Differential Cryptanalysis of Q*

Eli Biham[1], Vladimir Furman[1], Michal Misztal[2], and Vincent Rijmen[3]**

[1] Computer Science Department, Technion – Israel Institute of Technology, Haifa
32000, Israel. {biham,vfurman}@cs.technion.ac.il,
http://www.cs.technion.ac.il/~biham/.
[2] Institute of Mathematics and Operational Research
Military University of Technology, 00-908 Warsaw, ul. Kaliskiego 2, Poland
[3] ESAT/COSIC, K.U.Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium

Abstract. Q is a block cipher based on Rijndael and Serpent, which was submitted as a candidate to the NESSIE project by Leslie McBride. The submission document of Q describes 12 one-round iterative characteristics with probability 2^{-18} each. On 7 rounds these characteristics have probability 2^{-126}, and the author of Q claims that these are the best 7-round characteristics. We find additional one-round characteristics that can be extended to more rounds. We also combine the characteristics into differentials. We present several differential attacks on the full cipher. Our best attack on the full Q with 128-bit keys (8 rounds) uses 2^{105} chosen plaintexts and has a complexity of 2^{77} encryptions. Our best attack on the full Q with larger key sizes (9 rounds) uses 2^{125} chosen ciphertexts, and has a complexity of 2^{96} for 192-bit keys, and 2^{128} for 256-bit keys.

1 Introduction

Q is a block cipher submitted as a candidate to the NESSIE project [6] by Leslie McBride [5]. The best previous differential attack on Q is presented in its submission document (see [2] for more details about differential cryptanalysis). The documentation of Q describes 12 one-round iterative differential characteristics with probability 2^{-18} each. The author of Q claims that the best 7-round characteristics of Q are constructed by concatenating these one-round characteristics. These 7-round characteristics have probability 2^{-126}.

In this paper we show additional one-round characteristics that can be extended to more rounds. We also join the characteristics into differentials [4]. Subsequently, we present several differential attacks on full Q. A straightforward attack on 8 rounds uses 2^{124} chosen plaintexts and has negligible complexity. A more advanced attack on 8 rounds uses 2^{105} chosen plaintexts and has

* The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-1999-12324 and by the fund for the promotion of research at the Technion.

** FWO research assistant, sponsored by the Fund for Scientific Research - Flanders (Belgium). This research was sponsored in part by GOA project Mefisto 2000/06.

M. Matsui (Ed.): FSE 2001, LNCS 2355, pp. 174–186, 2002.

a complexity of 2^{77}. Our advanced attack on 9 rounds uses about 2^{125} chosen ciphertexts and has a complexity of 2^{128}. We discuss how to improve the complexity of this attack to 2^{96} in the case of 192-bit keys. In the 8- and 9-round attacks, we can reduce the complexity by using more plaintexts/ciphertexts, or reduce the number of chosen plaintexts/ciphertexts with a larger complexity of analysis.

The paper is organized as follows: In Section 2 we describe the structure of Q. In Section 3 we show the characteristics and the differentials of Q, for various numbers of rounds. In Section 4 we present the straightforward attack. In Section 5 we describe the more advanced attacks. We present a short conclusion in Section 6. Finally, in Appendix A we show the key scheduling of Q.

2 The Structure of Q

The cipher has 128-bit blocks and the key size may be 128 bits, 192 bits or 256 bits. The block is divided into four 32-bit words (a word consists of four bytes):

03	13	23	33
02	12	22	32
01	11	21	31
00	10	20	30
word 0	word 1	word 2	word 3

where ‘00’ is the least significant byte, and ‘33’ is the most significant byte. Groups of four bytes taken from different words, but from the same position in the words, are called *rows* (for example, the group of bytes ‘03’,‘13’,‘23’ and ‘33’ is a row).

