A New Meet-in-the-Middle Attack on the IDEA Block Cipher

Hüseyin Demirci[1], Ali Aydın Selçuk[2], and Erkan Türe[3]

[1] Tübitak UEKAE, 41470 Gebze, Kocaeli, Turkey
huseyind@uekae.tubitak.gov.tr
[2] Department of Computer Engineering
Bilkent University, 06800, Ankara, Turkey
selcuk@cs.bilkent.edu.tr
[3] Marmara University, Faculty of Engineering
Göztepe Campus, 81040 Kuyubaşı, İstanbul, Turkey
ture@eng.marmara.edu.tr

Abstract. In this paper we introduce a novel meet-in-the-middle attack on the IDEA block cipher. The attack consists of a precomputation and an elimination phase. The attack reduces the number of required plaintexts significantly for 4 and 4.5 rounds, and, to the best of our knowledge, it is the first attack on the 5-round IDEA.

1 Introduction

Events that happen with probability one (or zero) have been widely used in cryptanalysis. The cryptanalysis of Enigma is a historical example whereas the impossible differential attacks [1, 2] on the block ciphers Skipjack, IDEA, Khufu, and Khafre are more recent examples. In this paper, we present a new attack on the reduced-round versions of IDEA which generalizes the idea of "an event with probability one" to "a set of candidate events with probability one". The attack utilizes a set of events among which one is sure to happen. Key candidates are checked against this criterion, and those which do not realize any of the events are eliminated as wrong candidates.

The attack presented in this paper is a chosen-plaintext, meet-in-the-middle attack consisting of two phases: First there is a precomputation phase where a "sieving set" whose elements are the possible outcomes of a specific event is generated. Then there is the key elimination phase where the candidate keys which do not produce any of the events in the sieving set are eliminated. This attack can be considered as a combination of the Chaum-Evertse's meet in the middle approach [5] and the multi-set approach by Gilbert and Minier [9] to form collisions in the inner rounds of the cipher.

This paper proceeds as follows: In Section 2, we briefly describe the IDEA block cipher. In Section 3, we prove certain properties of IDEA essential to our attack. In Section 4, we develop the attack, beginning with three rounds and gradually advancing it to five rounds. In Section 5, we analyze the complexity of the attack. Finally in Section 6, we conclude the paper with possible directions for future research.

1.1 Notation

Throughout this paper we will be using the following notation: We use the symbol $\oplus$ for the bitwise exclusive-or (XOR) and $\boxplus$ for the modular addition, for both 8- and 16-bit variables; carry$(x \boxplus y)$ denotes the overflow carry bit from the modular addition of x and y. We denote the plaintext by (P_1, P_2, P_3, P_4) and the ciphertext by (C_1, C_2, C_3, C_4) where the separated parts show the 16-bit subblocks. The round numbers are denoted by superscripts. As an example, $C_1^{(2)}$ denotes the first subblock of the ciphertext after 2 rounds. In the i-th round we use six round-key subblocks where $K_1^{(i)}, K_2^{(i)}, K_3^{(i)}, K_4^{(i)}$ are the round key subblocks used in the transformation part and $K_5^{(i)}, K_6^{(i)}$ are the round key subblocks used in the MA-box. The first input of the MA-box $(P_1 \odot K_1) \oplus (P_3 \boxplus K_3)$ is denoted by p and the second input $(P_2 \boxplus K_2) \oplus (P_4 \odot K_4)$ is denoted by q. The output words of the MA-box are denoted by t and u, respectively.

The least significant and the most significant bits of a variable x are denoted by lsb(x) and msb(x), respectively. The least significant eight bits of x are denoted by lsb$_8(x)$ and the most significant eight bits by msb$_8(x)$. The notation $(x|y)$ denotes the concatenation of x and y. Finally, $K_j^{(i)}[m \ldots n]$ means that the round key subblock $K_j^{(i)}$ is being considered which uses the bits from m to n of the master key.

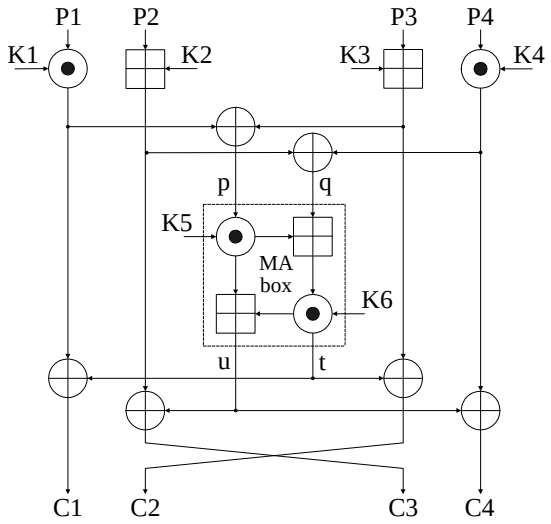

Fig. 1. One round of IDEA.

2 IDEA Block Cipher

The IDEA block cipher is a modified version of the PES block cipher [12, 13]. The main design concept is to mix operations from different algebraic groups. There are three different "incompatible" group operations on 16-bit subblocks: XOR, modular addition, and the "IDEA multiplication", which is a modified multiplication modulo $2^{16}+1$ where 0 is interpreted as 2^{16} to make the multiplication invertible. If the result of the multiplication is 2^{16}, it is converted to 0.

