Two Attacks on Reduced IDEA (Extended Abstract)

Johan Borst*[1], Lars R. Knudsen[2], Vincent Rijmen[2]**

[1] T.U. Eindhoven, Discr. Math., P.O. Box 513, NL-5600 MB Eindhoven, borst@win.tue.nl

[2] K.U. Leuven, Dept. Elektrotechniek-ESAT, Kard. Mercierlaan 94, B-3001 Heverlee, {lars.knudsen,vincent.rijmen}@esat.kuleuven.ac.be

Abstract. In 1991 Lai, Massey and Murphy introduced the IPES (Improved Proposed Encryption Standard), later renamed IDEA (International Data Encryption Algorithm). In this paper we give two new attacks on a reduced number of rounds of IDEA. A truncated differential attack on IDEA reduced to 3.5 rounds and a differential-linear attack on IDEA reduced to 3 rounds. The truncated differential attack contains a novel method for determining the secret key.

1 Introduction

The block cipher IDEA (International Data Encryption Algorithm) was proposed by X. Lai and J. Massey in [11] as a strengthened version of PES (for Proposed Encryption Standard) proposed by the same authors in [10]. The blocks are 64 bits and the keys are 128 bits. Both ciphers are based on the design concept of "mixing operations from different algebraic groups". IDEA was developed to increase the security against differential cryptanalysis. In [9] it was argued that for 3 rounds of IDEA there are no useful differentials and concluded that IDEA is resistant against a differential attack after 4 of its 8 rounds.

IDEA is an iterated cipher consisting of 8 rounds followed by an output transformation. We count the output transformation as an extra half round. The complete first round and the output transformation are depicted in the computational graph shown in Figure 1. The two multiplications and the two additions in the middle of the figure are called the MA-structure. The key schedule takes as input a 128 bit key and returns 52 subkeys, each of 16 bits.

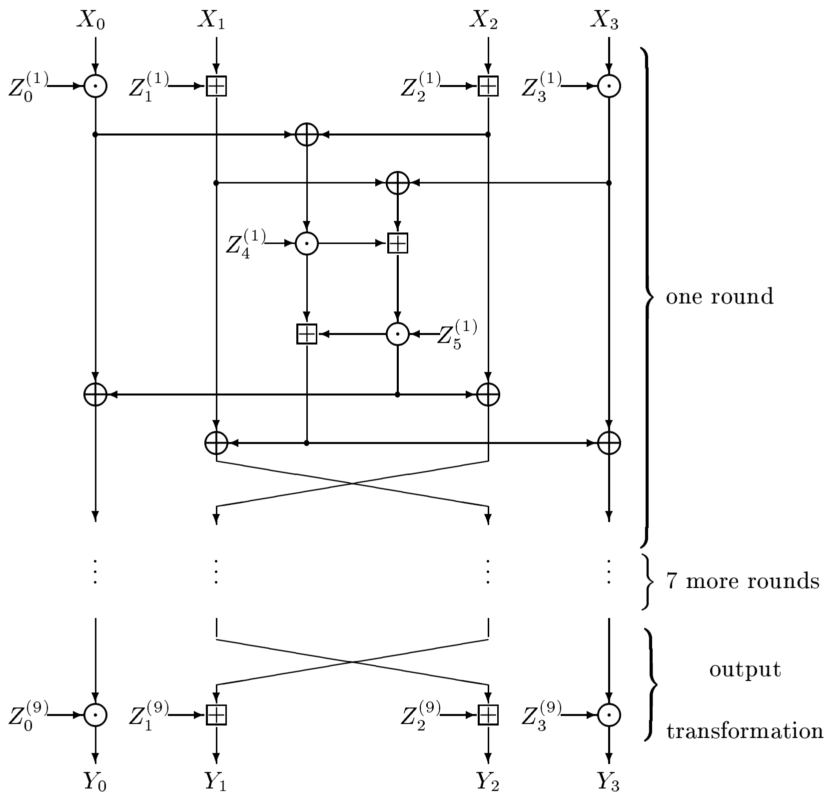
W. Meier cryptanalysed 2 rounds of IDEA in a differential-like attack using a partial distributive law [14]. J. Daemen found large classes of weak keys for IDEA [4] and also described an attack on 2.5 rounds of IDEA for all keys in [3].

Differential cryptanalysis was introduced by Biham and Shamir in [1]. In an attack on an iterated cipher one considers plaintext pairs P, P^* of a certain difference and the corresponding ciphertexts C and C^*. The main tool in the

---

* The work of the first author was done while visiting K.U. Leuven.

