Differential and Linear Attacks on the Full WIDEA-n Block Ciphers (under Weak Keys)

Jorge Nakahara Jr.

Université Libre de Bruxelles (ULB), Dept. of Computer Science, Belgium
jorge.nakahara@ulb.ac.be

Abstract. We report on differential and linear analysis of the full 8.5-round WIDEA-n ciphers for $n \in \{4, 8\}$, under weak-key assumptions. The novelty in our attacks include the use of differential and linear relation patterns that allow to bypass the diffusion provided by MDS codes altogether. Therefore, we can attack only a single IDEA instance out of n copies, effectively using a narrow trail for the propagation of differences and masks across WIDEA-n. In fact, the higher the value of n, the better the attacks become. Our analyses apply both to particular MDS matrices, such as the one used in AES, as well as general MDS matrices. Our attacks exploit fixed points of MDS matrices. We also observed a curious interaction between certain differential/linear patterns and the coefficients of MDS matrices for non-trivial fixed points. This interaction may serve as an instructive design criterion for block cipher designs such as WIDEA-n. The authors of WIDEA-n suggested a compression function construction using WIDEA-8 in Davies-Meyer mode. In this setting, the weaknesses identified in this paper can lead to free-start collisions and even actual collisions depending on the output transformation of the hash function.

Keywords: wide-block cipher, cryptanalysis, WIDEA-n, free-start collisions.

1 Introduction

In [6], Junod and Macchetti presented a Wide-block version of IDEA cipher [7] called WIDEA-n, combining n instances of the 8.5-round IDEA cipher joined by an $n \times n$ matrix derived from a Maximum Distance Separable (MDS) code. Their approach not only led to improved performance, due to the bit-slicing technique allowing parallel instances of IDEA to be evaluated altogether, but also showed wide-block cipher variants operating on bit strings whose size is a multiple of 64 bits (the original block size of IDEA).

The contributions of this paper include

- the first differential and linear distinguishers of the full 8.5-round WIDEA-n, for $n \in \{4, 8\}$, under weak-key assumptions. Actually, our attacks would hold even if n was allowed to be much larger than 8. **Weak-key assumptions mean that user keys in our attacks lead, through the key schedule,**

J. Pieprzyk, A.-R. Sadeghi, and M. Manulis (Eds.): CANS 2012, LNCS 7712, pp. 56–71, 2012.

to round subkeys which are either 0 **or** 1 **in specific positions along differential or linear trails** [2]. In this way, both the user key and some subkeys are weak. For this reason, we refer equally to weak-key and weak-subkeys assumptions.
- differential and linear distinguishers that apply both in a secret-key and in a hash/compression function settings, assuming WIDEA-n becomes the compression function in Davies-Meyer (DM) mode [8].
- we show how to bypass the MDS matrices using trivial and non-trivial fixed points. This procedure is possible due to carefully chosen differences and linear masks that lead to trivial differences and masks at the input to the MA/MAD-boxes. Therefore, avoiding these diffusion components in every round of WIDEA-n, for any n, means that we restrict the propagation of differences and masks to one single IDEA instance, out of n. The larger n is, the better the attacks become. Previous analyses using fixed points include [1,3,11], but the latter worked on fixed points for an entire block. As far as we aware of, this paper presents the first use of fixed points (in differential and linear settings) that bypass MDS codes in block cipher designs such as WIDEA-n.
- we show how and why some matrices from MDS codes can rather help the cryptanalysis of WIDEA-n, depending on where they are placed in a block cipher design such as WIDEA-n, and even depending on the exclusive-or sum of its coefficients. See Sect. 4.2.

This paper is organized as follows: Sect. 2 briefly details WIDEA-n; Sect. 3 describes the key schedule algorithms of WIDEA-n; Sect. 4 describes differential attacks on WIDEA-n; Sect. 5 details linear attacks on WIDEA-n. Sect. 6 describes attacks on WIDEA-n used in compression function constructions. Sect. 7 discusses weak keys. We conclude in Sect. 8.

2 The WIDEA-n Block Ciphers

WIDEA-n, $n \in \{4, 8\}$, stands for two Wide-block variants of the IDEA cipher [7] operating on $64n$-bit blocks. The rationale is to join n instances of the IDEA cipher using an $n \times n$ matrix derived from an MDS code, placed inside the MA-box (Multiplication-Addition) in each round of each IDEA instance (see Fig. 1). Thus, the original MA-box in IDEA becomes a so called MAD-box (Multiply-Add-Diffuse) [6] in WIDEA-n. The key size is $128n$ bits and WIDEA-n iterates 8.5 rounds. Fig. 1 depicts one round of WIDEA-4 cipher. For WIDEA-8, there are eight copies of IDEA side-by-side, connected by an 8×8 MDS matrix (Fig. 2). The MDS matrix in WIDEA-4 is the one used in the AES cipher [4]:

$$\begin{pmatrix} 2\,3\,1\,1 \\ 1\,2\,3\,1 \\ 1\,1\,2\,3 \\ 3\,1\,1\,2 \end{pmatrix}. \tag{1}$$

In WIDEA-4, the matrix (1) is multiplied on the right by a 4×1 vector $(a, b, c, d)^t$ containing part of the internal state of four IDEA instances, where t denotes vector transpose. This matrix product operation will be denoted $\mathrm{MDS}(a, b, c, d)^t$.

The MDS matrix in WIDEA-8 comes from the W cipher in the Whirlpool hash function [5], and the semantics for matrix multiplication is the same as explained for (1):

$$\begin{pmatrix} 1&1&4&1&8&5&2&9\\ 9&1&1&4&1&8&5&2\\ 2&9&1&1&4&1&8&5\\ 5&2&9&1&1&4&1&8\\ 8&5&2&9&1&1&4&1\\ 1&8&5&2&9&1&1&4\\ 4&1&8&5&2&9&1&1\\ 1&4&1&8&5&2&9&1 \end{pmatrix}. \tag{2}$$

In both (1) and (2), matrix coefficients and operations are performed over $\mathrm{GF}(2^{16}) = \mathrm{GF}(2)[\mathrm{x}]/(p(x))$, where $p(x) = x^{16} + x^5 + x^3 + x^2 + 1$ is an irreducible polynomial over GF(2).