IDEA admits a 128-bit key, has a 64-bit block size and consists of 8.5 rounds. The data is processed in 16-bit words. The round function consists of two parts: First there is a transformation part where each plaintext subblock is operated with a round key subblock, i.e.,

$$T : (P_1, P_2, P_3, P_4) \rightarrow (P_1 \odot K_1, P_2 \boxplus K_2, P_3 \boxplus K_3, P_4 \odot K_4).$$

In the second part, there is a multiplication-addition structure which is called the MA-box. MA-box uses two 16-bit inputs $p = (P_1 \odot K_1) \oplus (P_3 \boxplus K_3)$ and $q = (P_2 \boxplus K_2) \oplus (P_4 \odot K_4)$ to produce two 16-bit output words t and u. The output words are calculated as $t = ((p \odot K_5) \boxplus q) \odot K_6$ and $u = (p \odot K_5) \boxplus t$. Then the outputs of the MA-box are XORed with the outputs of the transformation part, and the two middle subblocks are exchanged. After one round, the ciphertext is of the form (C_1, C_2, C_3, C_4), where

$$\begin{aligned} C_1 &= (P_1 \odot K_1) \oplus t, \\ C_2 &= (P_3 \boxplus K_3) \oplus t, \\ C_3 &= (P_2 \boxplus K_2) \oplus u, \\ C_4 &= (P_4 \odot K_4) \oplus u. \end{aligned}$$

The encryption algorithm consists of eight full rounds and an extra transformation part. The key schedule processes the 128-bit master key into an array of 16-bit round subkeys by cyclic shifts; 16 bits are taken from this array each time a new round key subblock is required in the algorithm.

Decryption is done using the same algorithm, but with different round key subblocks. In the transformation part the multiplicative and additive inverses of the round key subblocks are used, whereas the same key subblocks are used in the MA-Box since it is an involution.

Following its proposal, various cryptanalytic attacks have been applied on reduced-round versions of IDEA. These include differential [14, 6], linear [11], differential-linear and truncated-differential [4], impossible differential [2], and square [15, 8] attack techniques. There are also related key attacks [15], and some classes of weak keys have been observed [7, 10, 3].

Currently the most effective attack on IDEA is due to Biham, Biryukov, and Shamir [2]. They used the impossible differential technique to sieve the key space for 3.5, 4, and 4.5 rounds. The 4.5-round attack requires the encryption of 2^{64} plaintexts which is the whole space.

Table 1 gives a comparison of the performance of the attacks described in this paper and of the attacks developed earlier.

Paper	Rounds	Type	No. of C. Plaintexts	Memory	Total Complexity
[14]	2	differential	2^{10}	2^{32}	2^{42}
[8]	2	square-like	23	small	2^{64}
[14]	2.5	differential	2^{10}	2^{96}	2^{106}
[6]	2.5	differential	2^{10}		2^{32}
[15]	2.5	square	3.2^{16}	small	$3.2^{63} + 2^{48}$
[8]	2.5	square-like	55	small	2^{81}
[4]	3	differential-linear	2^{29}	2^{16}	2^{44}
[8]	3	square-like	71	small	2^{71}
This paper	3	collision	2^{33}	2^{58}	2^{64}
[4]	3.5	truncated-differential	2^{56}	2^{32}	2^{67}
[2]	3.5	impossible-differential	$2^{38.5}$	2^{48}	2^{53}
[8]	3.5	square-like	2^{34}	small	2^{82}
[8]	3.5	square-like	103	small	2^{103}
This paper	3.5	collision	2^{24}	2^{58}	2^{73}
[2]	4	impossible-differential	2^{37}	2^{48}	2^{70}
[8]	4	square-like	2^{34}	small	2^{114}
This paper	4	collision	2^{24}	2^{58}	2^{89}
[2]	4.5	impossible-differential	2^{64}	2^{32}	2^{112}
This paper	4.5	collision	2^{24}	2^{58}	2^{121}
This paper	5	collision	2^{24}	2^{58}	2^{126}

Table 1. Plaintext, memory, and time complexity of chosen plaintext attacks on reduced-round versions of IDEA

3 Some Properties of IDEA

In this section, we present some observations on the IDEA block cipher. Theorem 1 states the main result of this section which plays a key role in our attack:

Theorem 1. *Let $\mathcal{P} = \{(P_1, P_2, P_3, P_4)\}$ be a set of 256 plaintexts such that*

- *P_1, P_3, $\mathrm{lsb}_8(P_2)$ are fixed,*
- *$\mathrm{msb}_8(P_2)$ takes all possible values over $0, 1, \ldots, 255$,*
- *P_4 varies according to P_2 such that $(P_2 \boxplus K_2^{(1)}) \oplus (P_4 \odot K_4^{(1)})$ is fixed.*

For $p^{(2)}$ denoting the first input of the MA-box in the second round, the following properties will hold in the encryption of the set $\mathcal{P}$:

- *$\mathrm{lsb}_8(p^{(2)})$ is fixed,*
- *$\mathrm{msb}_8(p^{(2)})$ takes all possible values over $0, 1, \ldots, 255$.*

Moreover, the $p^{(2)}$ values, when ordered according to their plaintexts' $\mathrm{msb}_8(P_2)$*, beginning with* $\mathrm{msb}_8(P_2) = 0$*, will be of the form*

$$(y_0|z), (y_1|z), \ldots, (y_{255}|z)$$

for some fixed, 8-bit z, and $y_i = (((i \boxplus a) \oplus b) \boxplus c) \oplus d$, for $0 \leq i \leq 255$ and fixed, 8-bit a, b, c, d.

