New Attacks against Reduced-Round Versions of IDEA

Pascal Junod

École Polytechnique Fédérale de Lausanne (Switzerland)
pascal@junod.info

Abstract. In this paper, we describe a sequence of simple, yet efficient chosen-plaintext (or chosen-ciphertext) attacks against reduced-round versions of IDEA (with 2, 2.5, 3, 3.5, and 4 rounds) which compare favourably with the best known attacks: some of them decrease considerably the time complexity given the same order of data at disposal while other ones decrease the amount of necessary known- or chosen-plaintext pairs under comparable time complexities. Additionally, we show how to trade time and memory for some of the known-plaintext attacks of Nakahara *et al.*

Key words: Block ciphers, IDEA, Demirci-Biryukov relation.

1 Introduction

Although the IDEA block cipher [10–12] is one of the oldest proposals of alternative to DES [18], it has withstood all kinds of cryptanalytical attacks surprinsingly well until now. Its strength is certainly due to an elegant and simple design approach which consists in mixing three algebraically incompatible group operations, namely the addition of vectors over $\mathrm{GF}(2)^{16}$, denoted "$\oplus$", the addition of integers over $\mathbb{Z}_{2^{16}}$, denoted "$\boxplus$", and the multiplication in $\mathrm{GF}\left(2^{16}+1\right)^*$, denoted "$\odot$". Despite the popularity of IDEA (due surely to the fact that it was chosen as the block cipher in the first versions of the software *Pretty Good Privacy (PGP)* [7] by Zimmerman), its cryptanalysis process has been a rather lengthy process. To the best of our knowldege, Meier [14] was the first to propose an attack based on differential cryptanalysis against up to 2.5 rounds running faster than an exhaustive search. Then, Borst *et al.* [3] presented a differential-linear attack against 3 rounds and a truncated differential attack on 3.5 rounds; Biham *et al.* [1] managed to break 4.5 rounds using impossible differentials. Motivated by a paper of Nakahara *et al.* [15] explaining how to break 2.5 rounds using an integral attack, Demirci [5] was able to break up to 4 rounds; one year later, these results were extended [6] using meet-in-the-middle techniques to break up to 5 rounds slightly faster than an exhaustive search. Very recently, Nakahara *et al.* [16] devised known-plaintext attacks against reduced-round versions of IDEA using ideas of Demirci as well as an (unpublished) observation of Biryukov. Other papers [2, 4, 8] present attacks against the full version of IDEA, but these attacks fortunately work only for a negligible fraction of the keys.

Contributions of this paper: Inspired by some of the ideas in the paper of Nakahara *et al.* [16], we describe a sequence of new attacks against reduced-round versions of IDEA, up to 4 rounds; these attacks are mainly based on the *Biryukov-Demirci relation*. Some of them, given a comparable computational complexity, reduce considerably the amount of necessary chosen plaintexts, while other attacks, given a comparable amount of chosen plaintexts, decrease favourably the computational complexity; additionaly, we show how to trade time and memory for some of the known-plaintext attacks of Nakahara *et al.* Furthermore, we explain how to combine some of these attacks with other known attacks, which allows in some cases to gain more key bits with a lesser complexity, or to avoid the use of both encryption and decryption oracles. This paper is organized as follows: we recall briefly in §2 the inner details of IDEA, and the attacks are described in §3. Finally, we compare our results to the best known attacks in §4.

2 The IDEA block cipher

IDEA encrypts 64-bit data blocks under a 128-bit key; it consists of eight identical rounds and a final half-round (a key addition layer similar to those in a full round). Figure 1 illustrates the computational flow of one round. Round r transforms a 64-bit input represented as a vector of four 16-bit words to an output vector of the same size: $(X_1^{(r)}, X_2^{(r)}, X_3^{(r)}, X_4^{(r)}) \mapsto (Y_1^{(r)}, Y_2^{(r)}, Y_3^{(r)}, Y_4^{(r)})$. This process is parametered by six 16-bit subkeys denoted $Z_i^{(r)}$, with $1 \leq i \leq 6$, which are derived from the master 128-bit key by means of the key-schedule algorithm. One evaluates the three IDEA algebraic operations as follows: $\oplus$ is a simple exclusive-or operation, $\boxplus$ is the addition modulo 2^{16} and $\odot$ is the common multiplication modulo $2^{16}+1$ (where 0 is considered as the number 2^{16}). First, two intermediate values $\alpha^{(r)}$ and $\beta^{(r)}$ are computed:

$$\alpha^{(r)} = \left(X_1^{(r)} \odot Z_1^{(r)}\right) \oplus \left(X_3^{(r)} \boxplus Z_3^{(r)}\right)$$
$$\beta^{(r)} = \left(X_2^{(r)} \boxplus Z_2^{(r)}\right) \oplus \left(X_4^{(r)} \odot Z_4^{(r)}\right)$$

These two values form the input of the *multiplication-addition box (MA-box)* which provides two 16-bit outputs $\gamma^{(r)}$ and $\delta^{(r)}$:

$$\delta^{(r)} = \left(\left(\alpha^{(r)} \odot Z_5^{(r)}\right) \boxplus \beta^{(r)}\right) \odot Z_6^{(r)}$$
$$\gamma^{(r)} = \left(\alpha^{(r)} \odot Z_5^{(r)}\right) \boxplus \delta^{(r)}$$

