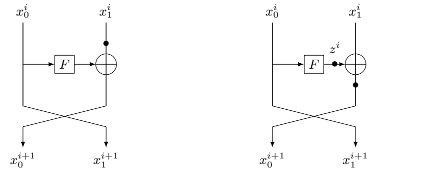
5.1 A Matrix Representation for Analyzing Key-Recovery Attacks

In the same way, we can derive ID, ZC and INT distinguishers of a Feistel cipher using a matrix representation of the round function, we show in this section how we can find the key-words involved in the attack using a matrix method. In this section, the vectors V_0 and W_0 represent respectively the input and output of the distinguishers. The vectors V_i and W_j represent the state when $i \leq r_{\text{in}}$ rounds are added at the beginning of a distinguisher and $j \leq r_{\text{out}}$ rounds are added at the end of this distinguisher. The knowledge of the value of the vectors $\{V_i\}_{i<r_{\text{in}}}$ and $\{W_j\}_{j<r_{\text{out}}}$ will be necessary to compute the number of attacked rounds in the ID, ZC and INT contexts. The operations on these vectors correspond to those given in Section 2.

In that case, the vectors $V_i, W_j \in \{(0, \tilde{A}_a, A_a, val)\}^b$ where $A_i, \tilde{A}_i$ denote a difference-word or a state-word which has to be computed and 0 denote words with no difference or a state-word with value unnecessary for the attack. We use the following arithmetic similar to the one given in Table 2: $0 \oplus A_a = A_a$, $A_a \oplus A_{a'} = A_{a''}$, $F(0) = 0$ and $F(A_a) = A_{a'}$. The description of *val* is given later in this section and is only used in the ID context. As explained in Section 3, the notation $\tilde{A_a}$ is necessary for the type-2 and type-3 ID and ZC distinguishers, which hold when the input and output differences (masks) should be respectively strictly different or strictly equal.

In an INT attack, using the partial sum technique introduced in [26] and described in Figure 7, the key-recovery attack could be divided in two searching paths reducing the overall time complexity. The computational time required by the second path is marginal as it usually allows to gain at least one round for free. More sophisticated attacks such as meet in the middle attacks could be mount using INT distinguishers (see for example [23]). For comparison purposes, we do not integrate those tricks in our study of the key-words involved in the attack. In all the cases, considering only the primary branch of the partial sum technique, we show that the number of involved key-words in the INT context is always lower than the number of key-words involved in the ZC context and in the ID context considering only the r_{out} direction.

Moreover, as all the INT distinguishers obtained using the matrix method are in fact higher order integral distinguishers with $b-1$ or $b-2$ active words in input, the computational cost to pay when trying to add one round at the beginning of this distinguisher is really high (except when some particular tricks such as the ones described in [18] could be considered which is not always the case) and to stay as generic as possible, we only consider a key-recovery attack on r_{out} rounds at the end of the INT distinguisher.



Test if $\sum x_1^i \overset{?}{=} 0$ Test if $\sum x_0^{i+1} \overset{?}{=} \sum z^i$

Fig. 7: On the left, the test to perform without the partial sum. On the right, the test to perform with the partial sum divides the computations into two steps.

Illustrated by the example provided in Figure 3, the sieving and guessed key-words can be found as follows.

Lemma 4. *In the ZC context,*

- *Given the vector V_{i-1}. Let $V_A = \mathcal{P}^{-1} \cdot V_{i-1}$.*
  - *The key-word $K_i[p]$ is a sieving key-word if $V_A[2p+1] \neq 0$ and $V_A[2p] = 0$.*
  - *The key-word $K_i[p]$ is a guessed key-word if $V_A[2p+1] \neq 0$ and $V_A[2p] \neq 0$.*
  - *The vector V_i is computed as $V_i = \mathcal{F}^T \cdot V_A = \mathcal{F}^T \cdot \mathcal{P}^{-1} \cdot V_{i-1}$.*
  - *The number of active words corresponds to the weight of V_i.*
- *Given the vector W_{j-1}.*
  - *The key-word $K_j[p]$ is a sieving key-word if $W_{j-1}[2p+1] \neq 0$ and $W_{j-1}[2p] = 0$.*
  - *The key-word $K_j[p]$ is a guessed key-word if $W_{j-1}[2p+1] \neq 0$ and $W_{j-1}[2p] \neq 0$.*

  - *The vector W_j is computed as $W_j = \mathcal{M} \cdot W_{j-1} = \mathcal{P} \cdot \mathcal{F}^T \cdot W_{j-1}$.*
  - *The number of active words corresponds to the weight of W_j.*

Usually in the ID context the matrix method allows us to only study the propagation of the differences. However, in the partial encryption/decryption process there are cases where even if the difference is equal to 0, the knowledge of the value is necessary to complete the key-recovery. We introduce the notation *val*, to represent such a word with difference 0. The weight of a vector V corresponds to the number of non-zero differences.

