# A New Attack on 6-Round IDEA

Eli Biham*[1] Orr Dunkelman*[2] Nathan Keller**[3]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
biham@cs.technion.ac.il
[2]Katholieke Universiteit Leuven, Dept. of Electrical Engineering ESAT/SCD-COSIC.
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
orr.dunkelman@esat.kuleuven.be
[3]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. Since its introduction in 1991, IDEA has withstood extensive cryptanalytic effort, but no attack was found on the full (8.5-round) variant of the cipher.
In this paper we present the first known attack on 6-round IDEA faster than exhaustive key search. The attack exploits the weak key-schedule algorithm of IDEA, and combines Square-like techniques with linear cryptanalysis to increase the number of rounds that can be attacked. The attack is the best known attack on IDEA. We also improve previous attacks on 5-round IDEA and introduce a 5-round attack which uses only 16 known plaintexts.

## 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 [19]. Due to its inclusion in several cryptographic packages, such as PGP , IDEA is one of the most widely used block ciphers. Since its introduction, IDEA resisted intensive cryptanalytic efforts [1–3, 5, 6, 9–11, 13–17, 21–23]. The best published chosen-plaintext attack on IDEA is an attack on 5-round IDEA that requires $2^{19}$ chosen plaintexts, and has time complexity of $2^{103}$ encryptions [3]. The best published related-key attack is an attack on 7.5-round IDEA that requires $2^{43.5}$ known plaintexts and has a time complexity of $2^{115.1}$ encryptions [3]. 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 membership test requires $2^{16}$ adaptive chosen plaintexts and ciphertexts and has a time complexity of $2^{16}$ encryptions [6].

* 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.

In this paper we present the first known attacks against 5.5-round and 6-round IDEA. These higher-order differential-linear [4] attacks consist of three components:

1. Constructing linear equations involving the least significant bits of the intermediate values of the cipher. We note that this idea was proposed and used in [3, 17, 23].
2. Using a higher-order differential (or a Square property) to simplify the linear equations. We note that this modification was proposed in [17]. However, as we show later, the Square distinguisher used in [17] is incorrect, and hence, we replace it by another distinguisher.
3. Taking advantage of the weak key schedule of IDEA — we observe that in some cases, guessing only 103 of the 128 key bits of IDEA is sufficient for encrypting two full rounds of the cipher, and even more than that.

The 5.5-round attack requires $2^{32}$ chosen plaintexts and has a time complexity of about $2^{127}$ encryptions, about twice faster than exhaustive key search. The 6-round attack requires almost the entire code book and has a time complexity similar to that of the 5.5-round attack. We note that the time complexity of the attacks could be improved significantly if the Square distinguisher could be replaced by a better one, like the one presented in [17]. However, we were not able to find such a distinguisher at this stage.

We then show two improvements to the 5-round attack presented in [3]. The first improvement reduces the data complexity of the attack by a factor of $\sqrt{2}$ to $2^{18.5}$ known plaintexts, without affecting the time complexity of the attack of $2^{103}$ encryptions. The second improvement reduces the data complexity to 16 known plaintexts, while raising the data complexity to $2^{114}$ encryptions.

The complexities of the new attacks, along with selected previously known attacks, are summarized in Table 1.

The paper is organized as follows: In Section 2 we briefly describe the structure of IDEA. In Section 3 we present the new attack on 5.5-round IDEA. In Section 4 we extend the 5.5-round attack to an attack on 6-round IDEA. We present improved attacks on 5-round IDEA in Section 5. Finally, Section 6 summarizes the paper.

## 2 Description of IDEA and the Notations Used in the Paper

IDEA [19] 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

| Rounds | Attack Type | Complexity | | Source |
|---|---|---|---|---|
| | | Data | Time | |
| 4 | Impossible Differential | $2^{37}$ CP | $2^{70}$ | [2] |
| 4 | Linear | 114 KP | $2^{114}$ | [23] |
| 4 | Square | $2^{32}$ CP | $2^{114}$ | [13] |
| 4 | Square | $2^{23}$ CP | $2^{98}$ | [17]$^{\dagger}$ |
| 4.5 | Impossible Differential | $2^{64}$ CP | $2^{112}$ | [2] |
| 4.5 | Linear | 16 CP | $2^{103}$ | [3] |
| 5 | Meet-in-the-Middle Attack | $2^{24}$ CP | $2^{126}$ | [14] |
| 5 | Meet-in-the-Middle Attack | $2^{24.6}$ CP | $2^{124}$ | [1] |
| 5 | Linear | $2^{19}$ KP | $2^{103}$ | [3] |
| 5 | Linear | $2^{18.5}$ KP | $2^{103}$ | Section 5 |
| 5 | Linear | 16 KP | $2^{114}$ | Section 5 |
| 5.5 | Higher-Order Differential-Linear | $2^{32}$ CP | $2^{126.85}$ | Section 3 |
| 6 | Higher-Order Differential-Linear | $2^{64} - 2^{52}$ KP | $2^{126.8}$ | Section 4 |