Finally, the output of the round r is given by

$$Y_1^{(r)} = \left(X_1^{(r)} \odot Z_1^{(r)}\right) \oplus \delta^{(r)}\ , \quad Y_2^{(r)} = \left(X_2^{(r)} \boxplus Z_2^{(r)}\right) \oplus \gamma^{(r)}$$
$$Y_3^{(r)} = \left(X_3^{(r)} \boxplus Z_3^{(r)}\right) \oplus \delta^{(r)}\ , \quad Y_4^{(r)} = \left(X_4^{(r)} \odot Z_4^{(r)}\right) \oplus \gamma^{(r)}$$

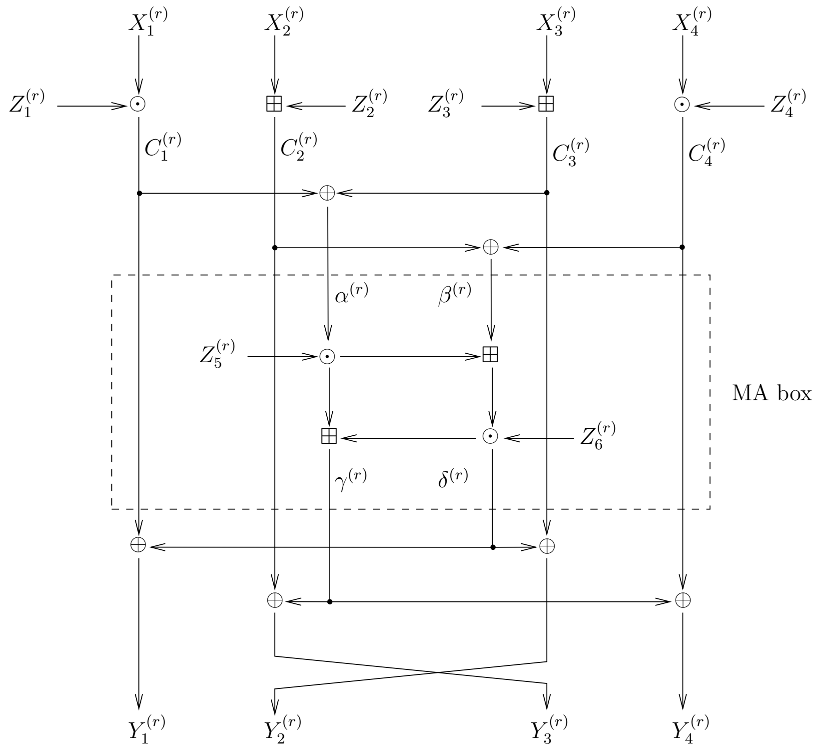

Fig. 1. Round r of IDEA

A *half-round* is defined to be the key-addition layer; we denote its output $(C_1^{(r)}, C_2^{(r)}, C_3^{(r)}, C_4^{(r)})$. The key-schedule of IDEA allows to derive fifty-two 16-bit subkeys out of the 128-bit key Z. Its description is straightforward; first, order the subkeys as

$$Z_1^{(1)}, \ldots, Z_6^{(1)}, Z_1^{(2)}, \ldots, Z_6^{(2)}, \ldots, Z_1^{(9)}, \ldots, Z_4^{(9)}$$

partition Z into eight 16-bit blocks, and assign these blocks directly to the first eight subkeys. Then, do the following until all remaining subkeys are assigned: rotate Z left 25 bits, partition the result, and assign these blocks to the next eight subkeys. In Figure 2, we give explicitely the value of the subkeys (where $Z_{[0\ldots15]}$ means the bits 0 to 15 (inclusive) of Z, $Z_{[117\ldots4]}$ means the bits 117-127 and 0-4 of Z, and where the leftmost bit of Z is numbered with 0).

3 Description of the Attacks

In this section, we describe new attacks breaking 2-rounds, 2.5-rounds, 3-rounds, 3.5-rounds, and 4-rounds IDEA, and we compute their complexity. But first of

Round r	$Z_1^{(r)}$	$Z_2^{(r)}$	$Z_3^{(r)}$	$Z_4^{(r)}$	$Z_5^{(r)}$	$Z_6^{(r)}$
1	$Z_{[0...15]}$	$Z_{[16...31]}$	$Z_{[32...47]}$	$Z_{[48...63]}$	$Z_{[64...79]}$	$Z_{[80...95]}$
2	$Z_{[96...111]}$	$Z_{[112...127]}$	$Z_{[25...40]}$	$Z_{[41...56]}$	$Z_{[57...72]}$	$Z_{[73...88]}$
3	$Z_{[89...104]}$	$Z_{[105...120]}$	$Z_{[121...8]}$	$Z_{[9...24]}$	$Z_{[50...65]}$	$Z_{[66...81]}$
4	$Z_{[82...97]}$	$Z_{[98...113]}$	$Z_{[114...1]}$	$Z_{[2...17]}$	$Z_{[18...33]}$	$Z_{[34...49]}$
5	$Z_{[75...90]}$	$Z_{[91...106]}$	$Z_{[107...122]}$	$Z_{[123...10]}$	$Z_{[11...26]}$	$Z_{[27...42]}$
6	$Z_{[43...58]}$	$Z_{[59...74]}$	$Z_{[100...115]}$	$Z_{[116...3]}$	$Z_{[4...19]}$	$Z_{[20...35]}$
7	$Z_{[36...51]}$	$Z_{[52...67]}$	$Z_{[68...83]}$	$Z_{[84...99]}$	$Z_{[125...12]}$	$Z_{[13...28]}$
8	$Z_{[29...44]}$	$Z_{[45...60]}$	$Z_{[61...76]}$	$Z_{[77...92]}$	$Z_{[93...108]}$	$Z_{[109...124]}$
8.5	$Z_{[22...37]}$	$Z_{[38...53]}$	$Z_{[54...69]}$	$Z_{[70...85]}$		

Fig. 2. Complete Key-Schedule of IDEA

all, we recall what is the Biryukov-Demirci relation, as it builds the core of our distinguishers.

