Improved Impossible Differentials on Twofish*

Eli Biham† Vladimir Furman‡

Abstract

Twofish is one of the AES finalists. The best up-to-date analysis of Twofish was presented by its designers, whose complexity on 6-round Twofish with 128-bit keys is $4.6 \cdot 2^{128}$ one-round computations. In this paper we present an improvement of the attack on 6-round Twofish, whose complexity is $1.84 \cdot 2^{128}$ one-round computations. For other key sizes, our results have the complexity $13 \cdot 2^{160}$ one-round computations for 192-bit keys and $24.2 \cdot 2^{192}$ one-round computations for 256-bit keys. For 7-round Twofish, the designers mentioned an attack, and estimated its complexity to be about 2^{256} simple steps for 256-bit keys, and for other key sizes have a complexities that exceed exhaustive search. We present an improvement of the attack on 7-round Twofish, whose complexity is $2^{226.48}$ one-round computations for 256-bit keys. We also show, for the first time, that this attack is faster than exhaustive search for 192-bit keys for which it breaks 7-round Twofish in $2 \cdot 2^{192}$ one-round computations, while exhaustive search takes $7 \cdot 2^{192}$.

Keywords: cryptanalysis, impossible differential, Twofish.

1 Introduction

Twofish[4] is a block cipher designed by Counterpane Systems Group as a candidate for the Advanced Encryption Standard selection process, and was accepted as one of the five finalists.

The best up-to-date attacks on Twofish breaking 6 rounds for all key sizes (128, 192 and 256) and 7 rounds for 256-bit keys only was presented by Ferguson in [2]. It uses a 5-round impossible differential (see [1, 3] for more details on attacks using impossible differentials). In this paper we present an improvement, based on same 5-round impossible differential and an additional 4-round impossible differential. Our improvement reduces the total complexity of the attack on 6-round Twofish for all key sizes and on 7-round Twofish for 256-bit keys. In addition, we show, for the first time, that the attack on 7-round Twofish for 192-bit keys is efficient as well.

Note that Ferguson made the full analysis only for 6-round Twofish with 128-bit keys and for other cases gave only their estimations about the complexity of the attacks. All results of Ferguson presented in this paper are computed by us, to allow fair comparison of their and our methods.

Our paper is organized as follows: Section 2 describes the attack on 6-round Twofish with 128-bit keys. Section 3 describes the attacks on 6-round Twofish with 192-bit and 256-bit keys. Finally, Section 4 describes the known plaintext attacks on 7-round Twofish with 192-bit and 256-bit keys.

*This work was supported by the European Union fund IST-1999-12324 - NESSIE.

†Computer Science Department, Technion - Israel Institute of Technology, Haifa 32000, Israel. biham@cs.technion.ac.il, http://www.cs.technion.ac.il/~biham/.

‡Computer Science Department, Technion - Israel Institute of Technology, Haifa 32000, Israel. vfurman@cs.technion.ac.il.

