Cryptanalysis of Block Ciphers Using Almost-Impossible Differentials

Hamid Mala, Mohammad Dakhilalian, and Mohsen Shakiba

Cryptography & System Security Research Laboratory, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran
{hamid_mala@ec, mdalian@cc, m.shakiba@ec}.iut.ac.ir

Abstract. In this paper, inspired from the notion of impossible differentials, we present a model to use differentials that are less probable than a random permutation. We introduce such a distinguisher for 2 rounds of Crypton, and present an attack on 6 rounds of this predecessor AES candidate. As a special case of this idea, we embed parts of the additional rounds around the impossible differential into the distinguisher to make a probabilistic distinguisher with more rounds. We show that with this change, the data complexity is increased but the time complexity may be reduced or increased. Then we discuss that this change in the impossible differential cryptanalysis is commodious and rational when the data complexity is low and time complexity is marginal.

1 Introduction

New attacks on iterated block ciphers intend to distinguish some rounds of a cipher from a random permutation. By adding some additional rounds to such a distinguisher and guessing their subkeys, the attacker makes a key-recovery attack. Differentials of high probability and differentials of zero probability are the distinguishers of differential cryptanalysis [2] and impossible differential cryptanalysis [1, 4], respectively. For a random permutation the probability of the differential lies between the probability of these two distinguishers. Now suppose the case where the probability of a differential lies between zero (the probability for an impossible differential) and the trivial probability (i.e., the probability for a random permutation). Such a differential is less probable than a random permutation, so in the rest of this paper we call it the *almost-impossible differential*. Consequently the key-recovery attack that uses these differentials is called *almost-impossible differential cryptanalysis* or *AIDC*. To the best of our knowledge, the first trials for using low-probability differentials go back to [3] and [5] for cryptanalysis of IDEA and DFC, respectively. Here we present a systematized model to construct and use low-probability differentials for cryptanalysis of block ciphers.

To construct its distinguisher, impossible differential attack uses two differentials with probability one in the reverse directions which contradict in the middle. Inspired from this miss in the middle method, we present a similar model for almost-impossible differentials, in which one differential has probability 1, and

the other differential, which is in the reverse direction, is probabilistic. Based on this model we introduce an almost-impossible differential for 2 rounds of Crypton, and propose a key-recovery attack for 6 rounds of this block cipher. Then we discuss the data and time complexity and success rate of the attack.

The rest of this paper is organized as follows. Section 2 provides a brief description of Crypton. The model of AIDC is introduced in Section 3. As an example of the application of this model, an attack on 6-round Crypton is presented in Section 4. Section 5 describes an analyses the complexity of a special class of almost-impossible differential attack whose distinguisher is constructed directly from an impossible differential. Finally, we conclude the paper in Section 6.

1.1 Notations

In this paper we use the following notations.

$a\|b$	: bit string concatenation of a and b,
$\oplus$	: bit-wise exclusive OR operation,
$\overline{\Delta x}$	: Any difference except Δx.

2 A Brief Description of Crypton

The 128-bit block cipher Crypton [6] has a 12-round SPN structure that supports key sizes of 128, 192, and 256 bits. Let us represent a 128-bit data by a 4×4 matrix A of bytes $A[i][j], i, j = 0, 1, 2, 3$, where the row with index $i = 0$ is the bottommost row and the column with index $j = 0$ is the leftmost column. Each round of Crypton applies the following 4 transformations to this state matrix.

- γ_o and γ_e (or simply γ) are byte-wise S-box layers which are applied to the 128-bit datd in odd and even rounds, respectively.
- π_o and π_e (or simply π) are linear column-wise permutations with differential branch number of 4 for odd and even rounds, respectively.
- τ is a linear transformation that transposes the state matrix.
- σ is a bitwise key XOR. When the given round key of round r is K_r, the notation σ_{K_r} is used.

The encryption consists of a whitening with the subkey K_0, 12 consecutive round functions $\rho_r(x) = \sigma_{K_r}(\tau(\pi(\gamma(x)))), r = 1, ..., 12$, and finally the final transformation $\varphi_e(x) = \tau(\pi_e(\tau(x)))$. Note that the round function $\rho_r(x)$ has an equivalent form of $\rho_r^{eq}(x) = \tau(\pi(\sigma_{K_r^{eq}}(\gamma(x))))$, where the equivalent round key is $K_r^{eq} = \tau(\pi^{-1}(K_r))$.

To denote the intermediate values after the application of γ, π, τ and σ in round r, we use the notations x_r^γ, x_r^π, x_r^τ and x_r^σ, respectively. The row i and column j of some intermediate value x in round r are denoted by $x_{r,col(j)}$ and $x_{r,row(i)}$, respectively.

Here we remind two properties of the diffusion transformation of Crypton from [7]. We will use these properties in the proposed attack of Section 4.

