Differential Cryptanalysis of the ICE Encryption Algorithm

Bart Van Rompay[1]*, Lars R. Knudsen[2]**, Vincent Rijmen[1]***

[1] K.U. Leuven, ESAT-COSIC, K. Mercierlaan 94, B-3001 Heverlee, Belgium
{bart.vanrompay,vincent.rijmen}@esat.kuleuven.ac.be
[2] Dept. of Informatics, University of Bergen, Hi-techcenter, N-5020 Bergen, Norway
larsr@ii.uib.no

Abstract. ICE is a 64-bit block cipher presented at the Fast Software Encryption Workshop in January 1997. It introduced the concept of a keyed permutation to improve the resistance against differential and linear cryptanalysis. In this paper we will show however that we can use low Hamming weighted differences to perform a practical, key dependent, differential attack on ICE. The main conclusion is that the keyed permutation is not as effective as it was conjectured to be.

1 The ICE Algorithm

ICE [7], which stands for *Information Concealment Engine*, is a 64-bit Feistel block cipher with a structure similar to DES, the *Data Encryption Standard* [5]. The standard ICE algorithm takes a 64-bit key and uses 16 subkeys in 16 rounds. There is a fast variant, Thin-ICE, which uses 8 rounds with a 64-bit key, and there are open-ended variants ICE-n which use $16n$ rounds and $64n$-bit keys.

Description of the round function The ICE round function F maps 32-bit inputs to 32-bit outputs, using a 60-bit subkey. First the 32-bit input is expanded to a 40-bit value. A 20-bit subkey performs a keyed permutation and a 40-bit subkey is exored to the resulting value. Finally it uses four 10 to 8-bit S-boxes and a permutation to obtain the 32-bit result of the round function.

Notation In this paper bits are numbered from right to left, starting at bit zero. So the rightmost bit of an n-bit value V is V_0, while the leftmost bit is V_{n-1}. The four S-boxes used in the round function are labeled $S0$, $S1$, $S2$ and $S3$.

* Sponsored by the Timesec project of the Federal Office for Scientific, Technical and Cultural Affairs (OSTC), Belgium.

** This author's work was done during his stay in Leuven as a postdoctoral fellow of the Research Council of the K.U. Leuven.

*** F.W.O. research assistant, sponsored by the Fund for Scientific Research, Flanders — Belgium.

The expansion function The 32-bit input I to the F function is expanded to four 10-bit values $E0, E1, E2, E3$.

The keyed permutation A 20-bit subkey performs a keyed permutation on the expanded 40-bit text, swapping bits between $E0$ and $E2$, and between $E1$ and $E3$. If permutation key bit $10 + i (i < 10)$ is set, bit i of $E0$ and $E2$ will be swapped. If permutation key bit $i (i < 10)$ is set, bit i of $E1$ and $E3$ will be swapped.

The exor operation The 40-bit result from the keyed permutation is exored with a 40-bit subkey.

The S-boxes ICE uses four 10 to 8-bit S-boxes to map the 40-bit value to a 32-bit value. These S-boxes are similar in structure to those used in LOKI [4, 3] in their use of Galois Field exponentiation. From the 10-bit input X, we concatenate X_9 and X_0 to form the row selector R. Bits $X_8 ... X_1$ form the column selector C. For each row there is a XOR offset value O_R, and a Galois Field prime (irreducible polynomial) P_R. The 8-bit output of an S-box for an input X is given by $(C \oplus O_R)^7 \bmod P_R$, under Galois Field arithmetic.

The permutation function Finally the four 8-bit S-box outputs are combined via a P-box into the 32-bit output of the F function.

The key schedule The ICE key scheduling algorithm maps a 64-bit key to 16 60-bit subkeys. Each subkey bit is dependent on only one key bit. The Thin-ICE key schedule is simply the first eight rounds of the standard ICE key schedule. The ICE-n key schedules build on the ICE key schedule and map a $64n$-bit key to $16n$ 60-bit subkeys.