3.1 The Biryukov-Demirci Relation

A crucial observation on which our attacks is based is that there exists a linear-like expression holding with probability one on *any* number of rounds. Nakahara *et al.* [17] name it *Biryukov-Demirci relation.* It is actually a combination of two facts, one of these being the following observation by Demirci [5].

Lemma 1 (Demirci [5]). *For any round number r of the* IDEA *block cipher,*

$$\operatorname{lsb}\left(\gamma^{(r)} \oplus \delta^{(r)}\right) = \operatorname{lsb}\left(\alpha^{(r)} \odot Z_5^{(r)}\right) \tag{1}$$

where $\operatorname{lsb}(a)$ *denotes the least significant (rightmost) bit of a.*

Using this theorem, one can easily set up a distinguisher using a few known triplets $(\alpha^{(r)}, \gamma^{(r)}, \delta^{(r)})$ which works as follows: for each possible value of $Z_5^{(r)}$, check whether Eq. (1) hold for the known triplets; this allows to sieve wrong values of $Z_5^{(r)}$ from the right one. Actually, one gets *two* candidates for $Z_5^{(r)}$, as observed by Demirci: if $Z_5^{(r)} \notin \{0, 1\}$, this distinguisher eliminates all keys except the correct one and a "conjugate" $2^{16} + 1 - Z_5^{(r)}$. Otherwise, it eliminates all keys except 0 and 1.

The second (unpublished) observation[1] states that the two middle words in a block of data are only combined either with subkeys or with internal cipher data, via group operations (namely $\oplus$ and $\boxplus$) which are GF(2)-linear when considering their least significant (rightmost) bit; this fact is valid across the full cipher (and is actually independent of the number of rounds). Combining this observation and Lemma 1, one easily obtain the *Biryukov-Demirci relation.*

[1] According to [17], this observation is credited to Biryukov.

Theorem 1 (Biryukov-Demirci relation). *For any number of rounds n in the* IDEA *block cipher, the following expression is true with probability one:*

$$\mathrm{lsb}\left(\bigoplus_{i=1}^{n}\left(\gamma^{(i)} \oplus \delta^{(i)}\right) \oplus X_2^{(1)} \oplus X_3^{(1)} \oplus Y_2^{(n+1)} \oplus Y_3^{(n+1)}\right) = \mathrm{lsb}\left(\bigoplus_{j=1}^{n}\left(Z_2^{(j)} \oplus Z_3^{(j)}\right)\right)$$

Note that Theorem 1 can easily be extended when a final half-round (key-addition layer) is present by adding the two relevant key bits.

3.2 Retrieving All Key Bits for 1.5 Rounds

The simplest attack described in [17] is built on top of the following expression holding with probability one; it is a straightforward application of Theorem 1 to 1.5-rounds IDEA.

$$\begin{aligned}\mathrm{lsb}\Big(X_2^{(1)} \oplus X_3^{(1)} \oplus C_2^{(2)} \oplus C_3^{(2)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus \\ Z_5^{(1)} \odot \left(\left(X_1^{(1)} \odot Z_1^{(1)}\right) \oplus \left(X_3^{(1)} \boxplus Z_3^{(1)}\right)\right)\Big) = 0 \qquad (2)\end{aligned}$$

By taking into account the key-schedule algorithm and guessing key bits numbered (see Figure 2) 0-15, 32-47, 64-79, which represent 48 unknown key bits, one can recover these right key bits and $\mathrm{lsb}\left(Z_2^{(1)} \oplus Z_2^{(2)}\right)$ with probability larger than 0.99 in roughly $3 \cdot \frac{12}{30} \cdot 2^{48} \approx 2^{48.26}$ 1.5-rounds IDEA evaluations if 55 known plaintext-ciphertext pairs are available using Alg. 1. The complexity of this at-

---

Algorithm 1 Attack breaking 1.5-round IDEA

1: **Input**: An oracle Ω implementing encryption by 1.5-rounds IDEA under a fixed, unknown key.
2: Query the ciphertexts corresponding to 55 different, uniformly distributed plaintexts P_i to Ω.
3: **for** all possible subkey candidates $(Z_1^{(1)}, Z_3^{(1)}, Z_5^{(1)})$ **do**
4: Check whether the expression

$$\begin{aligned}\mathrm{lsb}\Big(X_2^{(1)} \oplus X_3^{(1)} \oplus C_2^{(2)} \oplus C_3^{(2)} \oplus \\ Z_5^{(1)} \odot \left(\left(X_1^{(1)} \odot Z_1^{(1)}\right) \oplus \left(X_3^{(1)} \boxplus Z_3^{(1)}\right)\right)\Big) \qquad (3)\end{aligned}$$

gives the same bit for the two first pairs. If yes, take sequentially other pairs as long as Eq. (3) evaluates to a constant. If it holds for all 55 pairs, output "Key candidate".
5: **end for**

---

tack can be evaluated as follows: for each key candidate, one needs to evaluate

Eq. (3) at least two times, three times with probability $\frac{1}{4}$, four times with probability $\frac{1}{8}$, and so on, which results in an average of three evaluations of Eq. (3); as in [17], we assume furthermore that a $\odot$ operation is equivalent to three $\oplus$ (or three $\boxplus$) operations: thus, one evaluation of Eq. (3) costs 12 simple operations while a full evaluation of 1.5-round IDEA costs 30 simple operations. Note that we may have adopted the strategy of [17], which consists in guessing $\text{lsb}\left(Z_2^{(1)} \oplus Z_2^{(2)}\right)$ as well and evaluating Eq. (2). In this case, one would need one pair of known plaintext-ciphertext more to ensure the same success probability, and the complexity would have been equal to $2 \cdot \frac{14}{30} \cdot 2^{49} \approx 2^{48.90}$, which is slightly worse.