Property 1: Let $n_{i,j}$ be the number of 4-byte words with i non-zero bytes that after the application of π are converted to 4-byte words of j non-zero bytes. The values $n_{i,j}$ are represented in Table 1. Using $n_{i,j}$ values, the probability that the transformation π transforms a 4-byte word with i non-zero bytes in fixed positions into a word with j non-zero bytes in fixed positions is equal to $p_{i,j} = \frac{n_{i,j}/\binom{4}{i}\binom{4}{j}}{255^i}$.

Table 1. $n_{i,j}$ values for calculating the transition probabilities of π transformation

i	j				
	0	1	2	3	4
0	1	0	0	0	0
1	0	0	0	48	972
2	0	0	108	5760	384282
3	0	48	5760	1024800	65294892
4	0	972	384272	65294892	4162570479

Property 2: Let $y = \pi_e(\pi_o(x))$, then $y_{row(i)} = x_{row(i+2 \mod 4)}$. This property implies that the transformation $\pi_e(\pi_o(x))$ is equivalent to a row-wise circular shift. One can easily observe that when the equivalent round is used in the last round of Crypton, the transformations π and τ in that round and the Final transformation φ_e are easily simplified.

3 Almost-Impossible Differential Cryptanalysis

In this section we introduce the notion of almost-impossible differential cryptanalysis. In our model, represented in Figure 1, the distinguisher is constructed similar to the miss in the middle approach, but the key-recovery step is performed similar to that of differential attack.

Let $E = E_f \circ E_1 \circ E_0$ be the target cipher. Suppose we have already found an almost-impossible differential for $E_1 \circ E_0$, thus E_f is the additional rounds we want to recover some of their subkey bits (in a more general model we can add some additional rounds in the beginning of the cipher). The parameters and properties involved in the model are defined as below:

- E_0 is a part of cipher that conducts the differential Δx_I to the differential Δx_M in the encryption direction with probability q.
- E_1 is a part of cipher that conducts the differential Δx_O to the differential Δx_M in the decryption direction with probability 1.
- N is the number of plaintexts with the difference $\Delta P = \Delta x_I$.

- $p_{\Delta C}$ is the probability that a plaintext pair with the difference ΔP results in the difference ΔC for the ciphertext pair. Supposing that E is an ideal cipher, $p_{\Delta C}$ is independent of ΔP and easily calculated from ΔC.
- p_{wk} is the probability that a ciphertext pair with the difference ΔC after partial decryption through E_f^{-1} with a wrong subkey meets the difference Δx_O.
- p_{ck} is the probability that a ciphertext pair with the difference ΔC after partial decryption through E_f^{-1} with the correct subkey meets the difference Δx_O.
- q is the probability that a pair with the difference Δx_I after partial encryption through E_0 meets the difference Δx_M.
- p is the probability that an intermediate pair with the difference Δx_M after partial encryption through E_1 meets the difference Δx_O.
- r is the probability that an intermediate pair with the difference Δx_O after partial encryption through E_f meets the difference ΔC.

Corollary 1: for $E_1 \circ E_0$ the path from Δx_I to Δx_O inevitably passes from the difference Δx_M in the middle.

Now we discuss on the conditions that make $E_1 \circ E_0$ a distinguisher. The condition for $\Delta x_I \xrightarrow{E_1 \circ E_0} \Delta x_O$ to be a distinguisher is that p_{wk} and p_{ck} must not be equal to each other. When a wrong key is used through E_f^{-1} the relation between plaintexts and ciphertexts is intercepted, so p_{wk} is easily calculated just by considering the structure of E_f^{-1}. For computing p_{ck}, we should compute the fraction of proper pairs (i.e., pairs with the difference ΔP in the plaintext pair and ΔC in the ciphertext pair) that pass from Δx_O to the total number of proper pairs. Based on the parameters defined in the model, the number of proper pairs is $N \times p_{\Delta C}$, and considering Corollary 1, the number of pairs that pass from Δx_O in the output of E_1 is $N \times q \times p$, among them about $N \times q \times p \times r$ pairs meet ΔC. Thus p_{ck} is calculated as:

$$p_{ck} = \frac{N \times q \times p \times r}{N \times p_{\Delta C}} = \frac{q \times p \times r}{p_{\Delta C}}$$

Note that when $p_{wk} < p_{ck}$, the distinguisher is the well-known differential distinguisher, but we are interested in the case where $p_{ck} < p_{wk}$. In this case we have a distinguisher with the probability less than the probability for a random permutation. In the following, we present such a distinguisher for the block cipher Crypton, and exploit it to present an AIDC attack on 6 rounds of this cipher.