2 Differential Cryptanalysis

Differential cryptanalysis was introduced by Biham and Shamir [2], and can be used to perform chosen plaintext attacks. The basic idea is that two chosen plaintexts with a certain difference $P' = P_1 \oplus P_2$ can encipher to two ciphertexts such that $C' = C_1 \oplus C_2$ has a specific value with non-negligible probability, and such a characteristic (P', C') is useful in deriving certain bits of the key. The heart of differential attacks is the finding and the use of characteristics with high probabilities.

The analysis of ICE in [7] considers only symmetric differences, which have equal left and right 16-bit halves of the 32-bit input to the F function. This is claimed to be the best strategy since they are the only differences not affected by the keyed permutation. As a consequence the attacker has to target at least two S-boxes at a time and the probabilities are too low to be used in a realistic attack.

The approach used in our attack is to use differences with a low Hamming weight (as low as possible). Whether they will be affected by the keyed permutation depends on the values of only a few key bits. The differences used address only one S-box in the round function. In this way we can find characteristics with a probability high enough to (theoretically) recover ICE keys for the algorithm reduced to 15 rounds, in time less than the expected cost for exhaustive search. The complication is that the attack becomes key dependent.

3 Differential Characteristics for ICE

As explained in the previous section we will focus on low Hamming weighted differences that address only one S-box in the round function. It is not possible to build a 2-round iterative characteristic like the one used for the analysis of DES [2], with a difference that addresses only one S-box (using only the middle 6 bits out of the 10 input bits to that S-box so that it is not affected by the expansion in the round function). We can however build 3-round iterative characteristics of the form specified in Figure 1.

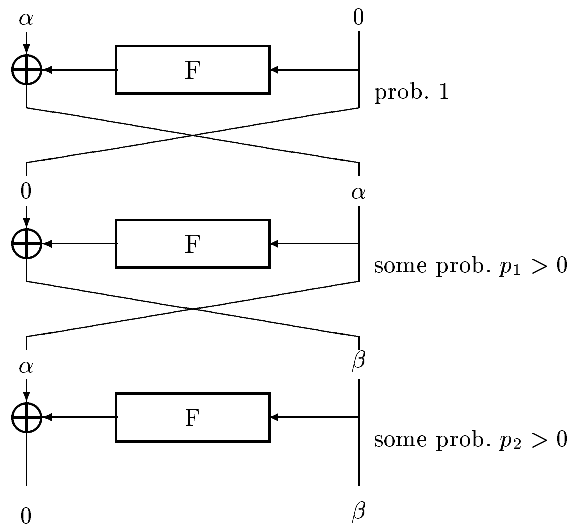

Fig. 1. 3-round iterative characteristic.

Because we restrict the differences α and β to one S-box, they can have a Hamming weight of no more than 4 each, since the 8-bit output of an S-box delivers, after the permutation and the key dependent permutation in the next round, up to 4 bits to an S-box in the next application of the round function.

The characteristic will be valid if α and β are not affected by the keyed permutation in the corresponding rounds. This happens if the permutation key bits used in the bit positions that are set in α and β, are equal to zero (so the difference will not be permuted from the left 20-bit half of the expanded text to the right or vice versa).

There are also characteristics that are valid only if certain permutation key bits are equal to one (corresponding to the bits set in α or β or both). In general we call these *conditional* characteristics (cf. the attack on Lucifer [1]), which have a certain probability with respect to a subset of the key space. Their usage is advisable when they improve the probability over the best probability of a non-conditional characteristic by a factor higher than the inverse of the *key fraction* (the ratio between the size of the subset and the size of the key space), especially if several such characteristics can efficiently share the same structure of chosen plaintexts.

If we consider only differences with Hamming weight one there is a total of 105 conditional characteristics with

$$2^{-13} \geq p_1 \cdot p_2 \geq 2^{-18}.$$

Table 1 lists some of the differences of Hamming weight 1, which can be used to construct a 3-round characteristic with probability $p_1 \cdot p_2 \geq 2^{-15}$, together with the corresponding probabilities. By interchanging the values of α and β we get twice the number of characteristics (except for the fourth entry in the table which has $\alpha = \beta$).

