New Cryptanalytic Results on IDEA

Eli Biham*[1] , Orr Dunkelman*[1], Nathan Keller**[2]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
{biham,orrd}@cs.technion.ac.il
[2]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il

Abstract. IDEA is a 64-bit block cipher with 128-bit keys introduced by Lai and Massey in 1991. IDEA is one of the most widely used block ciphers, due to its inclusion in several cryptographic packages, such as PGP and SSH. The cryptographic strength of IDEA relies on a combination of three incompatible group operations – XOR, addition and modular multiplication. Since its introduction in 1991, IDEA has withstood extensive cryptanalytic effort, but no attack was found on the full variant of the cipher.
In this paper we present the first known non-trivial relation that involves all the three operations of IDEA. Using this relation and other techniques, we devise a linear attack on 5-round IDEA that uses 2^{19} known plaintexts and has a time complexity of 2^{103} encryptions. By transforming the relation into a related-key one, a similar attack on 7.5-round IDEA can be applied with data complexity of $2^{43.5}$ known plaintexts and a time complexity equivalent to $2^{115.1}$ encryptions. Both of the attacks are by far the best known attacks on IDEA.

1 Introduction

The International Data Encryption Algorithm (IDEA) is a 64-bit, 8.5-round block cipher with 128-bit keys proposed by Lai and Massey in 1991 [20]. Due to its inclusion in several cryptographic packages, such as PGP and SSH, IDEA is one of the most widely used block ciphers. Since its introduction, IDEA resisted intensive cryptanalytic efforts [1, 5, 6, 8–14, 16, 21, 22, 24]. The best published chosen-plaintext attack on IDEA is an attack on 5-round IDEA that requires 2^{24} chosen plaintexts, and has time complexity of 2^{126} encryptions [12]. The best published related-key attack is an attack on 6.5-round IDEA that requires $2^{57.8}$ chosen plaintexts encrypted under four related keys and has time complexity of $2^{88.1}$ encryptions [5]. Along with the attacks on reduced-round variants, several weak-key classes for the entire IDEA were found. The largest weak key class (identified by a boomerang technique) contains 2^{64} keys, and the

* This work was supported in part by the Israel MOD Research and Technology Unit.
** The research presented in this paper was supported by the Adams fellowship.

membership test requires 2^{16} adaptive chosen plaintexts and ciphertexts and has a time complexity of 2^{16} encryptions [6].

The cryptographic strength of IDEA relies on the combination of three incompatible group operations: bitwise XOR, modular addition in $Z_{2^{16}}$, and modular multiplication in $GF(2^{16}+1)$ where 0 is replaced by 2^{16}. All the three operations are essential for the security of the cipher. Indeed, if the multiplication is removed, then the cipher can be broken easily by examining the least significant bits of the words during the encryption. If the XOR is removed, then the cipher is affine over addition in $Z_{2^{16}}$, and hence, is easily breakable using only few known plaintexts. In [7, 26] it is shown that if the addition is removed then the cipher can be easily broken using multiplicative differentials.

In this paper we present the first known non-trivial relation that involves all the three different operations of IDEA. More precisely, we show that for the MA transformation of IDEA, that is composed of additions and multiplications, there exists an XOR differential with a non-trivial probability.

We use our new relation to devise several new attacks on IDEA based on various attack techniques: First, we devise linear-type attacks on reduced-round variants of IDEA that are similar to the attacks presented in [12, 16, 24]. The attacks are based on constructing linear approximations with bias 1/2 that relates the least significant bits of some words during the encryption process. We use our relation, along with differential techniques and partial key guessing, to improve the basic technique presented in [16, 24] and to establish the best known attack on 5-round IDEA. Our attack requires only 2^{19} known plaintexts and the time complexity is equivalent to 2^{103} encryptions. Both the data and the time complexities are smaller than the respective complexities of all the previously known attacks on 4.5 or 5 rounds of IDEA. Our attack also has a relatively small memory complexity, unlike the 5-round attack in [12]. We also devise realistic attacks on variants of IDEA with a small number of rounds: A distinguishing attack on 2.5-round IDEA requiring 2^{18} chosen plaintexts and time complexity of 2^{18} encryptions, and an attack on 3-round IDEA with data complexity of 2^{19} chosen plaintexts and time complexity of about $2^{48.5}$ encryptions. Both of the attacks are better in some of the parameters than all the known attacks on the respective variants of IDEA.

We also show how to use the same relation in the related-key model. Using two related keys, we are able to extend the linear property by 2.5 rounds. This gives rise to a 7.5-round attack on IDEA requiring $2^{43.5}$ known plaintexts and a time complexity of $2^{115.1}$ encryptions. It is also possible to use our new relation to improve the previously best known related-key attack on IDEA, using the related-key rectangle technique. These improvements can be used to construct a 7-round related-key rectangle attack on IDEA with data complexity of 2^{65} related-key chosen plaintexts and time complexity of $2^{104.2}$ 7-round IDEA encryptions. The complexities of the new attacks, along with selected previously known attacks, are summarized in Table 1.

Rounds	Attack Type	Complexity Data	Time	# of Affected Keys	Source
2	Differential	2^{10} CP	2^{42}	all	[21]
2.5	Differential	2^{10} CP	2^{106}	all	[21]
3	Differential-Linear	2^{29} CP	2^{44}	all	[8]
3.5	Linear	103 KP/CP	2^{97}	all	[16]
3.5	Square	2^{22} CP	2^{66}	all	[16]
4	Imppossible Differential	2^{37} CP	2^{70}	all	[1]
4	Linear	114 KP	2^{114}	all	[24]
4	Square	2^{23} CP	2^{98}	all	[16]
4.5	Impossible Differential	2^{64} CP	2^{112}	all	[1]
5	Meet-in-the-Middle Attack	2^{24} CP	2^{126}	all	[12]
6.5	Related-Key Rectangle	$2^{59.8}$ RK-CP	$2^{88.1}$	all	[5]
$2.5^{\dagger}$	Linear	2^{18} CP	2^{18}	all	Section 4.1
3	Linear	2^{19} CP	$2^{48.5}$	all	Section 4.2
4.5	Linear	16 CP	2^{103}	all	Section 4.3
5	Linear	2^{19} KP	2^{103}	all	Section 4.3
7.5	Related-Key Linear	$2^{43.5}$ RK-KP	$2^{115.1}$	all	Section 5
7	Related-Key Rectangle	2^{65} RK-CP	$2^{104.2}$	all	Appendix A.3

KP – Known plaintext, CP – Chosen plaintext, RK – Related key,
Time complexity is measured in encryption units
† – Distinguishing attack

Table 1. Selected Known Attacks on IDEA and Our New Results

We expect that the new relation can also be used to improve other attacks on IDEA, as well as attacks on other block ciphers that use the same operations, e.g., the MESH family of block ciphers [23].

The paper is organized as follows: In Section 2, we briefly describe the structure of IDEA. In Section 3 we present the new relation between the operations of IDEA. In Section 4 we present the new attack on 5-round IDEA. In Section 5 we transform this attack into a 7.5-round related-key attack on IDEA. Appendix A suggests a related-key rectangle attack on 7-round IDEA. Finally, Section 6 summarizes the paper.