4 Cryptanalysis of Reduced Crypton Using Almost-Impossible Differentials

In this section, based on the model presented in the previous section, first a 2-round differential with probability less that the probability of a random permutation is presented. Then an attack procedure appropriate for this distinguisher

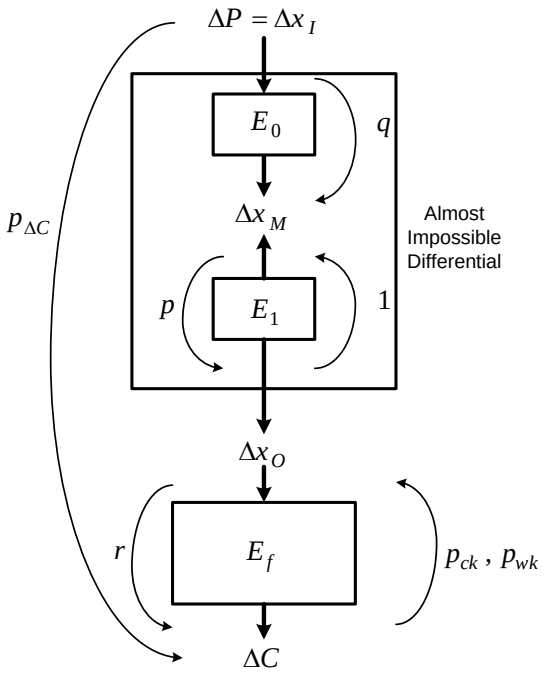

Fig. 1. Model and parameters of almost-impossible differential cryptanalysis

is applied to 6-round Crypton. Finally the attack complexity and its success rate are discussed.

4.1 An almost-impossible differential of Crypton

Figure 2 illustrates a 2-round almost-impossible differential of Crypton. The distinguisher concurs the model presented in Figure 1 in which E_0 is a 1-round probabilistic differential that converts a differential with 16 non-zero bytes to a differential with 4 non-zero bytes in a column, and E_1^{-1} is a 1-round deterministic differential with one special active bytes in the input and 4 active bytes in its output difference. From the properties of the transformation π (and π^{-1}), we know that when the input has only one active byte in a definite position, to ensure the existence of exactly 4 active bytes in the output of this transformation, the input active byte can take $n_{1,4}/4 = 243$ out of 255 possible values.

As mentioned in Section 2, for the π transformation, the probability that a column with i active bytes in fixed positions is converted to a column with j active bytes in fixed positions is denoted by $p_{i,j}$. Thus it is clear that for the distinguisher of Figure 2, the probability q is equal to $p_{4,1}^4$, and the probability p is $p_{4,1}$.

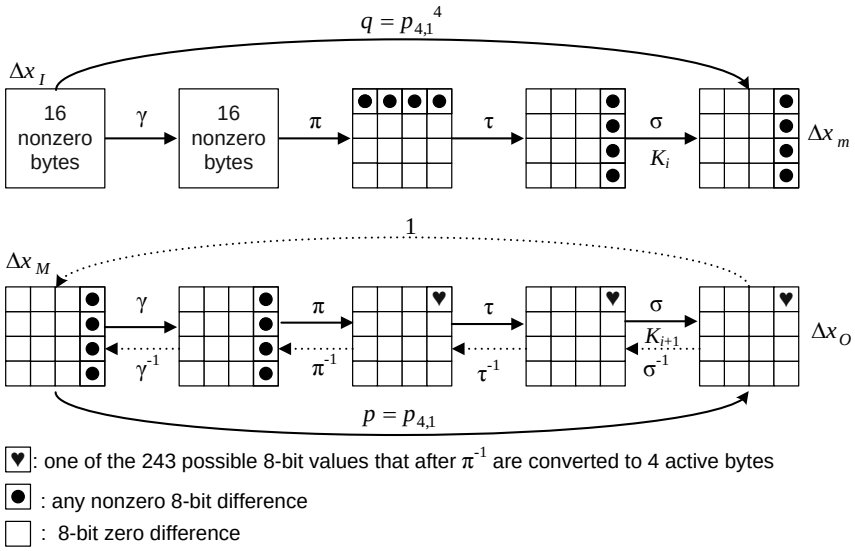

Fig. 2. 2-round almost-impossible differential of Crypton

4.2 The attack procedure

To apply a 6-round attack on Crypton, two rounds are added in the beginning and two other rounds (and the final transformation) are added at the end of the distinguisher. Based on the construction of the additional rounds at the ciphertext side it is easy to see that $r = 1$. In order to simply use Property 2 of the diffusion layer of Crypton, let us take these 6-round structure rounds 2 through 7 of Crypton. The attack is demonstrated in Figure 3.

