Improved DST Cryptanalysis of IDEA

Eyüp Serdar Ayaz and Ali Aydın Selçuk

Department of Computer Engineering
Bilkent University
Ankara, 06800, Turkey
{serdara,selcuk}@cs.bilkent.edu.tr

Abstract. In this paper, we show how the Demirci-Selcuk-Ture attack, which is currently the deepest penetrating attack on the IDEA block cipher, can be improved significantly in performance. The improvements presented reduce the attack's plaintext, memory, precomputation time, and key search time complexities. These improvements also make a practical implementation of the attack on reduced versions of IDEA possible, enabling the first experimental verifications of the DST attack.

1 Introduction

International Data Encryption Algorithm (IDEA) is one of the most popular block ciphers today, commonly used in popular software applications such as PGP. IDEA is known to be extremely secure too: Despite its relatively long history and numerous attempts to analyze it [1, 2, 3, 4, 5, 6, 8, 9, 10, 13, 14, 15], most known attacks on IDEA, which is an 8.5-round cipher, apply to no more than the cipher reduced to 4 rounds. The most effective attack currently known is due to Demirci, Selçuk, and Türe (DST) [7], which is a chosen plaintext attack effective on IDEA up to 5 rounds.

In this paper, we study the ways of enhancing the DST attack and improving its performance. The improvements discussed include shortening the variable part of the plaintexts, reducing the sieving set size, and utilizing previously unused elimination power of the sieving set. The improvements result in a reduction in the plaintext, memory, precomputation time, and key search time complexities of the attack and show that the DST attack can be conducted significantly more efficiently than it was originally thought.

The rest of this paper is organized as follows: In Section 2, we briefly describe the IDEA block cipher. In Section 3, we give an overview of the DST attack. In Section 4, we present several key observations on the DST attack and how to optimize the attack accordingly. In Section 5, we analyze the success probability of the attack according to these optimizations. In Section 6, we present our experimental results and compare them with our theoretical expectations. In Section 7, we calculate the total complexity of the revised attack. Finally in Section 8, we conclude with an overall assessment of the work presented.

E. Biham and A.M. Youssef (Eds.): SAC 2006, LNCS 4356, pp. 1–14, 2007.

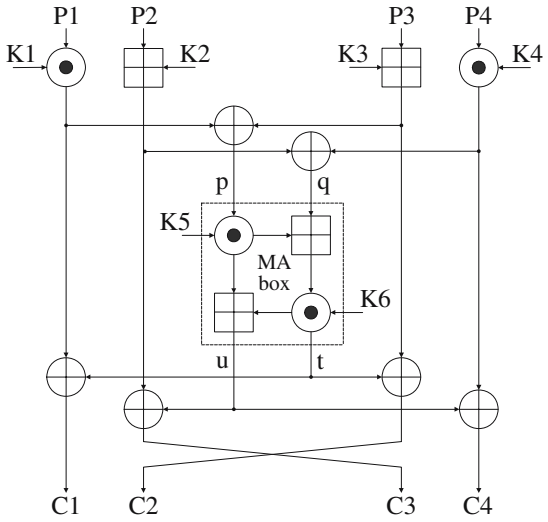

Fig. 1. One round of IDEA

1.1 Notation

We use the following notation in this paper: For modular addition and modular subtraction we use the symbols $\boxplus$ and $\boxminus$ respectively. Bitwise exclusive-or (XOR) is denoted by $\oplus$ and the IDEA multiplication is denoted by $\odot$. The plaintext is shown as (P_1, P_2, P_3, P_4) which is a concatenation of four 16-bit subblocks. Similarly the ciphertext is shown as (C_1, C_2, C_3, C_4). The superscripts in parenthesis denote the round numbers. There are six round-key subblocks for each round which are denoted by $K_1, K_2, K_3, K_4, K_5, K_6$. The inputs of the MA-box are denoted by p and q and the outputs are denoted by u and t.

The least significant bit of a variable x is denoted by $\mathrm{lsb}(x)$, the ith least significant bit is denoted by $\mathrm{lsb}_i(x)$, and the least significant i bits are denoted by $\mathrm{lsbs}_i(x)$. Similarly, the most-significant counterparts of these operators are respectively denoted by $\mathrm{msb}(x)$, $\mathrm{msb}_i(x)$, and $\mathrm{msbs}_i(x)$. Concatenation of two variables x, y is denoted by $(x|y)$. Finally, an inclusive bit interval between the mth and nth bits of a round-key subblock $K_j^{(i)}$ is denoted by $K_j^{(i)}[m \ldots n]$.