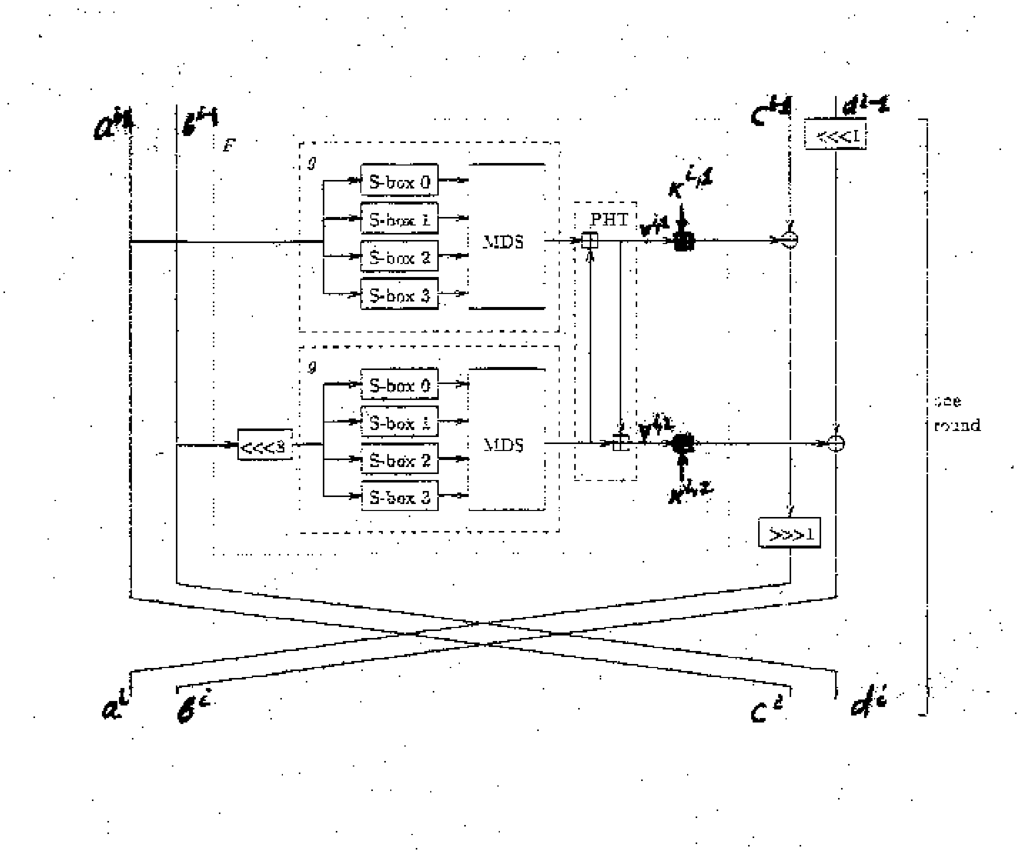

Figure 1: The round i

2 The 6-Round Attack on Twofish with 128-bit Keys

We denote the 128-bit word after round i by (a^i, b^i, c^i, d^i), the subkey of round i by $(K^{i,1}, K^{i,2})$ and the 64-bit PHT output word in round i by $(v^{i,1}, v^{i,2})$. Figure 1 outlines these notations for round i of Twofish. The full description of Twofish may be found in [4] and [2]. The following analysis is performed on an equivalent variant of Twofish without the one-bit rotations described in [2]. In [2], the authors show the equivalence between this variant of Twofish and full Twofish variant.

First, we describe the detailed analysis of Ferguson, as it is shown in [2], and then we describe our detailed analysis.

2.1 The Analysis of the Twofish Team

We know the 5-round impossible differential

$$(0, 0, \delta_1, \delta_2) \overset{5\ rounds}{\not\rightarrow} (0, 0, \delta_1, \delta_2),$$

where $(\delta_1, \delta_2) \neq (0, 0)$ [denoted in the equivalent variant].

Using this impossible differential a 6-round Twofish is attacked as follows: Look for plaintext pairs with a difference $(0, 0, \delta_1, \delta_2)$, where $(\delta_1, \delta_2) \neq (0, 0)$, that have a difference $(0, 0, \delta_1, \delta_2)$ after the 5th round when decrypting by some guessed subkey of the last round. If such a pair is found, the corresponding guessed subkey cannot be the correct subkey, because the differential is impossible. Therefore such a subkey is certainly wrong, and can be discarded from the list of the possible subkeys. If sufficiently many such pairs are analyzed, most subkeys can be discarded, and the remaining keys can be exhaustively searched.

A more detailed description of the attack is:

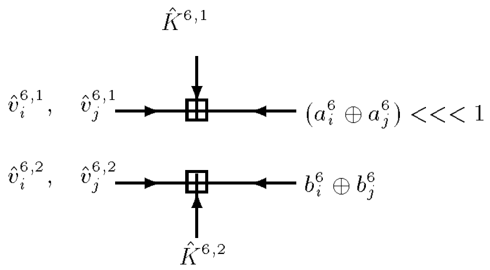

Figure 2: *The equation for finding last round subkeys, that are wrong due to 5-round impossible differential*

1. Fix a 64-bit value (a^0, b^0), and let (c^0, d^0) take all the 2^{64} possible values. Note that for this attack only a single such structure is needed.
2. Ask for the ciphertexts $(a_i^6, b_i^6, c_i^6, d_i^6)$ of the 2^{64} plaintexts (a^0, b^0, c_i^0, d_i^0), and insert them to a hash table indexed by $(c_i^0 \oplus c_i^6, d_i^0 \oplus d_i^6)$.
3. Any pair of encryptions without the same $(c_i^0 \oplus c_i^6, d_i^0 \oplus d_i^6)$ cannot make the differences before the last round be as required by 5-round impossible differential. These pairs are ignored, and only the pairs of encryptions with the same $(c_i^0 \oplus c_i^6, d_i^0 \oplus d_i^6)$ are analyzed. As there are 2^{127} pairs in total and 64-bit restrictions, the expectation is that about 2^{63} pairs remain.
4. The 64-bit S-box key S is guessed and the values $\hat{v}_i^{6,1}$, $\hat{v}_i^{6,2}$ (from c_i^6, d_i^6) are computed for each plaintext in the structure. Thus, for each remaining pair, the situation is as described on Figure 2, where the values $\hat{v}_i^{6,1}$, $\hat{v}_i^{6,2}$, $\hat{v}_j^{6,1}$, $\hat{v}_j^{6,2}$ and the differences $a^i \oplus a^j$, $b^i \oplus b^j$ are known. The subkeys $(\hat{K}^{6,1}, \hat{K}^{6,2})$ that satisfy the described requirement can be easily found. There is one such subkey $(\hat{K}^{6,1}, \hat{K}^{6,2})$ in average for each analyzed pair and each S-box key S guessing.
5. Each remaining pair eliminates a fraction of 2^{-64} of the possible keys for the last round. For each last round key exists one original key in average. Thus, a single structure of 2^{64} plaintexts reduces the number of the keys to $(1 - 2^{-64})^{2^{63}} = 1/\sqrt{e} = 0.6$ of the original number.
6. Test the remaining keys exhaustively.