We observe that one can actually apply a common trick[2] to the Biryukov-Demirci relation and thus extend Nakahara *et al.* attack: we can apply the relation in *two directions*, namely in the encryption or in the decryption direction. When applied to the decryption direction, the distinguisher Eq. (2) becomes

$$\begin{aligned}\text{lsb}\Big(X_2^{(1)} \oplus X_3^{(1)} \oplus C_2^{(2)} \oplus C_3^{(2)} \oplus Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus \\ Z_5^{(1)} \odot \left(\left(C_1^{(2)} \odot Z_1^{(2)}\right) \oplus \left(C_2^{(2)} \boxplus Z_2^{(2)}\right)\right)\Big) = 0 \qquad (4)\end{aligned}$$

Although it would not be more interesting to use Eq. (4) alone as distinguisher (since one should guess the same number of unknown key bits), one can use it after Eq. (2) to recover *all key bits* using roughly the same amount of computational effort. More precisely, once the key bits 0-15, 32-47, 64-79 are known, which actually fix $Z_1^{(1)}$ and $Z_5^{(1)}$, one can recover $Z_1^{(2)}$ and $Z_2^{(2)}$ (key bits numbered 96-127) in a $3 \cdot \frac{12}{30} \cdot 2^{32} \approx 2^{32.26}$ effort, derive the key bit 31, and search exhaustively for the remaining 47 unknown key bits. The overall complexity of this attack is approximately equal to $2^{48.26} + 2^{47} + 2^{32.26} \approx 2^{48.76}$ 1.5-round IDEA evaluations.

3.3 A New Chosen-Plaintext Attack Breaking 2 Rounds

Let us consider the relation Eq. (2) on 2 rounds, and let us fix $X_1^{(1)}$ and $X_3^{(1)}$ to arbitrary constants[3]. Our attack proceeds as follows and assumes that the adversary is able to encrypt about 62 chosen plaintexts: as first step, encrypt 23 chosen plaintexts with fixed $X_1^{(1)}$ and $X_3^{(1)}$, and guess $Z_5^{(2)}$. In a second step, guess $Z_6^{(2)}$ and test Eq. (2) on the partially decrypted ciphertext, and determine these unknown key bits with help of Eq. (2) by eliminating the candidates which do not render this expression constant, since the expression

$$\text{lsb}\left(Z_5^{(1)} \odot \left(\left(X_1^{(1)} \odot Z_1^{(1)}\right) \oplus \left(X_3^{(1)} \boxplus Z_3^{(1)}\right)\right)\right)$$

[2] This trick was proposed for the first time, as far as this author knows, by Matsui [13] in the linear cryptanalysis of DES.

[3] A similar technique was used by Knudsen and Mathiassen [9] to speed up by a small constant a linear cryptanalysis of DES.

provides an unknown, but *constant* bit to the cryptanalyst. This process gives us 4 candidates for the key bits 57-88 within a complexity of less than 2^{20} 2-rounds IDEA evaluations.

Once this process is achieved, one can use the attacks described in §3.2 to derive key bits 0-15 and 32-47 in a 2^{33} effort and key bits 96-127 in another 2^{33} effort with 39 additional chosen-plaintext. Hence, this attacks recovers all key bits (the 31 remaining ones with help of an exhaustive search) in a computational complexity approximately equal to 2^{34} 2-rounds IDEA evaluations. Thus, this attack compares quite favorably with Demirci's square-like attack [5] which requires roughly the same order of chosen-plaintexts and a 2^{64} computational effort to recover the whole key.

If a decryption oracle is available, instead of an encryption one, we can still mount a *chosen-ciphertext* attack based on the same properties. It would work as follows: fix $Y_1^{(2)}$ and $Y_3^{(2)}$ to an arbitrary constant, and guess $Z_1^{(1)}$, $Z_3^{(1)}$, and $Z_5^{(1)}$ (which represent 48 unknown key bits numbered 0-15, 32-47, and 64-79). Once these 48 bits recovered, after a 2^{48} process (provided 55 chosen plaintexts are available), one can recover 16 more bits (i.e. the still unknown bits of $Z_5^{(2)}$ and $Z_6^{(2)}$ in a second step, and 32 more (numbered 96-127 and corresponding to subkeys $Z_1^{(2)}$ and $Z_2^{(2)}$) in a third step; finally, the remaining ones can be found with help of an exhaustive search.