KP – Known plaintexts, CP – Chosen plaintexts
Time complexity is measured in encryption units
$^{\dagger}$ – As we show in Section 3.2, this attack does not work

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

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 to 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 the 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 omitted. 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.

## 3 A New Attack on 5.5-Round IDEA

In this section we present the new attack on 5.5-round IDEA. First we present the three components of the attack.

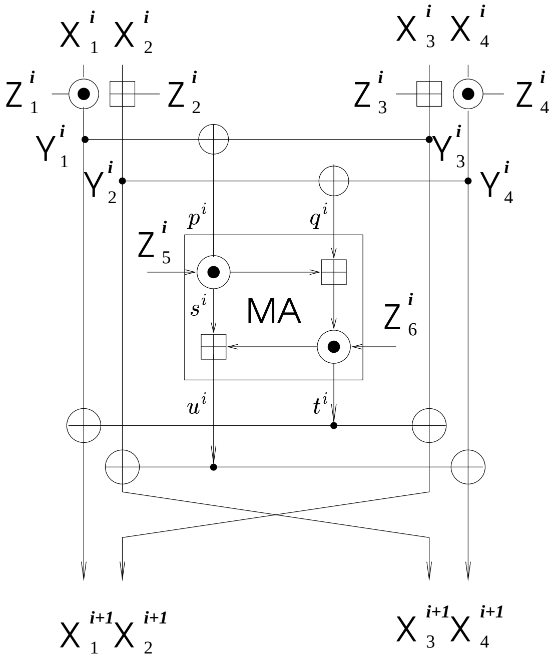


**Fig. 1.** One Round of IDEA

| 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

### 3.1 The First Component — A Linear Equation Involving the LSBs of the Intermediate Encryption Values

We start with an observation due to Biryukov (according to [23]) and Demirci [14]. 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

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

Similarly,

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

When we consider 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:

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

and

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

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 above two equations we obtain

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

This equation is called in [17] "the Biryukov-Demirci relation".

In order to simplify this equation, we consider the XOR of the intermediate values of several encrypted plaintexts. In [3] the XOR difference between two plaintexts is used; in our attack we use the XOR value of larger sets of plaintexts.

Consider a structure of plaintexts $P^1, \ldots, P^m$, where $m$ is an even integer. Then the XOR of the equations of the form (5) given by $P^1, \ldots, P^m$ gives

$$LSB\left(\bigoplus_{j=1}^{m}(P_2^j \oplus P_3^j) \oplus \bigoplus_{i=1}^{8} S^i\right) = LSB\left(\bigoplus_{j=1}^{m}(C_2^j \oplus C_3^j)\right), \quad (6)$$

where $S^i = \oplus_{j=1}^m s^i(P^j)$.

Equation (6) is the basic equation used in our attack, where $m$ and the exact structure of plaintexts are specified later.

### 3.2 The Second Component — A Square-Like Structure

In order to further simplify Equation (6) we want to use special structures of plaintexts, for which we will get $S^1 = S^2 = 0$, independently of the key. Our structures are higher-order differentials, a special case of Square-like structures that were used in [7, 12, 18, 20]. The structures we use and their properties are described in the following proposition.