We briefly describe an MA-box (Fig. 1): let the input to the i-th MA-box of the i-th IDEA instance, for $1 \le i \le n$, be denoted (p_i, q_i), its output be (r_i, s_i) and $(Z_{5,i-1}, Z_{6,i-1})$ be the round subkeys in it. We ignore the superscripts since they are irrelevant in this setting. The three group operations in IDEA and WIDEA-n are: $\oplus$ denote bitwise exclusive-or, $\boxplus$ denote addition modulo 2^{16} and $\odot$ denote multiplication in $\mathrm{GF}(2^{16}+1)$, with $0 \equiv 2^{16}$. Then, $s_i = (p_i \odot Z_{5,i-1} \boxplus q_i) \odot Z_{6,i-1}$ and $r_i = p_i \odot Z_{5,i-1} \boxplus s_i$, where $\odot$ has higher precedence than $\boxplus$.

Now, for the MAD-box (Fig. 2): the MDS matrix in WIDEA-n is placed inside the original MA-box of IDEA after $p_i \odot Z_{5,i-1} \boxplus q_i$. This way, a single MAD-box of WIDEA-n has output (r', s') such that

- $s' = MDS(p_i \odot Z_{5,i-1} \boxplus q_i) \odot Z_{6,i-1}$, and $r' = p_i \odot Z_{5,i-1} \boxplus s'$, where $1 \le i \le n$ and $MDS(p_i \odot Z_{5,i-1} \boxplus q_i)$ stands for the multiplication of an MDS matrix (1) or (2) by an $n \times 1$ vector containing the n values $p_i \odot Z_{5,i-1} \boxplus q_i$, for $1 \le i \le n$.
- every single output tuple (r', s') depends on all $(p_i, q_i, Z_{5,i-1})$, for $1 \le i \le n$, but not on $Z_{6,i-1}$.
- the placement of the MDS matrix also implies that its dependence on $(p_i, q_i, Z_{5,i-1})$, for $1 \le i \le n$, is spread to both (r', s') in all n IDEA instances in every round.
- the MDS matrix is preceded by $\boxplus$ and followed by $\odot$, while inside the matrix computation there is a combination of xor and multiplication in $\mathrm{GF}(2^{16})$. Except for the repeated xor in the matrix product, no other operation is repeated twice in a row in the MAD-box.
- since the half-round containing the MAD-box is an involution there is no need to compute the inverse MDS matrix for decryption.

To allow a compact representation for analysis, and taking into account the 3-dimensional structure of WIDEA-4, we denote the internal state of WIDEA-4 by the 4×4 matrix:

$$\begin{pmatrix} a_{12} & a_{13} & a_{14} & a_{15} \\ a_8 & a_9 & a_{10} & a_{11} \\ a_4 & a_5 & a_6 & a_7 \\ a_0 & a_1 & a_2 & a_3 \end{pmatrix}, \tag{3}$$

where each a_i, for $0 \leq i \leq 15$, is a 16-bit word and the numbering follows from Fig. 1, where $(a_{4(j-1)}, a_{4(j-1)+1}, a_{4(j-1)+2}, a_{4(j-1)+3})$ represent the state of the j-th IDEA instance for $1 \leq j \leq 4$. The MAD-boxes of the four IDEA instances are connected to each other via the 4×4 MDS matrix (1). Analogously, we denote the internal state of WIDEA-8 by the 8×4 matrix:

$$\begin{pmatrix} a_{28} & a_{29} & a_{30} & a_{31} \\ a_{24} & a_{25} & a_{26} & a_{27} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{16} & a_{17} & a_{18} & a_{19} \\ a_{12} & a_{13} & a_{14} & a_{15} \\ a_8 & a_9 & a_{10} & a_{11} \\ a_4 & a_5 & a_6 & a_7 \\ a_0 & a_1 & a_2 & a_3 \end{pmatrix}, \tag{4}$$

where each a_i, for $0 \leq i \leq 31$, is a 16-bit word. Word numbering follows Fig. 2 where $(a_{4(j-1)}, a_{4(j-1)+1}, a_{4(j-1)+2}, a_{4(j-1)+3})$ represents the state of the j-th IDEA instance, $1 \leq j \leq 8$. The MAD-boxes of the eight IDEA instances are connected by a single 8×8 MDS matrix (2) in every round.

3 The Key Schedule of WIDEA-n

Let Z_i, for $0 \leq i \leq 51$, denote the round subkeys used in 8.5-round WIDEA-n, $n \in \{4, 8\}$. The key schedule algorithm of WIDEA-4 is as follows [6]: due to the 4-way parallelism, each subkey has 64 bits. Thus, each subkey Z_i can be split into four slices $Z_{i,0}, \ldots, Z_{i,3}$ (see Fig. 1). Let K_i, for $0 \leq i \leq 7$, denote the eight 64-bit words representing the user key. The round subkeys are computed as follows:

- $Z_i = K_i$, for $0 \leq i \leq 7$.
- $Z_i = ((((Z_{i-1} \oplus Z_{i-8}) \overset{16}{\boxplus} Z_{i-5}) \overset{16}{\lll} 5) \lll 24) \oplus C_{i/8-1}$, for $8 \leq i \leq 51$, $i \equiv 0 \bmod 8$.
- $Z_i = ((((Z_{i-1} \oplus Z_{i-8}) \overset{16}{\boxplus} Z_{i-5}) \overset{16}{\lll} 5) \lll 24)$, for $8 \leq i \leq 51$, $i \not\equiv 0 \bmod 8$.

where operations superscripted with '16' indicate that the operation is actually carried out over 16-bit slices of Z_i. Othewise, the operation is carried out across 64-bit words, such as the bitwise left-rotation $\lll 24$. Following [6], $C_0, \ldots, C_5$ are constants inserted every eight rounds. This design using nonlinear feedback shift registers was inspired on the key schedule of MESH ciphers [9].

The key schedule algorithm of WIDEA-8 [6] follows an 8-way parallelism. Each 128-bit subkey Z_i can be split into eight slices $Z_{i,0}, \ldots, Z_{i,7}$ (see Fig. 2).

Let K_i, for $0 \leq i \leq 7$, denote the eight 128-bit words representing the user key. The round subkeys are computed exactly as for WIDEA-4, except that the subkeys and constants $C_{i/8-1}$ are 128 bits long.