3.4 A New Chosen-Plaintext Attack Breaking 2.5, 3, and 3.5 Rounds

If we apply the Demirci-Biryukov relation to 2.5-rounds IDEA, then one gets the following expression:

$$\begin{aligned} &\operatorname{lsb}\left(X_2^{(1)} \oplus X_3^{(1)} \oplus C_2^{(3)} \oplus C_3^{(3)} \oplus \bigoplus_{i=1}^{3}\left(Z_2^{(i)} \oplus Z_3^{(i)}\right)\right) \oplus \\ &\quad \operatorname{lsb}\left(Z_5^{(1)} \odot \left(\left(X_1^{(1)} \odot Z_1^{(1)}\right) \oplus \left(X_3^{(1)} \boxplus Z_3^{(1)}\right)\right)\right) \oplus \\ &\quad \operatorname{lsb}\left(Z_5^{(2)} \odot \left(\left(C_1^{(3)} \odot \overline{Z_1}^{(3)}\right) \oplus \left(C_2^{(3)} \boxminus Z_2^{(3)}\right)\right)\right) = 0 \end{aligned} \tag{5}$$

where $\overline{Z_1}^{(3)}$ denotes the inverse of $Z_1^{(3)}$ relatively to the group operation $\odot$. If we use the same trick than for 2 rounds and fix $X_1^{(1)}$ and $X_3^{(1)}$, an adversary can recover $Z_5^{(2)}$, $\overline{Z_1}^{(3)}$ and $Z_2^{(3)}$ (key bits 57-72 and 89-120) in a 2^{48} effort if 55 chosen-plaintexts are available (the success probability is then larger than 0.99). Once achieved, one can recover 39 key bits ($Z_1^{(1)}$, $Z_1^{(3)}$ and the remaining unknown bits of $Z_1^{(5)}$) numbered 0-15, 32-47 and 73-79 with the same distinguisher where Eq. (5) is fixed and known. For this, we need 46 additional known plaintexts. The remaining 41 key bits can be recovered with an exhaustive search within negligible computational complexity. Note that in this case, the Demirci-Biryukov relation applied on the decryption operation results in the same distinguisher. As far as we know, it is the fastest attack on 2.5-rounds IDEA not involving any weak-key assumption.

If a decryption oracle is available, instead of an encryption one, it is possible to mount a similar (chosen-ciphertext) attack: fix $C_1^{(2)}$ and $C_2^{(2)}$ to an arbitrary constant, and guess $Z_1^{(1)}$, $Z_3^{(1)}$, and $Z_5^{(1)}$. In a second step, guess the remaining unknown bits of $Z_1^{(3)}$, $Z_2^{(3)}$, and $Z_5^{(2)}$; one can finalize the attack using an exhaustive search.

We can extend to 3 rounds the attack previously described in a straightforward way: actually, if we fix $X_1^{(1)}$ and $X_3^{(1)}$ and guess $Z_5^{(2)}$, $\overline{Z_1}^{(3)}$, $Z_2^{(3)}$, $Z_5^{(3)}$ and $Z_6^{(3)}$ (which represent key bits numbered 50-81 and 89-120), one can recover 64 key bits in a 2^{64} process if 71 chosen-plaintext are available. Then, once $Z_5^{(2)}$, $\overline{Z_1}^{(3)}$, $Z_2^{(3)}$, $Z_5^{(3)}$ and $Z_6^{(3)}$ are known, one can apply the attack on 2.5 rounds to derive 49 more bits (numbered 0-15, 32-47, 73-79 and 127) with negligible complexity and the remaining 15 bits can finally be searched exhaustively.

The chosen-ciphertext version of this attack is clearly less effective, since one has to guess at least 96 unknown key bits (corresponding to subkeys $Z_5^{(3)}$, $Z_6^{(3)}$, $Z_1^{(3)}$, $Z_2^{(3)}$, $Z_5^{(2)}$, $Z_1^{(1)}$, $Z_3^{(1)}$, and $Z_5^{(1)}$; the unknown key bits are numbered 0-15, 32-47, 50-81,and 89-120).

For attacking 3.5 rounds, one uses a new time the distinguisher described above, one fixes $X_1^{(1)}$ and $X_3^{(1)}$ and one guesses furthermore all the keys of the last half-round; the subkeys under consideration are then $Z_5^{(2)}$, $Z_1^{(3)}$, $Z_2^{(3)}$, $Z_5^{(3)}$, $Z_6^{(3)}$, $Z_1^{(4)}$, $Z_2^{(4)}$, $Z_3^{(4)}$ and $Z_4^{(4)}$ (i.e. all the key bits but the interval 18-49, representing 96 key bits). The computational effort is approximately equal to 2^{97} if 103 chosen-plaintexts are available.

The same attack can be adapted for a decryption oracle, however resulting in a higher complexity: if 119 chosen-ciphertext are available to an attacker (where $C_1^{(4)}$ and $C_3^{(4)}$ are fixed to an arbitrary constant), then one can recover 112 key bits numbered 0-111 (corresponding to subkeys $Z_5^{(2)}$, $Z_1^{(2)}$, $Z_3^{(2)}$, $Z_5^{(1)}$, $Z_6^{(1)}$, $Z_1^{(1)}$, $Z_2^{(1)}$, $Z_3^{(1)}$, and $Z_4^{(1)}$).

3.5 Trading Time and Memory

We show now that it is possible under certain circumstances to trade memory and time complexities in the attacks of Nakahara *et al.* [17].