**Proposition 1.** *Let $T$ be a structure of $m = 2^{16}$ plaintexts, such that the intermediate values after the first $KA$ layer, denoted by $Y^{1,j}$, where $1 \leq j \leq m$, satisfy the following requirements:*

1. *$Y_2^{1,j}$ and $Y_4^{1,j}$ are fixed for all $j$.*
2. *The $2^{16}$ values $Y_1^{1,j}$ (for $1 \leq j \leq 2^{16}$) are different.*
3. *$Y_3^{1,j} = Y_1^{1,j} \oplus C$ for some fixed $C$.*

*Then for $T = \{P^1, \ldots, P^m\}$ the relation $S^1 = S^2 = 0$ holds, independently of the key.*

*Proof.* First, note that by the assumption, $(p^1, q^1)$ is fixed for all the $2^{16}$ elements of the structure. Hence, $s^1$ is fixed as well, leading to $S^1 = \bigoplus_{j=1}^{2^{16}} s^1(P^j) = 0$. The output values of the $MA$ structure, $(u^1, t^1)$, are also fixed. Thus, the values $X_3^2$ are constant for all the elements of the structure, as well as the values $X_4^2$. In addition, all the values $X_1^2$ are different, as well as the values $X_2^2$. As a result, all the values $Y_1^2$ are different and all the values $Y_3^2$ are constant. Hence, all the values $p^2 = Y_1^2 \oplus Y_3^2$ are different. Thus, all the values $s^2 = Z_5^2 \odot p^2$ are different. However, since there are only $2^{16}$ possible values of $s^2$, it means that $s^2$ assumes each possible value once and only once. Hence, $S^2 = \oplus_{j=1}^{2^{16}} s^2(P^j) = 0$, and this completes the proof.

In the sequel of the paper we call structures which satisfy the above conditions, “right structures”.

It follows from the proposition that if we take a right structure as $\{P^1, \ldots, P^m\}$, Equation (6) is simplified to

$$LSB\left(\bigoplus_{j=1}^{m}(P_2^j \oplus P_3^j) \oplus \bigoplus_{i=3}^{8} S^i\right) = LSB\left(\bigoplus_{j=1}^{m}(C_2^j \oplus C_3^j)\right). \tag{7}$$

We note that [17] makes use of a seemingly better Square-like structure that is described in the following statement [17, Section 3.7, Lemma 2]:

**Statement 1** *Let $L$ be a structure of $m = 2^{16}$ plaintexts, denoted by $P^1, \ldots, P^m$, having the following properties:*

1. *$P_1^j, P_2^j, P_3^j$ are fixed for all $j$.*
2. *The $2^{16}$ values $P_4^j$ (for $1 \leq j \leq 2^{16}$) are different.*

*Then for $T = \{P^1, \ldots, P^m\}$ we have $S^1 = S^2 = 0$, independently of the key.*

If this statement was correct, it could be used to improve significantly the time complexity of our attack. However, it appears that the statement is incorrect. For the structure described in the statement we indeed have $S^1 = 0$, but we do not have $S^2 = 0$, since nothing can be said about the values $s^2(P^j)$. The following set of constants is a counterexample for Statement 1:

$$P_1^j = F78b_x; \quad P_2^j = 245_x; \quad P_3^j = ABCD_x;$$
$$Z_1^1 = 8_x; \quad Z_2^1 = 567A_x; \quad Z_3^1 = 2C68_x; \quad Z_4^1 = 4_x;$$
$$Z_5^1 = 5_x; \quad Z_6^1 = 6_x; \quad Z_1^2 = 1238_x; \quad Z_2^2 = 999_x;$$

We note that $Z_2^2$ is not relevant to the approximation, but it is required for defining the key for which the above statement fails. Using the key schedule, we derive the following subkeys as well: $Z_3^2 = F458_x$, $Z_4^2 = D000_x$, $Z_5^2 = 800_x$, and $Z_6^2 = A00_x$.

### 3.3 The Third Component — Exploiting The Weak Key Schedule

The fact that the key schedule of IDEA is relatively weak has been known for a long time, and was already used to devise related-key attacks on reduced-round IDEA (e.g.,[5]) and to find large weak key classes for the entire cipher (e.g.,[6]). However, other key recovery attacks (e.g.,[2]) usually exploited other properties of IDEA and took only a small advantage of the key schedule.

In our attack, we use the weakness of the key schedule in order to calculate all the remaining $S^i$ values, while guessing a relatively small number of key bits.

Consider a 5.5-round variant of IDEA starting with the third round. We observe that in order to compute the values $S^5, S^6, S^7$ from the ciphertexts, it is sufficient to know only 103 key bits. The values $S^5, S^6, S^7$ are determined by the values of the ciphertext and of the subkeys $Z_4^8, Z_3^8, Z_2^8, Z_1^8, Z_6^7, Z_5^7, Z_4^7, Z_3^7, Z_2^7, Z_1^7, Z_6^6, Z_5^6, Z_1^6, Z_2^6, Z_5^5$. These subkeys are sufficient in order to partially decrypt the ciphertexts through the last two rounds and to find the value of $S^5$. Note that $S^5$ is independent of the values of $Z_3^6$ and $Z_4^6$. All the required 15 subkeys use only 103 bits of the master key, whereas bits 100–124 of the master key remain unused.