Proof. Consider the input of the second round $(P_1^{(2)}, P_2^{(2)}, P_3^{(2)}, P_4^{(2)})$. We have

$$p^{(2)} \;=\; (P_1^{(2)} \odot K_1^{(2)}) \oplus (P_3^{(2)} \boxplus K_3^{(2)}),$$

where

$$P_1^{(2)} = (P_1 \odot K_1^{(1)}) \oplus t^{(1)},$$
$$P_3^{(2)} = (P_2 \boxplus K_2^{(1)}) \oplus u^{(1)}.$$

Therefore,

$$p^{(2)} \;=\; (((P_1 \odot K_1^{(1)}) \oplus t^{(1)}) \odot K_1^{(2)}) \oplus (((P_2 \boxplus K_2^{(1)}) \oplus u^{(1)}) \boxplus K_3^{(2)})$$

where the only variable term is $\mathrm{msb}_8(P_2)$. If we order the $p^{(2)}$ values of the 256 plaintexts in $\mathcal{P}$ according to their plaintexts' $\mathrm{msb}_8(P_2)$, beginning with $\mathrm{msb}_8(P_2) = 0$, the resulting sequence will be $(y_0|z), (y_1|z), \ldots, (y_{255}|z)$, where

$$z = \mathrm{lsb}_8(((P_1 \odot K_1^{(1)}) \oplus t^{(1)}) \odot K_1^{(2)}) \oplus \mathrm{lsb}_8(((P_2 \boxplus K_2^{(1)}) \oplus u^{(1)}) \boxplus K_3^{(2)})$$

which is a constant over the set $\mathcal{P}$, and

$$y_i = (((i \boxplus a) \oplus b) \boxplus c) \oplus d$$

where the only variable term is i and

$$a = \mathrm{msb}_8(K_2^{(1)}) + \mathrm{carry}(\mathrm{lsb}_8(P_2) \boxplus \mathrm{lsb}_8(K_2^{(1)}))$$
$$b = \mathrm{msb}_8(u^{(1)})$$
$$c = \mathrm{msb}_8(K_3^{(2)}) + \mathrm{carry}(((\mathrm{lsb}_8(P_2) \boxplus \mathrm{lsb}_8(K_2^{(1)})) \oplus \mathrm{lsb}_8(u^{(1)})) \boxplus \mathrm{lsb}_8(K_3^{(2)}))$$
$$d = \mathrm{msb}_8(((P_1 \odot K_1^{(1)}) \oplus t^{(1)}) \odot K_1^{(2)})$$

are constants over $\mathcal{P}$. □

Theorem 2. *In the encryption of the plaintext set $\mathcal{P}$ defined in Theorem 1,* $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)})$ *equals either* $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ *or* $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)}) \oplus 1$ *for all the 256 plaintexts in $\mathcal{P}$.*

Proof. Note that

$$C_2^{(2)} = (((P_2 \boxplus K_2^{(1)}) \oplus u^{(1)}) \boxplus K_3^{(2)}) \oplus t^{(2)},$$
$$C_3^{(2)} = (((P_3 \boxplus K_3^{(1)}) \oplus t^{(1)}) \boxplus K_2^{(2)}) \oplus u^{(2)}.$$

Since the least significant bits of P_2, P_3, $t^{(1)}$, $u^{(1)}$ are all fixed, we have either $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)}) = \mathrm{lsb}(t^{(2)} \oplus u^{(2)})$ or $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)}) = \mathrm{lsb}(t^{(2)} \oplus u^{(2)}) \oplus 1$ for all plaintexts in $\mathcal{P}$. By Lemma 1 of [8], which we also state below, $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)}) = \mathrm{lsb}(t^{(2)} \oplus u^{(2)})$ and the result follows. □

Now observe that in the middle subblocks C_2 and C_3, only addition and XOR operations are used between the ciphertext, the round key, and the MA-box output subblocks. Since the only difference between addition and XOR is the carry bit, some information can leak from these variables. It seems that the variable $\mathrm{lsb}(C_2 \oplus C_3)$ is a candidate for being the Achilles' heel for the IDEA encryption algorithm.

Recall the following lemma from [8]:

Lemma 1. $\mathrm{lsb}(t \oplus u) = \mathrm{lsb}(p \odot K_5)$.

Proof. Since $u = t \boxplus (p \odot K_5)$ and the least significant bit XOR is the same as addition, we have $\mathrm{lsb}(t \oplus u) = \mathrm{lsb}(p \odot K_5)$. □

This property is useful for us because one bit of information related to the MA-box outputs can be obtained using only one input and one round key subblocks. This simple observation will play an important role in the attack.