2 Description of IDEA and the Notations Used in the Paper

IDEA [20] is a 64-bit, 8.5-round block cipher with 128-bit keys. It uses a composition of XOR operations, additions modulo 2^{16}, and multiplications over $GF(2^{16}+1)$.

Every round of IDEA is composed of two layers. The round input of round i is composed of four 16-bit words denoted by $(X_1^i, X_2^i, X_3^i, X_4^i)$. In the first layer, denoted by KA, the first and the fourth words are multiplied by subkey words (mod $2^{16}+1$) where 0 is replaced by 2^{16}, and the second and the third words

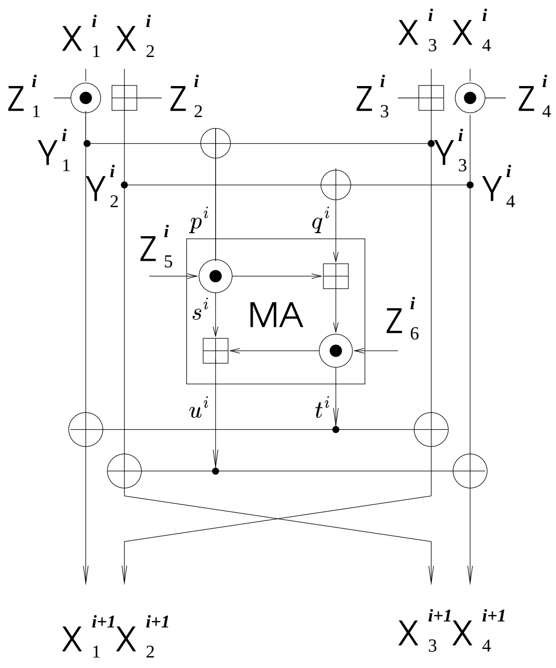

Fig. 1. One Round of IDEA

are added to subkey words in (mod 2^{16}). The intermediate values after this half-round are denoted by $(Y_1^i, Y_2^i, Y_3^i, Y_4^i)$. Formally, let Z_1^i, Z_2^i, Z_3^i, and Z_4^i be the four subkey words, then

$$Y_1^i = Z_1^i \odot X_1^i; \quad Y_2^i = Z_2^i \boxplus X_2^i; \quad Y_3^i = Z_3^i \boxplus X_3^i; \quad Y_4^i = Z_4^i \odot X_4^i$$

Then, $(p^i, q^i) = (Y_1^i \oplus Y_3^i, Y_2^i \oplus Y_4^i)$ enters the second layer, a structure composed of multiplications and additions denoted by MA. We denote the two output words of the MA transformation by (u^i, t^i). Denoting the subkey words that enter the MA function by Z_5^i and Z_6^i,

$$u^i = (p^i \odot Z_5^i) \boxplus t^i; \quad t^i = (q^i \boxplus (p^i \odot Z_5^i)) \odot Z_6^i$$

Another notation we use in the attack refers to an intermediate value in the MA layer: we denote the value $p^i \odot Z_5^i$ by s^i.

The output of the i-th round is $(Y_1^i \oplus t^i, Y_3^i \oplus t^i, Y_2^i \oplus u^i, Y_4^i \oplus u^i)$. In the last round (round 9) the MA layer is removed. Thus, the ciphertext is $(Y_1^9||Y_2^9||Y_3^9||Y_4^9)$. The structure of a single round of IDEA is shown in Figure 1.

IDEA's key schedule is linear: each subkey is composed of bits selected from the key. However, the exact structure of the key schedule is crucial for our attacks and hence the entire key schedule is described in Table 2.

Round	Z_1^i	Z_2^i	Z_3^i	Z_4^i	Z_5^i	Z_6^i
$i=1$	0–15	16–31	32–47	48–63	64–79	80–95
$i=2$	96–111	112–127	25–40	41–56	57–72	73–88
$i=3$	89–104	105–120	121–8	9–24	50–65	66–81
$i=4$	82–97	98–113	114–1	2–17	18–33	34–49
$i=5$	75–90	91–106	107–122	123–10	11–26	27–42
$i=6$	43–58	59–74	100–115	116–3	4–19	20–35
$i=7$	36–51	52–67	68–83	84–99	125–12	13–28
$i=8$	29–44	45–60	61–76	77–92	93–108	109–124
$i=9$	22–37	38–53	54–69	70–85		

Table 2. The Key Schedule Algorithm of IDEA

3 A New Non-Trivial Relation Between the Three Operations of IDEA

In this section we present the new non-trivial relation between the three different operations of IDEA. The relation we present is a property of the MA layer. Since the property is independent of the round number, in this section we omit the round index in all the notations. The property is related to the XOR difference between the values in two encryptions. We denote the difference in the word X by ΔX.

Observation 1 *Assume that the XOR difference between the two intermediate encryption values in the input to the MA layer is of the form $(\Delta p, \Delta q) = (0, \alpha)$ for some α. Assume also that there is no key difference in the key word Z_5 (but there is no assumption whether there is a key difference in the subkey word Z_6). Then:*

1. *The least significant bit of the value $\Delta u \oplus \Delta t$ equals zero.*
2. *The average probability of the event $(\Delta u, \Delta t) = (8000_x, 8000_x)$ over all the possible keys is 2^{-16} (if $\alpha \neq 0$ or if there is a key difference in Z_6).*
3. *If α is non-zero or if there is a difference in Z_6, then $\Sigma_{\nu,\tau} \Pr^2[(\Delta u, \Delta t) = (\nu, \tau)] = 2^{-23.72}$.*

We note that the first part of the observation is similar to observations that were used in [12, 16, 24].

If the MA layer was truly random, then the probability of the event $(\Delta u, \Delta t) = (8000_x, 8000_x)$ would be 2^{-32}. Hence, we have a differential with a much higher probability than expected.

The third part of the observation gives a much higher value than the corresponding value for a random function (which is 2^{-32}). The value discussed in the third part of the observation affects boomerang and rectangle attacks.

We shall now provide the proof of the observation: The proof uses the additive difference (module 2^{16}) between the two inputs, which we denote by δX. As there is no XOR difference in the first input word to the MA function ($\Delta p = 0$), then there is no additive difference as well, i.e., $\delta p = 0$. As there is no additive difference in the subkey Z_5, then $\Delta s = \delta s = 0$ as well. As $u = t \boxplus s$ then $\delta u = \delta t \boxplus \delta s = \delta t$. We use this relation in the proof:

1. $LSB(\Delta u) = LSB(\delta u) = LSB(\delta t) = LSB(\Delta t)$, where $LSB(w)$ denotes the least significant bit of the word w. Thus, $LSB(\Delta u \oplus \Delta t) = LSB(\Delta u) \oplus LSB(\Delta t) = 0$.
2. Since no assumption on α or the subkey difference in Z_6 was used (aside the fact that there is such a difference), we can assume that the value δt is randomly distributed. Hence, with probability 2^{-16} the difference is $\delta t = 8000_x$. In this case, $\delta u = 8000_x$ as well. However, $\delta t = 8000_x$ is equivalent to $\Delta t = 8000_x$. Thus, the probability of the event $(\Delta u, \Delta t) = (8000_x, 8000_x)$ is indeed 2^{-16}, as asserted.
3. We can write

$$\Sigma_{\nu,\tau} \Pr{}^2[(\Delta u, \Delta t) = (\nu, \tau)] = \Sigma_{\nu,\tau} \left(\Sigma_\delta \Pr[(\Delta u, \Delta t) = (\nu, \tau) \wedge (\delta t = \delta)]\right)^2 = \\ 2^{-32} \cdot \Sigma_{\nu,\tau}(\Sigma_\delta \Pr[(\Delta u, \Delta t) = (\nu, \tau))|(\delta t = \delta)])^2$$

where the last equality follows from the assumption that $\Pr[\delta t = \delta] = 2^{-16}$ for every δ. We calculated the last value explicitly by a computer program and got the value $\Sigma_{\beta,\gamma} \Pr{}^2[(\Delta u, \Delta t) = (\beta, \gamma)] = 2^{-23.72}$, as asserted.