2 IDEA Block Cipher

The IDEA block cipher is a modified version of the PES block cipher [11, 12]. IDEA has 64-bit blocks and takes 128-bit keys. The blocks are divided into four 16-bit words and all the operations are on these words. Three different "incompatible" group operations are performed on these words: Bitwise XOR, modular addition, and the *IDEA multiplication*, which is multiplication modulo $2^{16} + 1$ where 0 represents 2^{16}.

There are two parts in an IDEA round. The first is the transformation part:

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

In the second part, two inputs of the MA-box are calculated as $p = (P_1 \odot K_1) \oplus (P_3 \boxplus K_3)$ and $q = (P_2 \boxplus K_2) \oplus (P_4 \odot K_4)$. The outputs of the MA-box are $t = ((p \odot K_5) \boxplus q) \odot K_6$ and $u = (p \odot K_5) \boxplus t$. After these calculations t is XORed with the first and third output of the transformation part and u is XORed with the second and fourth. Finally, the ciphertext is formed by taking the outer blocks directly and exchanging the inner blocks.

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

IDEA consists of eight full rounds and an additional half round, which consists of one transformation part.

The key schedule creates 16-bit round subkeys from a 128-bit master key by taking 16 bits for a subkey and shifting the master key 25 bits after every 8th round key.

Decryption can be done using the encryption algorithm with the multiplicative and additive inverses of the round key subblocks in the transformation part and the same key subblocks in the MA-box.

3 The DST Attack

In this section, we give a brief overview of the DST attack with the relevant properties of the IDEA cipher.

3.1 Some Properties of IDEA

The following are some key observations of Demirci et al. [7] on the IDEA cipher which are fundamental to the DST attack. Proofs can be found in the original paper [7].

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, $\mathrm{lsbs}_8(P_2)$ are fixed,*
- *$\mathrm{msbs}_8(P_2)$ takes all possible values over $0, 1, \ldots, 255$,*
- *P_4 varies according to P_2 such that $q = (P_2 \boxplus K_2^{(1)}) \oplus (P_4 \odot K_4^{(1)})$ 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 $\mathcal{P}$:

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

Moreover, the $p^{(2)}$ values, when ordered according to the plaintext's $\mathrm{msbs}_8(P_2)$ *beginning with* $\mathrm{msbs}_8(P_2) = 0$*, 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.

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 $\mathcal{P}$.*

Lemma 1. *In the IDEA round function, the following property is satisfied:*

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

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

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

3.2 Attack on 3-Round IDEA

The DST attack starts with a precomputation phase where a "sieving set" is prepared which consists of 2^{56} elements of 256-bit strings

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

computed bitwise as

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

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

Once preparation of the sieving set is completed, the main phase of the attack follows. Below is a description of the basic attack on the 3-round IDEA:

1. The attacker takes a chosen plaintext set $\mathcal{R} = \{(P_1, P_2, P_3, P_4)\}$, where P_1, P_3, and $\mathrm{lsbs}_8(P_2)$ are fixed at an arbitrary value, and $\mathrm{msbs}_8(P_2)$ and P_4 take all possible values. All elements of $\mathcal{R}$ are encrypted with the 3-round IDEA.
2. For each value of $K_2^{(1)}$ and $K_4^{(1)}$, take a subset $\mathcal{P}$ of 256 plaintexts from $\mathcal{R}$ such that $\mathrm{msbs}_8(P_2)$ varies from 0 to 255 and P_4 is chosen to make $(P_2 \boxplus K_2^{(1)}) \oplus (P_4 \odot K_4^{(1)})$ constant.
3. For each value of $K_5^{(3)}$, a 256-bit string is formed by computing

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

   for each of the plaintexts in $\mathcal{P}$, ordered by $\mathrm{msbs}_8(P_2)$. If the current $(K_2^{(1)}, K_4^{(1)}, K_5^{(3)})$ triple is correct, this 256-bit string must be found in the sieving set. If it cannot be found, the key triple is eliminated.

4. If many key candidates survive this test, steps 1–3 can be repeated with a different plaintext set $\mathcal{R}$ until a single triple remains. We call one execution of steps 1–3 an *elimination round*.

This attack finds $K_2^{(1)}$, $K_4^{(1)}$, $K_5^{(3)}$ directly by exhaustive search. We can also find $K_5^{(2)}$ indirectly by storing the corresponding $K_5^{(2)}$ value along with each sieving set entry and returning its value in case of a sieving set hit.