Corollary 1. $\mathrm{lsb}(C_2^{(i)} \oplus C_3^{(i)} \oplus (K_5^{(i)} \odot (C_1^{(i)} \oplus C_2^{(i)}))) = \mathrm{lsb}(C_2^{(i-1)} \oplus C_3^{(i-1)} \oplus K_2^{(i)} \oplus K_3^{(i)})$.

Proof. By Lemma 1 we have $\mathrm{lsb}(t_i \oplus u_i) = \mathrm{lsb}(K_5^{(i)} \odot (C_1^{(i)} \oplus C_2^{(i)}))$. Consider the middle blocks $C_2^{(i)} = (C_3^{(i-1)} \boxplus K_3^{(i)}) \oplus t_i$ and $C_3^{(i)} = (C_2^{(i-1)} \boxplus K_2^{(i)}) \oplus u_i$. Since the least significant bit addition is equivalent to XOR, we have the result. □

By this corollary, we are able to relate the variables $\mathrm{lsb}(C_2^{(i-1)} \oplus C_3^{(i-1)})$ and $\mathrm{lsb}(C_2^{(i)} \oplus C_3^{(i)})$ of two successive rounds. We can generalize this idea. For two successive rounds, we have the following result:

Corollary 2. $\mathrm{lsb}(C_2^{(i)} \oplus C_3^{(i)} \oplus (K_5^{(i)} \odot (C_1^{(i)} \oplus C_2^{(i)}))) \oplus (K_5^{(i-1)} \odot (C_1^{(i-1)} \oplus C_2^{(i-1)}))) = \mathrm{lsb}(C_2^{(i-2)} \oplus C_3^{(i-2)} \oplus K_2^{(i)} \oplus K_3^{(i)} \oplus K_2^{(i-1)} \oplus K_3^{(i-1)})$.

Proof. Consider the middle blocks in the i-th round,

$$C_2^{(i)} = (((C_2^{(i-2)} \boxplus K_2^{(i-1)}) \oplus u_{i-1}) \boxplus K_3^{(i)}) \oplus t_i,$$
$$C_3^{(i)} = (((C_3^{(i-2)} \boxplus K_3^{(i-1)}) \oplus t_{i-1}) \boxplus K_2^{(i)}) \oplus u_i.$$

By Lemma 1, we have $\mathrm{lsb}(t_{i-1} \oplus u_{i-1}) = K_5^{(i-1)} \odot (C_1^{(i-1)} \oplus C_2^{(i-1)})$ and $\mathrm{lsb}(t_i \oplus u_i) = K_5^{(i)} \odot (C_1^{(i)} \oplus C_2^{(i)})$. Then the result follows. □

We will use Corollary 2 to get information about the second round variables using the output of the fourth round.

4 Attack on IDEA

In this section we describe our attack on IDEA using the results of the previous section. We first give a general outline, then describe the attack in detail.

4.1 The General Outline of the Attack

The first phase of the attack is a precomputation where all possible orderings of $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ are calculated for some particular sequence of plaintexts. Second there is a partial decryption phase. In this phase, different key candidates are tried to partially decrypt some particularly selected plaintext-ciphertext set. When the right key is tried, it is certain that there will be a match between the precomputed set and the set obtained by the partial decryption. Otherwise, it is extremely unlikely that such a match will occur by chance, and this criterion can be used safely to sieve out the wrong key candidates.

The construction of the precomputed set and the set used for the partial decryption is based on the results of the previous section. For a given set of 256 plaintexts as in the hypothesis of Theorem 1, we know, from Theorem 2, that in the second round, the variable $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ can be guessed from the variable $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)})$. Also from Theorem 1, we know that, when ordered according to $\mathrm{msb}_8(P_2)$, the sequence of the 256 $p^{(2)}$s must have the form

$$(y_0|z), (y_1|z), \ldots, (y_{255}|z).$$

In the precomputation phase of the attack, we compute and store all possible 256-bit sequences of the form

$$\mathrm{lsb}(k \odot (y_0|z)), \mathrm{lsb}(k \odot (y_1|z)), \ldots, \mathrm{lsb}(k \odot (y_{255}|z)),$$

for y_i, z as in Theorem 1 and for all possible k values.

On the other hand, we use Corollary 1 or Corollary 2 to make a partial decryption from the end of the cipher to get the bit sequence $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)})$, trying all possible values exhaustively for the key subblocks involved in this partial decryption. If the bit sequence obtained from such a decryption exists in the precomputed set, the key subblocks used for that decryption are a possible candidate. Otherwise, that key can be eliminated. This procedure is repeated until a single[1] candidate key remains.