The complexity of this attack is as follows: 2^{128} times (2^{64} possible S-box keys and 2^{64} plaintexts) one-round computations are spent, that reduces the number of the keys to 0.6 of the original number. So, there remain $0.6 \cdot 2^{128}$ undiscarded keys, whose exhaustive search takes $6 \cdot 0.6 \cdot 2^{128}$ one-round computations. In total, the complexity is $2^{128} + 3.6 \cdot 2^{128} = 4.6 \cdot 2^{128}$ one-round computations, while an exhaustive search takes $6 \cdot 2^{128}$ one-round computations. In average, these values are $2.8 \cdot 2^{128}$ and $3 \cdot 2^{128}$ respectively.

2.2 Our Analysis

We use the 5-round impossible difference:

$$(0, 0, \delta_1, \delta_2) \overset{5\ rounds}{\not\to} (0, 0, \delta_1, \delta_2),$$

like Ferguson does, together with the following additional 4-round impossible differential:

$$(0, 0, \delta_1, \delta_2) \overset{4\ rounds}{\not\to} (\delta_1, \delta_2, \delta_3, \delta_4),$$

for any δ_3, δ_4 such that $(\delta_3, \delta_4) \neq (0, 0)$[1].

[1] If $(\delta_3, \delta_4) = (0, 0)$ we get back to the 5-round impossible differential.

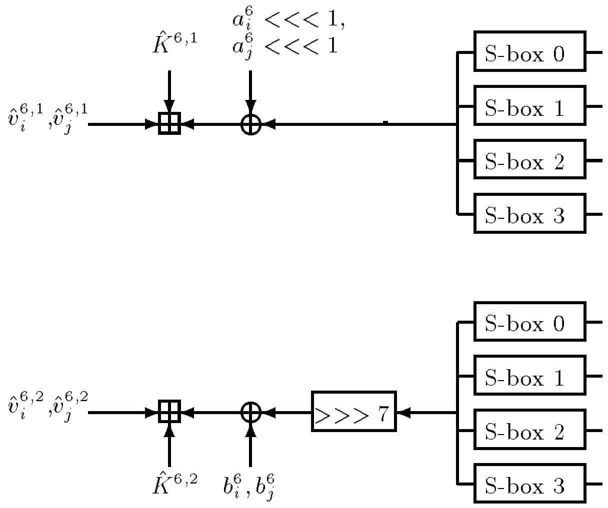

Figure 3: *The equation for finding last round subkeys, that are wrong due to 4-round impossible differential*

Similarly to the earlier analysis, for each pair of plaintexts $(a_i^0, b_i^0, c_i^0, d_i^0)$ and $(a_j^0, b_j^0, c_j^0, d_j^0)$ with a difference of $(0, 0, \delta_1, \delta_2)$, where $(\delta_1, \delta_2) \neq (0, 0)$, we ask for the ciphertexts $(a_i^6, b_i^6, c_i^6, d_i^6)$ and $(a_j^6, b_j^6, c_j^6, d_j^6)$ respectively.

We continue the analysis for each pairs for which one of the following conditions is satisfied:

- $c_i^0 \oplus c_i^6 = c_j^0 \oplus c_j^6$ and $d_i^0 \oplus d_i^6 = d_j^0 \oplus d_j^6$ (we can use the 5-round impossible differential). This is the case studied in the earlier analysis.
- $c_i^0 \oplus c_i^6 = c_j^0 \oplus c_j^6$ and $d_i^0 \oplus d_i^6 = d_j^0 \oplus d_j^6 \oplus 2^{31}$ (we can use the 4-round impossible differential).
- $c_i^0 \oplus c_i^6 = c_j^0 \oplus c_j^6 \oplus 2^{31}$ and $d_i^0 \oplus d_i^6 = d_j^0 \oplus d_j^6$ (we can use the 4-round impossible differential).
- $c_i^0 \oplus c_i^6 = c_j^0 \oplus c_j^6 \oplus 2^{31}$ and $d_i^0 \oplus d_i^6 = d_j^0 \oplus d_j^6 \oplus 2^{31}$ (we can use the 4-round impossible differential).

The last three cases are extend the earlier analysis using the 4-round impossible differential.