α	β	p_1 (-$\log_2$)	p_2 (-$\log_2$)	round 2	round 3
18	28	6	7	0	0
26	29	6	7.4	1	1
31	26	8	6	1	0
7	7	7	7	0	0
3	10	7.4	7.4	0	0
27	3	7.4	7.4	1	1
4	22	7.4	7.4	1	0
18	30	6	9	1	0
15	23	7	8	0	0
22	7	7	8	1	1

Table 1. Differential characteristics with Hamming weight 1 and $p_1 \cdot p_2 \geq 2^{-15}$. The differences α and β are noted by the bit position in the 32-bit value that is set at one. We also list the required value for the permutation key bits corresponding to α and β in the second and third round of the characteristic.

4 Differential attack on ICE

4.1 Attack on ICE with 6 rounds

We use the best 3-round characteristic, with $\alpha = 28$ and $\beta = 18$, followed by a trivial round with probability 1. The probability for this 4-round characteristic is $p_1 \cdot p_2 = 2^{-7} \cdot 2^{-6} = 2^{-13}$. It is valid if the bits set in the differences α and β are not permuted in the respective rounds (rounds 2 and 3). This can be translated to the following conditions:

- bit 14 = 0 for the permutation subkey in round 2.
- bit 2 = 0 for the permutation subkey in round 3.

Examination of the key scheduling algorithm shows the corresponding condition for the 64-bit user key:

- bit 20 = 1 and bit 12 = 0.

Figure 2 shows the 6-round algorithm and the 4-round characteristic. The expected input difference to the round function in round 5 is β, the expected output difference equals the difference in the right half of the ciphertext. This allows us to check if an arbitrary encrypted pair (with the right difference in the plaintext) is a right pair for the characteristic. The difference $\beta = 18$ delivers an input difference to S-box $S1$ or $S3$, depending on the value of the corresponding permutation key bit. So the output differences from S-boxes $S0$ and $S2$ have to be zero, as well as the output difference from either $S1$ or $S3$.

This corresponds to checking the values of $24 - 1 = 23$ bits. So a wrong pair has a probability 2^{-23} of surviving this filtering process. The probability of generating a right pair is much higher (2^{-13}), so when a pair survives the filtering, with a high probability it is a right pair.

For such a right pair we know the inputs and the difference at the output ($\Delta C_L \oplus \beta$) of the last round, and for all possible subkeys we can check whether they correspond. Repeat this for about four right pairs (we need to generate about $4 \cdot 2^{13} = 2^{15}$ pairs of plaintexts), the correct subkey will be suggested each time and can be distinguished from other suggested subkeys.

The signal-to-noise-ratio (the ratio of the number of times the correct key is suggested and the number of times an arbitrary key is suggested) for this attack can be calculated with the method described in [2]. It depends on the number of plaintext pairs m, the probability of the characteristic p, the number k of simultaneous key bits that we count on, the average count a per analysed pair, and the fraction b of the analysed pairs among all the pairs.

$$S/N = \frac{m \cdot p}{m \cdot a \cdot b / 2^k}.$$

In this case we have $m = 2^{15}$ and $p = 2^{-13}$. When concentrating on one S-box we are counting on $k = 20$ key bits (10 used for the permutation and 10 for the exor operation). The average count a equals 2^{12}, since we count on 2^{20}