Since in our case (for 5.5 rounds), Equation (7) is reduced to

$$LSB\left(\bigoplus_{j=1}^{m}(P_2^j \oplus P_3^j) \oplus \bigoplus_{i=5}^{7} S^i\right) = LSB\left(\bigoplus_{j=1}^{m}(C_2^j \oplus C_3^j)\right), \tag{8}$$

we can guess 103 key bits and check whether the equation holds.

### 3.4 The Basic 5.5-Round Attack

In this subsection we combine the components presented in the previous subsections to devise an attack on 5.5-round IDEA. As was shown in Section 3.3, once we have a right structure, it is possible to check the guess of 103 key bits.

This is done by partially decrypting the entire set under the subkey guess, and checking whether Equation (8) holds.

The problem in finding a right structure is the key addition layer, which prevents constructing the right structures immediately. The solution to this problem is to use more plaintexts, which increases the data complexity, but in exchange, the set is ensured to contain (several) right structures. We use a set of $2^{32}$ plaintexts, such that the second and the fourth words are fixed to some arbitrary constant, and the first and the third words obtain all possible values. In such a structure, for any given subkeys, it is possible to find $2^{16}$ right structures of $2^{16}$ plaintexts each.

The basic algorithm of the attack is the following:

1. **Data collection phase:** Ask for the encryption of a structure of chosen plaintexts, of the form $(x, B, y, D)$ for two randomly selected constants $(B, D)$ and all possible values of $x$ and $y$.
2. **Constructing a right structure:** For each possible value of bits 89–104 and 121–8 of the key, perform the following:
   (a) Choose an arbitrary 16-bit value $F_1$ (a candidate for $p^3$).
   (b) For all $0 \leq j \leq 2^{16} - 1$, partially decrypt the words $j$ under the guess of $Z_1^3$ and $F_1 \oplus j$ under the guess of $Z_3^3$. Denote the list of resulting values by $(A_1^j, C_1^j)$.
3. **Checking the linear equation:** Choose all plaintexts of the form $(A_1^j, B, C_1^j, D)$. For each possible value of key bits 0–104 and 121–127 (note that from these bits we already guessed key bits 89–104 and 121–8) partially decrypt the ciphertexts corresponding to the plaintexts of the right structure to get the values $S^5, S^6, S^7$. Check whether Equation (8) holds. If not, discard the partial key guess. If the equation holds, pass the key guess for further analysis.
4. **Filtering the remaining key guesses:** For the remaining key guesses, take another right structure of plaintexts, i.e., pick a different value for $F$, and repeat Steps 2–3, with three different selections. If a partial key guess passed all the four tests, perform exhaustive key search on the remaining key bits.

### 3.5 Analysis and Improvement of the Basic Attack

The attack requires one structure of $2^{32}$ plaintexts, and thus the data complexity of the attack is $2^{32}$ chosen plaintexts. The time complexity of Step (1) is $2^{32}$ encryptions.

The time complexity of Step (2) is $2^{48}$ partial decryptions. We note that this step can be performed as a precomputation, but there is no need to do it since the time complexity of this step is negligible with respect to the total time complexity of the attack.

The most time consuming part of the attack is Step (3) which is repeated $2^{32}$ times (for each guess of bits 89–104 and 121–8). In this step, the attacker guesses 80 key bits and performs a partial decryption of $2^{16}$ ciphertexts. The partial decryption includes two full rounds and three out of the eight operations of an additional round. Hence, the time complexity of this stage is about $0.43 \cdot 2^{128}$ encryptions in total. About half of the keys are expected to pass to Step (4). For these keys, the attacker performs three additional filterings in Step (4), with time complexity of $(0.22 + 0.11 + 0.05) \cdot 2^{128}$. The $2^{124}$ remaining keys are checked by exhaustive key search. Hence, the total time complexity of the attack is $(0.43 + 0.22 + 0.11 + 0.05 + 0.06) \cdot 2^{128} = 0.87 \cdot 2^{128}$ encryptions, which is only slightly better than exhaustive key search.