4 Differential Cryptanalysis of WIDEA-n

For a differential analysis, we start with Table 2 (in the appendix) that lists exhaustively all one-round characteristics of IDEA [2] using wordwise difference $\delta = 8000_x$. The subscript x indicates hexadecimal value. This difference propagates across $\oplus$ and $\boxplus$ with certainty and for any subkey value because the only active difference is in the most significant bit position. The arrows indicate difference propagation across one-round or across an MA/MAD-box in the encryption direction, depending on the context. All these characteristics hold with probability 1 under weak-key assumptions. Thus, the main purpose of weak keys is that they cause weak subkeys in specific positions inside WIDEA-n which allow straightforward propagation of differences (and bit masks). The third column in Table 2 shows the difference propagation inside the MA-box and consequently, if the MA-box is differentially active or not. An MA/MAD-box is differentially active if its input difference is nonzero. It is passive, otherwise.

We choose characteristics based on two criteria: (i) minimize the number of weak-key assumptions per round, and (ii) choose iterative difference patterns. Under these two conditions, the best choices include the 3-round characteristic

$$(0, 0, \delta, \delta) \rightarrow (0, \delta, \delta, 0) \rightarrow (0, \delta, 0, \delta) \rightarrow (0, 0, \delta, \delta), \tag{5}$$

with four weak-subkey assumptions[1]: $Z_{6(j-1)+3}$, $Z_{6(j-1)+4}$, Z_{6j+4}, $Z_{6(j+1)+3}$ starting from round j, for $j \geq 1$. All rotations of (5), for instance, starting from $(0, \delta, \delta, 0)$ instead of $(0, 0, \delta, \delta)$, result in equivalent characteristics.

Another relevant choice is the 1-round iterative characteristic

$$(\delta, \delta, \delta, \delta) \rightarrow (\delta, \delta, \delta, \delta), \tag{6}$$

with two weak-subkey assumptions: $Z_{6(j-1)}, Z_{6(j-1)+3}$ starting from round j.

We next describe attacks on WIDEA-n that bypass all the MDS matrices across 8.5-round WIDEA-n.

4.1 Differential Attack on One IDEA Instance Only

We use (6), a 1-round iterative characteristic whose differential trail does not include any MA-box, that is, all MA-boxes are passive. See Table 2. Extending it to WIDEA-4, one IDEA instance will follow the differential pattern (6), while

[1] We adapted the original terminology $Z_i^{(j)}$ that represents the i-subkey of the j-th round in [7] to the notation Z_l in WIDEA-n as described in Sect. 3, where $l = 6(j-1) + i - 1$, since there are six subkeys per round in IDEA.

the other three IDEA instances will have zero input difference. This means that all MAD-boxes will be passive. In other words, we exploit the (trivial) fixed point

$$MDS(0,0,0,0)^t = (0,0,0,0)^t,$$

where the superscript t denotes the transpose operation. In other words, if all weak-subkey assumption are satisfied, then the differential trail (6) concatenated with itself will propagate across a single 8.5-round IDEA instance in WIDEA-n, instead of (nonzero) differences spreading to all n IDEA instances, effectively bypassing the MDS diffusion layer in every round. This attack holds independent of which MDS matrix is used. Note that our attack does not contradict the branch number of the MDS matrix [4], but rather exploit a (trivial) fixed point.

Thus, we have the following 1-round iterative characteristic, using (6) in only a single IDEA instance in WIDEA-4:

$$\begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ \delta\,\delta\,\delta\,\delta \end{pmatrix} \rightarrow \begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ 0\,0\,0\,0 \\ \delta\,\delta\,\delta\,\delta \end{pmatrix}, \tag{7}$$

following the state representation (3). Without loss of generality, we picked the last row of the state as the active IDEA instance. The same reasoning applies if we had picked any of the other IDEA instances (but the weak key would not be the same). Thus, the fifteen most significant bits (MSB) of the following eighteen subkeys should be zero across 8.5 rounds of one IDEA instance: $Z_{6(j-1)}$, $Z_{6(j-1)+3}$, for $1 \leq j \leq 9$. Under these conditions, the output difference pattern in (7) will appear as ciphertext difference with certainty. If the key schedule of WIDEA-4 behaves as a random mapping and the weak-subkey conditions hold independently, then in WIDEA-4's key space of 2^{512} keys, the weak subkeys would represent a class of about $2^{512-15\cdot 18} = 2^{242}$ keys. The reasoning is that each 16-bit weak subkey can be either 0 or 1, and we need eighteen of them to be weak. For WIDEA-8, the same reasoning applies, but the weak-key class size is estimated at $2^{1024-15\cdot 18} = 2^{754}$.

In order to avoid this distinguishing attack using (7), more than $512/15 = 34$ weak-subkey conditions would be required, since each weak subkey implies the fifteen MSBs to be zero. This means WIDEA-4 would need more than $34/2 = 17$ rounds, which means more than double the original number of rounds, since each round requires two weak subkeys. But, the resulting performance would hardly be acceptable.

A key-recovery attack on the full 8.5-round WIDEA-4, using (7), can obtain the subkeys $Z_{48,0}$ and $Z_{51,0}$ of the last half-round. In this case, only sixteen subkeys need be weak, which imply a weak-key class of about $2^{512-15\cdot 16} = 2^{272}$ keys. The output difference after eight rounds, restricted to one IDEA instance, is $(\delta, \delta, \delta, \delta)$. For the additive subkeys, the δ difference propagates across to the ciphertext, but not for $Z_{48,0}$ and $Z_{51,0}$. This means a 16-bit condition for each 16-bit subkey piece. Thus, one chosen pair of texts is enough. Decrypting two multiplications in a half-round in one IDEA instance is equivalent to $\frac{1}{17\cdot 2\cdot 4}$ of

a full WIDEA-4, that is, the cost becomes $\frac{2^{32}}{17 \cdot 8} \approx 2^{25}$ WIDEA-4 encryptions. There are 17 half-rounds in 8.5-round IDEA. Memory cost is negligible. Time complexity for WIDEA-8 becomes 2^{24} encryptions since WIDEA-8 contains eight copies of IDEA. Moreover, the weak-key class size becomes $2^{1024-15 \cdot 16} = 2^{784}$. Recovering subkeys from the other IDEA instances is not possible because of the zero differences in (7). If we shift the $(\delta, \delta, \delta, \delta)$ pattern to another row of the state in (7), then we would need another key that has weak subkeys in that same part of the state. We leave the issue of a full key-recovery attack as an open problem.