Thus, in the 5th round, the output difference of the F function is either: $(0, 0)$ (first condition), $(0, 2^{31})$ (second condition), $(2^{31}, 0)$ (third condition) or $(2^{31}, 2^{31})$ (fourth condition). All these differences are preserved, when they pass through addition with the 5th round subkey $(K^{5,1}, K^{5,2})$, so the output difference of the PHT in the 5th round is either: $(0, 0)$, $(0, 2^{31})$, $(2^{31}, 0)$ or $(2^{31}, 2^{31})$, respectively. The PHT affects these differences in a predetermined way, thus the output differences of the g functions in the 5th round (i.e., the input difference of the PHT) are $(0, 0)$, $(2^{31}, 2^{31})$, $(0, 2^{31})$ and $(2^{31}, 0)$, respectively. The MDS matrix is linear, so we can compute the output difference of each S-box in the 5th round (by multiplication of the MDS^{-1} with the output difference of the g function). So we know the output difference of each S-box in 5th round.

We then guess the S-box key S and compute the values $\hat{v}_i^{6,1}$, $\hat{v}_i^{6,2}$ (from c_i^6, d_i^6) for each plaintext in the structure. Thus, for each remaining pair, we get the situation, described on Figure 3, where we know $\hat{v}_i^{6,1}$, $\hat{v}_i^{6,2}$, $\hat{v}_j^{6,1}$, $\hat{v}_j^{6,2}$, a_i^6, b_i^6, a_j^6, b_j^6, the S-box key S and the output differences of all the S-boxes. The only information that we do not know in Figure 3 is the 6th round subkey $(\hat{K}^{6,1}, \hat{K}^{6,2})$ that would satisfy these restrictions. Appendix A shows how we can easily find it by one operation on average.

A more detailed description of the attack is:

1. Fix a 64-bit value (a^0, b^0), and let (c^0, d^0) take all the 2^{64} possible values.

2. Ask for the ciphertexts $(a_i^6, b_i^6, c_i^6, d_i^6)$ of the 2^{64} plaintexts $(a_i^0, b_i^0, c_i^0, d_i^0)$, and insert them to a hash table indexed by $((c_i^0 \oplus c_i^6) \bmod 2^{31}, (d_i^0 \oplus d_i^6) \bmod 2^{31})$.

3. We have in total 2^{127} pairs and 62-bit restrictions, so we expect to get about 2^{65} right pairs (satisfying the condition described earlier in this section).

4. We guess the 64-bit S-box key S and calculate the supposed $(\hat{v}^{6,1}, \hat{v}^{6,2})$ for each plaintext from the structure.

5. Each analyzed pair eliminates a fraction of 2^{-64} of the possible keys for the last round. For each last round key exists one original key in average. Thus, a single structure of 2^{64} plaintexts reduces the number of the keys to $(1-2^{-64})^{2^{65}} = 1/e^2 = 0.14$ of the original number.

6. Test the remaining keys exhaustively.

The total complexity is 2^{128} (2^{64} plaintexts $\times$ 2^{64} S-box keys) $+6 \cdot 0.14 \cdot 2^{128}$ (an exhaustive search of $0.14 \cdot 2^{128}$ keys through 6 rounds) $= 1.84 \cdot 2^{128}$ one-round computations and 2^{42} preprocessing computations.

Actually, we can use a smaller structure of 2^{63} plaintexts. In this case, we get about 2^{63} pairs, satisfying the condition described in the beginning of this section. Thus, a single structure of 2^{63} plaintexts reduces the number of undiscarded keys to $(1-2^{-64})^{2^{63}} = 1/\sqrt{e} = 0.6$ of the original number. Hence, the total complexity is $2^{127} + 6 \cdot 0.6 \cdot 2^{128} = 4.1 \cdot 2^{128}$ one-round computations and 2^{42} preprocessing computations. This is still a smaller complexity than Ferguson describes, despite the fact that we use a smaller number of plaintexts.

We observe that we can enhance the attack using several structures of 2^{64} plaintexts. This enhancement can be applied both to our variant of attack and to the variant of Ferguson. Denote the number of structures used in attack by p. Then, the total complexity of the attack is

- Based on our method – $p \cdot 2^{128} + 6 \cdot e^{-2 \cdot p} \cdot 2^{128}$.
- Based on Ferguson method – $p \cdot 2^{128} + 6 \cdot e^{-p/2} \cdot 2^{128}$.

Figure 4 shows relation between the total complexity and the number of the structures for Ferguson method and for our method (the 1/2 structure point refers to the small half-size structure).

3 The 6-Round Attacks on Twofish with 192-bit and 256-bit Keys

The attack on 6-round Twofish, described earlier, works in similar way and similar reduction in the complexity when one structure is used and the key size is either 192 bits or 256 bits. However, for these cases, the same attack used in conjunction with several additional structures can further reduce the total complexity of the attack.

