Improving impossible-differential attacks against Rijndael-160 and Rijndael-224

Marine Minier

Improving Impossible Differential Attacks against Rijndael-160 and Rijndael-224*

Marine Minier

Université de Lyon, INRIA
INSA-Lyon, CITI, F-69621, Villeurbanne, France
marine.minier@insa-lyon.fr

Abstract. Impossible differential attacks are a very efficient form of cryptanalysis against block ciphers. In this paper, we improve the existing impossible differential attacks against Rijndael-160 and Rijndael-224.
Keywords: Rijndael, impossible differential attack, cryptanalysis, block ciphers.

1 Introduction

Impossible differential attack introduced in [1] is a powerful cryptanalytic method against block ciphers. Classically, impossible differential attacks take advantage of differentials which never occur for the studied permutations. From an impossible differential path, the idea is then to test if a key could verify or not the path, if yes the attacker could discard the tested key.

Rijndael [4] is an SPN block cipher designed by Vincent Rijmen and Joan Daemen. It has been chosen as the new advanced encryption standard by the NIST [6] in its 128-bit block size version for variable key lengths k, which can be set to 128, 192 or 256 bits. In its full version, the block length b is also variable and is equal to 128, 160, 192, 224 or 256 bits as detailed in [5]. We respectively called those versions Rijndael-b. The recommended number of rounds Nr is determined by b and k, and varies between 10 and 14.

Many cryptanalyses have been proposed against Rijndael for the different block sizes, the best impossible differential cryptanalyses are presented in [10,11]. In this paper, we improve the results of [10,11] against Rijndael-160 and Rijndael-224. In the case of Rijndael-160, we first introduce a new 5 rounds impossible differential that is the basis of our 8 rounds attack against Rijndael-160. For Rijndael-224, by exploiting the same 6 rounds impossible differential path than the one given in [10] and adding two rounds at the beginning and two rounds at the end, we are able to construct the first attack against 10 rounds of Rijndael-224. Table 1 sums up all the best attacks against Rijndael-160 and Rijndael-224 and gives the complexities of the new attacks presented here.

* This work was partially supported by the French National Agency of Research: ANR-11-INS-011.

Cipher	nb Rounds	Key Size	Data	Time	Memory	Type	Source
Rijndael-160	7	(all)	$2^{98.6}$ CP	$2^{98.6}$	small	Integral	[8]
	8	(192)	$2^{100.5}$ CP	$2^{174.5}$	small	Integral	[8]
	7	(192)	2^{147} CP	$2^{81.9}$	2^{64}	Imp. Diff.	[11]
	8	(192)	$2^{112.72}$ CP	$2^{158.1}$	$2^{128.72}$ bytes	Imp. Diff.	This paper
Rijndael-224	9	(256)	$2^{196.5}$ CP	$2^{196.5}$	small	Integral	[8]
	9	(256)	$2^{198.1}$ CP	$2^{195.2}$	$2^{140.4}$	Imp. Diff.	[10]
	10	(256)	$2^{185.38}$ CP	$2^{246.93}$	$2^{201.38}$ bytes	Imp. Diff.	This paper

Table 1. Summary of Best Attacks against Rijndael-160 and Rijndael-224.

This paper is organized as follows. In Section 2, we give a short description of the block cipher Rijndael-b. Section 3 recalls the complexity analysis of impossible differential attacks done in [2,3]. In Section 4 we present our 8 rounds impossible differential attack against Rijndael-160 whereas Section 5 is dedicated to the analysis of the impossible differential against 10 rounds of Rijndael-224. Finally, Section 6 concludes this paper.

2 Rijndael Description

Rijndael-b is an SPN block cipher designed by Joan Daemen and Vincent Rijmen [5]. It supports keys of length Nk and blocks of length b ranging from 128 up to 256 bits in steps of 32 bits. There are 15 instances of Rijndael. Among those 15 instances, the three versions with $b = 128$ and $Nk = 128, 192, 256$ are the AES.

The number of rounds Nr depends on the text size b and on the key size Nk and varies between 10 and 14 (see Table 2 for partial details). For all the versions, the current block at the input of the round r is represented by a $4 \times t$ with $t = (b/32)$ matrix of bytes $X^{(r)}$:

$$X^{(r)} = \begin{pmatrix} x_{0,0}^{(r)} & x_{0,1}^{(r)} & \cdots & x_{0,t}^{(r)} \\ x_{1,0}^{(r)} & x_{1,1}^{(r)} & \cdots & x_{1,t}^{(r)} \\ x_{2,0}^{(r)} & x_{2,1}^{(r)} & \cdots & x_{2,t}^{(r)} \\ x_{3,0}^{(r)} & x_{3,1}^{(r)} & \cdots & x_{3,t}^{(r)} \end{pmatrix}$$

The round function, repeated $Nr - 1$ times, involves four elementary mappings, all linear except the first one:

- `SubBytes (SB)`: a bytewise transformation that applies on each byte of the current block an 8-bit to 8-bit non linear S-box S.
- `ShiftRows (SR)`: a linear mapping that rotates on the left all the rows of the current matrix. the values of the shifts (given in Table 2) depend on b.
- `MixColumns (MC)`: a linear matrix multiplication; each column of the input matrix is multiplied by the matrix M that provides the corresponding column of the output matrix.

- `AddRoundKey (AK)`: a XOR operation between the current block and the subkey of the round r, K^r.