Q.E.D.

4 A New Attack on 5-Round IDEA

In this section we present new attacks on 2.5-round, 3-round and 5-round IDEA based on the first relation established in Section 3.

We start with an observation due to Biryukov (according to [24]) and Demirci [12]. Let us examine the second and the third words in all the intermediate stages of the encryption. There is a relation between the values of these words and the outputs of the MA layer in the intermediate rounds that uses only XOR and modular addition, but not multiplication. Let $P = (P_1, P_2, P_3, P_4)$ be a plaintext and let $C = (C_1, C_2, C_3, C_4)$ be its corresponding ciphertext, then

$$((((((((((((((((P_2 \boxplus Z_2^1) \oplus u^1) \boxplus Z_3^2) \oplus t^2) \boxplus Z_2^3) \oplus u^3) \boxplus Z_3^4) \oplus t^4) \boxplus Z_2^5) \oplus u^5) \\ \boxplus Z_3^6) \oplus t^6) \boxplus Z_2^7) \oplus u^7) \boxplus Z_3^8) \oplus t^8) \boxplus Z_2^9) = C_2. \tag{1}$$

Similarly,

$$((((((((((((((((P_3 \boxplus Z_3^1) \oplus t^1) \boxplus Z_2^2) \oplus u^2) \boxplus Z_3^3) \oplus t^3) \boxplus Z_2^4) \oplus u^4) \boxplus Z_3^5) \oplus t^5) \\ \boxplus Z_2^6) \oplus u^6) \boxplus Z_3^7) \oplus t^7) \boxplus Z_2^8) \oplus u^8) \boxplus Z_3^9) = C_3. \tag{2}$$

Now, if we are interested only in the value of the least significant bit (LSB) of the words, modular addition is equivalent to XOR and we can simplify the above equations into:

$$LSB(P_2 \oplus Z_2^1 \oplus u^1 \oplus Z_3^2 \oplus t^2 \oplus Z_2^3 \oplus u^3 \oplus Z_3^4 \oplus t^4 \oplus Z_2^5 \oplus u^5 \oplus Z_3^6 \oplus t^6 \oplus Z_2^7 \\ \oplus u^7 \oplus Z_3^8 \oplus t^8 \oplus Z_2^9) = LSB(C_2), \tag{3}$$

and

$$LSB(P_3 \oplus Z_3^1 \oplus t^1 \oplus Z_2^2 \oplus u^2 \oplus Z_3^3 \oplus t^3 \oplus Z_2^4 \oplus u^4 \oplus Z_3^5 \oplus t^5 \oplus Z_2^6 \oplus u^6 \oplus Z_3^7 \\ \oplus t^7 \oplus Z_2^8 \oplus u^8 \oplus Z_3^9) = LSB(C_3). \tag{4}$$

Since $u^i = t^i \boxplus s^i$ then $LSB(u^i) = LSB(t^i \boxplus s^i)$, thus, $LSB(u^i \oplus t^i) = LSB(s^i)$. Taking this into consideration and XORing the two above equations we obtain

$$LSB(P_2 \oplus P_3 \oplus Z_2^1 \oplus Z_3^1 \oplus s^1 \oplus Z_2^2 \oplus Z_3^2 \oplus s^2 \oplus Z_2^3 \oplus Z_3^3 \oplus s^3 \oplus Z_2^4 \oplus Z_3^4 \oplus s^4 \\ \oplus Z_2^5 \oplus Z_3^5 \oplus s^5 \oplus Z_2^6 \oplus Z_3^6 \oplus s^6 \oplus Z_2^7 \oplus Z_3^7 \oplus s^7 \oplus Z_2^8 \oplus Z_3^8 \oplus s^8 \oplus Z_2^9 \oplus Z_3^9) \\ = LSB(C_2 \oplus C_3). \tag{5}$$

This equation is called in [16] "the Biryukov-Demirci relation".

Consider two plaintexts P^1 and P^2. Denote the XOR difference between the encryptions of P^1 and P^2 (under the same secret key) in an intermediate value X by ΔX. Then, the XOR the equations given by P^1 and P^2 gives

$$LSB(P_2^1 \oplus P_3^1 \oplus P_2^2 \oplus P_3^2 \oplus \Delta s^1 \oplus \Delta s^2 \oplus \Delta s^3 \oplus \Delta s^4 \oplus \Delta s^5 \oplus \Delta s^6 \oplus \Delta s^7 \oplus \\ \Delta s^8) = LSB(C_2^1 \oplus C_3^1 \oplus C_2^2 \oplus C_3^2). \tag{6}$$

Equation (6) is the basic equation used in all our attacks in this section.

4.1 A Distinguishing Attack on 2.5-Round IDEA

Consider a 2.5-round variant of IDEA of the form $KA \circ MA \circ KA \circ MA \circ KA$. For sake of simplicity we assume that the attack is on the first 2.5 rounds of IDEA, but the same attack holds for any 2.5 consecutive rounds of this form.

For a 2.5-round IDEA, Equation (6) is reduced to

$$LSB(P_2^1 \oplus P_3^1 \oplus P_2^2 \oplus P_3^2 \oplus \Delta s^1 \oplus \Delta s^2) = LSB(C_2^1 \oplus C_3^1 \oplus C_2^2 \oplus C_3^2). \tag{7}$$

Note that by the first part of the observation in Section 3, if the input XOR difference to the MA layer is of the form $(\Delta p, \Delta q) = (0, \alpha)$ then $\Delta s = 0$. In order to use this property, we consider pairs of plaintexts (P^1, P^2) such that $\Delta(X_1^1, X_2^1, X_3^1, X_4^1) = (0, \beta, 0, \gamma)$ for arbitrary values of β and γ. For these pairs $\Delta Y_1^1 = \Delta Y_3^1 = 0$ (independent of the values Z_1^1, Z_3^1), and hence $\Delta p^1 = 0$. Therefore, the required property holds and $\Delta s^1 = 0$. We note that the same idea was used (to some extent) in [16].

Similarly, if we take only ciphertext pairs satisfying $\Delta(Y_1^3, Y_2^3, Y_3^3, Y_4^3) = (0, 0, \beta', \gamma')$ for arbitrary values of β' and γ', then $(\Delta p^2, \Delta q^2) = (0, \alpha')$ for some α', and hence $\Delta s^2 = 0$.