4.2 Differential Attack on All IDEA Instances

Let us analyse WIDEA-4. If we use the 3-round iterative linear relation (5), then there are active MAD-boxes along the differential trail. See Table 2. This means we need to exploit another fixed point of the AES MDS matrix (1):

$$MDS(\delta, \delta, \delta, \delta)^t = (\delta, \delta, \delta, \delta)^t.$$

When a MAD-box is active, we have to attack all four copies in WIDEA-4 at once so that the same value δ appears inside each MAD-box. This means that the input to the active MDS matrices is $(\delta, \delta, \delta, \delta)$. Applying it to the matrix in (1) results in $2 \cdot \delta \oplus 3 \cdot \delta \oplus \delta \oplus \delta = (2 \oplus 3 \oplus 1 \oplus 1) \cdot \delta = \delta$ in all four rows since the MDS matrix in AES is circulant. In other words, this fixed point exploits the fact that the exclusive-or of the coefficients in a line (or column) of the AES MDS matrix xor to 1. This is a new and surprising interaction between the differential pattern $(\delta, \delta, \delta, \delta)$ and the coefficients of the AES MDS matrix.

This attack does not contradict the branch number of the MDS matrix [4], but rather exploit a non-trivial fixed point. Note that this property does not hold for the 8×8 MDS matrix in (2) since in the latter, the exclusive-or sum of the coefficients in a line or column is 3. A consequence of this finding is an additional criterion for block cipher designs that employ matrices from MDS codes in the way they are used in WIDEA-4: carefully select the coefficients in these matrices in order to avoid fixed-point (differences).

We arrive at the following 3-round iterative characteristic:

$$\begin{pmatrix} 0 & \delta & 0 & \delta \\ 0 & \delta & 0 & \delta \\ 0 & \delta & 0 & \delta \\ 0 & \delta & 0 & \delta \end{pmatrix} \rightarrow \begin{pmatrix} 0 & 0 & \delta & \delta \\ 0 & 0 & \delta & \delta \\ 0 & 0 & \delta & \delta \\ 0 & 0 & \delta & \delta \end{pmatrix} \rightarrow \begin{pmatrix} 0 & \delta & \delta & 0 \\ 0 & \delta & \delta & 0 \\ 0 & \delta & \delta & 0 \\ 0 & \delta & \delta & 0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 & \delta & 0 & \delta \\ 0 & \delta & 0 & \delta \\ 0 & \delta & 0 & \delta \\ 0 & \delta & 0 & \delta \end{pmatrix}. \tag{8}$$

Thus, we exploit a combined symmetry in the AES MDS matrix, the differential pattern and the four identical copies of IDEA in WIDEA-4. Across 8.5 rounds, we need the following eleven 64-bit subkeys to be weak: Z_3, Z_9, Z_{10}, Z_{16}, Z_{21}, Z_{27}, Z_{28}, Z_{34}, Z_{39}, Z_{45} and Z_{46}, across all four IDEA instances. These weak subkeys imply conditions on $11 \cdot 4 \cdot 15 = 660 > 512$ bits. In a key space of 2^{512} keys we do not expect any weak-key class to satify all these conditions. It is a negative result, though. Analogous conclusions hold for WIDEA-8.

5 Linear Cryptanalysis of WIDEA-n

For a linear analysis, we listed exhaustively all one-round linear relations of IDEA [2] in Table 3 (in the appendix). All these linear relations hold with maximum bias 2^{-1} under weak-subkey assumptions.

The linear relations with the best profile for an attack on WIDEA-n have the same patterns as the characteristics used in Sect. 4, with γ instead of δ. From Table 3 we choose linear relations that: (i) minimize the number of weak-subkey assumptions per round, and (ii) are iterative. Under these two conditions, the best choices include the 3-round relation

$$(0, \gamma, \gamma, 0) \to (\gamma, 0, \gamma, 0) \to (\gamma, \gamma, 0, 0) \to (0, \gamma, \gamma, 0), \tag{9}$$

with only four weak-subkey assumptions: $Z_{6(j-1)+4}, Z_{6j}, Z_{6(j+1)}, Z_{6(j+1)+4}$ starting from round j. All rotations of (9), for instance, starting from $(\gamma, 0, \gamma, 0)$ instead of $(0, \gamma, \gamma, 0)$, also fulfill the same criteria.

Another relevant choice is the 1-round iterative characteristic

$$(\gamma, \gamma, \gamma, \gamma) \to (\gamma, \gamma, \gamma, \gamma), \tag{10}$$

with only two weak-subkey assumptions per round: $Z_{6(j-1)}, Z_{6(j-1)+3}$ starting from round j.

If we use linear relation (10) in a single IDEA instance in WIDEA-4, we arrive at the following 3-round iterative linear relation:

$$\begin{pmatrix} 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ \gamma\ \gamma\ \gamma\ \gamma \end{pmatrix} \to \begin{pmatrix} 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ \gamma\ \gamma\ \gamma\ \gamma \end{pmatrix}. \tag{11}$$

Similar to (7), there are no active MAD-boxes along (11) because all approximations at the input to the MAD-boxes are trivial: $(0, 0, 0, 0)$. Thus, once again, we exploit the fixed-point relation $\mathrm{MDS}(0, 0, 0, 0)^t = (0, 0, 0, 0)^t$. Concatenating (11) with itself across 8.5 rounds, this linear relation holds with maximum bias 2^{-1} as long as the following eighteen subkeys are weak: $Z_{6(j-1)}, Z_{6(j-1)+3}$ for $0 \leq j \leq 9$. The weak-key class for this linear relation has size $2^{512-15\cdot 18} = 2^{242}$ keys. Using relation (9) instead of (10) would require all IDEA instance to be attacked at once. This means using the fixed point $\mathrm{MDS}(\gamma, \gamma, \gamma, \gamma)^t = (\gamma, \gamma, \gamma, \gamma)^t$, which leads to the same problem as in Sect. 4.2: there are too many weak-subkey conditions because some MAD-boxes become active.

The linear relation (11) can be translated into $P \cdot \Gamma = E_K(P) \cdot \Gamma$, where $\Gamma = (\gamma, \gamma, \gamma, \gamma)$. This linear relation can be used to distinguish the full WIDEA-4 from a random permutation. A equally interesting consequence would be its implications in a hash function context. We discuss it in Sect. 6.

Applying (11) to WIDEA-8 gives even better results, since the later has key size of 1024 bits, and the same weak-subkey conditions lead to an estimated weak-key class of $2^{1024-15\cdot 18} = 2^{754}$ keys.