Figure 1 describes the round function of Q. The Keying layers XOR the corresponding subkeys (KA, KB or K_n) to the data, where KA and KB are fixed in all rounds, and K_r is a subkey used in round r only. The Byte Substitution layer is taken from Rijndael [3]. It substitutes the value of each byte in the data according to the S-Table. The Bit-Slice layer is taken from Serpent [1]: for $i = 0, \ldots, 31$, we construct nibble i by taking bit i from every word, then we replace each nibble according to the S-box (either A or B) and return the new bit values to their original places. The permutation changes the order of the bytes in the words in the following way: word 0 is not changed, word 1 is rotated by 1 byte (the bytes ‘10’,‘11’,‘12’,‘13’ become ‘13’,‘10’,‘11’,‘12’), word 2 is rotated by 2 bytes and word 3 is rotated by 3 bytes. To simplify the following analysis we divide the round operations into two parts, called ‘part I’ and ‘part II’, as shown in Figure 1.

The full cipher consists of 8 rounds in the case of 128-bit keys and 9 rounds for longer key lengths. The full cipher is illustrated in Figure 2.

3 Differentials of Q

Figure 3 describes the one-round characteristic with probability 2^{-18} presented

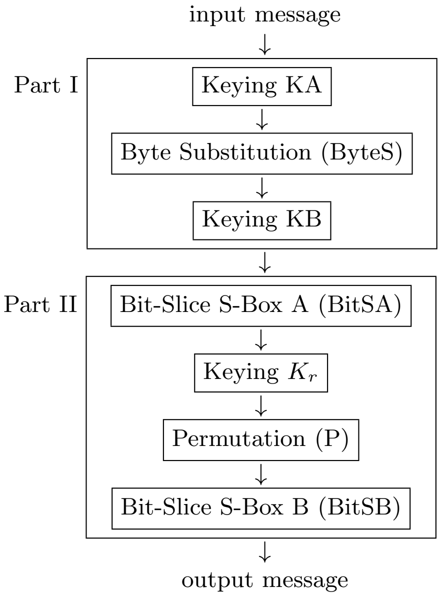

Fig. 1. The round function of Q

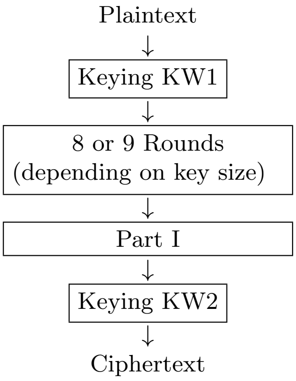

Fig. 2. Outline of Q

$$\begin{bmatrix} 0 & \delta_i & 0 & \delta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow[ByteS]{2^{-14}} \begin{bmatrix} 0 & \delta_i & 0 & \delta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow[BitSA]{2^{-2}} \begin{bmatrix} 0 & \delta_i & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\xrightarrow[P]{1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \delta_i & 0 & 0 \end{bmatrix} \xrightarrow[BitSB]{2^{-2}} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \delta_i & 0 & \delta_i \end{bmatrix}$$

Fig. 3. The one-round characteristic described by McBride

by McBride in [5], where δ_i is the 8-bit value 2^i ($i \in \{0, \ldots, 7\}$). The characteristic holds when $i \in \{0, 5, 7\}$. The input differences δ_i can be in any row. The row in which the output difference occurs will vary accordingly. Hence, in total $3 \times 4 = 12$ similar one-round characteristics were presented.

In the following we present additional one-round characteristics and expand the characteristics to one-round differentials.

3.1 One-Round Differentials

For simplicity of analysis we initially ignore the permutation layer, and present the analysis on a version of Q without this layer. Later, we will adapt the received results to the original Q, and discuss the influence of the permutation layer on the analysis. In our analysis, bits, bytes and words with non-zero difference will be called *active* bits, bytes or words, respectively.