** F.W.O. research assistent, sponsored by Funds for Scientific Research-Flanders (Belgium)



X_i : 16-bit plaintext subblock
Y_i : 16-bit ciphertext subblock
$Z_i^{(r)}$: 16-bit key subblock
$\oplus$: bit-by-bit exclusive-OR of 16-bit subblocks
$\boxplus$: addition modulo 2^{16} of 16-bit integers
$\odot$: multiplication modulo $2^{16}+1$ of 16-bit integers
with the zero subblock corresponding to 2^{16}

Fig. 1. Computational graph for the encryption process of the IDEA cipher.

differential attack is the characteristic, a list of the expected differences in the ciphertexts after each round of the cipher. Lai and Massey introduced the notions of differentials in [11, 9]. Later in [6] Knudsen extended the notions of differentials to that of *truncated differentials*, where only subsets of the differences are predicted. A *right* pair is a pair of plaintexts, for which the ciphertext differences follow the differential. In a differential attack an attacker needs to get at least one right pair. However, an attacker might not be able to determine which pairs are right pairs from the differences in the ciphertexts, but if the characteristic or differential predicts also the differences in (parts of) the ciphertexts, often an attacker can discard pairs, which are not right pairs. A *wrong* pair is a pair of plaintexts, for which the differences in the ciphertexts do not follow the differential, but which looks like a right pair to the attacker.

In the linear attack [12] by Matsui one considers linear combinations of some bits of the plaintext, the ciphertext and the key, and defines linear characteristics. Nyberg introduced the *linear hull* [15], the analogue to differentials in differential attacks.. In [5] Hellman and Langford combined the differential and the linear attack to the *differential-linear* attack, and applied it to 8 rounds of the DES.

In this paper we give two new attacks on IDEA. In Section 2 the differential attack using truncated differentials is described, which can be used to break 3.5 rounds of IDEA. In Section 3 the differential-linear attack is described, which can be used to break 3 rounds of IDEA and Section 4 gives concluding remarks. Full versions of the attacks in this paper are described in [2, 8].

2 Truncated Differential Attack

In this section we describe a differential attack on 3.5 rounds of IDEA using truncated differentials. We define the difference of two bit strings A and A^* of the same length as

$$\Delta A = A \oplus A^*. \tag{1}$$

For differential cryptanalysis of IDEA with other definitions of difference, we refer to [11, 14]. Under the definition of difference (1) IDEA is not a Markov cipher [11]. Also, as we will see, the probabilities of the differentials used depend very much on the key used in the encryptions. Thus, the hypothesis of stochastic equivalence [11], i.e., that the average probability of a differential taken over all keys is approximately the same as the probability for a fixed key for virtually all keys, does not hold for IDEA with difference (1).

Consider the following one-round differential for IDEA.

$$(a, b, c, d) \stackrel{p_1}{\to} (e, f, g, h) \xrightarrow{(e\oplus g, f\oplus h) \stackrel{p_2}{\to} (k,l)} (e \oplus l, g \oplus l, f \oplus k, h \oplus k)$$

(a, b, c, d) denotes the four-word input difference and (e, f, g, h) denotes the difference after the key addition. This transition has probability p_1. With probability p_2 the input difference $(e \oplus g, f \oplus h)$ to the MA-structure leads to an output difference (k, l). The output difference of the round is given as $(e \oplus l, g \oplus l, f \oplus k, h \oplus k)$.

The 3-round truncated differential used in our attack on IDEA is:

$$
\begin{array}{lcl}
(A,0,B,0) \overset{2^{-16}}{\to} (C,0,C,0) & \xrightarrow{(0,0)\overset{1}{\to}(0,0)} & (C,C,0,0) \\
(C,C,0,0) \overset{1}{\to} (D,E,0,0) & \xrightarrow{(D,E)\overset{2^{-32}}{\to}(E,D)} & (0,D,0,E) \\
(0,D,0,E) \overset{2^{-16}}{\to} (0,F,0,F) & \xrightarrow{(0,0)\overset{1}{\to}(0,0)} & (0,0,F,F) \\
(0,0,F,F) \overset{1}{\to} (0,G,0,H) & &
\end{array}
$$

where the words A to H represent any values. The average probability of the truncated differential is 2^{-64}. This probability is computed over all choices of the inputs to a round and to the MA-structure and over all choices of the round keys and where we have also assumed that the MA-structure acts like a random function.