Let us consider 2.5-rounds IDEA, and let us assume that we have 55 known plaintext-ciphertext pairs at disposal. For all possible values of $Z_1^{(1)}$, $Z_1^{(3)}$, and $Z_5^{(1)}$ (i.e. key bits numbered 0-15, 32-47, and 64-79), we can compute a guess for the value of the following expression.

$$\underbrace{Z_2^{(1)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_2^{(3)} \oplus Z_3^{(3)}}_{\text{constant}} \oplus \quad \text{lsb}\left(\delta^{(2)} \oplus \gamma^{(2)}\right) \tag{6}$$

The sub-sum depending only of the key bits is unknown but constant. Let us store all these guesses in a large hash table made of 2^{48} 55-bit words, As second step of the attack, one guesses the key bits $Z_5^{(2)}$, $Z_1^{(3)}$, and $Z_2^{(3)}$ (i.e. bits numbered 57-72 and 89-120): for all these guesses, and for the 55 ciphertexts, we can compute (by partially decrypting the ciphertexts) the value of $\mathrm{lsb}(\delta^{(2)} \oplus \gamma^{(2)})$ and checking whether this value (or its complement) is stored in the table or not. With high probability, the right subkey candidate will be determined by one of the few expected matches. This attack hence requires two times $55 \cdot 2^{48} \approx 2^{54}$ partial encryptions/decryptions, and 2^{48} memory cells, while the remaining 41 unknown bits can be recovered with an exhaustive search within negligible complexity.

The attack can be extended to more rounds in the following way. Using the same approach than for the 2.5-round case, one computes a hash table containing, for all possible values of $Z_1^{(1)}$, $Z_1^{(3)}$, and $Z_5^{(1)}$, a guess for

$$\underbrace{Z_2^{(1)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_2^{(3)} \oplus Z_3^{(3)}}_{\text{constant}} \oplus \quad \mathrm{lsb}\left(\delta^{(2)} \oplus \gamma^{(2)}\right) \oplus \mathrm{lsb}\left(\delta^{(3)} \oplus \gamma^{(3)}\right)$$

for 71 known plaintext-ciphertext pairs. In a second step, by guessing $Z_5^{(2)}$, $Z_1^{(3)}$, $Z_2^{(3)}$, $Z_5^{(3)}$, and $Z_5^{(3)}$ (i.e. key bits numbered 50-81 and 89-120), one can recover a total of 96 key bits, the remaining 32 ones with help of an exhaustive search, in an approximate overall computational complexity of 2^{70} operations.

Finally, this attack can be extended to 3.5 rounds if we guess the additional unknown key bits of $Z_1^{(4)}$, $Z_2^{(4)}$, $Z_3^{(4)}$, and $Z_4^{(4)}$ (i.e. bits numbered 0-17 and 121-127). One needs in this case 103 known plaintext-ciphertext pairs, 2^{48} 103-bit words of memory, and a computational complexity of about 2^{103} operations.

The same attack strategy on 4 rounds would imply guessing all the key bits, thus it is less efficient than an exhaustive key search.

3.6 Combination with other Attacks

Interestingly, we note that our attacks can be used in parallel with other attacks to gain more key bits. For instance, the attack on 3-rounds IDEA of Demirci *et al.*described in [6] is able to recover the values of $Z_2^{(1)}$, $Z_4^{(1)}$, $Z_5^{(2)}$, and $Z_5^{(3)}$ (which represents 41 key bits) in a 2^{42} effort (after a 2^{64} precomputation). Then, to derive 32 other key bits, the authors assume that a *decryption oracle* is available. If it is not the case, one can still relax this condition by applying the attack described in §3.4 and recover 41 additional key bits, namely those numbered 73-81 and 89-120, within negligible computational complexity. Similar considerations apply if *only a decryption oracle* is available.

Another interesting combination of known attacks and the ones described in this paper is the following: in [5], Demirci describes a square-like distinguisher which, with help of two sets of 2^{32} chosen-plaintexts, allows to recover $Z_5^{(3)}$ in about 2^{49} operations. If, in a second step, we plug the obtained value of $Z_5^{(3)}$ into the attack described in §3.4, we can derive 48 other key bits numbered 66-81, and 89-120 in a 2^{49} computational effort in a second step, and finally the remaining bits within negligible time. This defines an attack which derives all key bits within 2^{50} operations if 2^{33} chosen-plaintexts are available. This represents a computational complexity decrease by a factor of about 2^{32}. Unfortunately, the same strategy does only marginally improve the attack against 3.5 (or more rounds): one can replace the final exhaustive search of the remaining 80-bit keys by our more efficient attack.

3.7 A New Square-Like Distinguisher

As observed for the first time by Nakahara *et al.* [15] and later by Demirci [5], square-like distinguishers can be used with success to attack IDEA. We present now such a distinguisher which is somewhat simpler to use than the ones available in the literature.

Lemma 2 (Square-Like Distinguisher on 2.5-Round IDEA). *Let 2^{16} different inputs of 2.5-round IDEA be defined as follows: $X_1^{(1)}$, $X_2^{(1)}$, and $X_3^{(1)}$ are fixed to arbitrary constants, and $X_4^{(1)}$ takes all possible values. Then the XOR of the 2^{16} values of the equation*

$$\begin{aligned} X_2^{(1)} \oplus X_3^{(1)} \oplus C_2^{(1)} \oplus C_3^{(1)} \oplus \\ Z_2^{(1)} \oplus Z_3^{(1)} \oplus Z_2^{(2)} \oplus Z_3^{(2)} \oplus Z_2^{(3)} \oplus Z_3^{(3)} \oplus \\ \operatorname{lsb}\left(\gamma^{(1)} \oplus \delta^{(1)}\right) \oplus \operatorname{lsb}\left(\gamma^{(2)} \oplus \delta^{(2)}\right) \end{aligned} \tag{7}$$

is equal to 0 with probability one.