Note that to reduce the target key space and the attack complexity, for the last 2 rounds (i.e., rounds 6 and 7) the equivalent round structure is used. The attack procedure, inspired from the traditional differential attack, is as follows:

1. Take 2^n structures of plaintexts such that each structure contains 2^{32} plaintexts that have fixed values in the first 3 columns and take all the possible values in the last column (column 3). It is obvious that from each structure about $2^{32} \times 255^4/2 \approx 2^{62.98}$ plaintext pairs can be obtained such that these pairs have non-zero differences in the 4 bytes of column 3. Totally, we have 2^{n+32} plaintexts and we can collect about $2^{n+62.98}$ plaintext pairs (P, P') with the desired difference $\Delta P = P \oplus P'$ shown in Figure 3.
2. Obtain the ciphertexts of each structure, store them in a hash table indexed by their values in the 12 bytes of $(col(0, 2, 3))$, thus each two texts that lie in the same row ot this table form a proper pair (with the required ΔP and ΔC). Based on the form of ΔC, the probability of $p_{\Delta C}$ is the probability that a pair have zero difference in specified 12 bytes and at least 3 non-zero bytes in column 1. So this probability is:

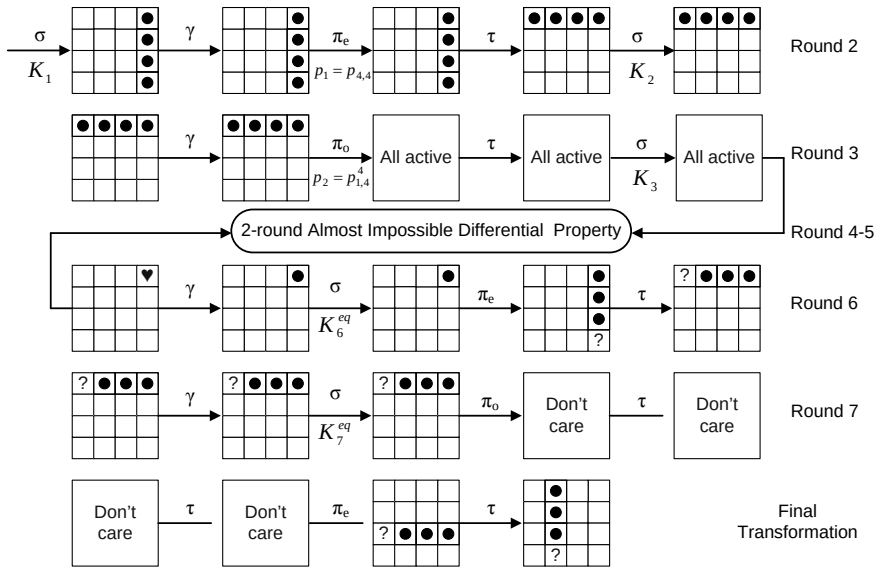

Fig. 3. AIDC of 6-round Crypton

$p_{\Delta C}$=pr{4 active bytes in column 1} $\times 2^{-8\times 12}$+pr{3 active bytes in column 1} $\times 2^{-8\times 13} = \frac{255^4}{256^4-1} \times 2^{-96} + \binom{4}{1} \times \frac{255^3}{255^3-1} \times 2^{-104} = 2^{-96.0001}$

Thus the expected number of obtained such pairs from all structures is $2^{n+62.98} \times p_{\Delta C} = 2^{n-33.0201}$.

3. Guess 32-bit values of $K_{1,col(3)}$, and for each guess partially encrypt all $2^{n-33.0201}$ pairs through σ, γ and π_e. Keep the pairs that result in 4 active bytes in the output. The probability of this event is $p_{4,4}$, thus the expected number of remaining pairs is $p_{4,4} \times 2^{n-33.0201}$. $p_{4,4}$ is computed based on Property 1 in Section 2.
4. For each of the 32-bit guesses of step 3, guess 32-bit values of $K_{2,row(3)}$, and for each guess partially encrypt all $p_{4,4} \times 2^{n-33.0201}$ pairs through σ, γ and π_o. Keep the pairs that result in 16 active bytes in the output. The probability of this event is $p_{1,4}^4$, thus the expected number of remaining pairs is $p_{4,4} \times p_{1,4}^4 \times 2^{n-33.0201}$.
5. Initialize 2^{32} counters with the initial value of 0 for the 32-bit possible values of $K_{7,row(3)}^{eq}$.
6. In this step, we use a well-known property of invertible S-boxes: Given an input and an output differences of the γ operation, there is on average one pair of actual values that satisfies these differences. Using this property, in the following way, will reduce the complexity of the attack. Based on Property 2 of Crypton, we know if in the final round, the equivalent round structure is used, then for any ciphertext pair with a difference ΔC, we

have $\Delta x^{\sigma_{eq}}_{7,row(i)} = \Delta C_{col(i+2 \mod 4)}, i = 0, 1, 2, 3$. Thus for each pair, we know the output difference of γ_7. Also $\Delta x^{\tau}_{6,row(3)}$ takes 4×255 possible differences (note that these are the differences which are generated from passing a column with 1 active byte through π in round 6). Thus for each of these 4×255 input/output difference pairs of γ_7 we obtain on average 1 pair of $(x^{\gamma}_{7,row(3)}, x^{'\gamma}_{7,row(3)})$, and since the corresponding $(x^{\sigma_{eq}}_{7,row(3)}, x^{'\sigma_{eq}}_{7,row(3)})$ is known from the ciphertext pair, we immediately obtain 4×255 values for $K^{eq}_{7,row(3)}$. Increase the counter of each of these 1020 values by 1.