3.3 Attack on 3.5-Round IDEA

The 3.5-round attack works similar to the 3-round attack. To find $\mathrm{lsb}(C_2^{(3)} \oplus C_3^{(3)} \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)})))$ we encrypt $\mathcal{P}$ with 3.5-round IDEA and decrypt $C_1^{(3.5)}$ and $C_2^{(3.5)}$ for a half-round by exhaustive search on $K_1^{(4)}$ and $K_2^{(4)}$. It is not necessary to find $C_3^{(3)}$ since $C_2^{(3)} \oplus C_3^{(3)}$ is equal to $C_2^{(3.5)} \oplus C_3^{(3.5)}$ or $C_2^{(3.5)} \oplus C_3^{(3.5)} \oplus 1$ for all 256 ciphertexts.

3.4 Attacks on Higher Number of Rounds

The attack on higher-round IDEA versions utilizes Corollary 2 to find $\mathrm{lsb}(C_2^{(2)} \oplus C_3(2))$ or its complement by computing $\mathrm{lsb}(C_2^{(4)} \oplus C_3^{(4)} \oplus (K_5^{(4)} \odot (C_1^{(4)} \oplus C_2^{(4)}))) \oplus (K_5^{(3)} \odot (C_1^{(3)} \oplus C_2^{(3)}))$.

In the 4-round attack, it is necessary to try exhaustively all possible values of $K_1^{(4)}$, $K_2^{(4)}$, $K_5^{(4)}$, and $K_6^{(4)}$ to find $C_1^{(3)} \oplus C_2^{(3)}$. For the 4.5-round attack, we need to search over $K_1^{(5)}$, $K_2^{(5)}$, $K_3^{(5)}$, $K_4^{(5)}$ to obtain the 4th round outputs. For the 5-round attack, $K_5^{(5)}$, and $K_6^{(5)}$ are also searched.

3.5 Complexity of the DST Attack

In these attacks, the space complexity and precomputation time are independent of the number of rounds while the key search time varies depending on the number of rounds attacked.

Memory required for the attack is determined by the size of the sieving set, which consists of 2^{56} elements of 256-bit strings.

Precomputation time is the time that is needed to prepare the sieving set. We need to calculate the f function once for each bit of the sieving set. There are 2^{56} elements of 256-bit strings, therefore the precomputation time complexity is 2^{64} f computations.

Complexity of the main phase of the attack, the key search time, is different in the 3-, 3.5-, 4-, 4.5- and 5-round attacks depending on the number of key bits searched. In each of these attacks, a lookup string is computed over 256 ciphertexts for each key candidate, contributing a complexity factor of 2^8. In the 3-round attack, the key searched is 34 bits, making the key search time complexity 2^{42} partial decryptions. The 3.5-round attack searches 32 more bits, making the time complexity 2^{74}. The 4-round attack needs 16 more key bits

which raises the time complexity to 2^{90}. We search 114 key bits for the 4.5-round attack and 119 bits for the 5-round attack, with the complexities of 2^{122} and 2^{127} partial decryptions respectively.

4 The Improved DST Attack

In this section we describe the improvements we have made on the DST attack which reduce the precomputation time, key search time, space, and plaintext complexities of the attack.

4.1 Shortening the Variable Parts

The original DST attack partitioned P_2 into 8-bit fixed and 8-bit variable parts, where the variable part took all possible 2^8 values over the chosen plaintext set $\mathcal{P}$. One can observe that in fact it is not necessary to have a balanced partition of P_2 and the attack works just as fine with an imbalanced partition. Accordingly, one can obtain significant savings in the attack by reducing the size of the variable part. For v denoting the number of most significant bits in the variable part of P_2, the sieving set for the attack becomes,

$$S = \{f(a, b, c, d, z, K_5^{(2)}) : \ 0 \leq a, b, c, d < 2^v, \ 0 \leq z < 2^{16-v}, \ 0 \leq K_5^{(2)} < 2^{16}\}.$$

Note that shortening the variable part of P_2 narrows the sieving set both vertically and horizontally. With a v-bit variable part, the sieving set entries will be 2^v bits each instead of 256 bits. Furthermore, the number of entries in the sieving set will be reduced by a factor of $2^{3(8-v)}$. This change also decreases the key search time by 2^{8-v}, since for each candidate key, we encrypt 2^v plaintexts to form the bit string to be searched in the sieving set instead of 256. We will see in Section 5 that having five variable bits is enough for an effective elimination. Therefore by an imbalanced partition of P_2, we obtain an improvement by a factor of 2^9 in precomputation time, 2^3 in key search time and 2^{12} in space.