We can then use this distinguisher to attack reduced-round versions of IDEA. To attack 3 rounds, encrypt 39 different structures of 2^{16} chosen plaintexts according to Lemma 2. Then, for all possible values of $Z_5^{(3)}$ and $Z_6^{(3)}$ (i.e. bits numbered 50-81), partially decrypt the ciphertext for the 39 structures using the same iterative strategy as in Alg. 1. This attack recovers 32 key bits, and with a few more chosen plaintexts, we can apply the attack on 2.5-rounds described in §3.4 to recover all the keys bits. In summary, this attack requires less than 2^{22} chosen-plaintexts and a computational complexity of approximately 2^{50} operations.

On 3.5 rounds, we can attack the round keys $Z_5^{(3)}$, $Z_1^{(4)}$, and $Z_2^{(4)}$ (i.e. 48 key bits numbered 50-65 and 82-113) in a similar fashion. In this case, we need 55 structures of 2^{16} chosen plaintexts (i.e. less than 2^{22} chosen plaintexts as well), and a computational complexity of approximately $3 \cdot 2^{16} \cdot 2^{48} \approx 2^{66}$ operations.

Finally, we can attack 4 rounds using the same strategy by guessing further key bits, i.e. those of $Z_5^{(4)}$ and of $Z_6^{(4)}$, which represents 80 unknown bits in total. Hence, we need about 87 structures of 2^{16} chosen plaintexts, which is less than 2^{23} chosen plaintexts, and a computational cost of about $3 \cdot 2^{16} \cdot 2^{80} \approx 2^{98}$ operations.

4 Summary of the Attacks

In this paper, we have used the same kind of properties derived by Demirci [5] and Nakahara et al. [16] to derive a sequence of simple, yet efficient attacks

Rounds	Data	Time	Attack type	Ref.	Note
2	2^{10} CP	2^{42}	differential	[14]	
2	62 CP	2^{34}	*linear-like*	*§3.3*	
2	23 CP	2^{64}	square-like	[5]	
2.5	2^{10} CP	2^{106}	differential	[14]	Memory: 2^{96}
2.5	2^{10} CP	2^{32}	differential	[4]	For one key out of 2^{77}
2.5	2^{18} CP	2^{58}	square	[15]	
2.5	2^{32} CP	2^{59}	square	[15]	
2.5	2^{48} CP	2^{79}	square	[15]	
2.5	2 CP	2^{37}	square	[15]	Under 2^{16} rel. keys
2.5	55 CP	2^{81}	square-like	[5]	
2.5	101 CP	2^{48}	*linear-like*	*§3.4*	
2.5	97 KP	2^{90}	linear-like	[16]	
2.5	55 KP	2^{54}	*linear-like*	*§3.5*	*Memory:* 2^{48}
3	2^{29} CP	2^{44}	differential-linear	[3]	
3	71 CP	2^{71}	square-like	[5]	
3	71 CP	2^{64}	*linear-like*	*§3.4*	
3	2^{33} CP	2^{64}	collision	[6]	Memory: 2^{64}
3	2^{33} CP	2^{50}	*linear-like + [5]*	*§3.6*	
3	2^{22} CP	2^{50}	*square-like*	*§3.7*	
3	71 KP	2^{70}	*linear-like*	*§3.5*	*Memory:* 2^{48}
3.5	2^{56} CP	2^{67}	truncated diff.	[3]	
3.5	$2^{38.5}$ CP	2^{53}	impossible diff.	[1]	Memory: 2^{48}
3.5	2^{34} CP	2^{82}	square-like	[5]	
3.5	2^{24} CP	2^{73}	collision	[6]	
3.5	2^{22} CP	2^{66}	*square-like*	*§3.7*	
3.5	103 CP	2^{103}	square-like	[5]	
3.5	103 CP	2^{97}	*linear-like*	*§3.4*	
3.5	119 KP	2^{112}	linear-like	[16]	
3.5	103 KP	2^{97}	*linear-like*	*§3.5*	*Memory:* 2^{48}
4	2^{37} CP	2^{70}	impossible diff.	[1]	Memory: 2^{48}
4	2^{34} CP	2^{114}	square-like	[5]	
4	2^{24} CP	2^{89}	collision	[6]	Memory: 2^{64}
4	2^{23} CP	2^{98}	*square-like*	*§3.7*	
4	121 KP	2^{114}	linear-like	[16]	
4.5	2^{64} CP	2^{112}	impossible diff.	[1]	
4.5	2^{24} CP	2^{121}	collision	[6]	Memory: 2^{64}
5	2^{24} CP	2^{126}	collision	[6]	Memory: 2^{64}

Fig. 3. Attacks against IDEA

against reduced-round versions of IDEA; the attacks against 2 and 2.5 rounds are the best known ones not involving any weak-key assumption, to the best of our knowledge. Some of them, given the same order of computational complexity, reduce the amount of necessary chosen plaintexts, while other attacks, given a comparable amount of chosen texts, decrease favorably the computational complexity; additionally, some tradeoffs between time and memory are presented, which lead to far less complex attacks using only known plaintext-ciphertext pairs. Furthermore, we showed how to use some of these attacks in combination with other known attacks, which allows sometimes to gain more key bits with a lesser complexity, or to avoid the use of both encryption and decryption oracles. The more important attacks against this block cipher are tabulated in Figure 3, where KP (resp. CP) means "known plaintext-ciphertext pairs" (resp. "chosen-plaintexts"), as well as their respective complexities. We observed that it is possible to dramatically decrease the complexity attacking IDEA by combining "independent" properties in a divide-and-conquer fashion. A nice illustration is certainly the attack on 3-rounds IDEA described in §3.6: it allows to reduce the computational complexity from 2^{82} down to 2^{50} and to somewhat approach the performances of the attack by Borst *et al.* [3] based on truncated differentials. In another case, we are able to relax some conditions, like the need of *two* oracles. Although such combinatorial properties (mainly due to the key-schedule algorithm) do not seem to result in a threat against the full version of the cipher, an important open question is to know whether such properties can be extended to attack more rounds.