4.2 Attack on 3-Round IDEA

To attack on 3 rounds of IDEA, we proceed through the following steps:

1. Prepare the *sieving set*, a set of 256-bit strings, as follows:

$$S = \{f(a, b, c, d, z, K_5^{(2)}) : \ 0 \le a, b, c, d, z < 2^8, \ 0 \le K_5^{(2)} < 2^{16}\}$$

[1] Actually, two candidates will remain: The right key and a "conjugate".

where f is a function, mapping a given $(a, b, c, d, z, K_5^{(2)})$ to a 256-bit string, defined bitwise by

$$f(a, b, c, d, z, K_5^{(2)})[i] = \mathrm{lsb}(K_5^{(2)} \odot (y_i | z))$$

for $y_i = (((i \boxplus a) \oplus b) \boxplus c) \oplus d$, $0 \le i \le 255$.

2. Take a set of 2^{24} plaintexts $\mathcal{P} = \{(P_1, P_2, P_3, P_4)\}$ such that P_1, P_3 and the least significant 8 bits of P_2 are fixed, and P_4 and the most significant 8 bits of P_2 take each of the possible 2^{24} values once. Encrypt this set with 3 rounds of IDEA.

3. For each value of $K_2^{(1)}$ and $K_4^{(1)}$, take 256 plaintexts from the set $\mathcal{P}$ such that the most significant 8 bits of P_2 change from 0 to 255 and $(P_2 \boxplus K_2^{(1)}) \oplus (P_4 \odot K_4^{(1)})$ are constant. For each candidate value for $K_5^{(3)}$, calculate

$$\mathrm{lsb}(C_2^{(3)} \oplus C_3^{(3)} \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)}))) \tag{1}$$

over the selected 256 plaintexts.

At this point, if the key value $K_5^{(3)}$ in (1) is correct, the computed $\mathrm{lsb}(C_2^{(3)} \oplus C_3^{(3)} \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)})))$s are all equal either to $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ or to $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)}) \oplus 1$, by Corollary 1.

4. Sort the 256 bits obtained in Step 3 according to the plaintexts' $\mathrm{msb}_8(P_2)$, for $0 \le \mathrm{msb}_8(P_2) \le 255$.

Recall that, by Theorem 2, $\mathrm{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ equals either $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)})$ or $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)}) \oplus 1$, and that the $p^{(2)}$ values, when sorted according to $\mathrm{msb}_8(P_2)$, follow the pattern given in Theorem 1. Therefore, the sorted 256-bit sequence that corresponds to the right choice of $(K_2^{(1)}, K_4^{(1)}, K_5^{(3)})$ must be present in the sieving set S.

Check whether the sorted 256-bit sequence is present in S. If not, eliminate the corresponding key combination $(K_2^{(1)}, K_4^{(1)}, K_5^{(3)})$.[2]

5. If more than two key combinations survive, return to Step 2 and change the plaintext set $\mathcal{P}$. Continue until only two combinations remain.

The attack finds the right combination of $K_2^{(1)}$, $K_4^{(1)}$, and $K_5^{(3)}$ explicitly. The correct value of $K_5^{(2)}$ is found implicitly from the element of the sieving set S that matches the remaining 256-bit string.

[2] As mentioned above, when the correct key values are tried, we will have $\mathrm{lsb}(C_2^{(3)} \oplus C_3^{(3)} \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)})))$ equal to either $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)})$ or $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)}) \oplus 1$. For the former, the 256-bit sequence obviously has to be in S. If the latter is the case, note that $\mathrm{lsb}(K_5^{(2)} \odot p^{(2)}) \oplus 1 = \mathrm{lsb}(k' \odot p^{(2)})$, for all $p^{(2)}$, where $k' = 2^{16} + 1 - K_5^{(2)}$. Hence, the sequence again has to be present in S. (This also implies a conjugate key triple $(K_2^{(1)}, K_4^{(1)}, K_5^{(3)\prime})$ that exists along with the right triple $(K_2^{(1)}, K_4^{(1)}, K_5^{(3)})$ which cannot be eliminated by sieving in S.)

4.3 Attacking the Decryption Operation

A chosen-ciphertext version of this attack is also possible which can be applied on the inverse cipher (i.e., the decryption operation) to obtain additional subblocks of the key. When the number of rounds is not an integer (i.e., 2.5, 3.5, etc.) the attack on the inverse cipher would proceed exactly like the one on the normal cipher, using the decryption subkeys instead of the encryption ones. When the number of rounds is an integer, a slight modification would be needed to deal with the effect of the MA-box half-round at the end. We suggest the following method on the 3-round IDEA, which makes use of the fact that the MA-box operation is an involution. The idea here is to obtain a set of 2^{24} chosen ciphertexts for the output of the first 2.5 rounds of the cipher that conforms to the plaintext specification of Theorem 1. Once such a set is obtained, the attack can be applied to the first 2.5 rounds of the cipher from the reverse direction.

We first generate a set C' of 2^{24} 64-bit blocks in the form of the set $\mathcal{P}$ in Step 2 of the original attack. Then we try the possible values for $K_5^{(3)}$ and $K_6^{(3)}$ and encrypt C' with an MA-box half-round using the guessed $K_5^{(3)}$ and $K_6^{(3)}$ values. Then we decrypt this set with the 3-round IDEA. If the values tried for $K_5^{(3)}$ and $K_6^{(3)}$ are correct, this combined operation of the half-round encryption and the 3-round decryption will be equivalent to a 2.5-round IDEA decryption, and the original attack can be applied in the reverse direction, using C' instead of $\mathcal{P}$. If wrong values are tried for $K_5^{(3)}$ and $K_6^{(3)}$, no meaningful results will be obtained.