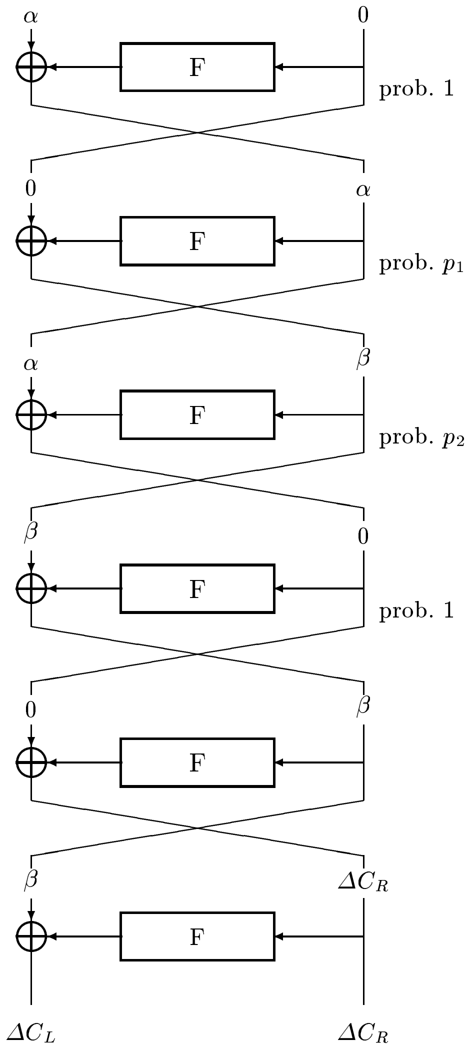

Fig. 2. The characteristic for an attack on 6 rounds.

subkeys and check an 8-bit value (difference at the output of the S-box). The fraction b (filtering) equals 2^{-23}. Hence the signal-to-noise-ratio is:

$$S/N = \frac{2^{15} \cdot 2^{-13}}{2^{15} \cdot 2^{12} \cdot 2^{-23}/2^{20}} = \frac{2^2}{2^{-16}} = 2^{18},$$

and similarly for the other three S-boxes. However $S0$ and $S2$, as well as $S1$ and $S3$, use the same permutation key bits, which we have to determine only once. For the second S-box which uses these permutation key bits we can count on just the 10 exor key bits. In this way we determine all 60 bits of the subkey. The remaining 4 bits of the user key can easily be found by exhaustive search.

4.2 Attack on ICE with 8 rounds (Thin-ICE)

We can extend the previous attack in a straightforward manner, using a 6-round characteristic with a probability $p_1 \cdot p_2 \cdot p_2 \cdot p_1 = 2^{-7} \cdot 2^{-6} \cdot 2^{-6} \cdot 2^{-7} = 2^{-26}$. The attack can be improved however by inserting a round before the first round of the characteristic without reducing the probability, like in the attack on DES [2]. The assumed evolution of differences (during the encryption of a right pair) is shown in Figure 3. In the first round the difference α at the input of the round function is an input difference to S-box $S0$ or $S2$, depending on the value of the corresponding permutation key bit. We guess this bit and repeat the attack if we have guessed wrong. We compensate the difference at the output of the round function of round 1 by using a structure of 2^9 plaintexts:

$$P_i = P \oplus (v_i, 0), \bar{P}_i = P \oplus (v_i, 0) \oplus (0, \alpha) \qquad \text{for } 0 \leq i < 2^8,$$

with v_i denoting all the possibilities for the 8 bits that are exored with the output bits from $S0$ or $S2$; (l, r) denotes the left and right 32-bit halves of a 64-bit text.

The probability for the characteristic is $p_1 \cdot p_2 \cdot p_2 = 2^{-7} \cdot 2^{-6} \cdot 2^{-6} = 2^{-19}$. The conditions for it to be valid are:

- bit $14 = 0$ for the permutation subkey in round 3.
- bit $2 = 0$ for the permutation subkey in round 4.
- bit $2 = 0$ for the permutation subkey in round 6.

The corresponding condition for the 64-bit user key is:

- bit $3 = 0$, bit $59 = 1$ and bit $25 = 1$.