4.2 Size of the Sieving Set

Another reduction in the size of the sieving set comes from the identical entries yielded by different (a, b, c, d) quadruples, i.e., the *collisions*. In the DST attack all the elements of the sieving set were thought to be distinct [7]. We have found that actually a significant number of collisions exist among the sieving set entries. Some of these collisions were found analytically and some were observed empirically. The analytical findings were obtained according to the y_i values:

Definition 1. *We call two* (a, b, c, d) *quadruples,* $0 \leq a, b, c, d < 2^v$*, equivalent if they give the same* $y_i = (((i \boxplus a) \oplus b) \boxplus c) \oplus d$ *value for all* $0 \leq i < 2^v$*.*

Lemma 2. *For any quadruple* (a, b, c, d)*, complementing the most significant bit of any two or four of* a, b, c, d *yields an equivalent quadruple.*

Proof. We are working in modulo 2^v, so there is no carry bit for addition on the most significant bit. This means changing the most significant bit of a variable in the addition operation changes only the most significant bit of the result. Exclusive-or has the same effect on all bits. So, in an expression of addition and exclusive-or operations, changing one of the variables' most significant bit flips the most significant bit of the result. Changing the most significant bit of an even number of the variables leaves the result unchanged. □

This property gives $\binom{4}{0}+\binom{4}{2}+\binom{4}{4}=8$ equivalent (a,b,c,d) quadruples. Another equivalence is related to the complement operation:

Lemma 3. *(a,b,c,d) is equivalent to $(a,\overline{b},\overline{c}\boxplus 1,\overline{d})$ for $0\le a,b,c,d<2^v$.*

Proof.

$$\begin{aligned}(((i\boxplus a)\oplus\overline{b})\boxplus\overline{c}\boxplus 1)\oplus\overline{d} &= (\overline{((i\boxplus a)\oplus b)}\boxplus\overline{c}\boxplus 1)\oplus\overline{d}\\ &= ((2^v-1-((i\boxplus a)\oplus b))\boxplus(2^v-c))\oplus\overline{d}\\ &= (2^{v+1}-1-(((i\boxplus a)\oplus b)\boxplus c))\oplus\overline{d}\\ &= \overline{(((i\boxplus a)\oplus b)\boxplus c)}\oplus\overline{d}\\ &= (((i\boxplus a)\oplus b)\boxplus c)\oplus d\end{aligned}$$ □

This relation can be applied to the 8 equivalent quadruples found in Lemma 1 yielding 16 equivalent quadruples.

The third equivalence is related to the second most significant bit:

Lemma 4. *(a,b,c,d) is equivalent to $(a\boxplus 2^{v-2},b,c\boxplus 2^{v-2},d)$ if $\mathrm{msb}_2(b)=1$, and to $(a\boxplus 2^{v-2},b,c\boxminus 2^{v-2},d)$ if $\mathrm{msb}_2(b)=0$.*

Proof. Assume $\mathrm{msb}_2(b)=1$ and consider $((((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxplus 2^{v-2})\boxplus c$. Obviously $\mathrm{msb}_2((i\boxplus(a\boxplus 2^{v-2}))\oplus b)=\mathrm{msb}_2(i\boxplus a)$. As for the most significant two bits, if there is a carry in the outer addition of $(i\boxplus a)\boxplus 2^{v-2}$, there will also be a carry on the outmost addition of $(((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxplus 2^{v-2}$ since $\mathrm{msb}_2(b)=1$. Similarly, if there is no carry in the outer addition of $(i\boxplus a)\boxplus 2^{v-2}$, there will also be no carry on the outmost addition of $(((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxplus 2^{v-2}$. So the most significant bit of the result is not changed. The second most significant bit is complemented twice, so it also remains same. Hence in both cases $((i\boxplus(a\boxplus 2^{v-2}))\oplus b)\boxplus(c\boxplus 2^{v-2})=((i\boxplus a)\oplus b)\boxplus c$.

Now, assume $\mathrm{msb}_2(b)=0$ and consider $((((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxminus 2^{v-2})\boxplus c$. Obviously $\mathrm{msb}_2((i\boxplus(a\boxplus 2^{v-2}))\oplus b)=\mathrm{msb}_2(\overline{i\boxplus a})$. As for the most significant two bits, if there is a carry in the outer addition of $(i\boxplus a)\boxplus 2^{v-2}$, then there will be no carry on the outmost addition of $(((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxplus 2^{v-2}$ since $\mathrm{msb}_2(b)=0$. Similarly, if there is no carry in the outer addition of $(i\boxplus a)\boxplus 2^{v-2}$, then there will be a carry on the outmost addition of $(((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxplus 2^{v-2}$. So the most significant bit of the result is changed in the operation $((((i\boxplus a)\boxplus 2^{v-2})\oplus b)\boxminus 2^{v-2})$. Adding 2^{v-1} will neutralize this, so the most significant bit of the result will remain the same. The second most significant bit is complemented twice, so it will be unchanged. Hence in both cases $((i\boxplus(a\boxplus 2^{v-2}))\oplus b)\boxplus(c\boxplus 2^{v-2}\boxplus 2^{v-1})$ $=((i\boxplus(a\boxplus 2^{v-2}))\oplus b)\boxplus(c\boxminus 2^{v-2})=((i\boxplus a)\oplus b)\boxplus c$. □

When Lemma 4 is applied to all 16 equivalent quadruples, the size of the equivalence class is doubled, yielding 32 equivalent quadruples.