A (partial) key-recovery attack on the full 8.5-round WIDEA-4, using (11), can recover subkeys $Z_{48,0}$ and $Z_{51,0}$. In this case, we attack the last half-round and only sixteen subkeys need be weak: $Z_{6(j-1)}, Z_{6(j-1)+3}$ for $0 \leq j \leq 8$, which implies a weak key class of about $2^{512-15\cdot 16} = 2^{272}$ keys. Using Matsui's estimation for a high-success rate attack, $8(2^{-1})^{-2} = 32$ known plaintexts are enough. The effort is equivalent to 2^{32} multiplications per subkey, which is equivalent to fraction of $\frac{1}{17\cdot 2\cdot 4}$ of a full WIDEA-4 computation, or $2^{32}/(17\cdot 2\cdot 4) \approx 2^{25}$ WIDEA-4 encryptions. The memory needed is 32 counters. Recovery of the remaining subkeys has the same problems as in the key-recovery attack in Sect. 4.1. The time complexity for WIDEA-8 becomes 2^{24} WIDEA-8 encryptions since there are eight IDEA instances, Also, the weak-key class size is $2^{1024-15\cdot 16} = 2^{784}$.

6 WIDEA-*n* in Davies-Meyer Mode

In [6], the authors suggested to use WIDEA-n as a compression function in Davies-Meyer (DM) mode, since the key size is double the block size[2]. The hash digest could range from 224 bits up to 512 bits, as in the SHA-2 hash function family [10] by truncation of the last chaining variable. The DM mode for a compression function construction is as follows [8]: the i-th chaining value is

$$H_i = H_{i-1} \oplus E_{m_i}(H_{i-1}), \tag{12}$$

where $H_0 = IV$ is the initial value, m_i is the i-th message block and $E_x(y)$ is a block cipher with key x and plaintext y. In particular, E is WIDEA-n, $|m_i|$ is $128n$ bits, $|H_i|$ is $64n$ bits.

The issue of weak subkeys in WIDEA-n is even more relevant in a hash function setting. In this case, the message to be hashed becomes the key input and can be chosen by the adversary.

We point to the following consequences from the results in the previous sections when WIDEA-n is used in DM mode:

- semi free-start collision: suppose we can set H_{i-1} with difference (7) for WIDEA-4. If m_i is a weak key that leads to weak subkeys as required in Sect. 4.1, then $H_{i-1} = E_{m_i}(H_{i-1})$, that is, H_i contains only zero word differences according to (12). It is a semi free-start collision because only the chaining variable has nonzero difference [8]. The same reasoning applies to WIDEA-8, using the same difference (7), but extended to a 1024-bit state. Note that this attack is independent of the MDS matrix used.
- truncation: suppose the output transformation in a (hypothetical) hash function using WIDEA-n in the compression function simply truncates the output to the least significant 192 bits for WIDEA-4 (3), or to the least significant 448 bits for WIDEA-8 (4). In both cases, we assume that at least 64 bits are cut off from the last chaining variable. Alternatively, more bits

[2] The number of rounds was increased from 8.5 to 10.5 to provide some security margin.

can be dropped, but 64 bits is enough for our attack. Suppose we have a differential trail like (7) in H_{i-1} but with weak subkeys up to the 8th round. These trails reduce the number of required weak subkeys to sixteen instead of eighteen (increasing the weak-key class size), but the input and output difference patterns are not the same. This fact implies that in DM mode, the exclusive-or between H_{i-1} and $E_{m_i}(H_{i-1})$ will not vanish. But, on the other hand, the nonzero difference words are isolated in a single 64-bit piece of the state. If that 64-bit piece is in the most significant part of the state, it will be truncated and we have a collision since the rest of the state has only zero word difference. In this way, we use the output transformation of the hash function to an attacker's advantage, if we can control the difference to remain in the part of the state that is going to be truncated[3].

– a linear relation such as

$$\begin{pmatrix} 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ \gamma\ \gamma\ \gamma\ \gamma \end{pmatrix} \rightarrow \begin{pmatrix} 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0 \\ \gamma\ \gamma\ \gamma\ \gamma \end{pmatrix}, \tag{13}$$

for WIDEA-4 $= E$, using relation (10), imply the linear relation $H_{i-1} \cdot \Gamma_1 = E_{m_i}(H_{i-1}) \cdot \Gamma_1$, where Γ_1 is any of the masks in a state in (13). Applying this iterative relation to a single compression function in DM mode, leads to

$$H_i \cdot \Gamma_1 = 0, \tag{14}$$

that is, the linear relation does not depend on H_{i-1} due to feedforward in the DM mode and the splitting of the Γ_1 mask. Relation (14) could be used to distinguish the compression function using WIDEA-4 in DM mode from a random function. Note that (14) depends only on the output H_i, and could be applied to the hash digest only if the masked bits are not truncated. This linear relation have implications for WIDEA-n in applications such as pseudorandom number generation, since the masked bit would leak information in the output bitstream.

Similar reasoning applies for WIDEA-8 in place of WIDEA-4 since the reasoning concerns one single IDEA instance (out of n).

Another example of collision using only two text blocks is in DES using the complementation property: suppose two texts P and $\overline{P}$ encrypted under an arbitrary key K and its bitwise complement $\overline{K}$. The corresponding ciphertexts are $C = \mathrm{DES}_K(P)$ and $\overline{C} = \mathrm{DES}_{\overline{K}}(\overline{P})$. In DM mode, $\Delta H_{i-1} = \Delta P = P \oplus \overline{P} = ffffffffffffffff_x$ and $\Delta H_i = \Delta H_{i-1} \oplus DES_K(P) \oplus$

[3] This collision has to happen in the last message block hashed. If the message length is say at most $2^{128} - 1$ bits, then the last 128 bits are reserved for the message length. Assume the first 64 bits are variable, so we can control the difference in it. For WIDEA-4, the remaining 256-64-128=64 bits are padding. For WIDEA-8, the remaining 1024-64-128=832 bits are padding. So, this is feasible, since we only need nonzero difference in the most significant 64 bits, while the rest of the state has zero difference.

$DES_{\overline{K}}(\overline{P}) = ffffffffffffffff_x \oplus ffffffffffffffff_x = 0$, which is a free-start collision (we have nonzero difference in both the key and the plaintext, which corresponds to message and chaining variable). This is yet another example of how a weakness in the key schedule turns into a weakness in a hash function setting, jeopardizing potential applications of a block cipher in a hash function setting.