In the defined structure there are 2^{16} pairs, of which 2^8 satisfy the first round. These can be isolated in 2^8 time as follows. Since the expected output differences from S-boxes $S1$ and $S3$ in round 7 are zero, we sort the texts according to the values of the corresponding bits in the right half of the ciphertext and find the matching values. We filter these further by S-box $S0$ or $S2$ (like in the 6 round attack), and can expect $2^8 \cdot 2^{-19} = 2^{-11}$ right pairs in a structure. By using 2^{13} structures 4 right pairs are expected. In total however there are $2^{16} \cdot 2^{13} = 2^{29}$ pairs. After filtering for 23 bits we expect there will remain $2^6 = 64$ wrong

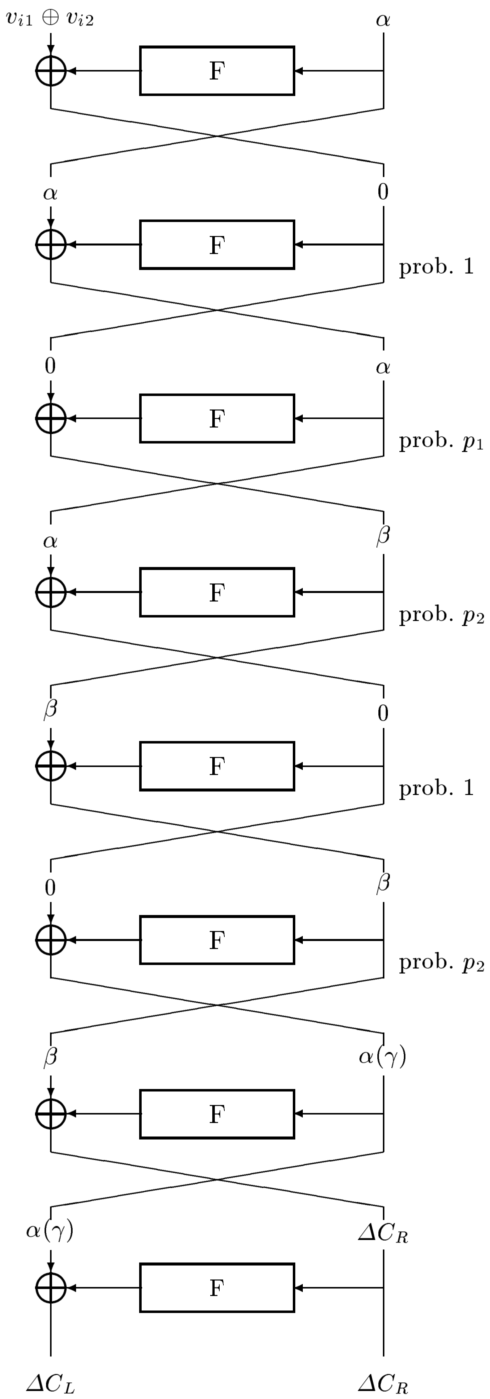

Fig. 3. The characteristic for an attack on 8 rounds.

pairs. For this mixture of right and wrong pairs we try all possible subkeys, concentrating on one S-box at a time.

In the calculation of the signal-to-noise-ratio for this attack there is an extra factor 2^{-8}, imposed by the first round structure ($m = 2^{13} \cdot 2^{16}$, but only $2^{13} \cdot 2^{8}$ pairs satisfy the first round):

$$S/N = \frac{2^{13} \cdot 2^{8} \cdot 2^{-19}}{2^{13} \cdot 2^{16} \cdot 2^{12} \cdot 2^{-23}/2^{20}} = \frac{2^{2}}{2^{-2}} = 2^{4}.$$

We can easily extend this attack to make it valid for twice as many keys. Just guess the value of bit 2 of the permutation subkey in round 6. If it equals 1 instead of 0, the round function in that round delivers an output exor different from $\alpha = 28$. With probability 2^{-6} this output exor will be $\gamma = 30$ and we can perform the attack in a similar way. The condition for the 64-bit user key is:

– bit 3 = 0 and bit 59 = 1.

Alternatively we can do an eight round attack using the characteristic with $\alpha = 31$ and $\beta = 26$. According to Table 1 the probability of this characteristic is $p_1 \cdot p_2 \cdot p_2 = 2^{-8} \cdot 2^{-6} \cdot 2^{-6} = 2^{-20}$. The conditions for the permutation key bits translate to the following condition for the user key:

– bit 48 = 1, bit 18 = 1 and bit 48 = 1.