Note that in the original attack, $K_5^{(3)}$ was among the key subblocks discovered. Moreover, seven bits of $K_6^{(3)}$, namely $K_6^{(3)}[67 \ldots 73]$, are also known since they are in common with the already discovered key subblock of $K_5^{(2)}$. Hence, it suffices to guess only the remaining nine bits of $K_6^{(3)}$. This makes it possible to launch the attack with a set of $2^9 \times 2^{24} = 2^{33}$ ciphertexts.

This decryption attack ends up discovering the key subblocks of $K_5^{(3)}[51 \ldots 66]$, $K_6^{(3)}[67 \ldots 82]$, $K_2^{(3)}[106 \ldots 121]$, $K_4^{(3)}[10 \ldots 25]$, which, together with the 3-round encryption attack, provides 73 bits of the master key.

4.4 Attack on 3.5-Round IDEA

In the attack on the 3.5-round IDEA, Steps 1 and 2 are identical to that of the 3-round attack, except that the plaintexts are encrypted with 3.5 rounds instead of 3. Then the attack proceeds as follows:

3. As in the 3-round attack, for every value of $K_2^{(1)}$ and $K_4^{(1)}$, take the 256 plaintext blocks from $\mathcal{P}$ that keep $(P_2 \boxplus K_2^{(1)}) \oplus (P_4 \odot K_4^{(1)})$ constant. For every value of the round key subblocks $K_1^{(4)}$ and $K_2^{(4)}$, do a partial decryption of the ciphertexts to obtain the $C_1^{(3)}$ and $C_2^{(3)}$ values. Then calculate, for each candidate $K_5^{(3)}$,

$$\mathrm{lsb}(C_2^{(3.5)} \oplus C_3^{(3.5)} \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)}))). \tag{2}$$

Note that $\text{lsb}(C_2^{(3.5)} \oplus C_3^{(3.5)})$ is either $\text{lsb}(C_2^{(3)} \oplus C_3^{(3)})$ or $\text{lsb}(C_2^{(3)} \oplus C_3^{(3)}) \oplus 1$, and the bit computed in (2) equals either $\text{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ or $\text{lsb}(C_2^{(2)} \oplus C_3^{(2)}) \oplus 1$ for all the ciphertexts. If the choices for $K_2^{(1)}$, $K_4^{(1)}$, $K_1^{(4)}$, $K_2^{(4)}$, and $K_5^{(3)}$ are correct, the derived 256-bit sequence must exist in the set S. Steps 4 and 5 are executed as in the 3-round attack and the key elimination is carried out. The remaining key bits are found with an exhaustive search.

4.5 Attack on 4 and 5 Rounds of IDEA

The attack on the 4-round IDEA follows the same logic. The only difference is in the partial decryption part in Step 3. In this part, we first make a partial decryption to find out $C_1^{(3)}$ and $C_2^{(3)}$ using the round key subblocks $K_1^{(4)}$, $K_2^{(4)}$, $K_5^{(4)}$, and $K_6^{(4)}$. Then we calculate the 256 values of

$$\text{lsb}(C_2^{(4)} \oplus C_3^{(4)} \oplus (K_5^{(4)} \odot (C_1^{(4)} \oplus C_2^{(4)}))) \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)}))). \quad (3)$$

By Corollary 2, we have $\text{lsb}(C_2^{(4)} \oplus C_3^{(4)} \oplus (K_5^{(4)} \odot (C_1^{(4)} \oplus C_2^{(4)}))) \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)})))$ equal either to $\text{lsb}(C_2^{(2)} \oplus C_3^{(2)})$ or to $\text{lsb}(C_2^{(2)} \oplus C_3^{(2)}) \oplus 1$ for all the 256 ciphertexts. From these bits, the 256-bit sequence is produced by sorting the bits according to their plaintexts' $\text{msb}_8(P_2)$; and the key elimination is carried out as in the aforementioned attacks.

To attack on 4.5 and 5 rounds of IDEA, in Step 3 we first make a decryption to reach the outputs of round 4, and then continue the same steps as in the 4-round attack. For 4.5 rounds of IDEA, we search for the round key subblocks $K_1^{(5)}$, $K_2^{(5)}$, $K_3^{(5)}$, and $K_4^{(5)}$, whereas for 5 rounds we search for $K_5^{(5)}$ and $K_6^{(5)}$ in addition to these subblocks to reach the end of round 4.

5 Complexity of The Attack

To assess the elimination power of the attack, we need to calculate the probability of a wrong key avoiding elimination by chance. Given a random sequence of 256 bits, what is the probability that this sequence exists in a set of 2^{56} elements whose elements are again random bit sequences of 256 bits? The probability that two random 256-bit sequences are equal is $1/2^{256}$. The probability that a 256-bit sequence does not exist in a set of 2^{56} elements is $(1-1/2^{256})^{(2^{56})} \approx e^{-2^{-200}} \approx 1$. The probability that all wrong keys will be eliminated by sieving in the set S at the first trial, with n_k denoting the number of key bits tried in the attack, is approximately

$$\left(\left(1 - \frac{1}{2^{256}}\right)^{(2^{56})}\right)^{(2^{n_k})} \approx e^{-2^{-200+n_k}}$$

which is ≈ 1 when $n_k \leq 128$, which will always be the case for the 128-bit key size of IDEA.