The additional one-round characteristics that we present are similar to the characteristics presented by McBride. In Part I, McBride uses only characteristics where the active bits of the input difference and the active bits of the output difference are the same. We, in contrast, allow that the active bits of input and of output differences are not the same. Figure 4 describes the characteristic of Part I. In McBride's characteristics i always equals j, while in our characteristics this constraint is removed.

$$\begin{bmatrix} 0 & \delta_i & 0 & \delta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \xrightarrow[Part\ I]{2^{-14}} \begin{bmatrix} 0 & \delta_j & 0 & \delta_j \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 4. The characteristic of Part I

Table 1 shows the probabilities for all 1-bit input differences to cause 1-bit output differences in the 8-bit S-box S that is used in the Byte Substitution layer. All empty entries have probability 0. In the characteristics presented in [5], only the entries laying on the diagonal are used. Our analysis uses all the non-zero entries. The characteristic over Part I uses the same entry of Table 1 twice, so we convert this table to the Part I table by squaring the probability of every non-zero entry. The resulting matrix is denoted by M_I.

In Part II, McBride uses only the characteristic where an input difference that has two active bits in one nibble (bits 1 and 3) causes a difference with one active bit (bit 1) after the Bit-Slice A layer, and an output difference with two active bits in one nibble (again bits 1 and 3). In the following, we describe additional characteristics over Part II, where an input difference with two active bits in one nibble causes an output difference with two active bits in one nibble. The active bits in the output difference are not necessarily the same bits as in the input difference.

Table 1. 8-bit S-box S: The probability that an input difference with exactly one active bit (i) causes an output difference with exactly one active bit (j).

i	j							
	0	1	2	3	4	5	6	7
0	**2^{-7}**		2^{-7}	2^{-7}				2^{-7}
1			2^{-7}				2^{-7}	2^{-7}
2						2^{-7}		
3		2^{-7}			2^{-7}			
4	2^{-7}			2^{-7}		2^{-7}		
5		2^{-7}	2^{-7}			**2^{-7}**	2^{-7}	2^{-7}
6	2^{-7}				2^{-7}	2^{-7}		
7		2^{-7}		2^{-7}				**2^{-7}**

Table 2. 4-bit S-box A: The probabilities that input differences with two active bits cause output differences with exactly one active bit.

M_A :

active	active output bit			
input bits	0	1	2	3
0,1	2^{-3}			2^{-2}
0,2				2^{-2}
1,2	2^{-3}			
0,3		2^{-2}	2^{-3}	
1,3		**2^{-2}**	2^{-3}	
2,3	2^{-3}	2^{-2}	2^{-3}	

Table 2 describes the probabilities that input differences with two active bits cause output differences with exactly one active bit, for the 4-bit S-box A. The empty entries denote zero probability. After the Bit-Slice A layer the difference has one active bit, so the input difference of the Bit-Slice B layer also has one active bit.

Table 3. 4-bit S-box B: The probabilities that input differences with one active bit cause output differences with two active bits.

M_B :

active	active output bits					
input bit	0,1	0,2	1,2	0,3	1,3	2,3
0	2^{-3}	2^{-2}	2^{-3}			2^{-3}
1		2^{-3}		2^{-3}	**2^{-2}**	
2			2^{-2}		2^{-3}	2^{-3}
3	2^{-3}		2^{-3}	2^{-3}		2^{-3}

Table 3 describes the probabilities that input differences with one active bit cause output differences with two active bits, for the 4-bit S-box B. McBride uses only the entries that are in bold in both tables (Table 2 and Table 3).

Table 4. Part II: The probabilities that input differences with two active bits cause output differences with two active bits.

$M_{II} = M_A \cdot M_B$:

active	active output bits					
input bits	0,1	0,2	1,2	0,3	1,3	2,3
0,1	$3 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$3 \cdot 2^{-6}$	$2 \cdot 2^{-6}$		$3 \cdot 2^{-6}$
0,2	$2 \cdot 2^{-6}$		$2 \cdot 2^{-6}$	$2 \cdot 2^{-6}$		$2 \cdot 2^{-6}$
1,2	$1 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$1 \cdot 2^{-6}$			$1 \cdot 2^{-6}$
0,3		$2 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$5 \cdot 2^{-6}$	$1 \cdot 2^{-6}$
1,3		$2 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$5 \cdot 2^{-6}$	$1 \cdot 2^{-6}$
2,3	$1 \cdot 2^{-6}$	$4 \cdot 2^{-6}$	$3 \cdot 2^{-6}$	$2 \cdot 2^{-6}$	$5 \cdot 2^{-6}$	$2 \cdot 2^{-6}$

Table 4 is the product of the matrixes M_A and M_B, $M_{II} = M_A \cdot M_B$. It describes the probabilities that input differences with two active bits cause output differences with two active bits in Part II. The empty entries denote zero probability. The entry (1,3),(1,3), for example, has probability $5 \cdot 2^{-6}$, because in addition to the McBride's characteristic described earlier we have the characteristic that goes $(1,3) \rightarrow (2) \rightarrow (1,3)$ and has probability 2^{-6}.