Those $Nr - 1$ rounds are surrounded at the top by an initial key addition with the subkey K^0 and at the bottom by a final transformation composed by a call to the round function where the `MixColumns` operation is omitted. The key schedule derives $Nr + 1$ b-bits round keys K^0 to K^{Nr} from the master key K of variable length.

We will denote by $X^{(r)I}, X^{(r)SB}, X^{(r)SR}, X^{(r)MC}$ and $X^{(r)AK}$ the input of round r and the intermediate values after the application of `SB`, `SR`, `MC` and `AK` of round r, respectively. We will also use the notation $x^{(r)}_{col(i)}$ of the internal state $X^{(r)}$ to designate the column number i starting at 0.

	AES	Rijndael-160	Rijndael-192	Rijndael-224	Rijndael-256
`ShiftRows`	(1,2,3)	(1,2,3)	(1,2,3)	(1,2,4)	(1,3,4)
Nb rounds (Nk=128)	10	11	12	13	14
Nb rounds (Nk=192)	12	12	12	13	14
Nb rounds (Nk=256)	14	14	14	14	14

Table 2. Parameters of the Rijndael block cipher where the triplet (i, j, k) for the `ShiftRows` operation designated the required number of byte shifts for the second, the third and the fourth rows.

In conclusion, the essential differences between the AES and the other Rijndael versions concern the number of rounds Nr and the `ShiftRows` parameters.

3 Recall on the Complexity Analysis of an Impossible Differential Attack from [2,3]

An impossible differential attack is a particular attack that works well against block ciphers. Impossible differential cryptanalysis have been introduced in parallel in [1] and [7]. The main idea of these attacks is to exploit particular differentials that are impossible, i.e. differentials that never occur. Based on such a distinguisher, the attacker could add at the beginning or/and at the end, some rounds to guess keybits. In this case, keybits that validate the impossible differentials are certainly wrong key guess. In this section, we recall the complexity analysis of an impossible differential attack as described in [2,3].

An impossible differential attack could be divided as shown in Fig. 1 into two main steps:

– First, an impossible differential distinguisher is built on an impossible differential path with probability 0 on r_Δ rounds with input differences Δ_X and Δ_Y output differences.

– Then, a key recovery part on r_{in} rounds added at the beginning and on r_{out} rounds added at the end could be implemented. In the backward direction, the r_{in} added rounds lead to a difference Δ_{in} computed from the difference Δ_X with probability 1. In the same way, for the forward direction, the r_{out} rounds lead to a difference Δ_{out} computed from the difference Δ_Y with probability 1.

A candidate key that verifies both differentials $\Delta_{in} \rightarrow \Delta_X$ and $\Delta_{out} \rightarrow \Delta_Y$ for some plaintext/ciphertext pairs is certainly a wrong key as it means this candidate key verifies a differential path which is in fact impossible.

During the key recovery process, the complexity of the attack is determined by the number of key bits that intervene into the computations to get Δ_X from Δ_{in} and to get Δ_Y from Δ_{out}. We call k_{in} the set of key bits that allow the computation of Δ_X from Δ_{in} (its cardinality is thus $|k_{in}|$) and k_{out} the set of key bits that allow the computation of Δ_Y from Δ_{out} (its cardinality is thus $|k_{out}|$). Finally, we denote by k the set $k_{in} \cup k_{out}$.

Moreover, for a given key, the differential paths from Δ_{in} to Δ_X and Δ_Y to Δ_{out} are verified by testing some bit-conditions. We call c_{in} the number of bit-conditions in the backward direction and c_{out} the number of bit-conditions in the forward direction. In other words, the probability to go from Δ_{in} to Δ_X (resp. from Δ_{out} to Δ_Y) is equal to $2^{-c_{in}}$ (resp. $2^{-c_{out}}$).

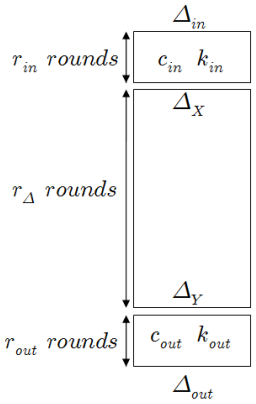

Fig. 1. Overview of an Impossible Differential Attack.

Following the study of [2,3], we could derive from those notations, the following formulas:

– The probability that, for a given key, a pair of inputs with differences $(\Delta_{in}, \Delta_{out})$ verifies all the bit-conditions and can be rejected is $2^{-(c_{in}+c_{out})} = 2^{-c}$.
– the required number of input (or output) pairs N must be such that the false positive probability (i.e. a wrong trial key stays in the candidate key

set) must be as small as possible. This probability is equal to $P = (1 - 2^{-(c_{in}+c_{out})})^N$. So, we should choose N such that

$$P = (1 - 2^{-(c_{in}+c_{out})})^N = \frac{1}{2^{|k_{in} \cup k_{out}|}}$$

Thus, we obtain:

$$\begin{aligned} e^{N \times \ln(1-2^{-c})} &= 2^{-|k|} \\ e^{-N2^{-c}} &\approx 2^{-|k|} \\ N2^{-c}/\ln(2) &\approx |k| \end{aligned}$$

leading to

$$N \approx 2^{c+\log_2(|k|\ln(2))} \tag{1}$$

We will denote by ϵ the value $\log_2(|k|\ln(2))$.