Steps 1 and 2 are the same in every attack. In Step 1, we first make 2^{64} precomputations to form the sieving set. We also need 2^{64} bits of memory to store these computations—equivalent to the storage of 2^{58} IDEA blocks. In Step 2 we make 2^{24} encryptions.

In Step 3, for the 3-round version of the attack, we try all possibilities for the round key subblocks,

$$K_2^{(1)}[17\ldots32], K_4^{(1)}[49\ldots64], K_5^{(3)}[51\ldots66].$$

which altogether make up 34 distinct bits of the master key. For each different combination of these key bits, we compute $\mathrm{lsb}(C_2^{(3)} \oplus C_3^{(3)} \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)})))$ over 256 ciphertexts, which makes $2^{34} \times 2^8 = 2^{42}$ computations in total. Moreover, each of the 2^{34} 256-bit strings computed must be compared against the sieving set for a possible match. This can be done efficiently by using a hash table for storing and searching the sieving set. Once the correct key value is found, the key subblock $K_5^{(2)}[58\ldots73]$ can be deduced from the matching string, providing another distinct 7 bits of the master key.

After the 3-round encryption attack is completed, the attack can be repeated on the decryption operation as described in Section 4.3, providing the subkey blocks $K_5^{(3)}[51\ldots66]$, $K_6^{(3)}[67\ldots82]$, $K_2^{(3)}[106\ldots121]$, $K_4^{(3)}[10\ldots25]$. The two attacks together provide 73 bits of the master key with a complexity of about 2^{41} partial encryptions and the remaining 55 key bits can be found by exhaustive search. However, there is also the complexity of computing the sieving set S that needs to be considered. This precomputation phase takes 2^{64} encryptions, dominating the complexity of the 3-round attack.

Consider the attack on 3.5-round IDEA. In Step 3, we use the round key subblocks

$$K_2^{(1)}[17\ldots32], K_4^{(1)}[49\ldots64], K_5^{(3)}[51\ldots66], K_1^{(4)}[83\ldots98], K_2^{(4)}[99\ldots114]$$

to find the sequences. Therefore, there are $2^{66} \times 2^8 = 2^{74}$ partial decryptions and 2^{66} comparisons. Seven additional master key bits will come from $K_5^{(2)}[58\ldots73]$, and the remaining 55 bits will have to be searched exhaustively. In this case, the computational complexity of the attack is dominated by the partial decryption phase. If we consider the complexity of a partial decryption to be half of the complexity of an encryption, the computational complexity of the attack is about 2^{73} encryptions.

Consider the attack on the 4-round IDEA. We use the round key subblocks $K_2^{(1)}$, $K_4^{(1)}$, $K_5^{(3)}$, $K_1^{(4)}$, $K_2^{(4)}$, $K_5^{(4)}$, and $K_6^{(4)}$ for obtaining the bit sequences. Although we are searching seven subblocks, because of the simple cyclic structure of the key schedule, these subblocks provide only 82 distinct bits of the master key. Therefore in the inner-most loop we are doing $2^{82} \times 2^8 = 2^{90}$ partial decryptions and 2^{82} comparisons against the set S. The main work is about 2^{89} encryptions.

For the 4.5 round IDEA, we additionally search for the round key subblocks

$$K_1^{(5)}[76\ldots91], K_2^{(5)}[92\ldots107], K_3^{(5)}[108\ldots123], K_4^{(5)}[124\ldots11]$$

in Step 3 of the attack. Most of these key bits are among those previously mentioned. There are 114 bits to be searched in total. The computational complexity is about $2^{114} \times 2^8 = 2^{122}$ partial decryptions. The data complexity is 2^{24} chosen plaintext blocks, which is a significant reduction from the 2^{64} chosen plaintexts of the best previously known attack [2]. But this reduction is at the expense of computational complexity, which is higher by a factor of 2^9.

Finally, for 5 rounds of IDEA, we also search the key subblocks $K_5^{(5)}[12 \ldots 27]$ and $K_6^{(5)}[28 \ldots 43]$. This brings 5 extra bits to search. The total complexity is about $(2^{119} \times 2^8) = 2^{127}$ partial decryptions. The data complexity is again 2^{24}.

Note that the decryption attack described in Section 4.3 on the 3-round IDEA can also be utilized to obtain additional key bits in higher-round attacks. However, the gain from those decryption attacks would be marginal. This is due to the fact that the encryption attacks on the IDEA versions with more than 3 rounds provide more than half of the 128 key bits with a complexity of between 2^{73}–2^{126} encryptions, making the complexity of searching the remaining key bits in those attacks relatively insignificant.

6 Conclusion

We introduced a new meet-in-the-middle attack against the reduced-round versions of the IDEA block cipher. The 4- and 4.5-round versions of the attack provide a significant reduction in the attack's data complexity over all previously known IDEA attacks. As for the 5-round version, this is the first attack developed against 5 rounds of IDEA faster than exhaustive search, to the best of our knowledge.