Now we extend the one-round characteristics to one-round differentials. To build a one-round differential, we compute the tensor product of M_I and M_{II}: $M_R = M_I \cdot M_{II}$. The matrix M_R contains the probabilities of corresponding one-round differentials, where every such differential may start from any row (i.e., we take any non-zero entry from Table 1 and any non-zero entry from Table 4). We obtain $24 \cdot 29 \cdot 4 = 2784$ one-round differentials (24 non-zero entries in Table 1, 29 non-zero entries in Table 4 and 4 options for the starting row). The best one-round differentials have probability $2^{-14} \cdot 5 \cdot 2^{-6} = 5 \cdot 2^{-20}$. One of them is shown on Figure 5 where i does not necessarily equal j.

$$\begin{bmatrix} 0 & \delta_i & 0 & \delta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \underset{\text{Part I}}{\overset{2^{-14}}{\rightarrow}} \begin{bmatrix} 0 & \delta_j & 0 & \delta_j \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \underset{\text{Part II}}{\overset{5 \cdot 2^{-6}}{\rightarrow}} \begin{bmatrix} 0 & \delta_j & 0 & \delta_j \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 5. One-round differential with probability $5 \cdot 2^{-20}$

3.2 Differentials over Several Rounds of Q

In this subsection we extend the one-round differentials to more rounds. For this analysis we assume that the rounds act independently, such that the probability of a multiple-round characteristic can be approximated by the product of the probabilities of one-round characteristics.

Part I changes only the index of active bits in the two bytes (of course, it must change to the same bit in both bytes), but does not change which bytes are active. Part II changes only which bytes in the row are active, but does not change the active row, nor the index of the active bit inside the bytes. Hence, the differentials of Part I over several rounds and the differentials of Part II over several rounds may be built independently, by making sure that both are satisfied together.

Thus, in order to obtain the probability of differentials on i rounds, we compute $(M_R)^i = (M_I)^i \times (M_{II})^i$. For example, Table 5 shows $(M_I)^2$ and $(M_{II})^2$.

Table 5. The matrixes M_I and M_{II} for 2-round differentials

$$(M_I)^2 = (2^{-14})^2 \cdot \begin{pmatrix} 1&2&1&2&1&1&0&2 \\ 1&1&0&1&1&2&0&1 \\ 0&1&1&0&0&1&1&1 \\ 1&0&1&1&0&0&2&1 \\ 2&1&1&1&2&1&0&1 \\ 1&2&2&1&1&3&2&3 \\ 2&1&2&2&0&1&2&2 \\ 0&2&1&1&1&0&1&2 \end{pmatrix}$$

$$(M_{II})^2 = (2^{-6})^2 \cdot \begin{pmatrix} 19&28&29&20&25&24 \\ 10&20&18&12&20&14 \\ 9&8&11&8&5&10 \\ 7&22&23&20&40&15 \\ 7&22&23&20&40&15 \\ 16&30&34&28&45&25 \end{pmatrix}$$

Consulting Table 5, we see that the best probability of two-round differential is $3 \cdot (2^{-14})^2 \cdot 45 \cdot (2^{-6})^2 = 135 \cdot 2^{-40} \cong 2^{-33}$.