- Thus, at this step, we need to find N pairs that verify a given truncated differential. From [2,3] using the limited birthday problem, the cost C_N for constructing N such pairs is:

$$C_N = \max\left\{ \min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{ \sqrt{N2^{n+1-|\Delta|}} \right\}, N2^{n+1-|\Delta_{in}|-|\Delta_{out}|} \right\}. \tag{2}$$

  verifying that $C_N < 2^n$ where n is the size of the block to cipher.
- Finally, the overall complexity C_T of the attack is given by

$$C_T = \left(C_N + \left(N + 2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}} \right) C'_E + 2^{|K|} P \right) C_E \tag{3}$$

  where C'_E is the ratio of the cost of partial encryption to the full encryption; C_E is the cost of one encryption; and finally $2^{|K|}P$ designates the time required for the complete exhaustive key search of K after the impossible differential attack. Note that, most of the time, as $C_N \times C_E$ and $2^{|K|}P$ are not the complexity bottlenecks, C_T could be approximate by $\left(N + 2^{|k_{in} \cup k_{out}|} \frac{N}{2^{c_{in}+c_{out}}}\right) C'_E \times C_E$.
- The memory complexity is determined by the number of N pairs we have to store and is thus equal to N.

4 8 Rounds Impossible Differential Attack on Rijndael-160

In this Section, we will describe the impossible differential attack on 8 rounds of Rijndael-160 using a new impossible differential path. Both, the complete attack on 8 rounds and the impossible differential path are presented in Fig. 2. Note also that using this impossible differential path to attack 7 rounds only marginally improves the attack complexity given in [11]. Thus, we will not describe this 7 rounds attack.

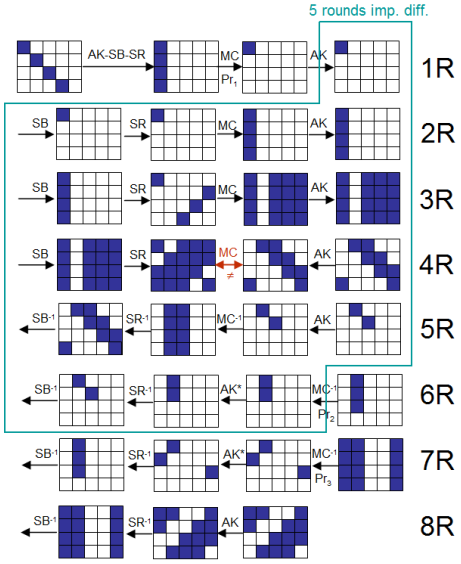

Fig. 2. The Complete Impossible Differential Attack on Rijndael-160. The impossible differential path on 5 rounds is surrounded by one round at the beginning and by 2 rounds at the end. Bytes with differences are the colored ones. The `AK*` operation stands for the classical case where the order of the `AK` and of the `MC` operations is inverted. This could be down up to a linear transformation of the current subkey.

4.1 The Used Impossible Differential Path

The impossible differential path presented in Fig. 2 works on 5 rounds of Rijndael-160. It is different from the one used in [11] because instead of using a path with $|\Delta_X| = 8$ and $|\Delta_Y| = 8$ leading to a 7 rounds attack with $|\Delta_{in}| = 32$ and $|\Delta_{out}| = 32$, our path has two bytes of differences in the output (i.e. $|\Delta_Y| = 16$) leading to a contradiction in the middle coming from the fact that a complete column of non-zero differences - generated through the `MC` operation by a single active byte - could not be equal to a 0 byte difference.

The main advantage of this path is the fact that we could reduce the number of active bytes on 8 rounds. If we use the impossible differential path of [11], we have $|\Delta_{in}| = 32$ and $|\Delta_{out}| = 32$ for a 7 rounds attack whereas our impossible differential path leads to $|\Delta_{in}| = 32$ and $|\Delta_{out}| = 24$ on 7 rounds and to $|\Delta_{in}| = 32$ and $|\Delta_{out}| = 96$ on 8 rounds. This last property allows to mount an attack on 8 rounds whereas the complexity of the attack built on 7 rounds with our impossible differential path is only marginally improved.

4.2 Attack Description on 8 rounds

From the 5 rounds impossible differential path shown in Fig. 2, we could add one round at the beginning and two rounds at the end (without the `MixColumns` operation in the 8th round). We first construct 4 precomputed tables to reduce the overall complexity of the attack following the same methodology as the one described in [9]. We then describe the attack and evaluate its complexity following also the work [9]. A summary of the attack complexity is given in Table 3.

Construction of Precomputed Tables. The precomputed tables will be denoted T_1, T_2, T_3 and T_4 and will help us during the key guess steps of the attack to retrieve the subkey bytes of K^8 : $K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2}$ for T_1 and $K^8_{0,4}, K^8_{1,3}, K^8_{2,2}, K^8_{3,1}$ for T_2 and $K^8_{0,1}, K^8_{1,0}, K^8_{2,4}, K^8_{3,3}$ for T_3 ; and of $K*^7$: $K*^7_{0,1}, K*^7_{1,0}, K*^7_{2,4}$ for T_4.