This differential has a mirror image with the same probability:

$$
\begin{array}{lcl}
(0,A,0,B) \overset{2^{-16}}{\to} (0,C,0,C) & \xrightarrow{(0,0)\overset{1}{\to}(0,0)} & (0,0,C,C) \\
(0,0,C,C) \overset{1}{\to} (0,0,D,E) & \xrightarrow{(D,E)\overset{2^{-32}}{\to}(E,D)} & (D,0,E,0) \\
(D,0,E,0) \overset{2^{-16}}{\to} (F,0,F,0) & \xrightarrow{(0,0)\overset{1}{\to}(0,0)} & (F,F,0,0) \\
(F,F,0,0) \overset{1}{\to} (G,0,H,0) & &
\end{array}
$$

These differentials are called truncated differentials, since we predict only two of the four words, the zeros, of the differences after each round.

We consider the attack also for reduced versions of IDEA, that operate on four nibbles, IDEA(16) and on four bytes IDEA(32), respectively, instead of four 16-bit words [9]. These reductions allow us to actually implement the attack and experimentally verify our results. The above differentials are defined similarly for the reduced versions. The average probabilities are 2^{-16} respectively 2^{-32}.

2.1 Description of the attack

First the attack on IDEA (full block length) is described. A structure of plaintexts consists of 2^{32} texts: p_1 and p_3 are fixed, p_0 and p_2 take on all possible values. We can use every combination of two texts as a pair. This means we generate $2^{32} \cdot (2^{32}-1)/2 \approx 2^{63}$ pairs from a structure. For every structure the expected number of right pairs is 0.5. The differential requires that Δc_0 and Δc_2 are equal to zero, and only such pairs are considered. On the average only one out of 2^{32} pairs will survive this test. For each surviving pair do the following: for all possible keys $Z_0^{(1)}, Z_2^{(1)}$ check whether

$$(p_0 \odot Z_0^{(1)}) \oplus (p_0^* \odot Z_0^{(1)}) = (p_2 \boxplus Z_2^{(1)}) \oplus (p_2^* \boxplus Z_2^{(1)}) \,. \tag{2}$$

On the average, this holds for 2^{16} values of $(Z_0^{(1)}, Z_2^{(1)})$. Similarly we check for which keys in the output transformation, it holds that

$$(c_1 \odot (Z_1^{(4)})^{-1}) \oplus (c_1^* \odot (Z_1^{(4)})^{-1}) = (c_3 \boxminus Z_3^{(4)}) \oplus (c_3^* \boxminus Z_3^{(4)}) \,. \tag{3}$$

Note that for a right pair these tests are successful for the correct value of the key. In total it can be expected that each pair suggests 2^{32} 64-bit key values and therefore every structure will suggest 2^{63} keys. Therefore every value of the key will be suggested 0.5 times per used structure and, as indicated above, every structure will suggest the correct value of the key 0.5 times. One might expect that among all the key values suggested by wrong pairs is also the correct value of the key. However, a wrong pair in the above attack will not suggest the correct value of the key. For a non-discarded pair of plaintexts and their ciphertexts a key will be suggested if the tests (2) and (3) succeed. For the correct value of the key this means that the input difference to the second round will be $(\tilde{C}, \tilde{C}, 0, 0)$. The output difference of the third round will be $(0, 0, \tilde{F}, \tilde{F})$, and the input difference of the third round will be $(0, \tilde{D}, 0, \tilde{E})$. Thus, the difference in the second round after the key addition will be $(D', E', 0, 0)$ and the output difference of the round is $(0, \tilde{D}, 0, \tilde{E})$. But this implies that $D' = \tilde{D}$ and $E' = \tilde{E}$, because of the structure of the round function of IDEA. It follows that if the correct value of the key is suggested for a pair of plaintexts, this must be a right pair. Summing up, for every structure in the attack there will be 0.5 right pairs, which suggest the correct value of the key, and 2^{31} wrong pairs, which on the average suggest a wrong value of the key 0.5 times. Thus, for the above attack the traditional method of Biham-Shamir [1] will not work, the S/N-ratio is 1, meaning that the correct value of the key cannot be distinguished from any other value of the key.

However, as we will see, the probability of the above differentials used in the attack depends very much on the secret key. For some keys the probability is less than the average probability and for other keys it is larger. We extend the key search method of a differential attack to the cases where the probability of the differential for the correct value of the secret key is different from the average probability over all keys. The bigger this difference the faster the attack. If the difference is big enough and if we assume that wrong values of the secret key is suggested randomly and uniformly by the attack, the correct value of the key will be found using sufficiently many plaintext pairs. This is a novel approach in differential attacks and reminiscent of the key search method in a linear attack [12].