Lemma 5. *In the ID context,*

- *Given the vector V_{i-1} and $V_A = \mathcal{P}^{-1} \cdot V_{i-1}$.*
  - *The key-word $K_i[p]$ is a sieving key-word if $V_A[2p] \neq 0$ and $V_A[2p+1] = 0$ or val.*
  - *The key-word $K_i[p]$ is a guessed key-word if $V_A[2p+1] \neq 0$.*
  - *In the case where $V_A[2p+1] \neq 0$ and $V_A[2p] = 0$, update $V_A[2p]$ to val.*
  - *The vector V_i is computed as $V_i = \mathcal{F} \cdot V_A$.*
  - *The number of active words corresponds to the weight of $\mathcal{F} \cdot V_A$.*
- *Given the vector W_{j-1}.*
  - *The key-word $K_j[p]$ is a sieving key-word if $W_{j-1}[2p] \neq 0$ and $W_{j-1}[2p+1] = 0$ or val.*
  - *The key-word $K_j[p]$ is a guessed key-word if $W_{j-1}[2p+1] \neq 0$.*
  - *In the case where $W_{j-1}[2p+1] \neq 0$ and $W_{j-1}[2p] = 0$, update $W_{j-1}[2p]$ to val.*
  - *The vector W_j is computed as $W_j = \mathcal{P} \cdot \mathcal{F} \cdot W_{j-1}$.*
  - *The number of active words corresponds to the weight of $\mathcal{P} \cdot \mathcal{F} \cdot W_{j-1}$.*

Proof. The proof is similar to the one in the ZC case but the reasoning is done with differentials instead of with the state-values. The difficult case corresponds to the third point. Indeed, when $V_A[2p+1] \neq 0$ and $V_A[2p] = 0$, it means that the value $V_A[2p]$ needs to be known even if the difference is 0 and thus for that purpose, we decide to use the *val* notation.

Lemma 6. *In the INT context, only the output direction on r_{out} rounds is considered. Given the distinguisher (V_0^{INT}, Y_0^{INT}) we define W_0^{INT} such that $W_0^{INT}[p] = 0$ if $Y_0^{INT}[p] = U_u$ and $W_0^{INT}[p] = A_a$ if $Y_0^{INT}[p] = A_a$ or $Y_0^{INT}[p] = \oplus_i A_i$.*

- *The key recovery is similar to the one in the ZC context, starting from the vector W_0^{INT}.*
- *The partial sum technique, can be applied from writing $W_1^{INT} = W_{1,1}^{INT} + W_{1,2}^{INT}$ and splitting the key recovery for the two vectors $W_{1,1}^{INT}$ and $W_{1,2}^{INT}$.*

5.2 Relations between the Key-Words of the Different Attacks

From the rules expressed in the previous section, we can derive the following relations between the key-words involved in the different attacks for related and strongly related matrix-method-derived distinguishers.

Lemma 7. *For a Type-II GFN, given a matrix-method-derived ZC distinguisher and its strongly related ID distinguisher, the key-words involved in a ZC key-recovery attack is a subset of the key-words involved in an ID key-recovery attack. The number of sieving key-words is identical.*

Proof. In this proof, we define the matrix $\mathcal{Q}$ such that $\mathcal{F}^T = \mathcal{Q} \cdot \mathcal{F} \cdot \mathcal{Q}^{-1}$ (see Remark 1). According to the hypothesis made in the beginning of this section in order to have the same key-words involved in an attack on 1+1 rounds, the vectors V_0^{ID} and V_0^{ZC} are such that $V_0^{ZC} = \mathcal{Q} \cdot V_0^{ID}$. We denote by $\tilde{V}_i^{ZC} = \mathcal{Q} \cdot V_i^{ZC}$. In this case the first part of Lemma 4 can be rewritten as:
• The key-word $K_i[p]$ is a sieving key-word if $\tilde{V}_A^{ZC}[2p] \neq 0$ and $\tilde{V}_A^{ZC}[2p+1] = 0$.
• The key-word $K_i[p]$ is a guessed key-word if $\tilde{V}_A^{ZC}[2p] \neq 0$ and $\tilde{V}_A^{ZC}[2p+1] \neq 0$.
• The vector V_i is computed as $V_i = \mathcal{F}^T \cdot V_A = \mathcal{Q} \cdot \mathcal{F} \cdot \tilde{V}_A^{ZC}$.
From the relation $V_0^{ZC} = \mathcal{Q} \cdot V_0^{ID}$ and the previous observations, we deduce that the number of sieving key-words among the key K_1^{in} is the same in both ZC and ID attacks. The same observation can be done on W_0^{ZC} and W_0^{ID}.