3.1 192-bit Keys

The total complexity of the attack on 192-bit keys variant using p structures is

- Based on our method – $p \cdot 2^{160} + 6 \cdot e^{-2 \cdot p} \cdot 2^{192}$.
- Based on Ferguson method – $p \cdot 2^{160} + 6 \cdot e^{-p/2} \cdot 2^{192}$.

Figure 5 shows the relation between the number of structures used in the attack and the total complexity of the attack for the attack presented by Ferguson and for our improvement. From Figure 5, we can see that our best result is received given 12 structures ($12 \cdot 2^{64}$ chosen plaintexts), and the complexity in this case is about $2^{163.7}$ one-round computations. Figure 5 also shows that

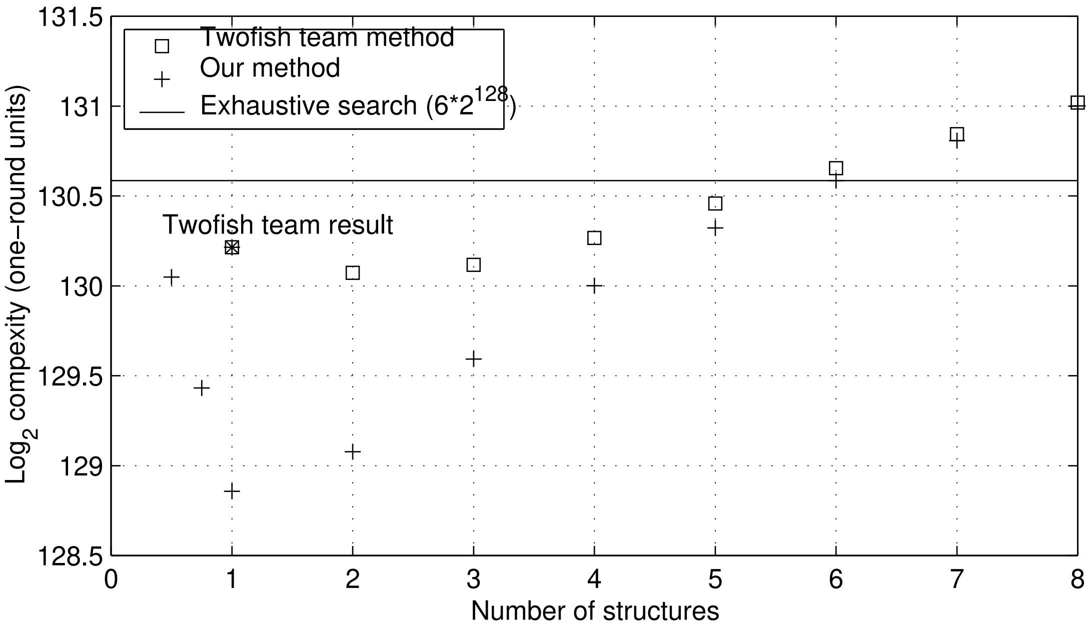

Figure 4: *The total complexity of attack on 6-round Twofish with 128-bit keys*

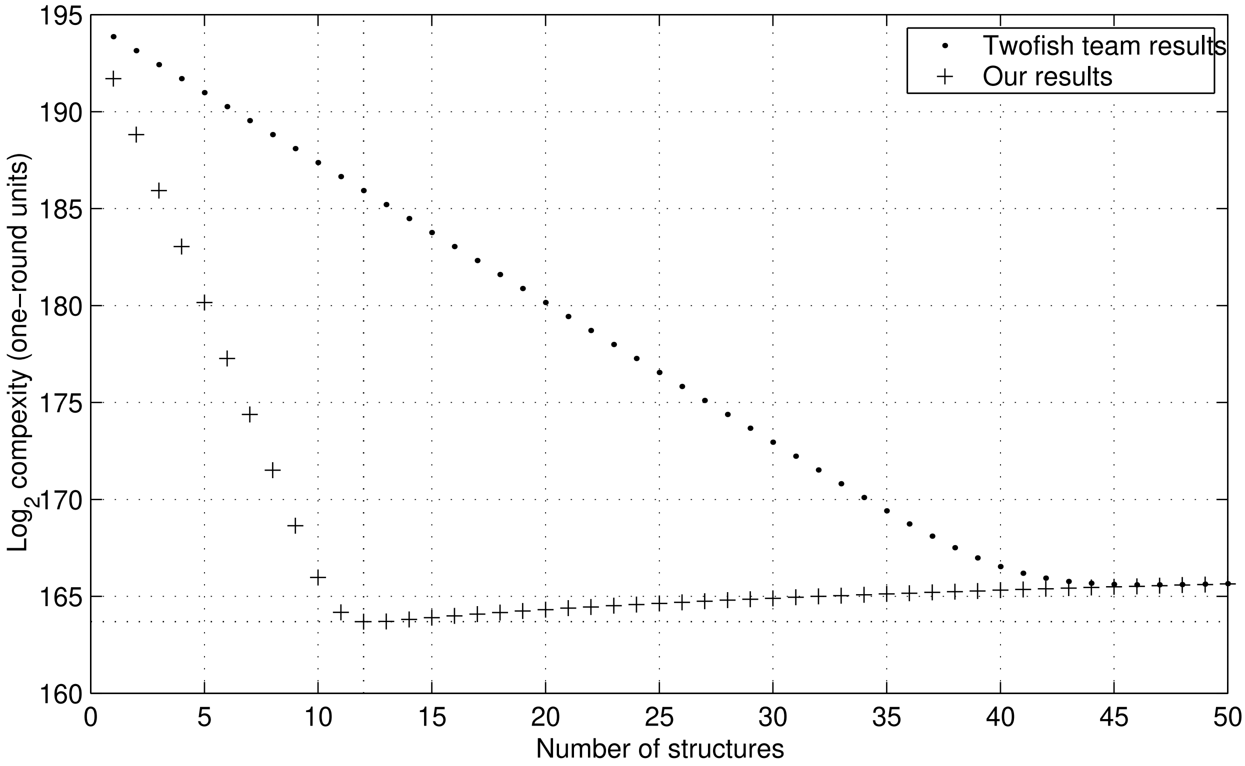

Figure 5: *The total complexity of attack on 6-round Twofish with 192-bit keys*