For the actual attack, there is an overlap between the key bits we count on in the first round and the bits we count on in the last round. Furthermore, because of the absence of a carry bit after the highest order bit of the modular addition, we are unable to distinguish keys that differ only in these bits, so we will regard these two values of the key as one. These two observations are very important to reduce the memory requirements when we implement the attack. Using the first differential above 14 key bits overlap and two bits are indistinguishable for IDEA, which means that we would search for only 48 bit key values. For the reduced versions of IDEA we implemented key schedules, such that relatively as many key bits overlap. For IDEA(32) and IDEA(16) seven and three bits overlap, respectively. This means that in these cases we search for only 23 bit

#Keys/All keys	Probability
13%	0
12%	$0 < p \leq 2^{-18}$
21%	$2^{-18} < p \leq 2^{-17}$
30%	$2^{-17} < p \leq 2^{-16}$
14%	$2^{-16} < p \leq 2^{-15}$
10%	$2^{-15} < p \leq 1$

Table 1. Probability of the used differential for IDEA(16) with 3.5 rounds for classes of the secret key.

and 11 bit key values. To find other key bits a similar attack with the second differential above can be executed.

2.2 Experimental verification

We implemented the attack using the first differential on IDEA(16). First we calculated the probability of the differential for all keys by exhaustive search. Table 1 shows these probabilities for different classes of keys. The average probability over all keys was estimated to $2^{-16.5}$. The key-dependency of the probabilities stems mostly from the second round of the differential, where a difference (D, E) in the inputs to the MA-structure must result in difference (E, D) in the outputs of the MA-structure. Of most interest are the classes of keys that deviate most from the average probability. It is interesting to see that for about 1 in every 8 possible values of the secret key the probability of the used differential is zero. The numbers in Table 1 also indicate that the attack will not work for some classes of keys, namely the classes of keys for which the probabilities are too close to the average probability over all choices of the keys.

In Table 2 we list the results of 1000 implementations of our attack on IDEA(16) for increasing number of chosen plaintexts. We used key rankings as in [13] and tested whether the correct value of the key was among the eight least and eight most suggested values, thus the attack returns 16 suggestions for 11 bits of the secret key. As seen, using all plaintexts the correct value of the key is among those 16 values in about 67% of all cases. Note that there are a total of 2^{16} plaintexts of IDEA(16) and that an exhaustive search for the key will take the time of about 2^{32} encryptions.

Next we implemented attacks on IDEA(32). First we estimated the probabilities of the used differentials for different classes of keys. The result follows from Table 3. Based on the results of 160 experiments with random keys, we estimated the average probability over all keys to $2^{-32.7}$. Note that this is slightly less than first estimation made in the beginning of this section. This difference is caused by the fact that the MA-structure is not a random mapping. We implemented the attack for 100 different randomly chosen keys using up to 2048 structures. The results are given in Table 4. Using the above results on reduced versions of IDEA, we estimate the number of chosen plaintexts needed in our

#Keys/All keys	# Structures	# Chosen plaintexts
25%	16	2^{12}
40%	32	2^{13}
51%	64	2^{14}
59%	128	2^{15}
67%	256	2^{16}

Table 2. Average number of chosen plaintexts needed in the attack on IDEA(16) with 3.5 rounds in 1000 attacks.

#Keys/All keys	Probability
14%	$0 \leq p \leq 2^{-35}$
10%	$2^{-35} < p \leq 2^{-33.0}$
31%	$2^{-33.0} < p \leq 2^{-32.5}$
45%	$2^{-32.5} < p$

Table 3. Probability of the used differential for IDEA(32) for classes of the secret key.

attack on IDEA. From Table 2 it follows that one finds 25% and 51% of the keys using $2^{3n/4}$ respectively $2^{7n/8}$ chosen plaintexts for $n = 16$ for IDEA(16). From Table 4 it follows that one finds 1% and more than 83% of the keys using $2^{5n/8}$ respectively $2^{7n/8}$ chosen plaintexts for $n = 32$ for IDEA(32). As can be seen the number of keys we can recover increases for larger block sizes with relatively the same amount of data. We predict that a similar increase will occur for the attack on IDEA. Next we consider the workload and the amount of memory needed. One needs enough memory to store one structure. Once one structure has been analysed it is thrown away and a new structure analysed. Thus, the memory requirement for the attack on IDEA is 2^{32} words of each 64 bits. The workload is the estimated number of operations needed to perform the attack, measured as the number of encryptions of the cipher. The 2^{32} ciphertexts in a structure are hashed on the values of c_0 and c_2, since for a right pair the pairs of these values are equal. The workload of the hashing and storing of the ciphertexts is

#Keys/All keys	# Structures	# Chosen plaintexts
1%	16	2^{20}
7%	64	2^{22}
15%	128	2^{23}
31%	256	2^{24}
54%	512	2^{25}
65%	1024	2^{26}
83%	2048	2^{27}

Table 4. Average number of chosen plaintexts needed in the attack on IDEA(32) with 3.5 rounds in 100 attacks.