Note that if $V_i^{ZC} = \mathcal{Q} \cdot V_i^{ID}$ then the number of guessed key-words in the ID context is larger than in the ZC context since the rule to determine if a key is a guessed key in the ZC context are more restrictive than the one in the ID context. Remain to analyze the general relation between V_i^{ZC} and V_i^{ID}.
Assume that $V_{i-1}^{ZC} = \mathcal{Q} \cdot V_{i-1}^{ID}$ (this is true for $i = 1$), meaning that $\tilde{V}_A^{ZC} = V_A^{ID}$ before update of this one. According to the third bullet of Lemma 5, after update of V_A^{ID}, we have $\tilde{V}_A^{ZC} = V_A^{ID} + V'$ where V' is a vector of b elements all equal to 0 or val. We conclude from the fact that operations on the matrix are additive.

Lemma 8. *For Type-II GFNs fulfilling one of the conditions given in Theorem 1 and for related ZC and ID distinguishers, the key-words involved in the encryption (resp. decryption) side of the ZC attack correspond to a subset of the key-words involved in the encryption (resp. decryption) side of the ID attack.*

Proof. The proof is similar to the one of Lemma 7 using the tools of the proof of Theorem 1.

The proof of the following results can be found in the extended version [5].

Lemma 9. *Given a ZC distinguisher and a strongly related INT distinguisher, without the partial sum technique, the key-words involved in the two attacks on r_{out} rounds added at the end of the distinguisher $\mathcal{K}_{r_{\text{out}}}^{ZC}$ and $K_{r_{\text{out}}}^{INT}$ are, modulo the round index, the same. For an alternating Type-II GFN, when considering the dominant part of the partial sum PS technique , we have: $|\mathcal{K}_{r_{\text{out}}+1}^{INT_{PS}}| = |\mathcal{K}_{r_{\text{out}}}^{INT}| + 1$.*

Lemma 10. *Given a ZC distinguisher, for an alternating Type-II GFN, denoting by $|\mathcal{K}_i^{ZC_{in}}|$ (resp. $|\mathcal{K}_j^{ZC_{out}}|$) the number of key-words involved in the i first rounds* *(resp. in the j last rounds) of the attack we have* $|\mathcal{K}^{ZC_{out}}_{r_{\text{out}}}| \geq |\mathcal{K}^{ZC_{in}}_{i}| + |\mathcal{K}^{ZC_{out}}_{j}|$ *with* $i + j = r_{\text{out}}$. *If* $HW(W_0^{INT}) = HW(W_0^{ZC}) = 1$, *for* $r_{\text{out}} \geq 3$, *we have* $|\mathcal{K}^{INT_{PS}}_{r_{\text{out}}+1}| \geq |\mathcal{K}^{ZC}_{i}| + |\mathcal{K}^{ZC}_{j}|$.

6 A ZC Attack on LBlock

In Section 4 we show a relation between the key-words involved in a key-recovery attack on LBlock. In this section we present a ZC attack on 23 rounds of this cipher when taking into consideration the key-schedule of this cipher.

In the INT attack of [25], 69 key-bits are involved and among them, 55 are sieving key-bits. The ZC attack of [27] requires to guess $2^{4 \cdot 14} = 2^{56}$ keys. Complexities of the best attacks on LBlock are resumed in Table 1.
In the analysis provided in [14], using a different ID distinguisher leads to only guess 71 key-bits in the attack on 22 rounds. A similar analysis shows that only 2 extra bits need to be guessed in the attack on 23 rounds. As done in [14] and [25] in the ID and INT context, from an analysis of the key-schedule we can derive a ZC attack on 23 rounds of LBlock.

The description of the attack is given in the extended version of this paper [5]. In Table 5, we provide the complexity for a number M of linear approximations between 2^4 and 2^8.The results show that even if the key-dependency is stronger in the ID context we can also take advantage of this key-dependency in a ZC attack to obtain an attack with a better time complexity. Due to the use of tricks to reduce the time complexity, the memory complexity of the ID attack of [14] is larger than the one of the ZC attack.