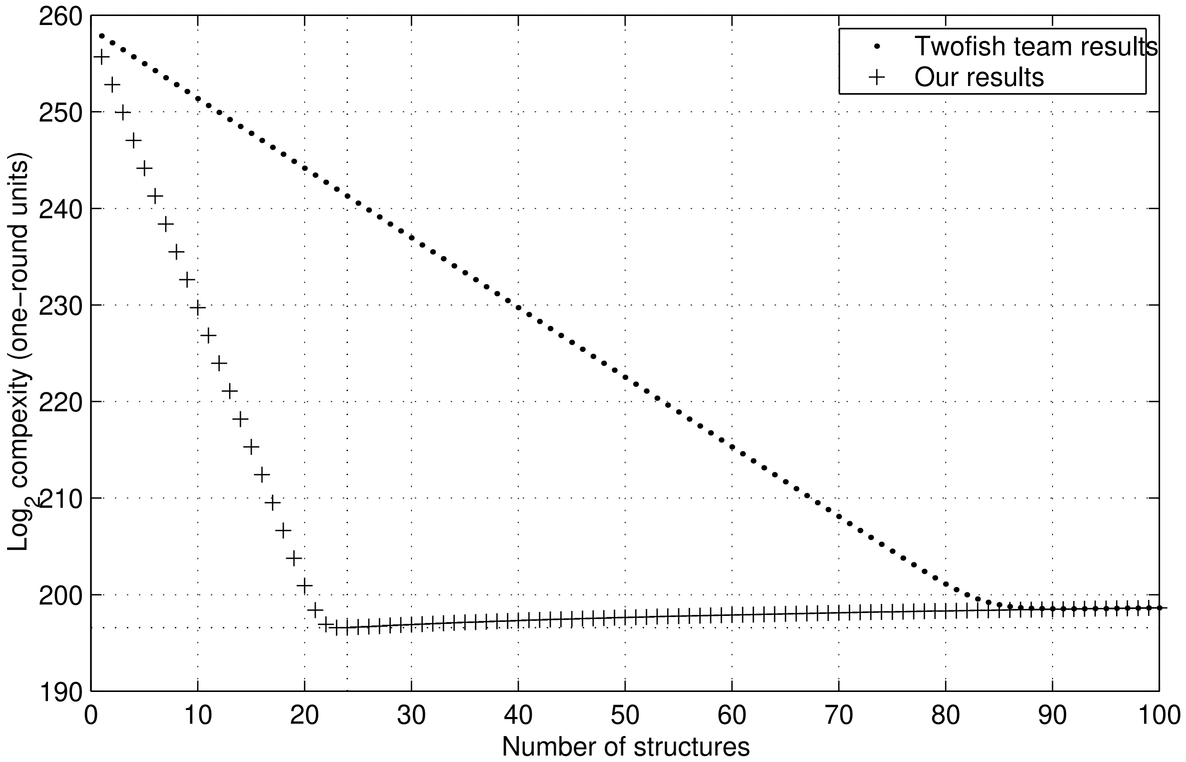

Figure 6: *The total complexity of attack on 6-round Twofish with 256-bit keys*

Key size in bits	Our results		Ferguson's results	
	complexity	chosen plaintexts	complexity	chosen plaintexts
128	$1.84 \cdot 2^{128}$	2^{64}	$4.6 \cdot 2^{128}$	2^{64}
192	$13 \cdot 2^{160}$	$12 \cdot 2^{64}$	$48.6 \cdot 2^{160}$	$47 \cdot 2^{64}$
256	$24.2 \cdot 2^{192}$	$24 \cdot 2^{64}$	$93 \cdot 2^{192}$	$91 \cdot 2^{64}$

Table 1: Comparing table between our and Ferguson best results

the complexity of our analysis increased when more than 12 structures are used (and similarly for more than 48 structures with Ferguson analysis). This phenomena is due to the larger number of chosen plaintexts, which cause the time required to discard the wrong keys to be larger than the time required to exhaustively search the remaining keys.