#Keys/All keys	# Structures	# Chosen plaintexts	Workload
>1%	2^8	2^{40}	2^{51}
>31%	2^{16}	2^{48}	2^{59}
>83%	2^{24}	2^{56}	2^{67}

Table 5. Estimated number of chosen plaintexts needed in the attack on IDEA with 3.5 rounds with 2^{32} words of memory.

small compared to the time of the rest of the attack. For each pair that survives the filtering process we try all possible 2^{16} values of the affected keys of each side of Eq. (2). These tests can be sped up by pre-calculating a table to avoid the expensive multiplication operation. This table would be of size 2^{32} 16-bit words. We estimate that a multiplication takes the equivalent of 3.5 additions, and that an addition, an exclusive-or and a table-lookup take about the same time. The workload is about 2^{12} encryptions of IDEA with 3.5 rounds for every analysed pair. Totally, the workload is about 2^{43} encryptions for every structure. Because of the overlap of key bits in this first round test with the key bits in the output transformation, the second part of the key search, i.e. using Equation (3), is much faster than the first and can be ignored in the workload estimation.

The estimated number of chosen plaintexts and the workload for our attack on IDEA is given in Table 5. Note that an exhaustive search for the key of IDEA takes the time of about 2^{128} encryptions of IDEA. Finally we discuss how to find additional key bits. The attack outlined above finds 48 bits of the 128 bit key of IDEA. However, once these key bits have been found, one can do a similar attack using the second truncated differential. As noted earlier the key-dependency of the probability of the first differential comes mostly from the second round of the differentials. Since the second round is the same for the two differentials, one can expect that for a fixed key the probabilities of the two differentials are very close. After doing the attack with the second differential one has all 64 key bits in the beginning of the first round and all 64 key bits of the output transformation. Subsequently, one can do similar attacks on a further reduced version of IDEA.

3 A Differential-linear Attack

In this section we give a differential-linear attack on IDEA reduced to 3 rounds. We will use the notation $P = (p_0, p_1, p_2, p_3)$, $C = (c_0, c_1, c_2, c_3)$ to describe plaintexts, ciphertexts and their 16-bit subblocks. The version we look at is 3 rounds without the output transformation and where we omit the swapping of the second and third ciphertext blocks. We will write $A[i]$ to indicate the i^{th} bit of A, where $A[0]$ is the least significant bit (LSB) of A and $A[15]$ denotes the most significant bit (MSB) for a 16-bit word A. These indices will be omitted whenever the context makes it clear which bit(s) we are considering. With $A[i, \dots, j]$ we will indicate the row of bits $A[i] \dots A[j]$. Also, we define some special 16-bit symbols μ_i for $i = 0, \dots, 15$, where $\mu_i[i] = 1$ and $\mu_i[j] = 0$ for $j \neq i$.