7. The counter with the minimum number determines the correct key with high probability.

4.3 Analysis of the attack

Here, p_{wk} is the probability that a ciphertext pair (C, C') with the difference $\Delta C = C \oplus C'$, after partial decryption with a wrong key guess $K^{eq}_{7,row(3)}$ reach a difference which has only one active byte in the output of π^{-1} in round 6. This probability which depends only on ΔC and the properties of π is computed as below:

$$p_{wk} = \frac{2^{-96} \times 255^4/(256^4-1)}{p_{\Delta C}} \times p_{4,1} + \frac{4 \times 2^{-104} \times 255^3/(256^3-1)}{p_{\Delta C}} \times p_{3,1} = 2^{-24.0055}$$

For each 64-bit value of the guessed subkeys of the first 2 rounds, $N' \triangleq p_{4,4} \times p^4_{1,4} \times 2^{n-33.0201} \approx 2^{n-33.32}$ pairs are tried in step 6. Hence, it is expected that the counter of each wrong key has a value of $N' \times p_{wk} = N' \times 2^{-24.0055}$.
On the other hand, as discussed in Section 3, the probability that the correct key meets the criterion is as below:

$$p_{ck} = \frac{q \times p \times r}{p_{\Delta C}} = \frac{p^5_{4,1}}{2^{-96.0001}} = 2^{-24.2629}$$

So the condition of AIDC that is $p_{ck} < p_{wk}$ holds, and the expected value for the correct key counter is $N' \times p_{ck} = N' \times 2^{-24.2629}$. The attack has a probabilistic nature which urges us to compute its success probability. In the following, we try to find a relation between the success probability and the value of N' which is a multiple of required plaintext pairs.

The value of the correct key counter (n_{ckc}) follows a binomial distribution of parameters (N', p_{ck}), while the value of each wrong key counter (n_{wkc}) follows a binomial distribution of parameters (N', p_{wk}). Since N' is expected to be very big, we can approximate the distribution of n_{ckc} as a Gaussian distribution of parameters $(p_{ck}, N'p_{ck}(1-p_{ck}))$, and n_{wkc} as a Gaussian distribution with parameters $(p_{wk}, N'p_{wk}(1-p_{wk}))$. Based on this approximation, the probability that n_{wkc} be greater than n_{ckc} is computed as below:

$$r \triangleq \text{pr } \{\ n_{ckc} < n_{wkc}\ \} = \text{pr } \{\ n_{ckc} - n_{wkc} < 0\ \} =$$
$$\int_{-\infty}^{0} \frac{1}{\sqrt{2\pi[N'p_{ck}(1-p_{ck}) + N'p_{wk}(1-p_{wk})]}} exp(\frac{-(x-(N'p_{ck}-N'p_{wk}))^2}{2N'[p_{ck}(1-p_{ck})+p_{wk}(1-p_{wk})]})dx$$
$$= \int_{-\infty}^{T} \frac{1}{\sqrt{2\pi}} exp(\frac{-x^2}{2})dx = \varphi(T)$$

where $T = \frac{N'(p_{wk}-p_{ck})}{\sqrt{N'p_{ck}(1-p_{ck})+N'p_{wk}(1-p_{wk})}}$ and $\varphi(\cdot)$ is the cumulative function of a Gaussian random variable. Since we have $2^{96}-1$ wrong keys, the probability that the correct key counter has the minimum value, which is the success probability, is $r^{2^{96}-1}$. With $N' = 2^{36.2}$ the success probability gets very close to 1. So n is obtained equal to $log_2 N' + 33.32 = 69.52$, and consequently the number of required plaintext will be $2^{n+32} = 2^{101.52}$.

The time complexity of step 2 is about $2^{n+32} = 2^{101.52}$ encryptions and the same number of memory accesses for arranging the hash table. In step 3, about $\frac{1}{4} \times \frac{1}{6} \times 2^{32} \times 2^{n-33.02} = 2^{63.915}$ encryptions are performed. Step 4 performs about $\frac{1}{4} \times \frac{1}{6} \times 2^{64} \times p_{4,4} \times 2^{n-33.02} = 2^{95.89}$ encryptions. The time complexity of step 5 is negligible. Step 6 requires $4 \times 255 \times 2^{64} \times N' \approx 2^{110.2}$ memory accesses, which is the dominant part of the attack's time complexity.

5 A Special Class of Almost-Impossible Differentials

In this section we discuss a special class of almost-impossible differentials in which the distinguisher is constructed from an impossible differential. Then we compare the data and time complexity of the two attacks when they are applied to the same number of rounds of a block cipher. The general model of the impossible differential attack and its corresponding almost-differential attack are represented in Figure 4. We first introduce the parameters of the models.