It may be possible to generalize the logic of this attack to other ciphers: Choose a group of plaintexts that will guarantee the occurrence of a certain kind of event in the upper part of the cipher. Then, from the lower end of the cipher, search the key bits that would give a partial decryption providing a match with the events expected to happen in the upper part. The key combinations which do not give any such match will be discarded as wrong candidates. The elimination will continue until a single or just a few candidates remain. It is an interesting question how such events can be found and the sieving sets can be constructed for other block ciphers.

Acknowledgments

We would like to thank an anonymous referee for his many insightful comments on the paper, in particular for his suggestion of using the decryption operation for obtaining additional key bits.

References

[1] E. Biham, A. Biryukov, A. Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials*, LNCS 1592, Proceedings of EUROCRYPT' 99, pp. 12-23, Springer-Verlag, 1999.

[2] E. Biham, A. Biryukov, A. Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, LNCS 1636, Proceedings of Fast Software Encryption - 6th International Workshop, FSE' 99, pp. 124-138, Springer-Verlag, 1999.

[3] A. Biryukov, J. Nakahara Jr., B. Preneel, J. Vandewalle, *New Weak-Key Classes of IDEA*, LNCS 2513, ICICS'2002, pp. 315-326, Springer-Verlag, 2002.

[4] J. Borst, L. R. Knudsen, V. Rijmen, *Two Attacks on Reduced IDEA (extended abstract),* LNCS 1223, Advances in Cryptology - Proceedings of EUROCRYPT'97, pp. 1-13, Springer-Verlag, 1997.

[5] D. Chaum, J.H. Evertse, *Cryptanalysis of DES with a Reduced Number of Rounds: Sequences of Linear Factors in Block Ciphers,* LNCS 218, CRYPTO'85, pp. 192-211, Springer-Verlag, 1986.

[6] J. Daemen, R. Govaerts, J. Vandewalle, *Cryptanalysis of 2.5 round of IDEA (extended abstract),* Technical Report ESAC-COSIC Technical Report 93/1, Department Of Electrical Engineering, Katholieke Universiteit Leuven, March 1993.

[7] J. Daemen, R. Govaerts, J. Vandewalle, *Weak Keys of IDEA,* LNCS 773, CRYPTO'93, pp. 224-231, Springer-Verlag, 1994.

[8] H. Demirci, *Square-like Attacks on Reduced Rounds of IDEA*, LNCS 2595, SAC'2002, pp. 147-159, Springer-Verlag, 2003.

[9] H. Gilbert, M. Minier, *A Collision Attack on 7 Rounds of Rijndael,* AES Candidate Conference 2000, pp. 230-241.

[10] P. Hawkes, *Differential-Linear Weak Key Classes of IDEA,* LNCS 1403, EUROCRYPT'98, pp. 112-126, Springer-Verlag, 1998.

[11] P. Hawkes, L. O'Connor, *On Applying Linear Cryptanalysis to IDEA,* LNCS 1163, ASIACRYPT'96, pp. 105-115, Springer-Verlag, 1996.

[12] X. Lai, J. L. Massey, *A Proposal for a New Block Encryption Standard,* LNCS 473, Advances in Cryptology - Proceedings of EUROCRYPT'90, pp. 389-404, Springer-Verlag, 1991.

[13] X. Lai, J. L. Massey and S. Murphy, *Markov Ciphers and Differential Cryptanalysis,* LNCS 547, Advances in Cryptology - Proceedings of EUROCRYPT'91, pp. 17-38,Springer-Verlag, 1991.

[14] W. Meier, *On the Security of the IDEA Block Cipher,* LNCS 765, Advances in Cryptology - Proceedings of EUROCRYPT'93, pp. 371-385, Springer-Verlag, 1994.

[15] J. Nakahara Jr., P.S.L.M. Barreto, B. Preneel, J. Vandewalle, H.Y. Kim, *SQUARE Attacks Against Reduced-Round PES and IDEA Block Ciphers,* IACR Cryptology ePrint Archive, Report 2001/068, 2001.

A Implementing the Attack

It is desirable to verify the correctness of the attack and our estimates for its success probability with an experimental implementation, possibly on a reduced version of IDEA. A major handicap for such an implementation with the 64-bit IDEA is the size of the sieving set, which would take 2^{64} encryptions for creation and 2^{64} bits of memory for storage.

Implementing the attack on a reduced version of IDEA, possibly with an 8-, 16-, or 32-bit block size, can be a more feasible alternative. However, with these block sizes, it turns out that the attack looses almost all of its elimination power: For w denoting the bit length of an IDEA word, i.e., one quarter of a block, the search string in the attack is $2^{w/2}$ bits long, having a domain size of $2^{2^{w/2}}$. On the other hand, the sieving set consists of $2^{3.5w}$ such strings, covering virtually every possibility if we take $w = 2$, 4, or 8, and hence rendering the key elimination phase of the attack useless.

At the time of this writing, no practical ways are known for an experimental testing of the attack, and the authors would gratefully welcome every suggestion on this issue.