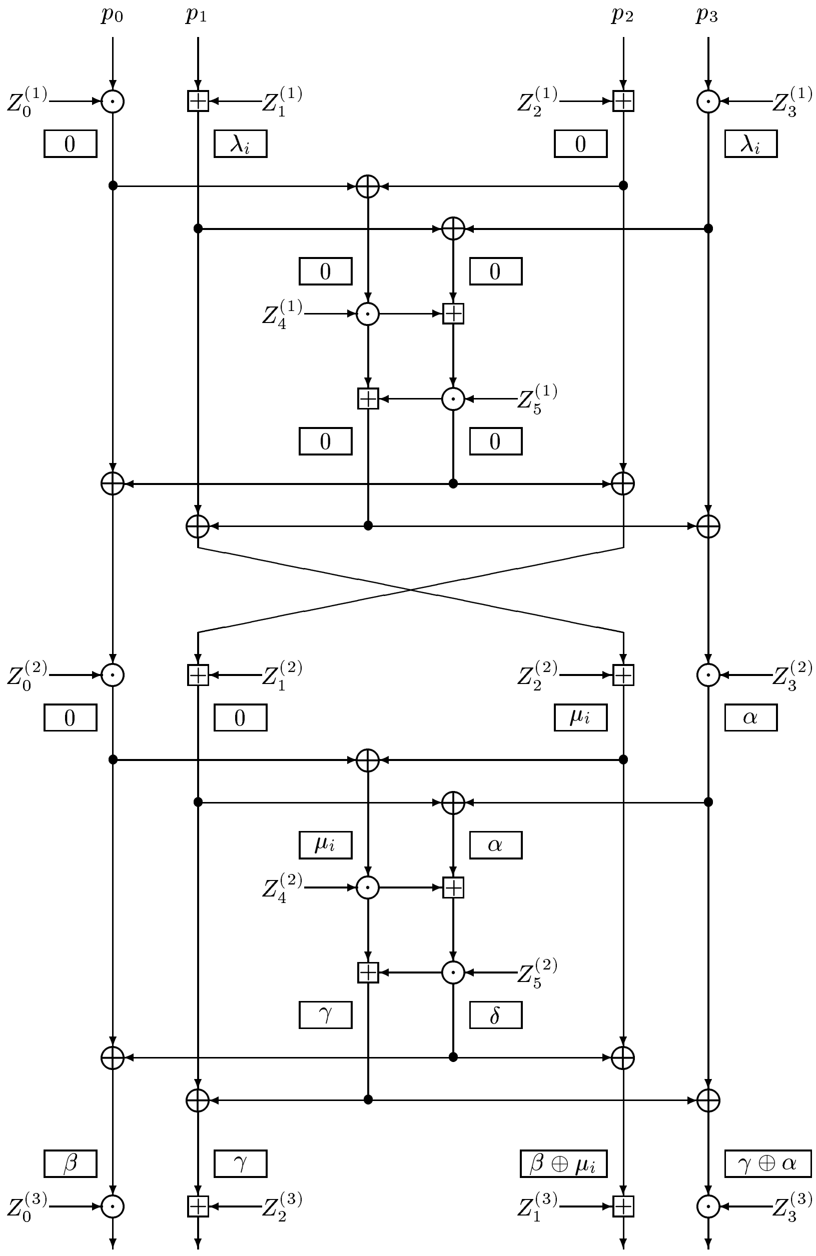


Fig. 2. Two rounds of IDEA.

3.1 Choosing plaintexts

Consider the two rounds of IDEA in Figure 2. The inserted boxes give the expected values of the differentials used in our attack.

We guess the value of $Z_3^{(1)}$. We encrypt a set of plaintexts (p_0, p_1, p_2, p_3), where p_0 and p_2 are fixed. With $\Delta p_1 = \mu_i$ one gets $\lambda_i = \mu_i$ with probability 0.5 (see e.g. [16, 7]), and similarly with probability 0.5 one gets the difference μ_i in outputs of the second addition in the second round, as indicated in Figure 2, thus this part of the differential has probability 1/4. A closer analysis shows that one can pick six plaintext pairs such that this part of the differential holds at least once. Details are given in the full paper [2]. The values of p_3 are chosen such that for the pairs we are going to analyse

$$(Z_3^{(1)} \odot p_3^*) \oplus (Z_3^{(1)} \odot p_3) = (p_1^* \boxplus Z_1^{(1)}) \oplus (p_1 \boxplus Z_1^{(1)}) = \lambda_i .$$

This ensures that the input difference of the MA-structure in the first round is zero. For one of the six pairs the difference after the key addition of the second round will be

$$(0, 0, \mu_i, \alpha). \tag{4}$$

3.2 Sets of linear relations

We concentrate on the first multiplication in the MA-structure of the second round and denote the input with $p^{(2)}$ and the output with $r^{(2)}$. Then

$$\Delta r^{(2)} = (Z_4^{(2)} \odot p^{(2)}) \oplus (Z_4^{(2)} \odot (p^{(2)} \oplus \mu_i)) .$$

We observed that for every choice of $Z_4^{(2)}$ there are several possible values for μ_i such that

$$\Delta r^{(2)}[0] = 0 \tag{5}$$

with a probability p, such that the bias $|p - 1/2| > 0.166$ over all $p^{(2)}$. Furthermore we observed that for all but 26 of the 2^{16} possible values of $Z_4^{(2)}$ there is at least one μ_i for which the bias is larger then 1/4.

We are going to use this in a linear attack. Instead of having one relation that holds with an average probability for each key, we are going to use a set of relations. For each key at least one of the relations has a large bias. This idea is central to our attack.

3.3 Propagation

¿From now on, we only consider the least significant bits of the various 16-bit intermediate results. For these bits the modular addition reduces to an exclusive-or. Denote by $t^{(2)}$ and $t^{(3)}$ the outputs of the second multiplication in the MA-structure of the second and third round, respectively. Using (4) we get for the difference after the second round

$$(\Delta t^{(2)}, \mu_i \oplus \Delta t^{(2)}, \Delta r^{(2)} \oplus \Delta t^{(2)}, \alpha \oplus \Delta r^{(2)} \oplus \Delta t^{(2)}).$$