- **Tables T_1, T_2 and T_3:** For all the $2^{32} \times 4 \times (2^8 - 1) \approx 2^{42}$ possible pairs $((x^{7AK}_{0,0}, x^{7AK}_{0,1}, x^{7AK}_{0,2}, x^{7AK}_{0,3}), (x'^{7AK}_{0,0}, x'^{7AK}_{0,1}, x'^{7AK}_{0,2}, x'^{7AK}_{0,3}))$ after the key addition with K^{*7} which have zero difference in exactly 3 out of the 4 bytes, compute the possible values of $((x^{8SR}_{0,0}, x^{8SR}_{1,4}, x^{8SR}_{2,3}, x^{8SR}_{3,2}), (x'^{8SR}_{0,0}, x'^{8SR}_{1,4}, x'^{8SR}_{2,3}, x'^{8SR}_{3,2}))$. Store the obtained pairs in an hash table T_1 indexed by their difference. T_1 has 2^{32} rows and on average about $2^{42}/2^{32} = 2^{10}$ pairs lie in each row. Do the same for T_2 and T_3 to store the possible values:
  - of $((x^{7AK}_{1,0}, x^{7AK}_{1,1}, x^{7AK}_{1,2}, x^{7AK}_{1,3}), (x'^{7AK}_{1,0}, x'^{7AK}_{1,1}, x'^{7AK}_{1,2}, x'^{7AK}_{1,3}))$ in T_2
  - and of $((x^{7AK}_{4,0}, x^{7AK}_{4,1}, x^{7AK}_{4,2}, x^{7AK}_{4,3}), (x'^{7AK}_{4,0}, x'^{7AK}_{4,1}, x'^{7AK}_{4,2}, x'^{7AK}_{4,3})$ in T_3.

  Once a byte position is determined by the first column key guess, the number of possible candidates for the two other columns of key guess becomes 2^8.
- **Table T_4:** For all of the about 2^{48} possible pairs of $(x^{7SR}_{0,1}, x^{7SR}_{1,0}, x^{7SR}_{2,4}, x'^{7SR}_{0,1}, x'^{7SR}_{1,0}, x'^{7SR}_{2,4})$ (4 other positions are also possible among the six possible such as $((1,1),(0,2),(3,4))$), compute the possible values of $(x^{6AK*}_{col(1)}, x'^{6AK*}_{col(1)})$ having zero difference in exactly 2 particular bytes among 4. Store the qualified pairs in an hash table T_4 indexed by their difference. We obtain about $2^{48} \times 2^{-16} \times 4 = 2^{34}$ such pairs. Thus, as T_4 has 2^{18} rows, and on average about $2^{34}/2^{18} = 2^{16}$ pairs lie in each row.

4.3 Attack Description.

The details of the attack are as follows:

1. First construct N pairs of plaintexts/ciphertexts that verify the input and output truncated differentials. The required number N of pairs will be determined by Equation (1) and the cost to generate them C_N by Equation (2).
2. We then follow the methodology proposed in [9] that uses the fact that given an input and an output difference of the Rijndael S-box, there is on average one pair of values that satisfies these differences. Indeed, $\Delta X^{(1)}$ is known

as the plaintext difference and the knowledge of $\Delta X^{(1)SB}_{SR^{-1}(col(0))}$ could help to find the correct input values and thus the corresponding 4 bytes of the subkey K^0. There are only $4 \times (2^8 - 1) \approx 2^{10}$ possible values of $\Delta X^{(1)MC}_{col(0)}$ with only one byte with a non-zero difference and thus the same number of possible values of $\Delta X^{(1)SB}_{SR^{-1}(col(0))}$. So, perform the following substeps:

(a) Initialize 2^{32} empty lists each corresponding to a different value of ($K^0_{0,0}$, $K^0_{1,1}$, $K^0_{2,2}$, $K^0_{3,3}$).

(b) For each pair of the N set and for each of the 2^{10} possible differences in $\Delta X^{(1)SB}_{SR^{-1}(col(0))}$, compute the keys which lead this specific plaintext pair to the wanted difference. Add this plaintext pair to the list corresponding to the key value.

For each of the N pairs, about 2^{10} values of $\Delta X^{(1)SB}_{SR^{-1}(col(0))}$ are examined. These $N \times 2^{10}$ possible values are distributed in 2^{32} lists. For a given subkey value $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,3})$, the number of stored pairs is $N \times 2^{-22}$.
For each of the possible subkey value $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,3})$ and for each of the $N \times 2^{-22}$ remaining plaintext pairs (P, P') in that list, perform the following steps:

3. Access the row with index $\Delta C_{((0,0),(1,4),(2,3),(3,2))}$ in table T_1. For each pair (y_1, z_1) in that row, select the value $C_{((0,0),(1,4),(2,3),(3,2))} \oplus y_1$. We thus expect to obtain about 2^{10} candidates for $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$ from the table T_1.
4. Repeat the same process using the tables T_2 and T_3 to determine the possible subkey bytes candidates $(K^8_{0,1}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$ and $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$. This time and as the first possible position of the byte difference is fixed only 2^8 candidates for both $(K^8_{0,1}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$ and $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$ will remain.
5. For each possible candidate of $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$, $(K^8_{0,1}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$ and $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$, perform the following substeps:
   (a) Partially decrypt the 8th round from each ciphertext pair to get ($x^{7AK*}_{0,1}$, $x^{7AK*}_{1,0}, x^{7AK*}_{2,4}$, $x'^{7AK*}_{0,1}, x'^{7AK*}_{1,0}$, $x'^{7AK*}_{2,4}$).
   (b) Access the corresponding row in T_4. For each pair (y_2, z_2) in that row, select the value $(x^{7AK*}_{0,1}, x^{7AK*}_{1,0}, x^{7AK*}_{2,4}) \oplus y_2$ as a key candidate for $(K^7_{0,1}, K^7_{1,0}, K^7_{2,4})$. We thus expect to obtain about 2^{16} candidates for $(K^7_{0,1}, K^7_{1,0}, K^7_{2,4})$ from the table T_4.
6. Final step: for each of the $N \times 2^{-22}$ possible pairs, we know $2^{10} \times 2^8 \times 2^8 \times 2^{16} = 2^{42}$ joint values of $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$, $(K^8_{0,1}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$, $(K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2})$ and $(K^7_{0,1}, K^7_{1,0}, K^7_{2,4})$ that result in the impossible differential. Remove this value from the list of all the 2^{120} possible values for these joint subkeys (Note that $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,3})$ has been previously guessed). After the trial of all the pairs, if the list is not empty, announce the values in the list along with the guess of $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,3})$ as the candidates for the correct target subkey.

Complexity of the Attack. Table 3 gives a complexity analysis step by step of the previous algorithm.

Step	Guessed Bytes	# Pairs Kept	Time Complexity
1	0	N	C_N
2	$K^0 : K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,3}$	$N2^{-22}$	$2^{10}N$ Mem. Acc.
3	$K^8 : K^8_{0,4}, K^8_{1,3}, K^8_{2,2}, K^8_{3,1}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}$ Mem. Acc.
4	$K^8 : K^8_{0,1}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^8$ Mem. Acc.
5	$K^8 : K^8_{0,0}, K^8_{1,4}, K^8_{2,3}, K^8_{3,2}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^82^8$ Mem. Acc.
5a		$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^82^8$ Part. Enc.
5b	$K*^7 : K*^7_{0,1}, K*^7_{1,0}, K*^7_{2,4}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^82^82^{16}$ Mem. Acc.
6	all	$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^82^82^{16}$ Mem. Acc.

Table 3. Summary of the 8 Rounds Attack Steps.

Clearly, the overall time complexity of the attack will be dominated by the steps 5b and 6, so the overall complexity of the attack is $N2^{53}$ memory access. As we could say that one round of Rijndael-160 has a cost of 25 memory accesses, the overall complexity of the attack is about $N2^{53} \times \frac{1}{8} \times \frac{1}{25} \approx N2^{45.35}$ 8 rounds Rijndael-160 encryptions.

Finally, to completely compute the data, time and memory complexities of the attack using the equations presented in Section 3, we need to determine the following unknowns: c_{in}, c_{out}, Δ_{in}, Δ_{out}, k_{in} and k_{out}. From Fig. 2 and the previous Subsection, we deduce that:

- $c_{in} = \log_2(1/Pr_1)$, $c_{out} = \log_2(1/Pr_3) + \log_2(1/Pr_2)$ with $Pr_1 = 2^{-22}$ (every possible byte could be active), $Pr_2 = 2^{-14}$ and $Pr_3 = 2^{-70}$. Thus $c = 22 + 14 + 70 = 106$.
- $|\Delta_{in}| = 32$, $|\Delta_{out}| = 96$.
- $|k_{in}| = 32$, $|k_{out}| = 120$. Thus $k = 152$.

Thus, from Equation (1), we have: $N = 2^{c+\epsilon} = 2^{106+\epsilon}$ where ϵ is equal to $\log_2(|k|\ln(2)) = \log_2(152 \times 0.69315) \approx 6.72$ to obtain $P = 2^{-152}$ and to discard all the wrong subkey bits as we need to guess 152 subkey bits. Thus, $N = 2^{106+\epsilon} = 2^{112.72}$. From Equation (2),

$$\begin{aligned} C_N &= \max\left\{ \min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{ \sqrt{N2^{n+1-\Delta}} \right\}, N2^{N+1-|\Delta_{in}|-|\Delta_{out}|} \right\} \\ &= \max\left\{ \sqrt{2^{112.72}2^{161-96}}, 2^{112.72}2^{161-32-96} \right\} \\ &= \max\left\{ 2^{88.86}, 2^{145.72} \right\} = 2^{145.72} \end{aligned}$$

From equation (3) and for a 192-bit key,

$$\begin{aligned} C_T &= \left(2^{145.72} + \left(2^{112.72} + 2^{112.72} \times \frac{2^{152}}{2^{106}} \right) \frac{1}{8} + 2^{192} \times 2^{-152} \right) \\ &\approx 2^{155.72} \text{ 8 rounds Rijndael-160 encryptions} \end{aligned}$$

Note that the complexity computed using Table 3 gives us a complexity equal to $C_T = N2^{45.35} \approx 2^{158.1}$ 8 rounds Rijndael-160 encryptions. We will keep this last

value as our attack is really based on the analysis given in Table 3. Note that the value of C_T given in Equation (3) is a theoretical value not always reachable by real attacks. The memory requirement is equal to $2 \times N2^{10} = 2^{123.72}$ (step 2) 160-bit plaintexts corresponding with $2^{128.72}$ bytes.