So the permutation key bit in round 6 doesn't impose an extra condition on the user key, and we don't have to guess this bit when using the characteristic with $\alpha = 31$ and $\beta = 26$.

4.3 Practical aspects of the analysis

The attacks for the 6-round version and the 8-round version (Thin-ICE) have been implemented and, on the average, work as predicted. However, using low Hamming weighted differences causes some complications. The input difference to the last round is caused by the output difference from the previous round. That output difference is caused by just one S-box and has a Hamming weight of no more than 8, with an average of 4.

Each S-box in the last round receives 2 bits from these 8 bits. Because the keyed permutation swaps bits between the 'partner' S-boxes $S0 - S2$ and $S1 - S3$, an S-box will finally receive between 0 and 4 from these bits at its input, depending on the value of the permutation subkey. Only these bits can cause an input difference. If S-box $S0$ or $S1$ gets k bits, then respectively $S2$ or $S3$ will get $4 - k$ bits. If a particular S-box gets k bits with a possible difference, the probability to get input difference zero is approximately 2^{-k}. In Table 2 we list the possible values for that probability, and the fraction of subkeys for which it holds.

If the input difference to an S-box is zero, all of the guesses for the permutation subkey that cause a zero difference will be counted, as will all possibilities

probability	fraction of subkeys
1	$1/2^4$
2^{-1}	$4/2^4$
2^{-2}	$6/2^4$
2^{-3}	$4/2^4$
2^{-4}	$1/2^4$

Table 2. Probabilities to get a zero input difference to an S-box.

for the exor subkey. Therefore the attack is less efficient and we have to look for some more right pairs for the characteristic (in practice between 4 and 8), hence use more plaintexts.

For a fraction 2^{-4} of the keys the input difference to the S-box will always be zero, so we can determine only some of the permutation key bits and none of the exor key bits. But then the partner S-box has a probability for zero input difference of only 2^{-4}. We determine the permutation key bits via this S-box, and the 10 exor key bits that we cannot determine can be looked for exhaustively after the differential attack (together with the 4 bits of the user key that are not used in the 60-bit subkey of the last round).

It is possible to exploit the occasions of zero input differences to improve our attack. If the input difference to an S-box in the last round is zero, the output difference is zero as well. In that case we know the corresponding difference at the input to the round function in the second to last round and we can check if its value corresponds to the value that is required for the characteristic. In this way we can do some extra filtering, which is important for the 8 round attack where we expect to get 64 wrong pairs. It will increase the signal-to-noise-ratio and reduce the number of required plaintexts.

4.4 Extending the attack

The 3-round iterative characteristic can be extended in a straightforward way to attack the ICE algorithm with an arbitrary number of rounds. But if the number of rounds exceeds 9, the signal-to-noise-ratio will drop below one, making the attack impossible (the last remark of previous section allows only a slight improvement by extra filtering). There are however several ways to improve the signal-to-noise-ratio.

Counting on more key bits When a pair survives the filtering (and is assumed to be a right pair, following the characteristic), we know the inputs and the difference at the output of the last round and check whether they correspond. In the basic attack we concentrate on one S-box and count on 20 subkey bits (10 used for the permutation and 10 for exoring).

Instead we can consider two partner S-boxes ($S0$ and $S2$, or $S1$ and $S3$) at the same time. They share 10 permutation key bits and both use 10 exor key bits. This allows us to count on 30 key bits, and results in an improvement of the signal-to-noise-ratio by a factor of 2^8 because we check the values of 8 more bits at the output of the second S-box (in the calculation of S/N we have $2^k = 2^{30}$ and $a = 2^{30}/2^{16} = 2^{14}$). In theory further improvements (by a factor of 2^{16} or 2^{24}) are possible by considering respectively three or four S-boxes (50 or 60 key bits).

Checking differences in the first round When a pair is assumed to follow the characteristic we can also check subkey bits in the first round of the algorithm. In this first round we use a special structure (cf. the attack on 8 rounds) and guess the value of the permutation key bit corresponding with the difference of Hamming weight one. Hence we can count on the 10 exor key bits of the S-box where the difference of Hamming weight 1 is located.