Because the ciphertext (c_0, c_1, c_2, c_3) equals the output of the third round, we can calculate

$$\begin{aligned}
\Delta t^{(3)} &= \Delta c_2 \oplus \Delta r^{(2)} \oplus \Delta t^{(2)} \\
\Delta r^{(3)} &= \Delta t^{(3)} \oplus \Delta c_1 \oplus \mu_i \oplus \Delta t^{(2)} = \Delta c_1 \oplus \Delta c_2 \oplus \mu_i \oplus \Delta r^{(2)},
\end{aligned}$$

where $r^{(3)}$ is defined in a similar way as $r^{(2)}$. In other words, we are able to predict the least significant bit of the output difference of the first multiplication of the MA-structure of the last round. The inputs of this multiplication are the subkey $Z_4^{(3)}$ and an intermediate result that equals $c_0 \oplus c_2$. For every ciphertext pair we can calculate $c_0 \oplus c_2$ and predict $\Delta r^{(3)}$ with a high probability. We keep a counter for every possible value of $Z_4^{(3)}$ and increment the counters of the key values that are compatible with the calculated $c_0 \oplus c_2$ and $\Delta r^{(3)}$.

Note that we don't know for which μ_i (5) holds with large probability. Therefore we have to repeat the attack for different values of μ_i. Also we guessed the value of $Z_3^{(1)}$. Our experiments suggest that for wrong guesses of $Z_3^{(1)}$ the algorithm fails to suggest a specific value for $Z_4^{(3)}$. Thus we can recognize wrong guesses. With this algorithm it is impossible to distinguish between the correct subkey values and their additive inverses modulo $2^{16} + 1$.

When $Z_3^{(1)}$ is guessed correctly, tests have shown that we need at most $9000 < 2^{14}$ pairs to determine $Z_4^{(3)}$. On the average we guess correctly after 2^{15} trials, therefore we need about 2^{29} plaintext pairs. Examining one plaintext pair takes a few exclusive-or operations and 2^{16} table look-ups, one for each value of $Z_4^{(3)}$. Since we examine 16 differentials, our attack needs totally about 2^{20} simple operations, i.e., addition or exclusive-or, for each pair and the total workload is therefore about 2^{49} simple operations. Using the estimate of Section 2 that an exclusive-or takes the same time as an addition and a multiplication takes 3.5 times as much time as either of them, the workload is about equal to $0.75 \cdot 2^{44}$ encryptions with 3 rounds of IDEA.

3.4 Finding additional key bits

In this paragraph we will describe how to find the subkeys $Z_0^{(3)}$ and $Z_5^{(3)}$ (or their additive inverses modulo $2^{16} + 1$). For this a method will be used similar to the main one described in [3]. First we will give a definition of compatibility.

Definition 1. A word A is said to be compatible with B modulo N if there exists a pair of words C,C^* with $C \oplus C^* = A$ and $C - C^* \pmod N = B$.

It is easy to see that a word A is compatible to at most 2^k words modulo N, where k is the Hamming weight of A. The probability that a randomly chosen word with Hamming weight k and another one are compatible modulo 2^{16} is therefore smaller or equal to 2^{k-16}.

For this part of the attack we will consider only the plaintext pairs that we already constructed with the correct guess for $Z_3^{(1)}$ (or its additive inverse modulo $2^{16}+1$) that yield μ_{15} after the key addition of the second round. The difference after the second round is $(\beta, \beta \oplus \mu_i, \gamma, \gamma \oplus \alpha)$, see Figure 2. Like α, γ and δ the difference β is unknown. However, since $Z_4^{(3)}$ is known (or $2^{16}+1-Z_4^{(3)}$), when also $Z_5^{(3)}$ and $Z_0^{(3)}$ would be known, we would be able to calculate β for each pair and the intermediate values $(q_0^{(3)}, q_1^{(3)}, q_2^{(3)}, q_3^{(3)})$ before the MA-structure of the last round. Then $\beta \oplus \mu_{15}$ must be compatible modulo 2^{16} with $(q_1^{(3)} \boxplus Z_1^{(3)}) \boxminus (q_1^{(3)*} \boxplus Z_1^{(3)}) = q_1^{(3)} \boxminus q_1^{(3)*}$. To find $Z_0^{(3)}$ and $Z_5^{(3)}$ we simply guess their values and for each guess check this compatibility requirement. It can be shown [3] that the expected number of pairs needed to eliminate a wrong guess for a pair $Z_0^{(3)}$,$Z_5^{(3)}$ is approximately equal to 1 divided by the probability that a random 16-bit word is compatible modulo 2^{16} to another one. Tests have shown that this number is between 1 and 5.