If we discard the two most significant bits of a and one most significant bit of b, c, d, we will find exactly one of these 32 equivalent quadruples, since the equivalent quadruples take all possible values over these five bits. Therefore, in the sieving set formation phase we do not have to search all combinations of (a, b, c, d); conducting the search on $\text{lsbs}_{v-2}(a)$, $\text{lsbs}_{v-1}(b)$, $\text{lsbs}_{v-1}(c)$, $\text{lsbs}_{v-1}(d)$ suffices. This reduction decreases both the precomputation time and the sieving set size by a factor of 2^5.

The collisions we dealt with in this section are exclusively based on equivalent (a, b, c, d) quadruples. As the experimental results in Section 6 show, there are other collisions as well and the actual collision rate can safely be assumed to be 2^6 or higher.

4.3 Indirect Elimination Power from the Sieving Set

The effectiveness of the DST attack can be improved significantly by using previously unutilized elimination power from the sieving set. When a lookup string is matched with a sieving set entry, we can do a further correctness test on the key by checking whether the key values used in obtaining the set entry matched are consistent with the round keys used in obtaining the lookup string.

First, we can check the $K_5^{(2)}$ found in a sieving set hit for consistency with the keys used in the partial decryption. The 3-round attack searches $K_2^{(1)}[17 \dots 32]$, $K_4^{(1)}[49 \dots 64]$, $K_5^{(3)}[51 \dots 66]$, which intersects with $K_5^{(2)}[58 \dots 73]$ on 9 bits over $[58 \dots 66]$. If we store the values of these nine bits of $K_5^{(2)}$ for each sieving set entry and compare them to the corresponding bits of the key candidate used in the partial decryption in case of a hit, a wrong key's chances of passing the sieving test will be reduced by a factor of 2^9.

The keys found in further round attacks—$K_1^{(4)}$, $K_2^{(4)}$ for 3.5-round attack, $K_5^{(4)}$, $K_6^{(4)}$ for 4-round attack, $K_1^{(5)}$, $K_2^{(5)}$, $K_3^{(5)}$, $K_4^{(5)}$ for 4.5-round attack and $K_5^{(5)}$, $K_6^{(5)}$—do not bring us any more bits intersecting with $K_5^{(2)}$.

The seven bits of $K_5^{(2)}$ that do not intersect with the searched round keys can be utilized to deduce the corresponding seven bits of the master key. Moreover, in attacks that use multiple elimination rounds, a check on these bits can be carried out to test the consistency of the sieving set hits across different elimination rounds. Either way, these seven bits can be used to reduce the set of key candidates by a factor of 2^7 per elimination round.

A similar consistency check can be applied also on the a values of the sieving set entries. Note that the 32 equivalent quadruples found in Section 4.2 have the same $\text{lsbs}_{v-2}(a)$ value. Hence, in case of a sieving set hit, the a value of the sieving set entry matched can be compared on the $v-2$ low order bits to the a value of the partial decryption,

$$a \;=\; \text{msbs}_v(K_2^{(1)}) + \text{carry}(\text{lsbs}_{16-v}(P_2) \boxplus \text{lsbs}_{16-v}(K_2^{(1)})),$$

which is fixed and known over the plaintext set $\mathcal{P}$. This extension brings an extra elimination power of 2^{v-2} to the attack while costing $v-2$ bits of storage per sieving set entry.

A similar check can be carried out over the c values. The 32 equivalent quadruples are equal to $\pm c \bmod 2^{v-2}$ over $\mathrm{lsbs}_{v-2}(c)$ while $\mathrm{msbs}_2(c)$ takes all possible four values. Moreover, for every value of c there are two possible values of $\mathrm{msbs}_v(K_3^{(2)})$ since

$$c = \mathrm{msbs}_v(K_3^{(2)}) + \mathrm{carry}(((\mathrm{lsbs}_{16-v}(P_2) \boxplus \mathrm{lsbs}_{16-v}(K_2^{(1)})) \oplus \mathrm{lsbs}_{16-v}(u^{(1)})) \boxplus \mathrm{lsbs}_{16-v}(K_3^{(2)}))$$

where the carry bit is an unknown. The key bits $\mathrm{msbs}_v(K_3^{(2)})$ are covered completely by $K_2^{(1)}$ for $v \leq 7$ which is the case in our attacks. Therefore, by conducting a consistency check between the key candidate tried and the c value of the sieving set entry matched, we can reduce the number of keys by an additional factor of 2^{v-4}. As in the case of a, this check on c costs an extra $v-2$ bits of storage per sieving set entry.