5 Impossible Differential Attack on Rijndael-224 up to 10 rounds

In this Section, we will briefly describe the impossible differential attacks on 10 rounds of Rijndael-224 using the same impossible differential path on 6 rounds than in [10]. Both, the complete attack on 10 rounds and the impossible differential path are presented in Fig. 3. The main difference between the attack proposed in [10] and the one presented here relies on the fact that using the probability Pr_4, we could add one more round at the end to build a 10 rounds attack.

As for the case of Rijndael-160, the attack could be divided into two steps. First, the attacker computes some pre-stored tables and then she launches the complete attack. As this attack is very similar to the previous case, we will not describe it in details. The process is about the same than for the attack on Rijndael-160 except that one more step is added at the beginning to enter the impossible differential path. The precomputed tables are the following ones:

- **Tables T_1, T_2 and T_3:** Those three tables are dedicated to the storage of the possible values $(K^0_{0,1}, K^0_{1,2}, K^0_{2,3}, K^0_{3,5})$, $(K^0_{0,3}, K^0_{1,4}, K^0_{2,5}, K^0_{3,0})$ and $(K^0_{0,6}, K^0_{1,0}, K^0_{2,1}, K^0_{3,3})$. For those tables, store all the $2^{32} \times (2^8 - 1) \approx 2^{42}$ possible pairs which have zero difference in exactly 3 pre-determined out of the 4 bytes. Each table has on average 2^{32} rows and on average about 2^8 pairs lie in each row.
- **Tables T_5:** This table is dedicated to the storage of possible values for $K^1_{0,6}, K^1_{1,0}, K^1_{2,1}, K^1_{3,2}$. As the previous tables, this table has on average 2^{32} rows and on average about 2^{10} pairs lie in each row.
- **Tables T_6 and T_7:** Those two tables are dedicated to the storage of the possible values $K^{10}_{0,2}, K^{10}_{1,3}, K^{10}_{2,4}, K^{10}_{3,6}$ and $K^{10}_{0,4}, K^0_{1,5}, K^0_{2,6}, K^0_{3,1}$. Those tables have 2^{32} rows and respectively about 2^{10} pairs and 2^8 pairs lying in each row.
- **Tables T_8:** This table is dedicated to the storage of possible values for $K^{*9}_{2,2}, K^{*9}_{3,4}$. This table is built on the 2^{32} possible pairs with two bytes with difference and stores the 2^{18} possible values according the 2^{18} possible differences which gives one possible candidate solution.

The summary of the attack is as follows:

1. First construct N pairs of plaintexts/ciphertexts that verify the input and output truncated differentials. The required number N of pairs will be determined by Equation (1) and the cost to generate them C_N by Equation (2).

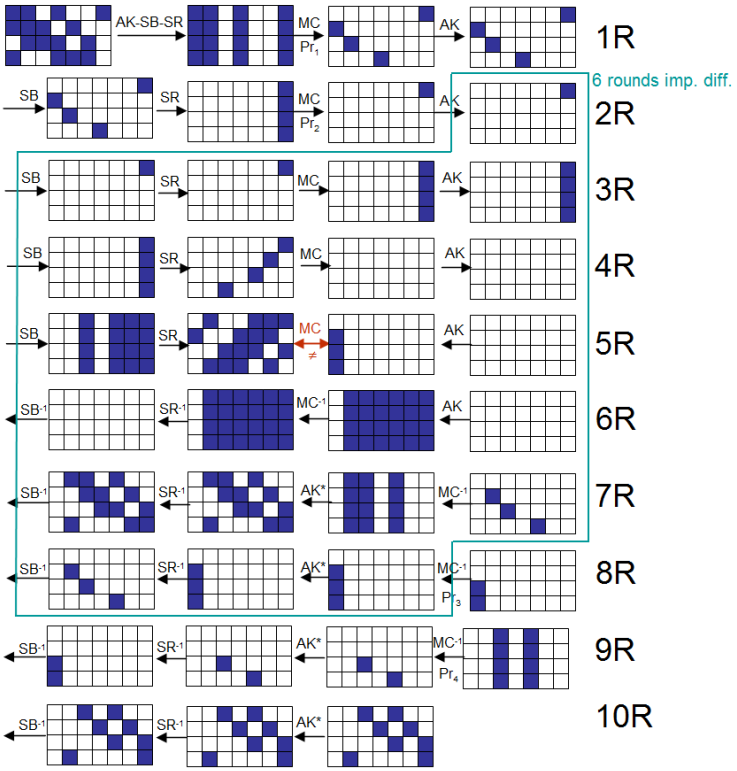

Fig. 3. The Complete Impossible Differential Attack on Rijndael-224. The impossible differential Path on 6 rounds is surrounded by 2 rounds at the beginning and 2 rounds at the end. Bytes with differences are the colored ones. The `AK*` operation stands for the classical case where the order of the `AK` and of the `MC` operations is inverted. This could be down up to a linear transformation of the current subkey.

2. Then, perform the following substeps:
   (a) Initialize 2^{32} empty lists each corresponding to a different value of $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,4})$.
   (b) For each pair of the N set and for each of the 2^{10} possible differences in $\Delta X^{(1)SB}_{SR^{-1}(col(0))}$, compute the keys which lead this specific plaintext pair to the wanted difference. Add this plaintext pair to the list corresponding to the key value.