In order to reduce the time complexity of the attack, we introduce a small change in Step (2). We observe, that there is no need to fix a concrete value of $F$ such that $p^3 = F$ for all the values of the structure; it is sufficient to have for all the plaintexts in the considered structure a fixed value for $p^3$. We exploit this observation by eliminating the need to guess bit 121 of the key in Step (2). We do not guess the value of this bit, but rather assume that its value is zero. As a result, when we decrypt the values $j \oplus F$ through the addition with $Z_3^3$, all the bits except for the MSB are correct, but the MSB might be wrong. However, if our assumption was incorrect and the values of the MSBs are wrong, this happens to all the elements of the structure simultaneously, since

$$x \boxplus (y \boxplus 8000_x) = (x \boxplus y) \boxplus 8000_x = (x \boxplus y) \oplus 8000_x. \tag{9}$$

Hence, if we take the plaintext structure $(A_1^j, B, C_1^j, D)$ as in the basic attack, we have two possibilities: If our assumption was correct, we get a right structure as in the basic attack. If our assumption was incorrect, then for all the elements of the structure we have $p^3 = F \oplus 8000_x$. However, $p^3$ still assumes the same value for all the elements of the structure, and hence, the structure is a right structure.

Therefore, we can obtain right structures without guessing the value of bit 121 of the key (by making sure to choose the $F$ values without using two values which differ only in the MSB). This improvement reduces the time complexity of all the steps of the attack (except for the final exhaustive key search) by a factor of two. Hence, the time complexity of the improved attack is $(0.22+0.11+0.05+0.03 + 0.06) \cdot 2^{128} = 0.45 \cdot 2^{128} = 2^{126.85}$, about twice faster than exhaustive key search.

It is also possible to use up to $2^{15}$ right structures from a given plaintext structure, and use them to filter wrong subkey guesses. When using $k$ right structures with the improved attack, the data complexity is $2^{32}$ chosen plaintexts, and the time complexity is

$$2^{111} \cdot 2^{16} \cdot \frac{2\frac{3}{8}}{5\frac{1}{2}} \cdot \left(1 + \frac{1}{2} + \cdots + \frac{1}{2^{k-1}}\right) + 2^{128-k}$$

encryptions. When using 16 right structures, the time complexity is $2^{126.8}$ encryptions. The memory requirements of the attack are mostly for storing the data, i.e., $2^{32}$ blocks of 128-bit each (or $2^{36}$ bytes).

## 4 The 6-Round Attack

In this section we extend the 5.5-round attack to an attack on 6-round IDEA. The variant we attack starts before the $MA$ layer of round 2 and ends before the $MA$ layer of round 8.

We observe that the key bits used in the $MA$ layer of round 2 (bits 57–88) are included in the bits that are guessed in the 5.5-round attack. Hence, we can add this half round in the beginning of the attack without enlarging the time complexity.

However, in this case it is much harder to construct right structures, and the data complexity is significantly increased. Our best method is to ask for the encryption of almost the entire code book, and look for the right structures after a partial encryption. We note that if a way to construct right structures is found, then the data complexity of the attack can be reduced. However, we did not find appropriate structures at this stage.

Assume that we start with $2^{64} - a$ known plaintexts, where $a$ is a constant we determine later. For a fixed value of the subkeys $Z_5^2, Z_6^2, Z_1^3, Z_3^3$, we can divide the plaintexts into $2^{48}$ classes according to the value of the triplet $(X_2^3, X_4^3, p^3)$ (obtained from the plaintexts using the fixed subkeys). Each class consists of $2^{16}$ plaintexts, and forms a right structure. Hence, $a$ has to be small enough, such that for almost every value of the subkeys, there will be at least four "full" classes in the pool of known plaintexts. In order to estimate the number of full classes, we look at the plaintexts that are not known to the attacker. Assuming that these plaintexts are uniformly distributed over the $2^{48}$ classes, the probability that none of them falls in some prescribed class is $(1-2^{-48})^a \approx e^{-a/2^{48}}$. Hence, the expected number of classes that are not ruined by missing plaintexts, i.e., the full classes, is $2^{48}e^{-a/2^{48}}$. If we take $a = 2^{53}$, this expected number is close to four. However, we want that for most of the possible values of the subkeys there will be four right structures. Hence, we take $a = 2^{52}$. In this case, the expected number of right structures is about $2^{25}$, and for most of the possible values of the subkeys, there will be enough right structures. If for some subkey guess, there are not enough right structures, the attacker has to check this guess separately by exhaustive key search over the remaining key bits.[1] We note that this improvement can also reduce the data complexity of the 5.5-round attack by about $2^{18}$ chosen plaintexts.