In this way, we can calculate the probabilities of the differentials over several rounds up to 9 rounds. We can also calculate the probabilities of the differentials over several rounds with an additional Part I in the end, or with an additional Part II in the beginning. For every such case there are $64 \cdot 36 \cdot 4 = 9216$ differentials (the computation is similar to the computation for the one-round case described earlier). Table 6 presents the best probabilities of the differentials for 6 rounds

Table 6. The best differential probabilities of Q (without permutation layer) for various numbers of rounds

Number of	Normal	With additional	With additional
rounds	case	Part I	Part II
6	$2^{-92.9}$	$2^{-105.35}$	$2^{-95.5}$
7	$2^{-107.9}$	$2^{-120.35}$	$2^{-110.5}$
8	$2^{-122.9}$	$2^{-135.35}$	$2^{-125.5}$
9	$2^{-137.9}$	$2^{-150.35}$	Irrelevant

and more. The best differentials are obtained when we start with δ_5 in bytes of the same row in words 2 and 3 (for example, in bytes '23' and '33'), and end with δ_7 in bytes of the same row in words 1 and 3. For the case that the matrix M_I is used j times and the matrix M_{II} is used k times, the appropriate entries are (5,7) in the $(M_I)^j$, and ((2,3),(1,3)) in $(M_{II})^k$. The probabilities of the other (not the best) differentials are close to the probability of the appropriate best differential presented in Table 6.

Now we discuss how the permutation layer influences the results. In the presented differentials, the permutation layer always works with one active byte only, and this layer only changes the order of the bytes. So, the permutation layer does not change the index of the active bit (i.e., has no influence on Part I). It also has no influence on Part II, because the permutation layer does not change the active word. Only the final row of the differential is influenced. In the variant of Q without a permutation layer, every differential has the active bits of the output difference in the same row as the active bits of the input difference (the difference cannot be in another row). In the original Q, however, different characteristics have the active bits of the output difference in different rows. These characteristics can be joined to 4 differentials (one for every possible active final row), where the sum of their probabilities equals the probability of the single differential that we obtain on the variant of Q without permutation.

Note that all the differentials described in this section may be taken in the backward direction. The probabilities of the differentials in the backward direction are equal to the presented probabilities for the forward direction.

4 A Basic Attack on Q

In this attack, we use a differential that spans the first 7 rounds, and part I and Bit-Slice A of the 8th round. In the first round, we can increase the probability by choosing input differences that produce the correct output difference with a higher probability than in the differentials presented in Section 3. Indeed, there exist differences ϵ such that the probability that the input difference ϵ may cause the output difference δ_i is higher than the probability that δ_j may cause δ_i. The optimal input difference is $\epsilon = \texttt{0xd8}$. In the 8th round, we do not specify the position of the active bits within the active bytes and consider all these possibilities as a generalized kind of differential, which may have different values for δ_i in the last round. The probability of the generalized differential is 2^{-121}. It has the following input difference:

$$\begin{bmatrix} 0 & 0 & \texttt{0xd8} & \texttt{0xd8} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

and after Bit-Slice A in the 8th round, it has one active bit, but we do not know the position of that bit. In order to simplify the attack, we specify the active row in the output. This reduces the probability of the differential by a factor of 1/4 (approximately).

Pairs that follow this differential have one active bit at the output of Bit-Slice A in the last round, hence one active row at the output of the 8th round, and hence one active row in the ciphertext. A pair that does not follow the characteristic has a similar output difference with probability 2^{-96}. Every pair that is not filtered suggests about 8 values for one word of $KW2 \oplus KB$. The signal-to-noise ratio of this attack is then:

$$S/N = \frac{2^{32} \cdot 2^{-123}}{8 \cdot 2^{-96}} = 4 \ .$$

The attack uses 2^{32} counters to count on 32 key bits. Since the S/N ratio is larger than 1, a counter with more than 4 hits is likely to belong to the correct key word. In order to get 4 right pairs, 2^{124} chosen plaintexts have to be encrypted. The plaintext requirements can be reduced by using structures of pairs. However, the attack discussed in the next section uses significantly fewer plaintexts.

5 Optimized Attacks on Q

Let G be the set of differences with exactly two active bytes, where both bytes are in the same row, and both have the same differential δ_i for any $i \in \{0, \ldots, 7\}$. In total, there are $\binom{4}{2} \cdot 4 \cdot 8 = 192$ differences in G (i.e., $\binom{4}{2}$ for choosing the affected bytes in the row, 4 for choosing the row and 8 for choosing i).