3.2 256-bit Keys

The total complexity of the attack on 256-bit keys variant using p structures is

- Based on our method – $p \cdot 2^{192} + 6 \cdot e^{-2 \cdot p} \cdot 2^{256}$.
- Based on Ferguson method – $p \cdot 2^{192} + 6 \cdot e^{-p/2} \cdot 2^{256}$.

Figure 6 is similar to Figure 5 for the case of 256-bit keys. Our best result in this case is about $2^{196.6}$ one-round computations using 24 structures.

Table 1 compares our best results to Ferguson best results (computed by us, as noted in the introduction) for each possible key size.

Note, that these attack are preserved when the 6-round variant of Twofish includes pre-whitening.

4 The 7-Round Known Plaintext Attacks on Twofish with 192-bit and 256-bit Keys

The 7-round attack is based on the same impossible differentials as the 6-round attack does. In this attack we use these impossible differentials in the middle rounds (starting from second round), in contrast to the 6-round attack, in which the impossible differentials start in the first round.

It is convenient to consider this attack as a variant of the 6-round attack, in which an additional round is prepended before the 6-round cipher. We want to get the difference $(0, 0, \delta_1, \delta_2)$ before the second round for some $(\delta_1, \delta_2) \neq (0, 0)$. Therefore, the difference before the first round must be $(\delta_1, \delta_2, \epsilon_1, \epsilon_2)$, where $(\epsilon_1, \epsilon_2) \neq (0, 0)$. In addition, the left half of the difference after the seventh round must be one of the following values: (δ_1, δ_2), $(\delta_1 \oplus 2^{31}, \delta_2)$, $(\delta_1, \delta_2 \oplus 2^{31})$ or $(\delta_1 \oplus 2^{31}, \delta_2 \oplus 2^{31})$ (otherwise we cannot use the impossible differentials). Hence, a pair of plaintexts may be used for this attack with probability 2^{-62} (the left part of the input difference must be equal to the left part of the output difference, except for their most significant bits). We call such a pair *a matching pair*.

For every matching pair, we guess the S-box key S and find the subkeys of the first and the last rounds, that leads to the impossible differential in the middle rounds. Such subkeys are wrong, and thus the combination of the initial keys that lead to these subkeys together with the guessed key S is wrong as well. We discard these keys from the list of the possible keys. Each matching pair eliminates about 2^{-128} fraction of the possible keys. Therefore, we need about 2^{128} matching pairs, i.e., about 2^{190} $(2^{128} \cdot 2^{62})$ pairs in total, which can be generated using $2^{95.5}$ known plaintexts.

In [2] Ferguson maintains that such an attack (using a 5-round impossible differential, of course) may be used only when the key size is 256 bits and it takes about 2^{256} steps. In Subsection 4.1 we show that such an attack is more efficient than exhaustive search even when the key size is 192 bits, and even if we use only a 5-round impossible differential, as Ferguson does. In Subsection 4.2 we show that the complexity of the attack on 256-bit keys variant is significantly less than 2^{256}.

4.1 192-bit Keys

As in the previous attack, the analysis consist of two stages: discarding most keys and exhaustively searching the remaining ones. We denote the number of the known plaintexts required for the attack by 2^p. Clearly, p must be close to 95.5 (according to above explanations). So, the first stage of the analysis takes 2^p (known plaintexts) $\cdot 2^{96}$ (guessing of the S-box key S) $\cdot 2$ (for the first and the last round) $= 2^{p+96+1} = 2^{p+97}$ one-round computations. It discards most keys as wrong, except for a fraction of $(1 - 2^{-128})^{2^{2 \cdot p - 1 - 62}} = e^{-2^{2 \cdot p - 63 - 128}} = e^{-2^{2 \cdot p - 191}}$ of the keys (where $2^{2 \cdot p - 1}$ pairs are received from 2^p known plaintexts and the probability of a pair to be a matching pair is 2^{-62}). Hence, the second stage of the attack takes $7 \cdot e^{-2^{2 \cdot p - 191}} \cdot 2^{192}$ one-round computations (where 7 is the number of rounds in discussed Twofish variant, $e^{-2^{2 \cdot p - 191}} \cdot 2^{192}$ is a number of the remaining keys, as described earlier). The total complexity of the attack is $2^{p+97} + 7 \cdot e^{-2^{2 \cdot p - 191}} \cdot 2^{192}$ one-round computations.