5 The Success Probability

As discussed in Section 4, we have found the actual size of the sieving set to be about 2^6 times smaller than what was thought previously, due to the collisions among the set entries. Hence, with a v-bit variable part of P_2, the expected size of the sieving set is about 2^{26+3v}. When a wrong key is checked against the sieving set, the probability of two random 2^v-bit strings matching by chance is 2^{-2^v}. With the indirect elimination power from $K_5^{(2)}$, $\mathrm{lsbs}_{v-2}(a)$, and $\mathrm{lsbs}_{v-2}(c)$, the probability of a random match between the lookup string and a particular sieving set entry is further reduced to $2^{-(2^v+2v+3)}$. Hence, the probability of a wrong key's passing the test (i.e., matching at least one entry in the sieving set) is now reduced to

$$1 - \left(1 - \frac{1}{2^{2^v+2v+3}}\right)^{(2^{26+3v})} \approx 2^{-2^v+v+23}$$

for a given v. Accordingly, $v = 5$ is the smallest value of v that gives a non-negligible elimination power, where a wrong key's probability of passing the test is 2^{-4}. This probability drops substantially by increasing v: For $v = 6$, it becomes 2^{-33}; for $v = 7$ it is 2^{-95}, and for $v = 8$ it is 2^{-221}.

The probability of elimination discussed above is for attacks with one elimination round (i.e., one pass of Steps 1–3 of the attack algorithm). In attacks that use several elimination rounds, a consistency check on $K_5^{(2)}[67 \ldots 73]$ is also possible in the elimination rounds after the first one. In this case, the probability of a wrong key's having a consistent match with a sieving set entry is further

Table 1. The actual sieving set sizes for 32-bit IDEA ($w = 8$) with $v = 5$. Each column shows the results for a particular combination of LS, $K_5^{(2)}$, a, c, included in the set entries. As more information is included, the collision rate approaches to the theoretical expectation given in the last column.

	LS	LS, $K_5^{(2)}$	LS, $K_5^{(2)}, a$	LS, $K_5^{(2)}, a, c$	$2^{2w+3v-6}$
$v = 5$	$2^{22.3}$	$2^{23.6}$	$2^{24.5}$	$2^{24.7}$	2^{25}

reduced to $2^{-(2^v+2v+10)}$. Hence, the probability of a wrong key's passing such an elimination round is

$$1 - \left(1 - \frac{1}{2^{2^v+2v+10}}\right)^{(2^{26+3v})} \approx 2^{-2^v+v+16}.$$

The probability of a wrong key's passing an elimination test with r rounds is therefore

$$2^{(-2^v+v+23)+(r-1)(-2^v+v+16)} = 2^{r(-2^v+v+16)+7}.$$

To successfully conclude an attack, we will need to run as many elimination rounds as needed to reduce the number of surviving key candidates to one. In the 3-round attack, 34 key bits are searched giving 2^{34} candidates in total. For $v = 5$, the probability of a wrong key's not being eliminated after r iterations is 2^{-11r+7}. Hence, four elimination rounds would suffice to eliminate virtually all wrong keys while keeping $v = 5$ in the 3-round attack. Similarly, two elimination rounds would suffice for $v = 6$ and one elimination round for $v = 7$.

6 Experimental Results

The improvements obtained have made a practical implementation of the DST attack possible on reduced versions of IDEA. We tested the attack on IDEA reduced to 3 rounds with a block size of 32 bits (i.e., word size $w = 8$). The key size is reduced accordingly to 64 bits; the key schedule rotates the master key 11 bits after every 8th subkey produced. The attack is tested with $v = 5$, since $v \geq 6$ is still beyond our limits of feasibility, and $v \leq 4$ does not produce a meaningful attack as the lookup string length, 2^v, is too short to give any significant elimination.

First we tested the size of the sieving set in comparison to our theoretical expectation $2^{2w+3v-6}$. The results, summarized in Table 1, show that the actual sieving set size is somewhat further smaller than our expectation due to unaccounted collisions, by a factor of 8 to 1.5, depending on the amount of extra information included—$K_5^{(2)}$, a,or c.[1]

[1] Tests were carried out for other combinations of $K_5^{(2)}$, a, and c not listed in Table 1 as well. Due to space limitations, only the most essential ones are listed here, according to their order of significance.