5.1 128-Bit Keys

In the case of 128-bit keys, the cipher consists of 8 rounds. We recover the key using about 2^{105} chosen plaintexts and 2^{77} complexity.

In this attack, we use the set of 7-round differentials described earlier, where the input difference is in the set of

$$\begin{bmatrix} 0 & 0 & \delta_i & \delta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

for all $i \in \{0, \ldots, 7\}$, as described in Section 3.2, and the output difference is in the set G. The sum of probabilities of the differentials in this set with any fixed input difference is approximately $2^{-104.35}$. So given about 2^{107} pairs with such an input difference we get an expected number of 6 pairs with output differences that belong to G. We use this set of 7-round differentials starting with Part II of the first round. The characteristic extends until the start of Part II in the last round.

We choose a structure of 2^{16} plaintexts by taking all the possibilities of bytes indexed ‘23’ and ‘33’, and all the other bytes are fixed to some randomly selected value. This structure includes 2^{31} pairs. The plaintexts give all the possible values of bytes '23' and '33' after Part I, and thus, they give $8 \cdot 2^{15} = 2^{18}$ pairs with the required difference (for the 7-round differential) after Part I. Hence, we need about $2^{107}/2^{18} = 2^{89}$ such structures in order to get an expected number of 6 pairs following the characteristic, i.e., we need about $2^{89} \cdot 2^{16} = 2^{105}$ chosen plaintexts. We call a pair with a difference that belongs to G after 7.5 rounds a *right pair*. We request to encrypt these 2^{105} chosen plaintexts to their ciphertexts under the unknown key.

How do we recognize the right pairs? Every right pair has difference δ_i (for some $i \in \{0, \ldots, 7\}$) in two bytes of the same row after 7.5 rounds, and zero differences in all other bytes. Different right pairs may have different i's. After an additional Part II, every right pair has either difference δ_i or 0 in every byte, but at least one row includes at least two bytes with difference δ_i. The received values become ciphertexts after an additional Part I. An input difference δ_i (for any fixed i) can result in approximately 128 possible output differences after S-box S. The set of possible output differences resulting from input difference δ_i is denoted by Z_i. Due to the fact that all the byte differences in the received values in the right pairs are either δ_i or 0, every byte of their ciphertext differences must either belong to Z_i or be 0 respectively. We compute Z_i for all $i \in \{0, \ldots, 7\}$ (this can be performed in a preprocessing stage). We check whether the output differences of all the bytes belong to the same Z_i. If they do, we continue the analysis. Otherwise this pair cannot be a right pair, and we discard it.

Wrong pairs satisfy the above condition for some specific Z_i with probability $(1/2)^{16} = 2^{-16}$ (1/2 is the probability that one byte difference belongs to this Z_i and 16 is the number of bytes). We select the pairs that satisfy the above condition for some Z_i. Thus about $2^{89} \cdot 2^{31} \cdot 2^{-16} \cdot 8 = 2^{107}$ pairs are selected.

Every selected pair suggests about 2^{16} possible values for $KW2 \oplus KB$. The 6 right pairs must suggest the right value of $KW2 \oplus KB$. So we look for values that are suggested 6 or more times. In total, about $2^{16} \cdot 2^{107} = 2^{123}$ values are suggested. We expect that approximately $\binom{2^{123}}{6} \cdot (2^{-128})^5 \cong 2^{89}$ values are suggested by 6 or more pairs.

The 6 right pairs must also suggest the right value of 16 bits of $KW1 \oplus KA$ (bytes '23' and '33'). Every selected pair suggests about $8^2 = 64$ possible such 16-bit values. So for every selected group of 6 pairs the probability to have at least one value suggested by all 6 pairs is $64^6 \cdot (2^{-16})^5 = 2^{-44}$. Thus, only $2^{89} \cdot 2^{-44} = 2^{45}$ groups of 6 pairs remain.