If the plaintext/ciphertext pair $((P^1, C^1), (P^2, C^2))$ satisfies both differential relations required above, Equation (7) is further reduced into

$$LSB(P_2^1 \oplus P_3^1 \oplus P_2^2 \oplus P_3^2) = LSB(C_2^1 \oplus C_3^1 \oplus C_2^2 \oplus C_3^2). \tag{8}$$

This is a simple linear relation that can be checked easily since only bits of the plaintexts and the ciphertexts are involved in the equation.

Based on these observations, we can mount a simple distinguishing attack on 2.5-round IDEA, using the following algorithm:

1. Ask for the encryption of 2^{18} plaintexts of the form (A, Z, B, W), where A and B are fixed and Z and W assume arbitrary random values.
2. Insert the ciphertexts into a hash table sorted by the first two words.
3. For every pair of ciphertexts in the same bin of the hash table, check whether Equation (8) holds for the corresponding plaintext/ciphertext pair.
4. If there is a pair for which the equation does not hold, conclude that the cipher is not 2.5-round IDEA. If there is no such pair, conclude that the cipher is 2.5-round IDEA.

Due to the structure of the plaintexts, for every pair of plaintexts the first differential requirement holds. For every pair of ciphertexts in the same bin of the hash table, the second requirement also holds. Hence, for all the checked pairs Equation (8) should be satisfied for 2.5-round IDEA.

The 2^{18} plaintexts can be combined into about 2^{35} possible pairs, and a fraction of 2^{-32} of them is expected to have ciphertext difference of the form $(0, 0, \beta', \gamma')$. Hence, the expected number of pairs analyzed in Step 3 is eight. If there is a pair for which the equation does not hold, we know for sure that the cipher is not 2.5-round IDEA. On the other hand, for a random permutation, the probability that the equation holds for all the eight pairs is $1/256$. Hence, the distinguisher succeeds with probability greater than 99.5%.

Since the second and the third steps of the attack are implemented using a hash table, the time complexity of the attack is dominated by the time complexity of the encryptions in the first step of the attack. Hence, the data complexity of the attack is 2^{18} chosen plaintexts and the time complexity is 2^{18} encryptions.

4.2 A Key Recovery Attack on 3-Round IDEA

The 2.5-round distinguisher can be extended to an attack on 3-round IDEA of the form $E = KA \circ MA \circ KA \circ MA \circ KA \circ MA$ by guessing the subkey of the last MA layer and applying the distinguishing attack to the first 2.5 rounds. In this case, the data complexity is slightly increased, since more pairs are required in the last step of the attack in order to discard all the wrong key values.

The attack algorithm is the following:

1. Ask for the encryption of 2^{19} plaintexts of the form (A, Z, B, W), where A and B are fixed and Z and W assume arbitrary random values.
2. For every guess of the 32-bit subkey of the last MA layer:

(a) Partially decrypt all the ciphertexts through the last MA layer and insert the resulting Y^3 values into a hash table sorted by the first 32 bits.
(b) For every pair of values in the same bin of the hash table, check whether Equation (8) holds for the corresponding plaintext/ciphertext pair.
(c) If there is a pair for which the equation does not hold, discard the subkey guess. Otherwise, keep the subkey guess.

3. Output all the subkey guesses that were not discarded.

Since there are 2^{19} plaintexts, then there are about 2^{37} possible pairs, and about 32 pairs are examined in Step 2(b). Hence, for a wrong key guess the probability that the equation holds for all the pairs is 2^{-32}. Therefore, only few possible key guesses remain, including the right key. The filtering can be further improved by enlarging the data structure by a small factor.

The time complexity of the attack is dominated by Step 2(b) which contains decrypting all ciphertexts under all the subkey guesses. The data complexity of the attack is 2^{19} chosen plaintexts and the time complexity of the attack is equivalent to $2^{19} \times 2^{32} \times (1/6) \approx 2^{48.5}$ 3-round encryptions. Note that the attack recovers only 32 bits of the master key and the rest of the key has to be found using other techniques.

We note that a similar attack can be mounted on a 3-round variant of IDEA of the form $E = MA \circ KA \circ MA \circ KA \circ MA \circ KA$. The only difference is that in this case the attack is performed in the decryption direction. The time and data complexities remain unchanged.

The two extensions can be combined to an attack on a 3.5-round variant of IDEA of the form $E = MA \circ KA \circ MA \circ KA \circ MA \circ KA \circ MA$. However, in this case the data and time complexities are worse than the complexities of the best known attack on 3.5-round IDEA. This follows from the fact that while in the 3-round attacks we could guarantee that one of the differential conditions holds, in the 3.5-round attack this is not the case.

4.3 Attack on 5-Round IDEA

In this section we devise an attack on a 5-round variant of IDEA starting with the second half of round 3. Choosing round 3 as the starting point of the attack is the optimal round, as described later.

First, we consider a 4.5-round attack starting at the beginning of round 4. For this variant, the Equation (6) is transformed into

$$LSB(P_2^1 \oplus P_3^1 \oplus P_2^2 \oplus P_3^2 \oplus \Delta s^4 \oplus \Delta s^5 \oplus \Delta s^6 \oplus \Delta s^7) = LSB(C_2^1 \oplus C_3^1 \oplus C_2^2 \oplus C_3^2). \quad (9)$$

In our attack we use pairs of plaintexts with XOR difference $\Delta(X_1^4, X_2^4, X_3^4, X_4^4) = (0, \beta, 0, \gamma)$, thus, $\Delta s^4 = 0$. In order to calculate Δs^i for $5 \leq i \leq 7$, we guess part of the master key and partially decrypt the ciphertexts through the last three rounds.

In order to calculate the required Δs^i values, we guess the subkeys Z_4^8,Z_3^8,Z_2^8, Z_1^8,Z_6^7,Z_5^7,Z_4^7,Z_3^7,Z_2^7,Z_1^7,Z_6^6,Z_5^6 that allow to partially decrypt two rounds, and the subkeys Z_1^6, Z_2^6, Z_5^5 that allow to calculate the value Δs^5. However, it appears that all these 15 subkeys use only 103 bits of the master key, whereas bits 100–124 of the master key remain unused. Hence, we can guess 103 bits of the master key, and for each guess we can check whether the equation holds for the plaintext/ciphertext pairs. We note that finding the right subkey requires about 128 pairs for the analysis, which can be constructed from about 16 chosen plaintexts. We also note that starting the attack in a different round would require guessing more subkey bits.

In order to extend the attack to 5 rounds, we guess the subkey of the MA layer in round 3. This does not increase the time complexity since the relevant subkey is composed of bits 50–81 of the master key that are included in the 103 bits we guess in the 4.5-round attack. However, this additional half round affects the data complexity of the attack.

The only remaining issue is getting pairs of plaintexts with difference $\Delta(X_1^4, X_2^4, X_3^4, X_4^4) = (0, \beta, 0, \gamma)$. Since for every guess of the MA layer of round 3 different plaintext pairs are needed to fulfill this differential requirement, this attack uses known plaintexts instead of chosen plaintexts. We start with 2^{19} known plaintexts that compose 2^{37} possible pairs. For each subkey guess of the MA layer of round 3, we partially encrypt all the plaintexts and choose the pairs that have difference $\Delta(X_1^4, X_2^4, X_3^4, X_4^4) = (0, \beta, 0, \gamma)$. We expect 32 such pairs, and these pairs are used in the sequel of the attack. The time complexity of this step is negligible compared to the time complexities of the other steps of the attack.