   For each of the N pairs, about 2^{10} values of $\Delta X^{(1)SB}_{SR^{-1}(col(0))}$ are examined. These $N \times 2^{10}$ possible values are distributed in 2^{32} lists. For a given subkey value $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,4})$, the number of stored pairs is $N \times 2^{-22}$.
   For each of the possible subkey value $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,4})$ and for each

of the $N \times 2^{-22}$ remaining plaintext pairs (P, P') in that list, perform the following steps:

3. Test the possible candidate keys $(K^0_{0,1}, K^0_{1,2}, K^0_{2,3}, K^0_{3,5})$, $(K^0_{0,3}, K^0_{1,4}, K^0_{2,5}, K^0_{3,0})$ and $(K^0_{0,6}, K^0_{1,0}, K^0_{2,1}, K^0_{3,3})$ using T_1, T_2 and T_3. The number of possible values is each time 2^8.
4. For each possible candidate of $(K^0_{0,1}, K^0_{1,2}, K^0_{2,3}, K^0_{3,5})$, $(K^0_{0,3}, K^0_{1,4}, K^0_{2,5}, K^0_{3,0})$ and $(K^0_{0,6}, K^0_{1,0}, K^0_{2,1}, K^0_{3,3})$ perform the following substeps:
   (a) Partially encrypt the second round from each plaintext pair to get $(x^{2MC}_{col(6)})$.
   (b) Access the corresponding row in T_5 and select the corresponding value as a key candidate for $K^1_{0,6}, K^1_{1,0}, K^1_{2,1}, K^1_{3,2}$. We thus expect to obtain about 2^{10} candidates.
5. Do the same in the deciphering direction: use tables T_6 and T_7 to determine from the ciphertexts the 2^{10} possible candidates for $(K^{10}_{0,2}, K^{10}_{1,3}, K^{10}_{2,4}, K^{10}_{3,6})$ and the 2^8 possible candidates for $(K^{10}_{0,4}, K^0_{1,5}, K^0_{2,6}, K^0_{3,1})$.
6. Then, for each possible of those values, perform the following substeps to determine the last two key bytes values $K^{*9}_{2,2}, K^{*9}_{3,4}$:
   (a) Partially decrypt the tenth round from each possible ciphertext pair to get $(x^{9AK*}_{(2,2)})$ and $(x^{9AK*}_{(3,4)})$.
   (b) Access the corresponding row in T_8 and select the corresponding value as a key candidate for $K^{*9}_{2,2}, K^{*9}_{3,4}$. We thus expect to obtain about 1 candidate on average.
7. Final step: for each of the $N \times 2^{-22}$ possible pairs, we know $2^8 \times 2^8 \times 2^8 \times 2^{10} \times 2^{10} \times 2^8 \times 2^8 = 2^{60}$ joint values of all the possible subkeys that result in the impossible differential path. Remove this value from the list of all the 2^{208} possible values for these joint subkeys (Note that $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,4})$ is previously guessed). After the trial of all the pairs, if the list is not empty, announce the values in the list along with the guess of $(K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,4})$ as the candidates for the correct target subkey.

Step	Guessed Bytes	# Paris Kept	Time Complexity
1	0	N	C_N
2	$K^0 : K^0_{0,0}, K^0_{1,1}, K^0_{2,2}, K^0_{3,4}$	$N2^{-22}$	$2^{10}N$ Mem. Acc.
3	$K^0 : K^0_{0,1}, K^0_{1,2}, K^0_{2,3}, K^0_{3,5}$	$N2^{-22}$	$2^{32}2^8N2^{-22} = 2^{18}N$ Mem. Acc.
3	$K^0 : K^0_{0,3}, K^0_{1,4}, K^0_{2,5}, K^0_{3,0}$	$N2^{-22}$	$2^{32}2^8 2^8 N2^{-22} = 2^{26}N$ Mem. Acc.
3	$K^0 : K^0_{0,6}, K^0_{1,0}, K^0_{2,1}, K^0_{3,3}$	$N2^{-22}$	$2^{32}2^8 2^8 2^8 N2^{-22} = 2^{34}N$ Mem. Acc.
4a		$N2^{-22}$	$2^{32}2^8 2^8 2^8 N2^{-22} = 2^{34}N$ Part. Enc.
4b	$K^1 : K^1_{0,6}, K^1_{1,0}, K^1_{2,1}, K^1_{3,2}$		$2^{32}2^8 2^8 2^8 N2^{-22}2^{10} = 2^{44}N$ Mem. Acc.
5	$K^{10} : K^{10}_{0,2}, K^{10}_{1,3}, K^{10}_{2,4}, K^{10}_{3,6}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}$ Mem. Acc.
5	$K^{10} : K^{10}_{0,4}, K^0_{1,5}, K^0_{2,6}, K^0_{3,1}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^8 = 2^{28}N$ Mem. Acc.
6a		$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^8 = 2^{28}N$ Part. Enc.
6b	$K^9 : K^{*9}_{2,2}, K^{*9}_{3,4}$	$N2^{-22}$	$2^{32}N2^{-22}2^{10}2^8 2^8 = 2^{36}N$ Mem. Acc.
7	all	$N2^{-22}$	$2^{32}N2^{-22}2^{34}2^{26} = 2^{70}N$ Mem. Acc.

Table 4. Summary of the 10 Rounds Attack Steps against Rijndael-224.

Table 4 sums up the complexities of each step of the previous algorithm step by step. From this table, we could see that the overall time complexity of the attack is dominated by step 7, so the overall complexity of the attack is $N2^{70}$ memory access. As we could say that one round of Rijndael-224 has a cost of 35 memory accesses, the overall complexity of the attack is about $N2^{70} \times \frac{1}{10} \times \frac{1}{35} \approx N2^{61.55}$ 10 rounds Rijndael-224 encryptions.