We observe that the subkey KB has only 2^{32} possible values (see Appendix A). For each guessed KB, we find $KW2$ by XOR-ing this KB with each of the 2^{45} combinations received from the 2^{45} groups. According to the key scheduling algorithm described in Appendix A, we can recover all the subkeys and the key given $KW2$ in an amount of time that is equivalent to the time taken to perform a single encryption. The wrong keys can be discarded by comparing the guessed KB with KB received from $KW2$. If more than one key remains after this check, then we discard all the remaining wrong keys by trial encryption.

The attack requires 2^{105} chosen plaintexts, and the complexity of analysis is about $2^{45} \cdot 2^{32} = 2^{77}$.

There is also a similar *chosen ciphertext attack* which requires the same number of chosen ciphertexts and has the same complexity.

5.2 256-Bit and 192-Bit Keys

In this case, the cipher consists of 9 rounds. We recover the key using about 2^{125} chosen ciphertexts and 2^{128} complexity.

The attack is similar to the attack from Section 5.1, except that this attack operates in the backward direction. According to Section 3.2 the best differentials in the backward direction start with active bytes in words 1 and 3. Thus, in this attack, we use the set of 8-round differentials in the backward direction, where the input difference is in the set of

$$\begin{bmatrix} 0 & \delta_i & 0 & \delta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

for all $i \in \{0, \ldots, 7\}$, and the output difference belongs to G. The sum of probabilities of the differentials in this set with any fixed input difference is approximately $2^{-119.35}$. So given about 2^{127} pairs with such an input difference we expect to have about 200 pairs ($200 \cong 2^{7.65}$) with output differences that belong to G. We use this set of 8-round differentials starting with Part II of the 9th round.

We build structures of 2^{16} ciphertexts each, as in the previous attack, but now we take all possibilities for bytes '13' and '33'. In this attack we need about $2^{127}/2^{18} = 2^{109}$ such structures to get an expected number of 200 right pairs after 8.5 rounds in the backward direction (or after the first round in the forward direction), i.e., we need about $2^{109} \cdot 2^{16} = 2^{125}$ chosen ciphertexts. Now we call a pair with difference that belongs to G after 8.5 rounds a *right pair.* We request to decrypt these 2^{125} chosen ciphertexts to their plaintexts under the unknown key.

The pair selection process is as described in the previous attack. Thus, about $2^{109} \cdot 2^{31} \cdot 2^{-16} \cdot 8 = 2^{127}$ pairs are selected.

The following calculations are similar to those described in the previous subsection. Every selected pair suggests about 2^{16} possible values for $KW1 \oplus KA$ and about 64 possible values of 16 bits (bytes '13' and '33') of $KW2 \oplus KB$. In total, about $2^{16} \cdot 64 \cdot 2^{127} = 2^{149}$ values are suggested. The 200 right pairs must suggest the right values of $KW1 \oplus KA$ and of 16 bits of $KW2 \oplus KB$. So we look for values that are suggested 200 or more times. We expect that approximately $\binom{2^{149}}{200} \cdot (2^{-144})^{199} \cong 2^{-101}$ wrong values are suggested by 200 or more pairs. We observe that the subkey KA has only 2^{32} possible values (see Appendix A). For each guessed KA, we find $KW1$ by XOR-ing this KA with the received combination. According to the key scheduling procedure described in Appendix A, we can recover all the subkeys and the key given any two consecutive 128-bit subkeys in time of a single encryption. $KW1$ and KAB are two consecutive subkeys. We receive $KW1$ and 32 bits from KAB (KA is word 0 of KAB taken four times), so it remains to guess additional 96 bits of KAB to recover all the subkeys and the original key. The wrong keys can be discarded by trial decryption.

The attack requires 2^{125} chosen ciphertexts, and the complexity of analysis is about $2^{32} \cdot 2^{96} = 2^{128}$.

5.3 Improved Attack for 192-Bit Keys

We can improve the complexity of the previous attack for 192-bit keys. This improvement uses the same structures and pair selection process as in the previous subsection, and changes only the key recovery process.