7 Weak Keys

An important question to address is whether weak keys exist in WIDEA-n that can generate the weak subkeys required in the attacks in Sect. 4, 5 and 6. Recalling the key schedule of WIDEA-4 in Sect. 3 and taking, for instance, the pattern (7) repeated over six rounds, requires that the most significant fifteen bits (corresponding to the first IDEA instance in Fig. 1) of the following subkeys to be zero: Z_0, Z_3, Z_6, Z_9, Z_{12}, Z_{15}, Z_{18}, Z_{21}, Z_{24}, Z_{27}, Z_{30} and Z_{33}. Satisfying Z_0, Z_3 and Z_6 is straightforward since they are part of the user key. There are nine subkey conditions left. Writing down the corresponding equations in the key schedule we got the following, where the variables in boldface are either 0 or 1. So far, we did not find any contradiction. That is, even though we could not yet find a 512-bit user key that leads to the eleven weak subkeys, we have found no reason these nine equations (15)–(23) cannot be satisfied.

$$\mathbf{Z}_9 = (((Z_8 \oplus Z_1) \overset{16}{\boxplus} Z_4) \overset{16}{\lll} 5) \lll 24, \tag{15}$$

$$\mathbf{Z}_{12} = (((Z_{11} \oplus Z_4) \overset{16}{\boxplus} Z_7) \overset{16}{\lll} 5) \lll 24, \tag{16}$$

$$\mathbf{Z}_{15} = (((Z_{14} \oplus Z_7) \overset{16}{\boxplus} Z_{10}) \overset{16}{\lll} 5) \lll 24, \tag{17}$$

$$\mathbf{Z}_{18} = (((Z_{17} \oplus Z_{10}) \overset{16}{\boxplus} Z_{13}) \overset{16}{\lll} 5) \lll 24, \tag{18}$$

$$\mathbf{Z}_{21} = (((Z_{20} \oplus Z_{13}) \overset{16}{\boxplus} Z_{16}) \overset{16}{\lll} 5) \lll 24, \tag{19}$$

$$\mathbf{Z}_{24} = ((((Z_{23} \oplus Z_{16}) \overset{16}{\boxplus} Z_{19}) \overset{16}{\lll} 5) \lll 24) \oplus C_2, \tag{20}$$

$$\mathbf{Z}_{27} = (((Z_{26} \oplus Z_{19}) \overset{16}{\boxplus} Z_{22}) \overset{16}{\lll} 5) \lll 24, \tag{21}$$

$$\mathbf{Z}_{30} = (((Z_{29} \oplus Z_{22}) \overset{16}{\boxplus} Z_{25}) \overset{16}{\lll} 5) \lll 24, \tag{22}$$

$$\mathbf{Z}_{33} = (((Z_{32} \oplus Z_{25}) \overset{16}{\boxplus} Z_{28}) \overset{16}{\lll} 5) \lll 24. \tag{23}$$

Assuming that the key schedule behaves as a random mapping, generating (approximately) uniformly distributed subkeys, we expect each subkey to have equal chance to assume a value in the range $[0, \ldots, 2^{64} - 1]$ for WIDEA-4, or $[0, \ldots, 2^{128} - 1]$ for WIDEA-8. Therefore, we assume each 16-bit subkey for each IDEA instance to have approximately the same chance to have values in the range $[0, \ldots, 2^{16} - 1]$. For the particular values 0 and 1 the chance is 2^{-16} for each. Under these assumptions, we estimated the weak-key classes in Sect. 4, 5 and 6.

In order to have some experimental evidence of the presence of weak subkeys, we searched for them in mini-versions of the WIDEA-4 key schedule. Recall that in WIDEA-4, the subkeys are 64-bit wide since each one of them has to key four IDEA instances at once. As such, the search effort is too big even though we are looking for a weak subkey value in a 16-bit piece of the 64-bit WIDEA-4 subkey. Taking into account that the key schedule operates wordwise, we shrank the word size from 16 to 4 bits. So, for instance, equations such as (15) would be modified to $Z_9 = (((Z_8 \oplus Z_1) \overset{4}{\boxplus} Z_4) \overset{4}{\lll} 1) \lll 6$, where the rotation amounts 1 and 6 were chosen to match the reduced word sizes. Attack simulations on such reduced scale equations shows weak 4-bit weak subkey values to appear for the first IDEA instance, as expected for the attacks in Sect. 4, 5 and 6. The same behavior was observed when the word size was reduced to 5 bits, leading to equations such as $Z_9 = (((Z_8 \oplus Z_1) \overset{5}{\boxplus} Z_4) \overset{5}{\lll} 2) \lll 7$. These experiments provide evidence that weak subkey values can and do appear in critical places in differential and linear trails, which gives some evidence for the propagation of differential and linear patterns.

8 Conclusions

This paper described the first differential and linear analyses of the full WIDEA-n ciphers [6], for $n \in \{4, 8\}$ under weak-key assumptions, both in the block cipher and in the hash function settings. Table 1 summarizes our attack complexities for WIDEA-n.

We exploited iterative differential characteristics and iterative linear relations that bypassed the MDS matrices in WIDEA-n by carefully choosing trails that input trivial differences or relations, such as $(0, 0, 0, 0)^t$, or symmetric ones such as $(\delta, \delta, \delta, \delta)^t$ to the MAD-boxes. The rationale is to exploit fixed points for the MDS matrix for these particular differences and masks. This effectively means that we found and exploited **narrow differential and linear trails**. This phenomenon was observed for the AES MDS matrix, for which $\text{MDS}(\delta, \delta, \delta, \delta)^t = (\delta, \delta, \delta, \delta)^t$. This simple observation allowed us to bypass all diffusion layers connecting the four IDEA instances in WIDEA-4 because the exclusive-or sum of the coefficients in the MDS matrix of AES equals one. This result does not hold for the MDS matrix used in WIDEA-8. Our attacks exploit structural weaknesses due to the way MDS matrices are placed inside WIDEA-n to connect n IDEA instances. These attacks do not appply to the AES cipher [4].