Acknowledgments

We would like to thank Willi Meier, Serge Vaudenay, Michael Hill as well as an anonymous reviewer for useful and interesting comments about this work.

References

1. E. Biham, A. Biryukov, and A. Shamir. Miss-in-the-middle attacks on IDEA and Khufru. In L. Knudsen, editor, *Fast Software Encryption: 6th International Workshop, FSE'99, Rome, Italy, March 1999. Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, pages 124–138. Springer-Verlag, 1999.
2. A. Biryukov, J. Nakahara, B. Preneel, and J. Vandewalle. New weak-key classes of IDEA. In R. Deng, S. Qing, F. Bao, and J. Zhou, editors, *Information and Communications Security: 4th International Conference, ICICS 2002, Singapore, December 9-12, 2002. Proceedings*, volume 2513 of *Lecture Notes in Computer Science*, pages 315–326. Springer-Verlag, 2002.
3. J. Borst, L. Knudsen, and V. Rijmen. Two attacks on reduced IDEA (extended abstract). In W. Fumy, editor, *Advances in Cryptology – Eurocrypt '97: International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 1997. Proceedings*, volume 1233 of *Lecture Notes in Computer Science*, pages 1–13. Springer-Verlag, 1997.

4. J. Daemen, R. Govaerts, and J. Vandewalle. Weak keys for IDEA. In D. Stinson, editor, *Advances in Cryptology* – CRYPTO *'93: 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993. Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 224–231. Springer-Verlag, 1994.
5. H. Demirci. Square-like attacks on reduced rounds of IDEA. In K. Nyberg and H. Heys, editors, *Selected Areas in Cryptography: 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 147–159. Springer-Verlag, 2003.
6. H. Demirci, A. Selçuk, and E. Türe. A new meet-in-the-middle attack on the IDEA block cipher. In *Selected Areas in Cryptography: 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 2003. Revised Papers*, volume 3006 of *Lecture Notes in Computer Science*, pages 117–129. Springer-Verlag, 2004.
7. S. Garfinkel. *PGP: Pretty Good Privacy.* O'Reilly and Associates, 1994.
8. P. Hawkes. Differential-linear weak key classes of IDEA. In K. Nyberg, editor, *Advances in Cryptology* – EUROCRYPT *'98: International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, May/June 1998. Proceedings*, volume 1403 of *Lecture Notes in Computer Science*, pages 112–126. Springer-Verlag, 1998.
9. L. Knudsen and J. Mathiassen. A chosen-plaintext linear attack on DES. In B. Schneier, editor, *Fast Software Encryption: 7th International Workshop, FSE 2000, New York, NY, USA, April 2000. Proceeding*, volume 1978 of *Lecture Notes in Computer Science*, pages 262–272. Springer-Verlag, 2001.
10. X. Lai. *On the design and security of block ciphers*, volume 1 of *ETH Series in Information Processing.* Hartung-Gorre Verlag, 1992.
11. X. Lai and J. Massey. A proposal for a new block encryption standard. In I. Damgård, editor, *Advances in Cryptology* – EUROCRYPT *'90: Workshop on the Theory and Application of Cryptographic Techniques, Aarhus, Denmark, May 1990. Proceedings*, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer-Verlag, 1991.
12. X. Lai, J. Massey, and S. Murphy. Markov ciphers and differential cryptanalysis. In D. Davies, editor, *Advances in Cryptology* – EUROCRYPT *'91: Workshop on the Theory and Application of Cryptographic Techniques, Brighton, UK, April 1991. Proceedings*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38. Springer-Verlag, 1991.
13. M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y. Desmedt, editor, *Advances in Cryptology* – CRYPTO *'94: 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994. Proceedings*, volume 839 of *Lecture Notes in Computer Science*, 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: Workshop on the Theory and Application of Cryptographic Techniques, Lofthus, Norway, May 1993. Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pages 371–385. Springer-Verlag, 1993.
15. J. Nakahara, P. Barreto, B. Preneel, J. Vandewalle, and Y. Kim. Square attacks on reduced-round PES and IDEA block ciphers. In B. Macq and J.-J. Quisquater, editors, *Proceedings of 23rd Symposium on Information Theory in the Benelux, Louvain-la-Neuve, Belgium, May 29-31, 2002*, pages 187–195, 2002.

16. J. Nakahara, B. Preneel, and J. Vandewalle. The Biryukov-Demirci attack on IDEA and MESH ciphers. Technical report, COSIC, ESAT, Katholieke Universiteit Leuven, Leuven, Belgium, 2003.
17. J. Nakahara, B. Preneel, and J. Vandewalle. The Biryukov-Demirci attack on reduced-round versions of IDEA and MESH block ciphers. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *Information Security and Privacy: 9th Australasian Conference, ACISP 2004, Sydney, Australia, July 13-15, 2004. Proceedings*, volume 3108 of *Lecture Notes in Computer Science*, pages 98–109. Springer-Verlag, 2004.
18. National Bureau of Standards, U. S. Department of Commerce. *Data Encryption Standard (DES)*, FIPS 46, 1977.