From the previous section, we expect to get about 1 value of $KW1 \oplus KA$ suggested by 200 or more pairs. We have 2^{32} possible combinations of subkeys ($KW1$,KA) as described in the previous subsection. According to the key scheduling algorithm, the knowledge of $KW1$ and KH is sufficient to recover all the other subkeys and the original key. Due to the fact that KH has only 64 non-zero bits for 192-bit keys, we can recover all the subkeys and the original key in $2^{32} \cdot 2^{64} = 2^{96}$ steps. Wrong keys can be discarded by comparing the guessed KA with KA received from $KW1$ and KH. If more than one key remains after this comparison then we discard all the remaining wrong keys by trial decryption.

The improved attack requires 2^{125} chosen ciphertexts, and the complexity of analysis is about 2^{96}.

6 Conclusions

In this paper we demonstrated that both the 8-round version and the 9-round version of Q are vulnerable to differential cryptanalysis. While these attacks are academic in nature, they show that the security analysis of the designer was insufficient. The most important improvements are the usage of differentials instead of characteristics, and the usage of characteristics that are not iterative, but 'almost iterative'.

The analysis shows that the combination of elements from two secure ciphers, in casu Rijndael and Serpent, does not necessarily result in a secure cipher. The round function of Q was composed by combining the nonlinear transformations of Rijndael and Serpent, and leaving out all the linear transformations except for the permutation. As a result, the diffusion properties of the round transformation of Q are suboptimal. Whereas the designers of Rijndael and Serpent ensured that there would exist no multi-round characteristics with a small number of active S-boxes in the original ciphers, several such characteristics can be defined for Q. Furthermore, the limited diffusion in the round transformation of Q allows combination of many characteristics into one differential.

References

[1] Ross Anderson, Eli Biham, Lars Knudsen, *Serpent: A proposal for Advanced Encryption Standard*, submitted to AES, 1998.

[2] Eli Biham, Adi Shamir, *Differential cryptanalysis of the Data Encryption Standard*, Springer Verlag, 1993.

[3] Joan Daemen, Vincent Rijmen, *The block cipher Rijndael*, Smart Card Research and Applications, LNCS 1820, J.-J. Quisquater and B. Schneier, Eds., Springer-Verlag, 2000, pp. 288-296.

[4] Xuejia Lai, James L. Massey,*Markov Ciphers and Differential Cryptanalysis*, proceedings of EUROCRYPT'91, LNCS 547, pp.17-38, 1991.

[5] Leslie 'Mack' McBride, *Q: A Proposal for NESSIE v2.00*, submitted to NESSIE, 2000.

[6] www.cryptonessie.org

A The Key Scheduling of Q

Generally, the cipher Q may work with a key of any length. Keys longer than 256 bits are reduced to 256 bits using the polynomial $X^{256} + X^{193} + X^{113} + X^6 + 1$ in GF(2^{256}). The other keys are expanded to the 256-bit keys by adding zeroes as the most significant bits. The 128 least significant bits of the received key material are called KL, and the 128 most significant bits are called KH. Thus for 128-bit keys $KH = 0$ and for 192-bit keys the 64 most significant bits of KH are zeroes.

Then the procedure described later runs $r + 4$ times, where r is a number of rounds in Q, and their 128-bit outputs are taken as subkeys in the following order:

$$Discard, KW1, KAB, K_0, K_1, \ldots, K_{r-1}, KW2.$$

The first output is discarded and the subkeys KA, KB are built from the words 0 and 1 of KAB respectively. In both, the corresponding 32-bit word is used four times with the permutation applied.

The procedure description:

1. A single byte counter is XOR-ed to the least significant byte of KL (a counter is started from 0, and is increased for every procedure running).
2. $KL = KL \oplus KH$.
3. Byte Substitution layer is performed on KL.
4. The constant $0x9e3779b9$ is XOR-ed to the word 0 of KL.
5. Bit-Slice layer (S-box C) is performed on KL.
6. Permutation layer is performed on KL.
7. Output KL.

Note that for 128-bit keys, the knowledge of any 128-bit subkey suffices to recover all other subkeys and the original key. For 192-bit keys and 256-bit keys the knowledge of any two consecutive 128-bit subkeys suffices to recover all other subkeys and the original key.