To completely compute the data (and N), time and memory complexities of the attack, we need to determine all the unknowns defined in Section 3. From Fig. 3, we could deduce that:

- $c_{in} = \log_2(1/Pr_1) \times \log_2(1/Pr_2)$, $c_{out} = \log_2(1/Pr_3) \times \log_2(1/Pr_4)$ with $Pr_1 = 2^{-94}$, $Pr_2 = 2^{-24}$, $Pr_3 = 2^{-14}$ and $Pr_4 = 2^{-46}$. Thus, $c_{in} = 118$, $c_{out} = 60$ and $c = 178$.
- $|\Delta_{in}| = 128$, $|\Delta_{out}| = 64$.
- $|k_{in}| = 160$, $|k_{out}| = 80$.

Thus, from Equation (1), we have: $N = 2^{178+\epsilon}$ where ϵ is equal to $\log_2(|k|\ln(2)) = \log_2(240 \times 0.69315) \approx 7.38$ to obtain $P = 2^{-240}$ and to discard all the wrong subkey bits as we need to guess 240 subkey bits. Thus, $N = 2^{178+\epsilon} = 2^{185.38}$.

From Equation (2),

$$\begin{aligned} C_N &= \max\left\{ \min_{\Delta \in \{\Delta_{in}, \Delta_{out}\}} \left\{ \sqrt{N2^{n+1-\Delta}} \right\}, N2^{N+1-|\Delta_{in}|-|\Delta_{out}|} \right\} \\ &= \max\left\{ \sqrt{2^{185.38}2^{225-128}}, 2^{185.38}2^{225-128-64} \right\} \\ &= \max\left\{ 2^{141.19}, 2^{218.38} \right\} = 2^{218.38} \end{aligned}$$

From equation (3) and for a 256-bit key,

$$\begin{aligned} C_T &= \left(2^{218.38} + \left(2^{185.38} + 2^{185.38} \times \frac{2^{240}}{2^{178}} \right) \frac{1}{10} + 2^{256} \times 2^{-240} \right) \\ &\approx 2^{244.06} \text{ 10 rounds Rijndael-224 encryptions} \end{aligned}$$

Note that the complexity computed using Table 4 gives us a complexity equal to $C_T = N2^{61.55} \approx 2^{185.38+61.55} = 2^{246.93}$ 10 rounds Rijndael-224 encryptions. We will keep this last value as our attack is really based on the analysis given in Table 4. The memory requirement depends on step 2 and is thus equal to $2 \times 2^{10} \times 2^{185.38}$ 224-bit words corresponding with $2^{201.38}$ bytes.

6 Conclusion

We have proposed in this paper two impossible differential attacks against respectively 8 rounds of Rijndael-160 and 10 rounds of Rijndael-224. As future works, we will try to improve the two proposed algorithms to decrease the memory and time complexities of the two attacks.

References

1. E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In *Advances in Cryptology - EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer, 1999.
2. Christina Boura, Marine Minier, María Naya-Plasencia, and Valentin Suder. Improved Impossible Differential Attacks against Round-Reduced LBlock. Cryptology ePrint Archive, Report 2014/279, 2014. `http://eprint.iacr.org/`.
3. Christina Boura, María Naya-Plasencia, and Valentin Suder. Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon. In *Advances in Cryptology - ASIACRYPT 2014*, volume 8873 of *Lecture Notes in Computer Science*, pages 179–199. Springer, 2014.
4. J. Daemen and V. Rijmen. AES Proposal: Rijndael. In *The First Advanced Encryption Standard Candidate Conference*. N.I.S.T., 1998.
5. J. Daemen and V. Rijmen. *The Design of Rijndael*. Springer-Verlag, 2002.
6. FIPS 197. Advanced Encryption Standard. Federal Information Processing Standards Publication 197, 2001. U.S. Department of Commerce/N.I.S.T.
7. L. Knudsen. DEAL-a 128-bit block cipher. *complexity*, 258(2), 1998.
8. Yan-Jun Li and Wen-Ling Wu. Improved integral attacks on rijndael. *J. Inf. Sci. Eng.*, 27(6):2031–2045, 2011.
9. Hamid Mala, Mohammad Dakhilalian, Vincent Rijmen, and Mahmoud Modarres-Hashemi. Improved impossible differential cryptanalysis of 7-round AES-128. In *Progress in Cryptology - INDOCRYPT 2010*, volume 6498 of *Lecture Notes in Computer Science*, pages 282–291. Springer, 2010.
10. Qingju Wang, Dawu Gu, Vincent Rijmen, Ya Liu, Jiazhe Chen, and Andrey Bogdanov. Improved Impossible Differential Attacks on Large-Block Rijndael. In *Information Security and Cryptology - ICISC 2012*, volume 7839 of *Lecture Notes in Computer Science*, pages 126–140. Springer, 2012.
11. Lei Zhang, Wenling Wu, Je Hong Park, Bonwook Koo, and Yongjin Yeom. Improved Impossible Differential Attacks on Large-Block Rijndael. In *Information Security - ISC 2008*, volume 5222 of *Lecture Notes in Computer Science*, pages 298–315. Springer, 2008.