Table 2. The experimental results for the DST attack with $v = 5$. The results in the table are the ratio of wrong keys passing the sieving set test uneliminated, obtained over 1000 runs of the attack, each containing 2^{18} keys tested. The theoretical results are the calculations in Section 5 according to the actual sieving set sizes in Table 1.

	LS	LS, $K_5^{(2)}$	LS, $K_5^{(2)}, a$	LS,$K_5^{(2)}, a, c$
Theoretical	$2^{-9.7}$	$2^{-13.4}$	$2^{-15.5}$	$2^{-16.3}$
Empirical	$2^{-9.6}$	$2^{-12.3}$	$2^{-13.2}$	$2^{-13.9}$

We implemented the DST attack with $v = 5$ to see its actual success. Table 2 summarizes the result of these tests, where the wrong keys are eliminated according to the lookup string (LS), $K_5^{(2)}$, a, and c; and the ratio of the uneliminated ones are listed. The test results are compared to the theoretical results calculated in Section 5.

An analysis of the experimental results reveals several key points. First and foremost, the DST attack works as expected. Especially when only LS is used in elimination, the expected and the actual results are almost identical. When $K_5^{(2)}$, a, and c are also included in the process, the power of the attack is significantly boosted. There appears to be a slight deviation from the expectations however, which probably results from some subtle correlations involved. Accordingly, there may be a few wrong keys left at the end of the attack, which can easily be removed by an extra elimination round or by exhaustive search.

7 Complexity of the Attack

The optimizations discussed in this paper provide significant reductions in the space, precomputation time, and key search time complexities of the DST attack. Space complexity of the attack is mainly the size of the sieving set. Each sieving set entry contains a 2^v-bit lookup string. Additionally we need to store the $K_5^{(2)}$, $\mathrm{lsbs}_{v-2}(a)$, and $\mathrm{lsbs}_{v-2}(c)$ values to have the extra elimination power, which costs us an extra $12+2v$ bits per entry. The number of entries in the set is about 2^{3v+26}. Thus the overall space requirement of the sieving set is $2^{3v+26} \cdot (2^v + 2v + 12)$ bits. In terms of the IDEA block size, this is less than 2^{41} IDEA blocks for $v = 5$.

Precomputation time complexity is the time required to calculate the sieving set. We need to compute the f function 2^v times for each sieving set entry. The number of entries calculated for the sieving set is $2^{3v+32-5}$ since the most significant bits of a, b, c, d and the second most significant bit of a need not to be searched. Hence the precomputation time complexity is 2^{4v+27} f computations which is roughly equivalent to 2^{4v+26} IDEA rounds. The precomputation time is the dominant time complexity only for the 3-round attack.

Key search time complexity depends on both the number of rounds attacked and the number of variable bits in P_2. For each candidate key set, we take 2^v values of $\mathrm{msbs}_v(P_2)$ and calculate the lookup string by partial decryptions. This procedure may need to be repeated several times if the attack requires multiple elimination rounds.

Table 3. A comparison of the complexities of the basic DST attack and the optimized version. The space complexity figures are in terms of one IDEA block (64 bits). The unit of precomputation time complexity is one computation of the f function. The key search complexities are compared in terms of the number of partial decryptions to be executed. The optimized attack figures are given for $v = 5, 6, 7$ which yield the best results.

	DST	$v = 5$	$v = 6$	$v = 7$
Space complexity	2^{58}	2^{41}	2^{45}	2^{49}
Precomputation	2^{64}	2^{47}	2^{51}	2^{55}
Key search, 3-round	2^{42}	2^{39}	2^{40}	2^{41}
3.5-round	2^{74}	2^{71}	2^{72}	2^{73}
4-round	2^{90}	2^{87}	2^{88}	2^{89}
4.5-round	2^{122}	2^{119}	2^{120}	2^{121}
5-round	2^{127}	2^{124}	2^{125}	2^{126}

Table 4. Plaintext complexities of the DST attack for different v. The improvements over the original attack ($v = 8$) in this respect, although non-trivial, is relatively less significant compared to the other improvements.