Other attacks, that hold for any MDS matrix in WIDEA-n, exploit iterative differential (and linear) patterns that avoid the MAD-boxes altogether in every round. Such patterns cause zero input differences or zero input masks into each MAD-box, and thus, exploit the all-zero fixed point: $\text{MDS}(0, 0, 0, 0)^t = (0, 0, 0, 0)^t$. This approach is much more effective than the previous one because: (i) there are many fewer weak-subkey restrictions, (ii) it applies to any MDS matrix, (iii) we only attack one IDEA instance instead of n, which reduces considerably the attack complexity and increases the weak-key class size for the attack. The larger the value of n, the better the attacks become.

The implications of weak differential and linear attacks are not restricted to the block cipher setting. In [6], the authors suggested to use WIDEA-n in a compression function in Davies-Meyer mode. Our attacks lead to semi free-start collisions (depending on truncation of the final hash digest) or distinguishing attacks on the compression function using WIDEA-n in Davies-Meyer mode.

Even though the weak-key classes correspond to a small fraction of the key space, their existence implies that WIDEA-n are not ideal ciphers, and as such cannot be used in cryptographic constructions that require tight security, in the same way as IDEA [12].

Table 1. Attack complexities for the full 8.5-round WIDEA-n with $n \in \{4, 8\}$

cipher	attack type	complexity			# weak keys	comment
		data	time	memory	(‡)	
WIDEA-4	DC (distinguishing)	2 CP	2	negl.	2^{242}	see (7)
	DC (key recovery)*	2 CP	2^{25}	negl.	2^{272}	see (7)
	LC (distinguishing)	32 KP	32	negl.	2^{242}	see (11)
	LC (key recovery)*	32 KP	2^{25}	negl.	2^{272}	see (11)
WIDEA-8	DC (distinguishing)	2 CP	2	negl.	2^{754}	see (7)
	DC (key recovery)*	2 CP	2^{24}	negl.	2^{784}	see (7)
	LC (distinguishing)	32 KP	32	negl.	2^{754}	see (11)
	LC (key recovery)*	32 KP	2^{24}	negl.	2^{784}	see (11)

CP: chosen plaintext; CC: chosen ciphertext; KP: known plaintext; *: partial key recovery; ‡: estimated

Open problems include: (i) how to recover the full (512- or 1024-bit) key of WIDEA-n; (ii) find weak keys that through the key schedule algorithms lead to weak round subkeys fitting in the requirements of our differential and linear distinguishers.

References

1. Courtois, N.: Algebraic complexity reduction and cryptanalysis of GOST, IACR ePrint archive 2011/626 (2011)
2. Daemen, J., Govaerts, R., Vandewalle, J.: Weak Keys for IDEA. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 224–231. Springer, Heidelberg (1994)
3. Dinur, I., Dunkelman, O., Shamir, A.: Improved attacks on full GOST, IACR ePrint archive, 2011/558 (2011)
4. FIPS197: Advanced Encryption Standard (AES), FIPS PUB 197 Federal Information Processing Standard Publication 197, U.S. Department of Commerce (2001)
5. ISO: Information Technology – Security Techniques – Hash functions – Part 3: Dedicated hash functions. ISO/IEC 10118-3:2004, International Organization for Standardization (2004)
6. Junod, P., Macchetti, M.: Revisiting the IDEA Philosophy. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 277–295. Springer, Heidelberg (2009)

7. Lai, X., Massey, J.L., Murphy, S.: Markov Ciphers and Differential Cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)
8. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press (1997)
9. Nakahara Jr., J., Rijmen, V., Preneel, B., Vandewalle, J.: The MESH Block Ciphers. In: Chae, K., Yung, M. (eds.) WISA 2003. LNCS, vol. 2908, pp. 458–473. Springer, Heidelberg (2004)
10. SHS: Secure Hash Standard, Federal Information Processing Standards, FIPS PUB 180-3 (October 2008)
11. Vaudenay, S.: Related-key attack against triple encryption based on fixed points. In: SECRYPT 2011, pp. 59–67. SciTPress (2011)
12. Wei, L., Peyrin, T., Sokolowski, P., Ling, S., Pieprzyk, J., Wang, H.: On the (in)security of IDEA in various hashing modes. IACR ePrint archive, 2012/264 (2012)

A Appendix

Table 2. One-round characteristics of IDEA using xor differences and $\delta = 8000_x$

1-round characteristics	weak subkeys j-th round	diff. in MA-box
$(0,0,0,\delta) \to (\delta,\delta,\delta,0)$	$Z_{6(j-1)+3}, Z_{6(j-1)+5}$	$(0,\delta) \to (\delta,\delta)$
$(0,0,\delta,0) \to (\delta,0,0,0)$	$Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\delta,0) \to (0,\delta)$
$(0,0,\delta,\delta) \to (0,\delta,\delta,0)$	$Z_{6(j-1)+3}, Z_{6(j-1)+4}$	$(\delta,\delta) \to (\delta,0)$
$(0,\delta,0,0) \to (\delta,\delta,0,\delta)$	$Z_{6(j-1)+5}$	$(0,\delta) \to (\delta,\delta)$
$(0,\delta,0,\delta) \to (0,0,\delta,\delta)$	$Z_{6(j-1)+3}$	**(0,0)→(0,0)**
$(0,\delta,\delta,0) \to (0,\delta,0,\delta)$	$Z_{6(j-1)+4}$	$(\delta,\delta) \to (\delta,0)$
$(0,\delta,\delta,\delta) \to (\delta,0,\delta,\delta)$	$Z_{6(j-1)+3}, Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\delta,0) \to (0,\delta)$
$(\delta,0,0,0) \to (0,\delta,0,0)$	$Z_{6(j-1)}, Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\delta,0) \to (0,\delta)$
$(\delta,0,0,\delta) \to (\delta,0,\delta,0)$	$Z_{6(j-1)}, Z_{6(j-1)+3}, Z_{6(j-1)+4}$	$(\delta,\delta) \to (\delta,0)$
$(\delta,0,\delta,0) \to (\delta,\delta,0,0)$	$Z_{6(j-1)}$	**(0,0)→(0,0)**
$(\delta,0,\delta,\delta) \to (0,0,\delta,0)$	$Z_{6(j-1)}, Z_{6(j-1)+3}, Z_{6(j-1)+5}$	$(0,\delta) \to (\delta,\delta)$
$(\delta,\delta,0,0) \to (\delta,0,0,\delta)$	$Z_{6(j-1)}, Z_{6(j-1)+4}$	$(\delta,\delta) \to (\delta,0)$
$(\delta,\delta,0,\delta) \to (0,\delta,\delta,\delta)$	$Z_{6(j-1)}, Z_{6(j-1)+3}, Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\delta,0) \to (0,\delta)$
$(\delta,\delta,\delta,0) \to (0,0,0,\delta)$	$Z_{6(j-1)}, Z_{6(j-1)+5}$	$(0,\delta) \to (\delta,\delta)$
$(\delta,\delta,\delta,\delta) \to (\delta,\delta,\delta,\delta)$	$Z_{6(j-1)}, Z_{6(j-1)+3}$	**(0,0)→(0,0)**