Therefore, our 6-round attack requires $2^{64} - 2^{52}$ known plaintexts. The first stage of the attack is slightly changed to the following:

1. Ask for the encryption of $2^{64} - 2^{52}$ arbitrary plaintexts.
2. For every possible value of key bits 57—104 and 121—8:
   (a) Repeat until a candidate right structure is found in the data set — Choose $(X_2^3, X_4^3, p^3)$ at random. Partially decrypt the set $(j, X_2^3, j \oplus p^3, X_4^3)$ for $j = 0, \ldots 2^{16} - 1$ under the subkey guess, and check that all

[1] The probability that for $2^{64} - 2^{52}$ random plaintexts, there exists such a key is about $e^{-33554367.2}$.

the resulting plaintexts are in the data set.[2] Stop once at least four such sets are found.

(b) Apply Steps (2) and (3) of the 5.5-round attack given the four right structures using the guessed key bits.

This algorithm allows to detect four right structures that are used in the same way as in the 5.5-round attack. We note that since the expected number of right structures is about $2^{25}$, we expect that after about $2^{25}$ checked triples $(X_2^3, X_4^3, p^3)$, four right structures are found with a high probability. Hence, the time complexity of this stage is $2^{64} \cdot 2^{25} \cdot 2^{16} = 2^{105}$ partial decryptions, which is negligible compared to Step (3) of the 5.5-round attack.

The rest of the attack is the same as in the 5.5-round attack. However, the time complexity measured in encryption units is reduced, since we still decrypt only 2.375 rounds, while the total number of rounds is increased to six. Hence, the time complexity of the attack is $(0.2 + 0.1 + 0.05 + 0.02 + 0.06) \cdot 2^{128} = 0.43 \cdot 2^{128} = 2^{126.8}$ six-round encryptions. Like in the 5.5-round attack, the memory complexity of the 6-round attack is dominated mostly by the memory required for the data itself, i.e., $2^{64} - 2^{52}$ blocks of 128 bits each. We note that as this value is very close to the entire code book, it can be improved by a factor of 2, by not storing the plaintexts themselves (and keeping only the ciphertexts).

## 5 An Improved 5-Round Attack

In [3] a 5-round attack on IDEA with data complexity of $2^{19}$ known plaintexts and time complexity of $2^{103}$ operations is presented. The attack is based on the Biryukov-Demirci relation, when two plaintexts $(P^1, P^2)$ are used. The relation is used for the four and a half rounds from the beginning of round 4 till after the key addition layer of round 8. For this case the Biryukov-Demirci relation is reduced to:

$$LSB(P_2^1 \oplus P_3^1 \oplus P_2^2 \oplus P_3^2 \oplus \Delta s^4 \oplus \Delta s^5 \oplus \Delta s^6 \oplus \Delta s^7) = LSB(C_2^1 \oplus C_3^1 \oplus C_2^2 \oplus C_3^2), \quad (10)$$

where $P_2^1, P_3^1, P_2^2$, and $P_3^2$ are taken at the beginning of round 4.

By requiring that $\Delta(X_1^4, X_2^4, X_3^4, X_4^4) = (0, \beta, 0, \gamma)$, the attacker ensures that $\Delta s^4 = 0$. Due to the key schedule of IDEA, it is sufficient to guess 103 bits of the key in order to compute $\Delta s^5, \Delta s^6$, and $\Delta s^7$. The attack is then quite a straightforward filtering of wrong subkey guesses which suggest that the Biryukov-Demirci relation does not hold.

In [3], the relation $\Delta(X_1^4, X_2^4, X_3^4, X_4^4) = (0, \beta, 0, \gamma)$ is achieved by using $2^{19}$ known plaintexts, which compose about $2^{37}$ pairs. On average, about 32 satisfy the requirement on $\Delta(X_1^4, X_2^4, X_3^4, X_4^4)$.

[2] We alert the reader to the abuse we have taken in the notations. While $j$ and $j \oplus p^3$ are given after the $KA$ layer of round 3, $X_2^3$ and $X_4^3$ are given before it. Thus, we partially decrypt two words through the $KA$ layer of round 3, and then continue to partially decrypt all four words under the $MA$ layer of round 2.