ID on 23 rounds from [14]

Data	Time	Memory
2^{57}	$2^{78.58}$	2^{72}
2^{59}	$2^{75.36}$	2^{74}
2^{61}	$2^{76.48}$	2^{76}
2^{63}	$2^{78.48}$	2^{78}

ZC on 23 rounds

Data	Time	Memory	a	M
$2^{61.90}$	$2^{76.78}$	2^{60}	5	2^8
$2^{62.16}$	$2^{76.47}$	2^{60}	7	2^8
$2^{62.61}$	$2^{75.95}$	2^{59}	7	2^7
$2^{63.87}$	$2^{76.51}$	$\mathbf{2^{56}}$	7	2^4
$2^{63.87}$	$\mathbf{2^{73.94}}$	2^{60}	7	2^4

Table 5: Data/Time trade-offs: comparison of the complexities of the ID attack of [14] and of the new ZC attack. M is the number of linear masks involved in the attack. a denotes the advantage of the attack, meaning that $\beta = 2^{-a}$.

7 Conclusion

In this paper, we show how we can use the matrix method to analyse the key-recovery of Feistel-like ciphers. Based on a matrix representation of the round function we determine the key-words involved in ID, ZC or INT key-recovery attacks. In particular we illustate, that in many cases, when the matrix-method-derived distinguishers are related the number of involved key-words is smaller in the ZC and INT context than in the ID context. Nevertheless, our analysis also show that when the same number of differential and linear approximations are used, the data complexity of a ZC attack is larger than the one of a related ID attack.

While most of the results of this paper are dedicated to Type-II GFN, similar results can be obtained for other ciphers. For example, ID and ZC attacks behave similarly against the Feistel-like ciphers, HIGHT [16,32] and SIMON [30] (see Table 1) which contain modular additions, AND and rotation operations. Whereas the INT distinguisher is less efficient for HIGHT than ZC and ID distinguishers, it is more efficient, at bit level, in the case of SIMON.

Acknowledgment. The authors would like to thank the anonymous referees for their helpful comments. We also wish to thank the Aalto Science Institute (AScI) for funding the research visit of Marine Minier at Aalto University.

References

1. T. P. Berger, M. Minier, and G. Thomas. Extended Generalized Feistel Networks Using Matrix Representation. In *SAC'13*, LNCS 8282, pages 289–305, 2013.
2. E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In *EUROCRYPT'99*, LNCS 1592, pages 12–23, 1999.
3. C. Blondeau, A. Bogdanov, and M. Wang. On the (In)Equivalence of Impossible Differential and Zero-Correlation Distinguishers for Feistel- and Skipjack-Type Ciphers. In *ACNS'14*, LNCS 8479, pages 271–288, 2014.
4. C. Blondeau, B. Gérard, and J.-P. Tillich. Accurate estimates of the data complexity and success probability for various cryptanalyses. *Des. Codes Cryptography*, 59(1-3):3–34, 2011.
5. C. Blondeau and M. Minier. Relations between Impossible, Integral and Zero-correlation Key-Recovery Attacks (extended version). Cryptology ePrint Archive, Report 2015/?, 2015. `http://eprint.iacr.org/`.
6. C. Blondeau and K. Nyberg. New Links between Differential and Linear Cryptanalysis. In *EUROCRYPT'13*, LNCS 7881, pages 388–404, 2013.
7. C. Blondeau and K. Nyberg. Links Between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities. In *EUROCRYPT'14*, LNCS 8441, 2014.
8. A. Bogdanov, H. Geng, M. Wang, L. Wen, and B. Collard. Zero-Correlation Linear Cryptanalysis with FFT and Improved Attacks on ISO Standards Camellia and CLEFIA. In *SAC'13*, LNCS 8282, 2014.
9. A. Bogdanov, G. Leander, K. Nyberg, and M. Wang. Integral and Multidimensional Linear Distinguishers with Correlation Zero. In *ASIACRYPT'12*, LNCS 7658, pages 244–261, 2012.
10. A. Bogdanov and V. Rijmen. Zero-Correlation Linear Cryptanalysis of Block Ciphers. *IACR Cryptology ePrint Archive*, 2011:123, 2011.
11. A. Bogdanov and M. Wang. Zero Correlation Linear Cryptanalysis with Reduced Data Complexity. In *FSE'12*, LNCS 7549, pages 29–48, 2012.
12. Andrey Bogdanov, Huizheng Geng, Meiqin Wang, Long Wen, and Baudoin Collard. Zero-Correlation Linear Cryptanalysis with FFT and Improved Attacks on ISO Standards Camellia and CLEFIA. In *Selected Areas in Cryptography - SAC 2013*, LNCS 8282, pages 306–323, 2013.
13. C. Bouillaguet, O. Dunkelman, P.-A. Fouque, and G. Leurent. New insights on impossible differential cryptanalysis. In *Selected Areas in Cryptography - SAC 2011*, LNCS 7118, pages 243–259, 2011.