Table 3. One-round linear relations of IDEA with $\gamma = 1$

1-round linear relations	weak subkeys j-th round	masks in MA-box
$(0,0,0,\gamma) \to (0,0,\gamma,0)$	$Z_{6(j-1)+3}, Z_{6(j-1)+5}$	$(0,\gamma) \to (\gamma,0)$
$(0,0,\gamma,0) \to (\gamma,0,\gamma,\gamma)$	$Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\gamma,\gamma) \to (0,\gamma)$
$(0,0,\gamma,\gamma) \to (\gamma,0,0,\gamma)$	$Z_{6(j-1)+3}, Z_{6(j-1)+4}$	$(\gamma,0) \to (\gamma,\gamma)$
$(0,\gamma,0,0) \to (0,0,0,\gamma)$	$Z_{6(j-1)+5}$	$(0,\gamma) \to (\gamma,1)$
$(0,\gamma,0,\gamma) \to (0,0,\gamma,\gamma)$	$Z_{6(j-1)+3}$	**(0,0)→(0,0)**
$(0,\gamma,\gamma,0) \to (\gamma,0,\gamma,0)$	$Z_{6(j-1)+4}$	$(\gamma,0) \to (\gamma,\gamma)$
$(0,\gamma,\gamma,\gamma) \to (\gamma,0,0,0)$	$Z_{6(j-1)+3}, Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\gamma,\gamma) \to (0,\gamma)$
$(\gamma,0,0,0) \to (0,\gamma,\gamma,\gamma)$	$Z_{6(j-1)}, Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\gamma,\gamma) \to (0,\gamma)$
$(\gamma,0,0,\gamma) \to (0,\gamma,0,\gamma)$	$Z_{6(j-1)}, Z_{6(j-1)+3}, Z_{6(j-1)+4}$	$(\gamma,0) \to (\gamma,\gamma)$
$(\gamma,0,\gamma,0) \to (\gamma,\gamma,0,0)$	$Z_{6(j-1)}$	**(0,0)→(0,0)**
$(\gamma,0,\gamma,\gamma) \to (\gamma,\gamma,\gamma,0)$	$Z_{6(j-1)}, Z_{6(j-1)+3}, Z_{6(j-1)+5}$	$(0,\gamma) \to (\gamma,0)$
$(\gamma,\gamma,0,0) \to (0,\gamma,\gamma,0)$	$Z_{6(j-1)}, Z_{6(j-1)+4}$	$(\gamma,0) \to (\gamma,\gamma)$
$(\gamma,\gamma,0,\gamma) \to (0,\gamma,0,0)$	$Z_{6(j-1)}, Z_{6(j-1)+3}, Z_{6(j-1)+4}, Z_{6(j-1)+5}$	$(\gamma,\gamma) \to (0,\gamma)$
$(\gamma,\gamma,\gamma,0) \to (\gamma,\gamma,0,\gamma)$	$Z_{6(j-1)}, Z_{6(j-1)+5}$	$(0,\gamma) \to (\gamma,0)$
$(\gamma,\gamma,\gamma,\gamma) \to (\gamma,\gamma,\gamma,\gamma)$	$Z_{6(j-1)}, Z_{6(j-1)+3}$	**(0,0)→(0,0)**

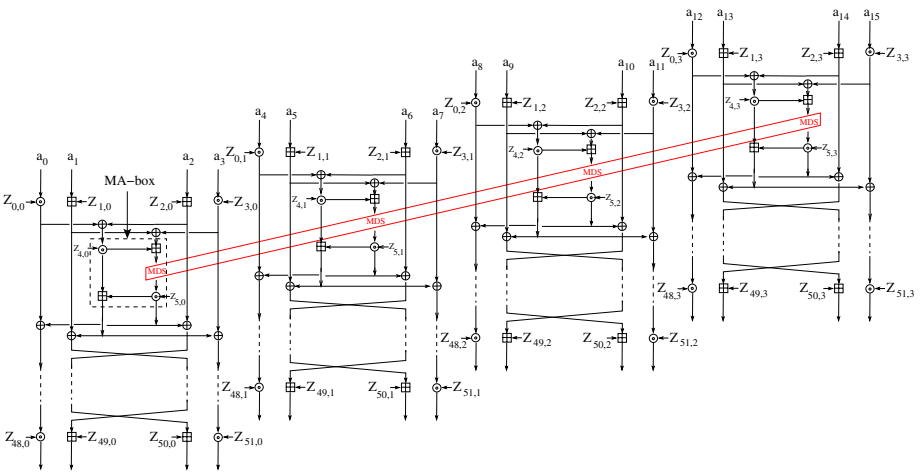

Fig. 1. Computational graph of the WIDEA-4 block cipher

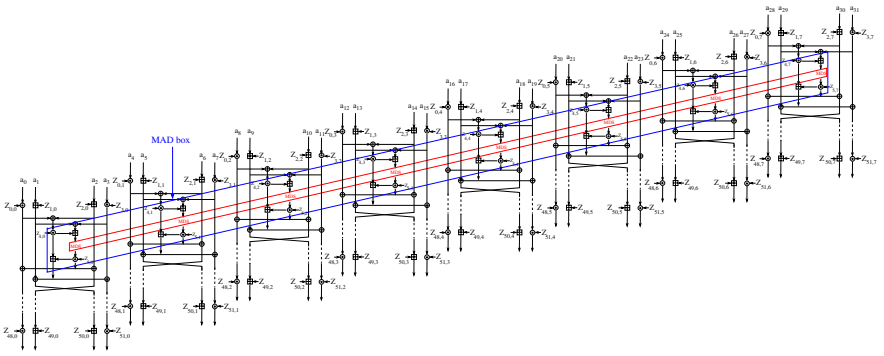

Fig. 2. Computational graph of the WIDEA-8 block cipher