As in the previous section this search method doesn't make a distinction between $Z_0^{(3)}$,$Z_5^{(3)}$ and their additive inverses modulo $2^{16}+1$. It takes two multiplications with $Z_5^{(3)}$ and $Z_4^{(3)}$ to find $\Delta q_0^{(3)}$ and $\Delta q_1^{(3)}$, but as $Z_4^{(3)}$ is fixed, multiplications with this key are many times the same. Then it takes one multiplication with $(Z_0^{(3)})^{-1}$ to find β. So finding $Z_0^{(3)}$ and $Z_5^{(3)}$ takes at most 2^{33} multiplications modulo $2^{16}+1$. According to the estimates earlier made this is about equal to $1.5 \cdot 2^{29}$ encryptions with 3 rounds of IDEA.

Finally, one can find the remaining key bits by doing additional attacks using similar characteristics as the above. The attack will have a better performance, since many key bits are already known.

4 Conclusions

We have presented two attacks on IDEA with a reduced number of rounds. The first attack finds the secret key of 3.5 rounds of IDEA in more than 86% of all cases using an estimated number of 2^{56} chosen plaintexts and a workload of about 2^{67} encryptions of 3.5 rounds of IDEA. With 2^{40} chosen plaintexts the attack works for 1% of all keys. The second attack finds the secret key of 3 rounds of IDEA. It needs at most 2^{29} chosen pairs of plaintext and a workload of about 2^{44} encryptions with 3 rounds of IDEA.

Although our attacks make use of some sophisticated techniques, the efficiencies, in particular the workloads, of the algorithms probably can be improved greatly. Further we think that similar attacks can be successful against more rounds of IDEA, but it is questionable if in this way anything substantial can be achieved against the full 8.5-rounds version of IDEA.

References

1. E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer Verlag, 1993.
2. J. Borst. *Differential-Linear Cryptanalysis of IDEA.* Technical Report ESAT-COSIC Report 96-2, Department of Electrical Engineering, Katholieke Universiteit Leuven, Febr. 1997.

3. J. Daemen, R. Govaerts, and J. Vandewalle. Cryptanalysis of 2,5 rounds of IDEA. Technical Report ESAT-COSIC Report 94-1, Department of Electrical Engineering, Katholieke Universiteit Leuven, March 1994.
4. J. Daemen, R. Govaerts, and J. Vandewalle. Weak keys for IDEA. In T. Helleseth, editor, *Advances in Cryptology - Proc. Eurocrypt'93, LNCS 773*, pages 224–231. Springer Verlag, 1994.
5. M.E. Hellman and S. K. Langford. Differential–linear cryptanalysis. In Y. G. Desmedt, editor, *Advances in Cryptology - Proc. Crypto'94, LNCS 839*, pages 26–39. Springer Verlag, 1994.
6. L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 196–211. Springer Verlag, 1995.
7. L.R. Knudsen and W. Meier. Improved differential attack on RC5. In Neal Koblitz, editor, *Advances in Cryptology - Proc. Crypto'96, LNCS 1109*, pages 216–228. Springer Verlag, 1996.
8. L.R. Knudsen and V. Rijmen. *Truncated Differentials of IDEA*. Technical Report ESAT-COSIC Report 97-1, Department of Electrical Engineering, Katholieke Universiteit Leuven, Febr. 1997.
9. X. Lai. *On the Design and Security of Block Ciphers.* PhD thesis, ETH, Zürich, Switzerland, 1992.
10. X. Lai and J.L. Massey. A proposal for a new block encryption standard. In I.B. Damgård, editor, *Advances in Cryptology - Proc. Eurocrypt'90, LNCS 473*, pages 389–404. Springer Verlag, 1991.
11. X. Lai, J.L. Massey, and S. Murphy. Markov ciphers and differential cryptanalysis. In D.W. Davies, editor, *Advances in Cryptology - Proc. Eurocrypt'91, LNCS 547*, pages 17–38. Springer Verlag, 1992.
12. M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - Proc. Eurocrypt'93, LNCS 765*, pages 386–397. Springer Verlag, 1993.
13. M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y. G. Desmedt, editor, *Advances in Cryptology - Proc. Crypto'94, LNCS 839*, pages 1–11. Springer Verlag, 1994.
14. W. Meier. On the security of the IDEA block cipher. In T. Helleseth, editor, *Advances in Cryptology - Eurocrypt'93, LNCS 765*, pages 371–385. Springer Verlag, 1993.
15. K. Nyberg. Linear approximations of block ciphers. In A. De Santis, editor, *Advances in Cryptology - Proc. Eurocrypt'94, LNCS 950*, pages 439–444. Springer Verlag, 1994.
16. R.A. Rueppel. *Analysis and Design of Stream Ciphers.* Springer Verlag, 1986.