Our first improvement reduces the data complexity of the attack. We note that $\Delta s^4 = 0$ is satisfied whenever $\Delta p^4 = 0$ holds. In addition, we observe that the subkey $Z_1^4$ is covered by the 103 key bits guessed during the attack. We also note that a difference in the MSB is preserved by addition with an unknown subkey. Using these three observations, we can enlarge the number of plaintext pairs that can be used in the attack. Instead of using only pairs for which $\Delta(X_1^4, X_3^4) = (0,0)$, we can use also pairs for which $\Delta Y_1^4 = 8000_x$ and $\Delta X_3^4 = 8000_x$, since for such pairs we also have $\Delta s^4 = 0$. As a result, we can start with $2^{18.5}$ known plaintexts, and out of the $2^{36}$ possible pairs we can still find 32 pairs satisfying $\Delta s^4 = 0$. The rest of the attack is similar to the attack algorithm of [3].

### 5.1 A 5-Round Attack Using only 16 Known Plaintexts

Our second improvement has much in common with the first improvement. We note that guessing the subkey $Z_3^4$ adds only eleven bits to the total number of guessed key bits. On the other hand, after guessing this subkey, we are able to compute the exact value of $p^4$ for all the plaintexts. Moreover, since the subkey $Z_5^4$ is also covered by the guessed key bits, we are able also to compute the exact values of $s^4$ for all the plaintexts, and hence to compute the value of $\Delta s^4$ for any pair of plaintexts. As a result, all the plaintext pairs, and not only a part of them, can be used to filter subkey candidates. Since each plaintext pair filters about half of the subkey candidates, 16 pairs are sufficient to reduce the number of possible keys to $2^{113}$, and these candidates can be checked by exhaustive key search.

We note that the 16 known plaintexts compose $16 \cdot 15/2 = 120$ pairs. However, only 15 of these pairs can be linearly independent, and hence only 15 pairs can be used for filtering key candidates. Using a smart ordering of the operations, it can be shown that the time complexity of this attack is $2^{114}$ encryptions.

## 6 Summary and Conclusions

In this paper we presented the first known attacks on 5.5-round and 6-round variants of IDEA. Our attack on 6-round IDEA has a time complexity of $2^{126.8}$ encryptions and data complexity of $2^{64} - 2^{52}$ known plaintexts. The attacks exploit three techniques: Constructing a linear equation involving the LSBs of several intermediate encryption values, using Square-like structures, and exploiting the weak key schedule. Each of these techniques was already used to attack reduced variants of IDEA, but the novel combination of the techniques allows to improve the previously known attacks significantly.

We also showed that it possible to attack 5-round IDEA using only 16 known plaintexts with time complexity of $2^{114}$ encryptions. The 5-round attack exploits the weakness of the key schedule of IDEA, that allows to recover many subkeys while guessing only a subset of the key bits.

## References


1. Eyup S. Ayaz, Ali A. Selçuk, *Improved DST Cryptanalysis of IDEA*, proceedings of Selected Areas in Cryptography 2006, to appear, Springer-Verlag.
2. 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.
3. Eli Biham, Orr Dunkelman, Nathan Keller, *New Cryptanalytic Results on IDEA*, Advances in Cryptology, proceedings of ASIACRYPT'06, Lecture Notes in Computer Science 4284, pp. 412–427, Springer-Verlag, 2006.
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. Alex Biryukov, Adi Shamir, *Structural Cryptanalysis of SASAS*, Advances in Cryptology, proceedings of EUROCRYPT '01, Lecture Notes in Computer Science 2045, pp. 394–405, Springer-Verlag, 2001.
8. 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.
9. 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.
10. 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.
11. 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.
12. Joan Daemen, Lars R. Knudsen, Vincent Rijmen, *The Block Cipher Square*, proceedings of Fast Software Encryption 4, Lecture Notes in Computer Science 1267, pp. 149–165, Springer-Verlag, 1997.
13. 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.
14. 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.
15. 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.
16. Philip Hawkes, Luke 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.

17. 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.
18. Lars R. Knudsen, David Wagner, *Integral Cryptanalysis*, proceedings of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 112–127, Springer-Verlag, 2002.
19. 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.
20. Stefan Lucks, *The Saturation Attack — A Bait for Twofish*, proceedings of Fast Software Encryption 8, Lecture Notes in Computer Science 2355, pp. 1–15, Springer-Verlag, 2002.
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., 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.
24. 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*.
25. 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.