The attack algorithm is as follows:

1. Ask for the encryption of 2^{19} known plaintexts.
2. For each guess of key bits 50–81, perform the following:
   (a) Partially encrypt the plaintexts through the MA layer of round 3 and insert the resulting X^4 values to a hash table indexed by the first and the third words.
   (b) For each guess of key bits 0–49,82–99,[1] and 125–127 and for all the colliding pairs, perform the following:
      i. Partially decrypt all the pairs through rounds 7 and 6, and the MA layer of round 5.
      ii. Verify that Equation (9) holds for all of the pairs. If no, discard the key guess.
   (c) If the key guess passed the filtering, perform exhaustive search on the remaining 25 key bits.

As we mentioned before, for every guess of key bits 50–81, we expect that 32 pairs are analyzed in Step 2(b) of the attack. Hence, the probability that a wrong key guess passes the filtering is 2^{-32}. Thus, we expect that about $2^{103} \cdot 2^{-32} = 2^{71}$ key guesses enter Step 2(c). Thus, the time complexity of Step 2(c) is expected to be equivalent to $2^{25} \cdot 2^{71} = 2^{96}$ encryptions in total.

Therefore, the time complexity of the attack is dominated by the partial decryptions of Step 2(b). We observe that this step can be optimized. Note that

[1] Note that key bits 50–81 are already guessed.

half of the key guesses are discarded after the first pair, half of the remaining key guesses are discarded after the second pair, etc. Hence, instead of decrypting all the pairs at once, the attacker can decrypt the first pair and check whether the equation holds, then (if the key guess was not discarded) decrypt the second pair and check the equation for it, etc. Using this improvement, the time complexity of this step is $2^{103}+2^{102}+2^{101}+\ldots \approx 2^{104}$ partial decryptions, which are roughly equivalent to 2^{103} full encryptions.

Hence, the data complexity of the attack is 2^{19} known plaintexts and the time complexity is 2^{103} encryptions.

5 Related-Key Attack on 7.5-Round IDEA

In this section we present a related-key attack on the first 7.5 rounds of IDEA. The 7.5-round related-key attack uses similar relations as the 5-round known plaintext attack. In the attack we use the difference between the keys to construct pairs of plaintexts for which the intermediate values (when encrypted under the two different keys) are equal for 2.5 rounds. For such pairs of plaintexts, Equation (6) is reduced to a much simpler one.

Let the K and K^* be two keys such that they are equal in all bits but bit 34 and any non-empty subset of bits $\{41, 42, \ldots, 49\}$. Let P and P^* be the two plaintexts, such that Y^2 and Y^{2*}, the corresponding intermediate encryption values after the KA layer of round 2, satisfy:

$$Y_1^2 = Y_1^{2*}; \quad Y_2^2 = Y_2^{2*}; \quad Y_3^2 = Y_3^{2*}; \quad Y_4^2 = Y_4^{2*} \tag{10}$$

In such pair, the intermediate encryption values are equal until the MA layer of round 4. In that MA layer, the input difference is $(\Delta p^4, \Delta q^4) = (0, 0)$ and the key difference affects only Z_6^4. Hence, by the observation presented in Section 3, $\Delta s^2 = \Delta s^3 = \Delta s^4 = 0$.

Therefore, for such pair Equation (6) is reduced to

$$LSB(P_2 \oplus P_3 \oplus P_2^* \oplus P_3^* \oplus \Delta s^1 \oplus \Delta s^5 \oplus \Delta s^6 \oplus \Delta s^7) = LSB(C_2 \oplus C_3 \oplus C_2^* \oplus C_3^*). \tag{11}$$

Hence, if the attacker is able to construct plaintext pairs satisfying Equation (10), he can partially encrypt/decrypt the plaintext/ciphertext pairs through rounds $1, 7, 6$, and 5 and check whether Equation (11) is satisfied. In order to do so, the attacker has to guess the subkeys Z_1^1, Z_3^1, Z_5^1 for the partial encryption and $Z_5^5, Z_1^6, Z_2^6, Z_5^6, Z_6^6, Z_1^7 – Z_6^7, Z_1^8 – Z_4^8$ for the partial decryption. However, these 18 subkeys use only 103 bits of the master key, and hence guessing these key bits and checking whether Equation (11) holds for some plaintext/ciphertext pairs satisfying Equation (10) yields an attack faster than exhaustive key search.

Constructing pairs of plaintexts satisfying Equation (10) is not a trivial operation. However, if we use the known plaintext model and take sufficiently many plaintexts, then Equation (10) may be satisfied sufficiently many times. A naive approach would be to partially encrypt all the given known plaintexts through round 1 and the KA layer of round 2, and to find the relevant pairs. However, even in an optimized manner, this approach would result in guessing 96 key bits, which combined with the known plaintext nature of the attack results in a time complexity of least 2^{128} 1-round IDEA encryptions.

Therefore, we use a modified approach. We use $2^{42.5}$ known plaintexts encrypted under two related keys (a total of $2^{43.5}$ related-key known plaintexts), and partially encrypt them through the KA layer of round 1. After the KA layer, we consider only the pairs that have difference $(0, 0040_x, 0, 0040_x)$. Such pairs have difference $(0, 0, 0040_x, 0040_x)$ at the input to the KA layer of round 2, independent of the value of the subkeys Z_5^1, Z_6^1. With probability 1/2 the difference in the third word is canceled by the key difference, and with probability 2^{-16} the difference in the fourth word is canceled by the key difference, leading to a pair that satisfies Equation (10). Hence, the required pairs are detected in a two steps algorithm. First the attacker guesses the values of the subkeys Z_1^1, Z_2^1, Z_3^1, and Z_4^1 and finds the pairs having difference $(0, 0040_x, 0, 0040_x)$ after the first KA layer. Most of the pairs are filtered at this stage. Then the attacker further guesses the values of the subkeys Z_5^1, Z_6^1, Z_3^2, and Z_4^2 and checks which of the remaining pairs satisfy Equation (10).

The attack algorithm on 7.5-round IDEA is as follows:

1. Ask for $2^{42.5}$ known plaintexts encrypted under K and denote the set of plaintexts and ciphertexts by $SetP$.
2. Ask for $2^{42.5}$ known plaintexts encrypted under K^* and denote the set of plaintexts and ciphertexts by $SetP^*$.
3. For each guess of the subkeys Z_1^1, Z_2^1, Z_3^1, and Z_4^1:
   (a) Partially encrypt all plaintexts in $SetP$ and in $SetP^*$ through the KA layer of round 1.
   (b) Find all pairs of Y^1 (encrypted under K) and Y^{1*} (encrypted under K^*) such that $Y^1 \oplus Y^{1*} = (0, 0040_x, 0, 0040_x)$.
   (c) For each such pair, and each guess of Z_5^1, Z_6^1, Z_3^2, and Z_4^2:
      i. If the pair satisfies Equation (10), guess $Z_5^5, Z_1^6, Z_2^6, Z_5^6, Z_6^6, Z_1^7$–$Z_6^7$, and Z_1^8–Z_4^8 and verify whether Equation (11) is satisfied.
      ii. If the equation is not satisfied — discard the subkey guess.
4. For each remaining subkey, exhaustively try all 25 remaining subkey bits, and output the remaining key.