14. C. Boura, M. Minier, M. Naya-Plasencia, and V. Suder. Improved Impossible Differential Attacks against Round-Reduced LBlock. Cryptology ePrint Archive, Report 2014/279, 2014. `http://eprint.iacr.org/`.
15. C. Boura, M. Naya-Plasencia, and V. Suder. Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In *ASIACRYPT 2014*, LNCS 8873, pages 179–199, 2014.
16. J. Chen, M. Wang, and B. Preneel. Impossible Differential Cryptanalysis of the Lightweight Block Ciphers TEA, XTEA and HIGHT. In *AFRICACRYPT'12*, LNCS 7374, pages 117–137, 2012.
17. J. Daemen, L. R. Knudsen, and V. Rijmen. The Block Cipher Square. In *FSE'97*, LNCS 1267, pages 149–165, 1997.
18. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting. Improved Cryptanalysis of Rijndael. In *FSE'00*, LNCS 1978, pages 213–230, 2000.
19. F. Karakoç, H. Demirci, and A. Harmanci. Impossible Differential Cryptanalysis of Reduced-round LBlock. In *WISTP'12*, LNCS 7322, pages 179–188. 2012.
20. J. Kim, S. Hong, J. Sung, C. Lee, and S. Lee. Impossible Differential Cryptanalysis for Block Cipher Structures. In *INDOCRYPT'03*, LNCS 2904, pages 82–96, 2003.
21. L. Knudsen. DEAL-a 128-bit block cipher. *complexity*, 258(2), 1998.
22. L. R. Knudsen and D. Wagner. Integral Cryptanalysis. In *FSE'02*, LNCS 2365, pages 112–127, 2002.
23. Jiqiang Lu, Yongzhuang Wei, Jongsung Kim, and Enes Pasalic. The higher-order meet-in-the-middle attack and its application to the Camellia block cipher. *Theor. Comput. Sci.*, 527:102–122, 2014.
24. Y. Luo, X. Lai, Z. Wu, and G. Gong. A Unified Method for Finding Impossible Differentials of Block Cipher Structures. *Inf. Sci.*, 263:211–220, 2014.
25. Y. Sasaki and L. Wang. Comprehensive Study of Integral Analysis on 22-Round LBlock. In *ICISC'12*, LNCS 7839, pages 156–169, 2012.
26. Y. Sasaki and L. Wang. Meet-in-the-Middle Technique for Integral Attacks against Feistel ciphers. In *SAC'12*, LNCS 7707, pages 234–251, 2012.
27. H. Soleimany and K. Nyberg. Zero-Correlation Linear Cryptanalysis of Reduced-round LBlock. *Des. Codes Cryptography*, 73(2):683–698, 2014.
28. T. Suzaki and K. Minematsu. Improving the Generalized Feistel. In *FSE'10*, LNCS 6147, pages 19–39, 2010.
29. P. M. Voutier. A New Approximation to the Normal Distribution Quantile Function. *ArXiv e-prints*, February 2010.
30. Q. Wang, Z. Liu, K. Varici, Y. Sasaki, V. Rijmen, and Y. Todo. Cryptanalysis of Reduced-round SIMON32 and SIMON48. Cryptology ePrint Archive, Report 2014/761, 2014. `http://eprint.iacr.org/`.
31. L. Wen, M. Wang, and A. Bogdanov. Multidimensional Zero-Correlation Linear Cryptanalysis of E2. In *AFRICACRYPT'14*, LNCS 8469, pages 147–164, 2014.
32. L. Wen, M. Wang, A. Bogdanov, and H. Chena. Multidimensional Zero-Correlation Attacks on Lightweight Block Cipher HIGHT: Improved Cryptanalysis of an ISO Standard. *Information Processing Letters*, 114(6):322–330, 2014.
33. L. Wen, M. Wang, A. Bogdanov, and H. Chena. Multidimensional Zero-Correlation Attacks on Lightweight Block Cipher HIGHT: Improved Cryptanalysis of an ISO Standard. *Information Processing Letters*, 114(6):322–330, 2014.
34. W. Wu and L. Zhang. LBlock: A Lightweight Block Cipher. In *ACNS'11*, LNCS 6715, pages 327–344, 2011.
35. W. Zhang, B. Su, W. Wu, D. Feng, and C. Wu. Extending Higher-Order Integral: An Efficient Unified Algorithm of Constructing Integral Distinguishers for Block Ciphers. In *ACNS'12*, LNCS 7341, pages 117–134, 2012.