Let $E = E_f \circ E_1 \circ E_0 \circ E_i$ be the target cipher. Suppose we have already found an impossible differential for $E_1 \circ E_0$, i.e., $\Delta x_I \overset{E_1 \circ E_0}{\nrightarrow} \Delta x_O$. E_i and E_f are the initial and final rounds added around the impossible differential, and we want to recover some of their subkey bits. The parameters and properties involved in the model are defined as below:

- E_0 and E_1 are the two deterministic differentials that contradict in the middle with probability 1.
- N is the number of plaintexts with the difference ΔP.
- $p_{\Delta C}$ is the probability that a plaintext pair with the difference ΔP results in the difference ΔC for the ciphertext pair. Supposing that E is an ideal cipher, $p_{\Delta C}$ is independent of ΔP and easily calculated from ΔC. This parameter is the same for the two attacks.
- p_i is the probability that a plaintext pair (P, P') with the difference $P \oplus P' = \Delta P$ after passing through E_i results in the difference Δx_I.
- p_f is the probability that a ciphertext pair (C, C') with the difference $C \oplus C' = \Delta C$ after passing through E_f^{-1} results in the difference Δx_O.
- k_i and k_f are the target subkeys of the initial and final rounds in the impossible differential attack, respectively; the size of these two subkey spaces is denoted by $|k_i|$ nad $|k_f|$, respectively.

The almost-impossible differential distinguisher is constructed by pushing parts of the additional rounds into the impossible differential. Here, without loss of generality, we embed the initial rounds into the distinguisher. This way, the

number of rounds is increased, and the size of target subkeys is reduced from $|k_i|.|k_f|$ to $|k_f|$.

The parameters of the corresponding almost-impossible differential are listed below:

- p_{wk} is the probability that a ciphertext pair with the difference ΔC after partial decryption through E_f^{-1} with a wrong subkey meets the difference Δx_O. This probability is equal to p_f
- p_{ck} is the probability that a ciphertext pair with the difference ΔC after partial decryption through E_f^{-1} with the correct subkey meets the difference Δx_O. This probability will be calculated later.
- $p_{\Delta x_O}$ is the probability that an intermediate pair with the difference $\overline{\Delta x_I}$ after partial encryption through $E_1 \circ E_0$ meets the difference Δx_O. To compute this probability, we assume that $E_1 \circ E_0$ acts like a random permutation when the input difference is not Δx_I.

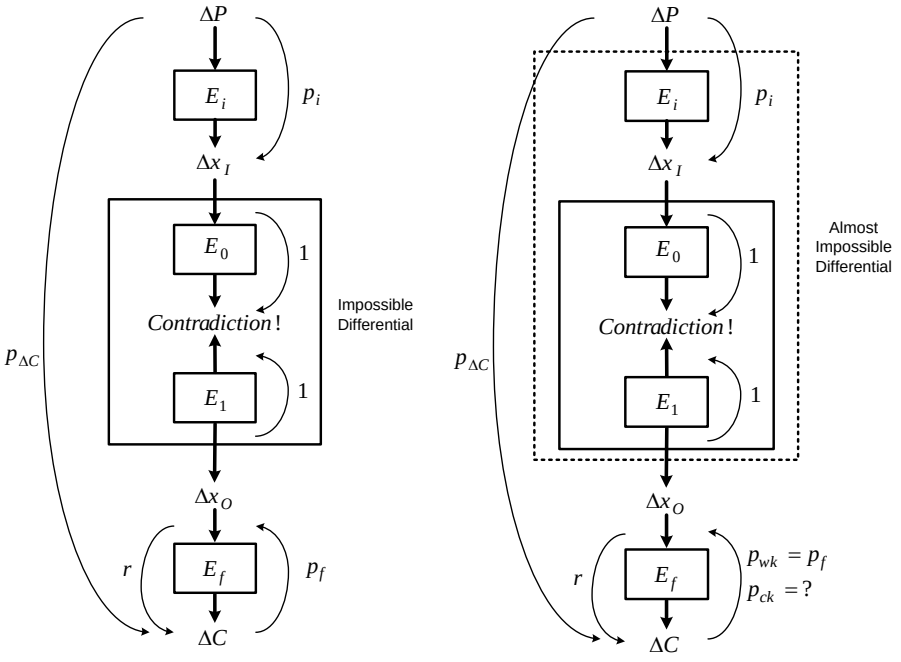

Fig. 4. Constructing almost-impossible differential cryptanalysis from impossible differential cryptanalysis

The condition for $\Delta P \xrightarrow{E_1 \circ E_0 \circ E_i} \Delta x_O$ to be a distinguisher is that p_{wk} and p_{ck} must not be equal to each other. When a wrong key is used through E_f^{-1}

the relation between plaintexts and ciphertexts is intercepted, so p_{wk} is easily computed just by considering the structure of E_f^{-1}. For computing p_{ck}, we should compute the fraction of proper pairs that pass from Δx_O to the total number of proper pairs. Based on the parameters defined in the model, the number of proper pairs is $N \times p_{\Delta C}$. The number of proper pairs that pass from Δx_O in the output of E_1 is computed as below:

$$p_{ck} = pr\{\Delta C \xrightarrow[k_{f,c}]{E_f^{-1}} \Delta x_O\} = \frac{N \times pr\{\Delta P \xrightarrow{E_1 \circ E_0 \circ E_i} \Delta x_O \xrightarrow{E_f} \Delta C\}}{N \times pr\{\Delta P \xrightarrow{E} \Delta C\}}$$