There are 2^{85} pairs of plaintexts, of which $2^{85} \cdot 2^{-64} = 2^{21}$ have difference $(0, 0040_x, 0, 0040_x)$ after the KA layer of round 1. For each guess of Z_5^1, Z_6^1, Z_3^2, and Z_4^2, about $2^{21} \cdot 2^{-17} = 16$ pairs have a zero difference after the KA layer of round 2, satisfying Equation (10). For a correct subkey guess, all these pairs should satisfy Equation (11). For wrong subkey guesses, the probability that Equation (11) is satisfied for all the pairs is 2^{-16}. There are 2^{103} possible subkeys, and hence the number of subkeys that enter Step 4 is expected to be $2^{103} \cdot 2^{-16} = 2^{87}$.

The time complexity of the attack is thus dominated by Step 3 (Steps 1 and 2 have time complexity of $2^{42.5}$ encryptions each, and Step 4 has time complexity of $2^{87} \cdot 2^{25} = 2^{112}$ trial encryptions). Step 3(a) is repeated 2^{64} times, and each time $2^{43.5}$ values are partially encrypted through one KA layer. Hence, the time complexity of this step is $2^{64} \cdot 2^{43.5} = 2^{107.5}$ partial encryptions. Step 3(b) can be executed efficiently using a hash table. In Step 3(c)(i) only 2^{21} pairs (or 2^{22} values) are analyzed but this step requires guessing 32 more bits (Z_3^2 and Z_4^2 are covered by the bits guessed in Step 3(a)). Thus, the time complexity of the first part of this step (finding the pairs satisfying Equation (10)) is $2^{64} \cdot 2^{22} \cdot 2^{32} = 2^{118}$ 1-round decryptions. The time complexity of the second part of Step 3(c)(i) (checking whether Equation (11) is satisfied) is much lower, as even though 9 more key bits are guessed, there are only 32 pairs (or 64 values) that enter this step. Thus, the total time complexity of the attack is about $2^{118} \cdot \frac{1}{7.5} = 2^{115.1}$ 7.5-round IDEA encryptions.

6 Summary and Conclusions

In this paper we presented several new results on the block cipher IDEA: The first non-trivial relation involving all the three different operations of IDEA, a known-plaintext 5-round attack, a related-key attack on 7.5-round IDEA (with two keys) and a related-key rectangle attack on 7-round IDEA (with four keys). These results are by far the best known attacks against reduced-round variants of the cipher.

Our paper shows that the linear key schedule of IDEA makes the cipher relatively vulnerable to attacks that guess vast amounts of the key. However, despite our findings, the full IDEA still resists all known attacks.

References

1. Eli Biham, Alex Biryukov, Adi Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.
2. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, proceedings of EUROCRYPT '01, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.
3. Eli Biham, Orr Dunkelman, Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceedings of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 1–16, Springer-Verlag, 2002.
4. Eli Biham, Orr Dunkelman, Nathan Keller, *New Combined Attacks on Block Ciphers*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 126–144, Springer-Verlag, 2005.
5. Eli Biham, Orr Dunkelman, Nathan Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology, proceedings of EUROCRYPT'05, Lecture Notes in Computer Science 3494, pp. 507-525, Springer-Verlag, 2005.
6. Alex Biryukov, Jorge Nakahara Jr., Bart Preneel, Joos Vandewalle, *New Weak-Key Classes of IDEA*, proceedings of Information and Communications Security 4, Lecture Notes in Computer Science 2513, pp. 315–326, Springer-Verlag, 2002.
7. Nikita Borisov, Monica Chew, Robert Johnson, David Wagner, *Multiplicative Differentials*, proceedings of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 17–33, Springer-Verlag, 2002.

8. Johan Borst, Lars R. Knudsen, Vincent Rijmen, *Two Attacks on Reduced Round IDEA*, Advances in Cryptology, proceedings of EUROCRYPT '97, Lecture Notes in Computer Science 1233, pp. 1–13, Springer-Verlag, 1997.
9. Joan Daemen, René Govaerts, Joos Vandewalle, *Cryptanalysis of 2.5 Rounds of IDEA (Extended Abstract)*, technical report 93/1, Department of Electrical Engineering, ESAT–COSIC, Belgium, 1993.
10. Joan Daemen, René Govaerts, Joos Vandewalle, *Weak Keys for IDEA*, Advances in Cryptology, proceedings of CRYPTO '93, Lecture Notes in Computer Science 773, pp. 224–231, Springer-Verlag, 1994.
11. Hüseyin Demirci, *Square-like Attacks on Reduced Rounds of IDEA*, proceedings of Selected Areas in Cryptography 2002, Lecture Notes in Computer Science 2595, pp. 147–159, Springer-Verlag, 2003.
12. Hüseyin Demirci, Ali A. Selçuk, Erkan Türe, *A New Meet-in-the-Middle Attack on the IDEA Block Cipher*, proceedings of Selected Areas in Cryptography 2003, Lecture Notes in Computer Science 3006, pp. 117–129, Springer-Verlag, 2004.
13. Philip Hawkes, *Differential-Linear Weak Keys Classes of IDEA*, Advances in Cryptology, proceedings if EUROCRYPT '98, Lecture Notes in Computer Science 1403, pp. 112–126, Springer-Verlag, 1998.
14. P. Hawkes, L. O'Connor, *On Applying Linear Cryptanalysis to IDEA*, Advances in Cryptology - Proceedings of ASIACRYPT'96, Lecture Notes in Computer Science 1163, pp. 105–115, Springer-Verlag, 1996.
15. Seokhie Hong, Jongsung Kim, Guil Kim, Sangjin Lee, Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 368–383, Springer-Verlag, 2005.
16. Pascal Junod, *New Attacks Against Reduced-Round Versions of IDEA*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 384–397, Springer-Verlag, 2005.
17. John Kelsey, Bruce Schneier, David Wagner, *Key-Schedule Cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, Advances in Cryptology, proceedings of CRYPTO '96, Lecture Notes in Computer Science 1109, pp. 237–251, Springer-Verlag, 1996.
18. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 75–93, Springer-Verlag, 2000.
19. Jongsung Kim, Guil Kim, Seokhie Hong, Sangjin Lee, Dowon Hong, *The Related-Key Rectangle Attack — Application to SHACAL-1*, proceedings of ACISP 2004, Lecture Notes in Computer Science 3108, pp. 123–136, Springer-Verlag, 2004.
20. Xuejia Lai, James L. Massey, Sean Murphy, *Markov Ciphers and Differential Cryptanalysis*, Advances in Cryptology - Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science 547, pp. 17–38, Springer-Verlag, 1992.
21. Willi Meier, *On the Security of the IDEA Block Cipher*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 371–385, Springer-Verlag, 1994.
22. Jorge Nakahara Jr., Paulo S.L.M. Barreto, Bart Preneel, Joos Vandewalle, Hae Y. Kim, *SQUARE Attacks Against Reduced-Round PES and IDEA Block Ciphers*, IACR Cryptology ePrint Archive, Report 2001/068, 2001.
23. Jorge Nakahara Jr., Vincent Rijmen, Bart Preneel, Joos Vandewalle, *The MESH Block Ciphers*, proceedings of Information Security Applications, 4th International Workshop, WISA 2003, Lecture Notes in Computer Science 2908, pp. 458–473, Springer-Verlag, 2004.