Attack	$v = 5$	$v = 6$	$v = 7$	$v = 8$
3-round	2^{23}	2^{22}	2^{23}	2^{24}
3.5-round	$2^{23.6}$	2^{23}	2^{23}	2^{24}
4-round	2^{24}	2^{23}	2^{23}	2^{24}
4.5-round	$2^{24.6}$	$2^{23.6}$	2^{23}	2^{24}
5-round	$2^{24.6}$	$2^{23.6}$	2^{23}	2^{24}

The effect of multiple elimination rounds on the attack's complexity is two fold. First, a different plaintext set $\mathcal{R}$ would be needed for each elimination round, making the total plaintext complexity of the attack $r \cdot 2^{16+v}$ for r denoting the number of elimination rounds to be applied. Second, the complexity of the key search phase would increase due to multiple repetitions of the elimination procedure. However, this increase can be expected to be relatively marginal, since the extra elimination rounds will be applied only to the keys that have passed the previous tests. Given that each elimination round will remove the vast majority of the wrong keys, the additional time complexity from the extra elimination rounds will be negligible.

The space and time complexities of the optimized DST attack in comparison to the basic attack are summarized in Table 3; the plaintext complexities are given in Table 4.

8 Conclusion

In this paper, we described several improvements on the DST attack [7] on IDEA and showed how the attack can be made significantly more efficient. The

improvements reduce the plaintext, memory, precomputation, and the time complexity of the attack. The new attack becomes the most efficient attack on all these four accounts on the 4.5- and 5-round IDEA, and the most efficient in plaintext complexity on the 4-round cipher along with [10].

With the current improvements, a practical implementation of the attack has also become feasible and we provided the first experimental verifications of the DST attack.

An even more significant improvement on the DST attack would be to extend it beyond 5 rounds of IDEA. Unfortunately, the round keys that need to be tried exhaustively in the partial decryption phase covers all the 128 key bits in the 5.5-round or higher round versions of the attack. Hence, no matter how much improvement is achieved on the core section of the attack, the overall attack cannot be made perform faster than exhaustive search on 5.5 or more rounds. We leave it as an open research problem to make the fundamental ideas of the DST attack work effectively on 5.5 or more rounds of the IDEA cipher.

Acknowledgments

We would like to thank Hüseyin Demirci for several helpful suggestions and comments on this paper.

References

[1] Biham, E., Biryukov, A., Shamir, A.: Miss in the Middle Attacks on IDEA and Khufu. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 124–138. Springer, Heidelberg (1999)

[2] Biryukov, A., Nakahara Jr., J., Preneel, B., Vandewalle, J.: New Weak-Key Classes of IDEA. In: Deng, R.H., Qing, S., Bao, F., Zhou, J. (eds.) ICICS 2002. LNCS, vol. 2513, pp. 315–326. Springer, Heidelberg (2002)

[3] Borst, J., Knudsen, L.R., Rijmen, V.: Two Attacks on Reduced IDEA (extended abstract). In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 1–13. Springer, Heidelberg (1997)

[4] Daemen, J., Govaerts, R., Vandewalle, J.: 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)

[5] Daemen, J., Govaerts, R., Vandewalle, J.: Weak Keys of IDEA. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 224–231. Springer, Heidelberg (1994)

[6] Demirci, H.: Square-like Attacks on Reduced Rounds of IDEA. In: Nyberg, K., Heys, H.M. (eds.) SAC 2002. LNCS, vol. 2595, pp. 147–159. Springer, Heidelberg (2003)

[7] Demirci, H., Selçuk, A.A., Türe, E.: A New Meet-in-the-Middle Attack on the IDEA Block Cipher. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 117–129. Springer, Heidelberg (2004)

[8] Hawkes, P.: Differential-Linear Weak Key Classes of IDEA. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 112–126. Springer, Heidelberg (1998)

[9] Hawkes, P., O'Connor, L.: On Applying Linear Cryptanalysis to IDEA. In: Kim, K.-c., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 105–115. Springer, Heidelberg (1996)

[10] Junod, P.: New attacks against reduced-round versions of IDEA. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 384–397. Springer, Heidelberg (2005)
[11] Lai, X., Massey, J.L.: A Proposal for a New Block Encryption Standard. In: Damgård, I.B. (ed.) EUROCRYPT 1990. LNCS, vol. 473, pp. 389–404. Springer, Heidelberg (1991)
[12] Lai, X., Massey, J.L., Murphy, S.: Markov Ciphers and Differential Cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)
[13] Meier, W.: On the Security of the IDEA Block Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 371–385. Springer, Heidelberg (1994)
[14] Nakahara Jr., J., Barreto, P.S.L.M., Preneel, B., Vandewalle, J., Kim, H.Y.: Square Attacks Against Reduced-Round PES and IDEA Block Ciphers. In: 23rd Symposium on Information Theory in the Benelux. Louvain-la-Neuve, pp. 187–195 (2002)
[15] Nakahara, J., Preneel, B., Vandewalle, J.: The Biryukov-Demirci attack on reduced-round versions of IDEA and MESH block ciphers. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 98–109. Springer, Heidelberg (2004)