Moreover, due to the key schedule some of these subkey bits in the first round represent the same user key bits as some of the subkey bits in the last round of the algorithm. This allows us to improve the signal-to-noise-ratio by a factor of 2^8 by counting on just a few more key bits. Note also that some of the key bits that we count on are already known, because of the condition on the user key for the characteristic.

Filtering in the last round The most important improvement can be made by adapting the characteristic. In the previous attacks we used the characteristics with the highest probabilities. The resulting attacks are called 2R-attacks (cf. Biham and Shamir [2]), because they don't make assumptions for the last two rounds of the algorithm.

Instead we can perform 1R-attacks, using a characteristic up to the last to one round. An example of the last rounds for such a characteristic is shown in Figure 4.

Although the probability of such a characteristic is generally lower than for a 2R-attack, it is useful because it allows much more filtering and an overall reduction of the signal-to-noise-ratio. In the last round we can check if the difference at the output of the round function (ΔC_L in Figure 4) is possible, like we did in the last to one round in the previous attacks. This corresponds to checking the values of 23 bits. But we can also check the difference in the right half of the ciphertext ($\Delta C_R = \alpha$ in Figure 4), thus filter for 32 more bits.

This results in an improvement of the signal-to-noise-ratio by a factor of $2^{32} \cdot p_c$, where p_c represents the factor by which the probability of the characteristic is reduced when we perform a 1R-attack instead of a 2R-attack.

Results for an arbitrary number of rounds Table 3 lists for each number of rounds: the probability of the characteristic, the required number of chosen plaintexts (assuming 4 right pairs for the characteristic are sufficient, and that we need both guesses for the permutation key bit in the first round structure), the number of subkey bits counted on (excluding key bits in the first round

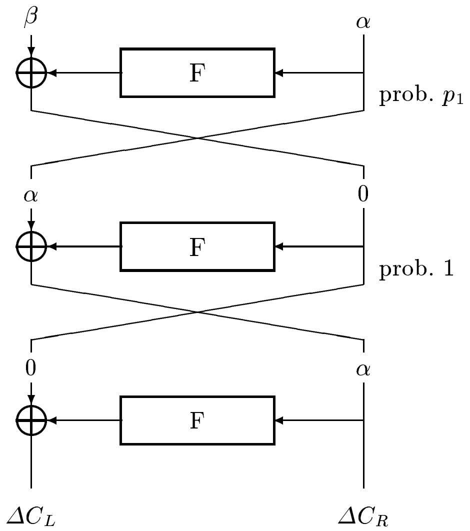

Fig. 4. Characteristic (last rounds) for a 1R-attack.

because of the overlap), the signal-to-noise-ratio and the fraction of keys that can be found with the attack.

When the number of rounds exceeds 9 we list two different attacks: a 2R-attack (where the signal-to-noise-ratio is improved by counting on more key bits and checking the difference in the first round), and a 1R-attack (which has a lower probability and requires more plaintexts). When the number of rounds is a multiple of 3, we have listed only the 1R-attack because it's probability is the same as for the 2R-attack ($p_c = 1$). In the other cases we have $p_c = p_1 = 2^{-7}$ or $p_c = p_2 = 2^{-6}$.

Note that the key fraction is lower for a 1R-attack, because the characteristic imposes more conditions on the user key (except when the number of rounds is a multiple of 3).

The table shows that the differential analysis works for up to 15 rounds of ICE: for this attack 8 key bits are fixed and an exhaustive search would have 2^{56} possibilities, our attack requires at most 2^{56} plaintexts. For the complete 16-round algorithm the number of plaintexts needed is still smaller than the total number available, but an exhaustive search in the covered fraction of the key space would be faster.