24. Jorge Nakahara Jr., Bart Preneel, Joos Vandewalle, *The Biryukov-Demirci Attack on Reduced-Round Versions of IDEA and MESH Ciphers*, proceedings of ACISP 2004, Lecture Notes in Computer Science 3108, pp. 98–109, Springer-Verlag, 2004.
25. NESSIE, *Performance of Optimized Implementations of the NESSIE Primitives*, NES/DOC/TEC/WP6/D21/a, available on-line at *http://www.nessie.eu.org/nessie*.
26. Havard Raddum, *Cryptanalysis of IDEA-X/2*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 1–8, Springer-Verlag, 2003.
27. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, 1999.

A A Related-Key Rectangle Attack on 7-Round IDEA

In this appendix we use the third part of the observation in Section 3 to improve the 6.5-round related-key rectangle attack presented in [5] and to devise a related-key rectangle attack on 7-round IDEA. First, we discuss the 5.5-round distinguishers used in the attacks.

A.1 Two Related-Key Boomerang Distinguishers for 5.5-Round IDEA

The 5.5-round distinguisher presented in [5] applies to rounds 2–6.5 and uses two related-key differentials.

The first related-key differential starts at the beginning of round 2 with the difference $(0, 0, 8000_x, 0)$ and ends after the MA layer of round 4. The key difference is in bit 25. Thus, the key difference cancels the data difference and the zero difference is preserved until the MA layer of round 4. In that MA layer, we assume that all the 2^{32} possible output differences are equiprobable and sum over all the possibilities. As a result, we obtain $\hat{p} = 2^{-16}$.

The second related-key differential is easier to present in the decryption direction. The differential ends after the KA layer of round 7 with data difference $(0, 0, 0100_x, 0, 0)$ and has a key difference in key bit 75. The key difference cancels the data difference with probability 1/2 and the zero difference is preserved until the KA layer of round 5. In that layer, there is key difference in the word Z_1^5. If all 2^{16} input differences of the multiplication with Z_1^5 are equiprobable (as expected for multiplication), by using all possible differences we obtain $\hat{q} = 2^{-9}$.

These differentials are combined to a 5.5-round related-key boomerang distinguisher that requires $2^{51.6}$ adaptive chosen plaintexts and ciphertexts ($2^{49.6}$ values are encrypted/decrypted using four different keys).

Using the third part of the observation in Section 3 we can easily improve this distinguisher. We replace the first differential by a similar one: The new differential also starts at the beginning of round 2, but has input difference $(0, 0, 0001_x, 0)$ and key difference is in bit 40. With probability 1/2, the key difference cancels the data difference and the zero difference is preserved until the MA layer of round 4. In this MA layer, unlike the former differential, the key difference affects only the subkey Z_6^4, while both Δp^4 and the key difference in Z_5^4 are zero. Hence, we can apply the third part of the observation in Section 3 to obtain $\hat{p} = 2^{-12.86}$. The second differential remains unchanged.

Using these new differentials, we obtain a 5.5-round related-key boomerang distinguisher that requires $2^{45.32}$ adaptive chosen plaintexts and ciphertexts ($2^{43.32}$ values are encrypted/decrypted using four different keys). The 6.5-round attack presented in [5] can be improved by a similar factor using the new differentials.

We now present a new 5.5-round distinguisher, applicable for rounds 1.5–6. The probability of the new distinguisher is lower than the probability of the former one, but it can be used to devise better key recovery attacks.

The first related-key differential starts after the KA layer of round 1 with the difference $(0, 0040_x, 0, 0040_x)$ and ends after the MA layer of round 4. The key difference is in bit 34, and any non-empty subset of bits $\{41, 42, \ldots, 49\}$. The second related-key differential starts at the beginning of round 5 with the difference $(0, 8000_x, 0, 0)$ and key difference in key bit 91. This difference evolves into a zero difference after the MA layer of round 6 with probability 1.

The second differential is quite standard. It is based on canceling the difference in the second word using the key difference in bit 91 (i.e., $\Delta K_1 = e_{91}$). Then, the zero difference is preserved until key bit 91 is used again in the subkey Z_4^7.

The first differential is a bit more complicated. A pair with input difference $\alpha = (0, 0040_x, 0, 0040_x)$ to the MA layer of round 1 has difference $(0, 0, 0040_x, 0040_x)$ after the MA layer with probability 1. With probability 1/2 the key difference cancels the data difference in the third word, and with probability 2^{-16} the key difference cancels the data difference in the fourth word. Thus, with probability 2^{-17}, the pair has a zero difference after the KA layer of round 2. This zero difference is preserved until the last multiplication in the MA layer of round 4. Hence, in that MA layer both Δp^4 and the key difference in Z_5^4 are zero. Thus, we can apply the observation in Section 3 to obtain $\hat{p} = 2^{-17} \cdot 2^{-11.86} = 2^{-28.86}$. In other words, taking two pairs with difference α before the MA of round 1, then they are expected to have the same difference after round 4 with probability of at least $\hat{p}^2 = 2^{-57.8}$. The key difference ΔK_0 can be any of 511 possible values. We use the value of $\Delta K_0 = e_{34,49}$, but it can be any of the other values without affecting our attack.

Using these differentials, we get a 5.5-round related-key boomerang distinguisher that uses $2^{59.32}$ adaptive chosen plaintexts and ciphertexts ($2^{57.32}$ values are encrypted/decrypted using four different keys).

A.2 A 6.5-Round Related-Key Rectangle Attack

We now present a related-key rectangle attack [5, 15, 19] on the first 6.5 rounds of IDEA based on the distinguisher presented above. The attack algorithm mostly follows the attack algorithm presented in [3] with the few modifications needed due to the related-key nature of the attack.