By conditioning whether the intermediate difference in the output of E_i is Δx_I or not we have:

$$\frac{pr\{\Delta P \to \Delta x_O \to \Delta C | \Delta P \to \Delta x_I\} \times pr\{\Delta P \to \Delta x_I\} + pr\{\Delta P \to \Delta x_O \to \Delta C | \Delta P \to \overline{\Delta x_I}\} \times pr\{\Delta P \to \overline{\Delta x_I}\}}{p_{\Delta C}}$$

From the model it is clear that $pr\{\Delta P \to \Delta x_O \to \Delta C | \Delta P \to \Delta x_I\} = 0$, and thus the relation is simplified to:

$$= \frac{pr\{\overline{\Delta x_I} \to \Delta x_O \to \Delta C | \Delta P \to \overline{\Delta x_I}\} \times (1-p_i)}{p_{\Delta C}} = \frac{pr\{\overline{\Delta x_I} \to \Delta x_O \to \Delta C\} \times (1-p_i)}{p_{\Delta C}} = \frac{pr\{\overline{\Delta x_I} \to \Delta x_O\} \cdot pr\{\Delta x_O \to \Delta C\} \cdot (1-p_i)}{p_{\Delta C}} = \frac{p_{\Delta x_O} \cdot r \cdot (1-p_i)}{p_{\Delta C}}$$

But $pr\{\overline{\Delta x_I} \to \Delta x_O \to \Delta C\}$ Can be obtained from another perspective as:

$$pr\{\overline{\Delta x_I} \xrightarrow{E_1 \circ E_0 \circ E_i} \Delta x_O \xrightarrow{E_f} \Delta C\} = pr\{\overline{\Delta x_I} \xrightarrow{E_f \circ E_1 \circ E_0 \circ E_i} \Delta C \xrightarrow{E_f^{-1}} \Delta x_O\} = pr\{\overline{\Delta x_I} \xrightarrow{E} \Delta C\} \cdot pr\{\Delta C \xrightarrow{E_f^{-1}} \Delta x_O\} = p_{\Delta C} \cdot p_{wk}$$

By replacing $r \cdot p_{\Delta x_O} = p_{\Delta C} \cdot p_{wk}$ into p_{ck} we get:

$$p_{ck} = \frac{r \cdot p_{\Delta x_O} \cdot (1-p_i)}{p_{\Delta C}} = (1-p_i) \cdot p_{wk} = (1-p_i) \cdot p_f$$

Thus $p_{ck} < p_{wk}$ and this special class is always a distinguisher. Note that for $p_i = 1$ we have $p_{ck} = 0$ and the almost impossible differential is transformed into impossible differential.

In [8], based on the same idea, an attack which is called *improbable differential cryptanalysis* has been applied to 13 rounds of CLEFIA-128.

5.1 On the strength and complexity of the special class of AIDC

In this section, we compare the complexity of the IDC and its corresponding AIDC when they are applied to the same number of rounds of a block cipher (see Figure 4). Then we discuss the conditions where AIDC can improve the complexity of its IDC, or even it can break the more number of rounds than its corresponding IDC.

Complexity of IDC. Based on the parameters of the IDC in Figure 4, the probability that a wrong joint subkey $k_i|k_f$ be eliminated with one of the N_{IDC} proper pairs is $p_i.p_f$. Thus the number of the wrong subkeys remaining at the end of the impossible differential attack is $\epsilon = |k_i| \times |k_f| \times (1 - p_i.p_f)^{N_{IDC}} \approx |k_i| \cdot |k_f| \times e^{-p_i p_f N_{IDC}}$. For the sake of simplicity let us take $\epsilon = 1$, thus:

$$N_{IDC} = \frac{Ln(|k_i|.|k_f|)}{p_i p_f}$$

The time complexity depends on whether the attacker can split the target subkeys into smaller parts and check partial tests on them or not. However, since it is also the case for the corresponding AIDC, let us do not consider it for both the attacks. But, as we will see after obtaining the result of this section, for a more efficient AIDC attack we should embed the parts of the additional rounds which offer us a bigger probability. So we assume that $p_i > p_f$. The time complexity for the filtration in the final rounds is $2 \times N_{IDC} \times |k_f|$ partial decryptions, that offer $N_{IDC} \times p_f$ pairs for further analysis in the plaintext side. The time complexity for the initial rounds is $2 \times |k_f| \times |k_i| \times p_f \times N_{IDC}$ (Note that in some cases we can reduce this complexity using some additional memory and precomputation).