4.5 Key dependency of the attacks

We described the previous attacks using the best conditional characteristic. In Table 3 we have listed for how many keys this works. For Thin-ICE the attack works for a fraction 2^{-2} of the keys. For other keys however we can use a different characteristic with a lower probability. We can use several characteristics with

# rounds	probability †	# plaintexts †	# counters †	S/N †	key fraction †
4	0	4	20	23	all
5	-6	10	20	17	all
6	-13	16	20	18	-2
7	-13	17	20	10	-2
8	-19	23	20	4	-2
9	-26	29	20	5	-4
10	-26	30	20	5	-4
10	-32	36	20	23	-5
11	-32	36	20	-1	-4
11	-39	42	20	24	-6
12	-39	43	20	16	-6
13	-39	43	30	0	-6
13	-45	49	20	10	-7
14	-45	49	50	2	-6
14	-52	55	20	11	-8
15	-52	56	20	3	-8
16	-52	56	60	3	-8
16	-58	62	30	5	-9

Table 3. Differential analysis for an arbitrary number of rounds. († $\log_2$)

the same set of plaintexts, if we use a special structure for these plaintexts. For two characteristics this is a quartet structure (like the one used for the analysis of DES in [2]), for three an octet structure and so on. The number of plaintexts we need depends on the characteristic with the lowest probability. If we want to be able to determine as many possible keys with as few possible plaintexts we use the characteristics with the highest probabilities. Table 4 shows the evolution of these numbers for the Thin-ICE algorithm.

# characteristics	# plaintexts ($\log_2$)	key fraction
1	23	25%
3	24	63%
5	25	81%
6	26	88%
8	27	95%

Table 4. For Thin-ICE (8 rounds) : the number of characteristics and plaintexts required versus the fraction of keys that can be found.

5 Conclusion

We have described attacks on the ICE algorithm using differential analysis. The main conclusion of this paper is that keyed permutation does not prevent differential cryptanalysis. Although the analysis is more complicated and becomes key dependent, in our opinion the intention of the design has not been reached.

The best 3-round iterative characteristic that can be used in our attack has a probability of 2^{-13}, which is higher than the probability of 2^{-16} of the best 3-round characteristic for LOKI'91 [6] (a similar block cipher that makes use of four identical 12 to 8-bit S-boxes).

We have also demonstrated a practical attack on the fast version Thin-ICE. In its basic form it finds the secret key in 25% of the cases using 2^{23} chosen plaintexts, and in 95% of the cases using 2^{27} plaintexts. The optimal characteristic for our attack on Thin-ICE has a probability of 2^{-19} which is much higher than the probability of the optimal characteristic based on a symmetric input difference, which was shown to be 2^{-56} in [7].

References

1. I. Ben-Aroya and E. Biham, "Differential Cryptanalysis of Lucifer," *Advances in Cryptology – Crypto '93 Proceedings, LNCS 773*, D. Stinson, Ed., Springer-Verlag, 1994, pp. 187–199.
2. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
3. L. Brown, M. Kwan, J. Pieprzyk and J. Seberry, "Improving resistance to differential cryptanalysis and the redesign of LOKI," *Advances in Cryptology – AsiaCrypt '91 Proceedings, LNCS 739*, H. Imai, R. Rivest, and T. Matsumoto, Eds., Springer-Verlag, 1993, pp. 36–50.
4. L. Brown, J. Pieprzyk and J. Seberry, "LOKI: A Cryptographic Primitive for Authentication and Secrecy Applications," *Advances in Cryptology – AusCrypt '90 Proceedings, LNCS 453*, J. Seberry and J. Pieprzyk, Eds., Springer-Verlag, 1990, pp. 229–236.
5. FIPS 46, *Data Encryption Standard*, Federal Information Processing Standard (FIPS), Publication 46, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., January 1977.
6. L. Knudsen, "Cryptanalysis of LOKI'91," *Advances in Cryptology – AusCrypt'92 Proceedings, LNCS 718*, J. Seberry and Y. Zheng, Eds., Springer-Verlag, 1993, pp. 196–208.
7. M. Kwan, "The Design of the ICE Encryption Algorithm," *Proceedings of the 4th Workshop on Fast Software Encryption, Haifa, Israel, LNCS 1267*, E. Biham, Ed., Springer-Verlag, 1997, pp. 69–82.