Let K_a, K_b, K_c, K_d be the related keys such that $K_b = K_a \oplus \Delta K_0$, $K_c = K_a \oplus \Delta K_1$, and $K_d = K_c \oplus \Delta K_0$. The attack algorithm is as follows:

1. **Data Collection Phase:**
   (a) Generate 2^{35} structures $S_1^a, \ldots, S_{2^{35}}^a$ of 2^{28} plaintexts each, where in each structure the first word, the six least significant bits of the second word, and the 14 least significant bits of the third word are fixed. Ask for the encryption of the structures under K_a.
   (b) Flip bit 6 of the second word and bit 13 of the third word of any plaintext encrypted under K_a, and ask for the encryption of the resulting plaintexts under K_b (to obtain $S_1^b, \ldots, S_{2^{35}}^b$).
   (c) Generate 2^{35} structures $S_1^c, \ldots, S_{2^{35}}^c$ of 2^{28} plaintexts each, where in each structure the first word, the six least significant bits of the second word, and the 14 least significant bits of the third word are fixed. Ask for the encryption of the structures under K_c.
   (d) Flip bit 6 of the second word and bit 13 of the third word of any plaintext encrypted under K_c, and ask for the encryption of the resulting plaintexts under K_d (to obtain $S_1^d, \ldots, S_{2^{39}}^d$).
2. **Finding Candidate Quartets:**
   (a) Find all pairs of ciphertexts $C_a \in S_i^a$ and $C_c \in S_j^c$, such that they have the same value in the first, the second, and the third words.
   (b) For each such pair, check whether there are pairs of ciphertexts $C_b \in S_i^b$ and $C_d \in S_j^d$, such that they have the same value in the first, the second, and the third words. If such a pair exists — transfer (P_a, P_b, P_c, P_d), the corresponding plaintexts, to analysis.
3. **Analysis of Candidate Quartets:**
   (a) Initialize 2^{64} counters, each corresponds to a different guess of $Z_1^2, Z_1^3, Z_1^4, Z_7^4$.
   (b) For each subkey guess of $Z_1^2, Z_1^3, Z_1^4, Z_7^4$ and each candidate quartet, check whether the partial encryption and partial decryption of the pairs of the quartet lead to the required differences. If this is the case increment the respective counter.
4. **Output:** Output all subkey guesses whose counter has values greater than 8.

The attack is expected to encounter more than 16 right quartets, thus, the counter that corresponds to the right subkey is expected to be greater than 8. The probability that a quartet satisfies the differences for a wrong subkey is 2^{-128}, and as for each subkey guess there are 2^{126} quartets, then the expected value of counters that correspond to wrong subkey values is $2^{126} \cdot 2^{-128} = 2^{-2}$. Thus, with probability 95.7% the right subkey is in the list of the subkeys returned in Step 4, and only 1 out of $2^{42.1}$ wrong subkeys is in the list. The remaining key bits can be found by exhaustive search (in a time complexity of about $2^{85.9}$ trial encryptions).

The time complexity of Step 1 is 2^{65} encryptions. The time complexity of Step 2(a) is the time required to insert 2^{64} ciphertexts into a hash table indexed by the first 48 bits, i.e., 2^{64} memory accesses. The expected number of collisions is $2^{63} \cdot 2^{63} \cdot 2^{-48} = 2^{78}$, and for each such collision, additional two memory accesses are needed (in Step 2(b)). Thus, the total time complexity of Step 2 is 2^{79} memory accesses.

We note that Step 2(b) usually finds about 256 pairs of ciphertexts from the corresponding Set_i^b and Set_j^d (as there are 2^{28} ciphertexts in each such set of ciphertexts). Thus, 2^{86} candidate quartets enter Step 3. We note that the analysis of each subkey word is independent of other subkey words, and thus, can be efficiently done using table look-ups. For example, consider the fourth word in the pairs (C_a, C_c) and (C_b, C_d). For the right subkey and a right quartet, the difference before the multiplication with Z_7^4 should be zero. It is possible to generate a table of 2^{32} values corresponding to the two words (the fourth word of C_a and the fourth word of C_c, or the fourth word of C_b and the fourth word of C_d) and the subkey values for which the pair has a zero difference before the multiplication. Thus, by two accesses to the table it is possible to check that both (C_a, C_c) and (C_b, C_d) suggest the same subkey value for Z_7^4.

We note that as multiplication has a relatively random behavior, then it is expected that for any subkey there is probability 2^{-16} that it decrypts the pair of values correctly. Thus, it is expected that each entry of the table contains one subkey on average, and the probability that the pairs (C_a, C_c) and (C_b, C_d) suggest the same subkey, for a wrong quartet, is 2^{-16}. If there is no subkey suggested by the two pairs, then the quartet is obviously wrong, and hence the quartet can be discarded from further analysis.

Thus, by a careful ordering of the memory accesses, and using four tables, one for each of the four key words (and the appropriate difference) it is possible to find the subkeys suggested by the quartet in approximately two memory accesses for each quartet, and thus, the total time complexity of Step 3 is about 2^{87} memory accesses.

The time complexity of Step 4 is 2^{64} memory accesses. We conclude that the attack requires 2^{87} memory access, and has a data complexity of 2^{65} related-key chosen plaintexts. Retrieving the remaining key material requires additional $2^{85.9}$ trial encryptions, but by considering the subkeys in the order of the counters (i.e., maximal counter first in a descending order) this value can be lowered by far. For example, only 1.66 wrong keys on average are expected to have counters whose value is greater than 13, while the correct subkey is expected to have such a counter with probability of 72.5%.

A.3 Attacking 7-Round IDEA

The main idea behind the 7-round attack is to partially decrypt all ciphertexts under all possible values of the key of the last MA layer, and then to apply the 6.5-round attack. A trivial implementation of this approach would lead to an attack that requires $2^{32} \cdot 2^{87} = 2^{119}$ memory accesses, and a data complexity of 2^{65} related-key chosen plaintexts.

However, we improve this result by observing that there are 12 shared bits between the subkeys Z_6^7 and Z_2^1. This allows us to filter most of the wrong candidate quartets, by evaluating the difference after the addition in the KA layer of round 1. Of course, we have to consider the cases when there is a carry from bit 3 and the cases when there is no carry, and as we do not know the correct case, we discard all the pairs whose difference is affected by the carry from bit 3.

The improved 7-round attack starts with Step 1 of the 6.5-round attack. Then, for each guess of Z_5^7, Z_6^7, all the ciphertexts are partially decrypted one half round. Then, Step 2(a) of the the 6.5-round attack is applied. Step 2(b) is altered a little bit. As noted earlier, the 12 most significant bits of Z_1^2 are known from the key guess, and the attacker can use them to find, for each plaintext $P_a \in S_i^a$ encrypted under K_a, only 2^{18} plaintexts $P_b \in S_i^b$ that may have the required difference with P_a after the KA layer of round 1. We note that a similar procedure can be done with P_c and P_d.

The number of pairs $(P_b \in S_i^b, P_d \in S_j^d)$ satisfying the differences with P_a and P_c, such that the corresponding ciphertexts have zero difference in the first, second, and third words for a given (P_a, P_c) pair that enters Step 2(b) is now reduced to $2^{18} \cdot 2^{18} \cdot 2^{-48} = 2^{-12}$. Thus, it is expected that only $2^{78} \cdot 2^{-12} = 2^{66}$ quartets enter Step 3 for each guess of Z_5^7, Z_6^7.

We note that this approach leads to discarding $1/4$ of the right pairs, i.e., pairs with difference $(0, 0040_x, 0, 0040_x)$ after the KA layer of Round 1. Thus, it is expected that we encounter only 9 right quartets. On the other hand, this approach also reduces by a factor of $9/16$ the number of quartets suggesting a wrong subkey.

Thus, the total time complexity of the new attack (without the data collection and the partial decryption steps) is:

$$2^{32} \cdot \left[2^{64} + 2^{79} + 2^{67} + 2^{64}\right] = 2^{111}$$

memory accesses.

Using the conversion of three clock cycles for one memory access, and the time measurements of the NESSIE project [25], these 2^{111} memory accesses are equivalent to $2^{104.2}$ 7-round IDEA encryptions.