Complexity of AIDC. Let us follow and use the results of the procedure we took for the AIDC on Crypton. The data complexity, which is used based on this method, will be very close to the data complexity obtained from the well-known threshold-using method. We previously obtained that:

$$r \triangleq \text{pr}\ \{\ n_{ckc} < n_{wkc}\ \} = \varphi(T) = \varphi(\frac{N_{AIDC}(p_{wk} - p_{ck})}{\sqrt{N_{AIDC}p_{ck}(1-p_{ck}) + N_{AIDC}p_{wk}(1-p_{wk})}})$$

and a value about 8 for T will make the success probability close to 1. So the number of required proper pairs is computed as below:

$$T = \frac{\sqrt{(N_{AIDC})}.(p_{wk} - p_{ck})}{\sqrt{p_{ck}(1-p_{ck}) + p_{wk}(1-p_{wk})}} \approx 8$$

Replacing $p_{wk} = p_f$ and $p_{ck} = (1 - p_i)p_f$ yields:

$$N_{AIDC} = 8^2 \cdot \frac{2p_f - 2p_f^2 - p_i.p_f + 2p_i.p_f^2 - p_i^2.p_f^2}{p_i^2.p_f^2} \approx \frac{2^7}{p_i^2.p_f}$$

The time complexity is about $2 \times N_{AIDC} \times |k_f|$ partial decryptions.

Discussion: The number of proper pairs required for AIDC is about $\frac{1}{p_i}$ times larger than the required proper pairs for the corresponding IDC. Since the two attacks use the same structures of plaintexts and have the same $p_{\Delta C}$, this relation holds for their data complexities (DC), i.e.:

$$N_{AIDC} \approx \frac{1}{p_i} \cdot N_{IDC} \Longrightarrow DC_{AIDC} \approx \frac{1}{p_i} \cdot DC_{IDC}$$

Thus, in order for the AIDC to have advantage over its corresponding IDC, the time complexity of the IDC (denoted by TC_{IDC}) must be larger than its data complexity (DC_{IDC}), and furthermore AIDC must be able to reduce TC_{IDC}.

$$TC_{IDC} = 2N_{IDC}|k_f| + 2N_{IDC}|k_f||k_i|p_f > TC_{AIDC} = 2N_{AIDC}|k_f|$$

By replacing $N_{AIDC} = \frac{1}{p_i} \cdot N_{IDC}$, we get:

$$1 + |k_i| \cdot p_f > \frac{1}{p_i}$$

Hence, the AIDC can reduce the complexity of its corresponding IDC (and even can be applied to more rounds of a block cipher) if:

1. The data complexity of the IDC is larger than its time complexity, and
2. The space of subkey space which is pushed into the distinguisher is large enough to satisfy the relation $1 + |k_i| \cdot p_f > \frac{1}{p_i}$.

Depending on the structure of a cipher, a lot of techniques, including the early abort, key schedule considerations, precomputation and hash tables, and ... may be used to decrease the time complexity of the impossible differential attack. Therefore the second criterion is not strict. In practice we should compare the time complexity of the two attacks after application of these techniques.

6 Conclusion

In this paper, first we introduced the notion of almost-impossible differential cryptanalysis. This attack uses differentials which are less probable than a random permutation. Constructing the distinguisher is very similar to the miss in the middle approach used in impossible differential attack, and the key-recovery step uses counters like in the well-known differential attack. As an example of AIDC, an attack on six rounds of Crypton with a success rate very close to 1 was proposed. The data and time complexity of the attack are $2^{101.52}$ chosen plaintexts and $2^{110.2}$ memory accesses, respectively. Finally we discussed a special class of AIDC in which the distinguisher is constructed from pushing some parts of the additional rounds around an impossible differential into the distinguisher. This way the data complexity is increased but the time complexity may decreased.

In this work, the model presented for AIDC considers cases where one of the two underlying differentials in the distinguisher are deterministic and the other is probabilistic. As a future work, the case that both of this differentials are probabilistic may be considered.

References

1. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12-23. Springer, Heidelberg (1999)
2. Biham, E., Shamir, A.: Differential Cryptanalysis of the Data Encryption Standard. Springer, Heidelberg (1993)
3. Borst, J., Knudsen, L.R., Rijmen, V.: Two Attacks on Reduced IDEA. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 1–13. Springer-Verlag (1997)

4. Knudsen, L.R.: DEAL – a 128-bit Block Cipher. Technical report, Department of Informatics, University of Bergen, Norway (1998)
5. Knudsen, L.R., Rijmen, V.: On the Decorrelated Fast Cipher (DFC) and Its Theory. In Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 81-94. Springer-Verlag (1999)
6. Lim, C.H.: Crypton: A New 128-bit Block Cipher. The first Advanced Standard Encryption Candidate Conference, NIST (1998)
7. Mala, H., Shakiba, M., Dakhilalian, M.: New impossible differential attacks on reduced-round Crypton. Computer Standards & Interfaces 32(4): 222-227 (2010)
8. Tezcan, C.: The Improbable Differential Attack: Cryptanalysis of Reduced Round CLEFIA. Cryptology ePrint Archive: Report 2010/435.