We can improve this result by observing the symmetry of the cipher: We set the 4-round impossible differential in the decryption direction from round six backwards and use a similar analysis replacing the first and last rounds. This improvement allows us to discard twice as many keys using the 4-round impossible differential (used in 3/4 of the cases), or a total of 7/4 of the number of keys discarded by each pair in the first stage of the attack. So it reduces the fraction of remaining keys to $(1 - 7/4 \cdot 2^{-128})^{2^{2 \cdot p - 1 - 62}} = e^{-7/4 \cdot 2^{2 \cdot p - 1 - 62 - 128}} = e^{-7 \cot 2^{2 \cdot p - 193}}$. Hence, the total complexity of the attack is $2^{p+97} + 7 \cdot e^{-7 \cdot 2^{2 \cdot p - 193}} \cdot 2^{192}$ one-round computations.

If the attack uses only the 5-round impossible differential (as in Ferguson method), then the probability for some pair to be a matching pair is 2^{-64}, and the total complexity of the attack is $2^{p+97} + 7 \cdot e^{-2^{2 \cdot p - 193}} \cdot 2^{192}$ one-round computations.

Figure 7 shows the total complexity of the three variants of the attack:

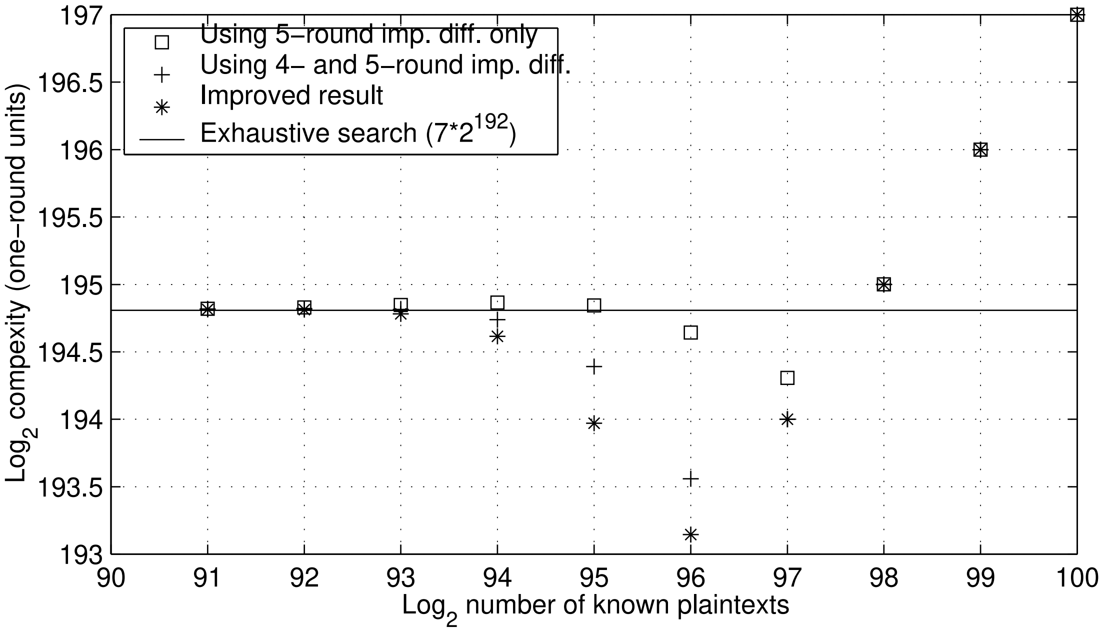

Figure 7: *The total attack complexity for 7-round Twofish with 192-bit keys*

1. Using only the 5-round impossible differential – the best result requires $4.95 \cdot 2^{192}$ one-round computations using $2^{96.95}$ chosen plaintexts.

2. Using the 5-round and the 4-round impossible differentials – the best result requires $2.79 \cdot 2^{192}$ one-round computations using $2^{96.25}$ chosen plaintexts.

3. Improved result using the 5-round and the 4-round impossible differentials – the best result requires $2 \cdot 2^{192}$ one-round computations using $2^{95.9}$ chosen plaintexts.

Note that exhaustive search takes $7 \cdot 2^{192} = 2^{194.81}$ one-round computations.

4.2 256-bit Keys

In this case, the analysis is similar to the previous one. The first stage of the analysis takes $2^p \cdot 2^{128} \cdot 2 = 2^{p+128+1} = 2^{p+129}$ one-round computations. This reduces the fraction of undiscarded keys to $(1-2^{-128})^{2^{2 \cdot p-1-62}} = e^{-2^{2 \cdot p-63-128}} = e^{-2^{2 \cdot p-191}}$. Hence, the second stage of the attack takes $7 \cdot e^{-2^{2 \cdot p-191}} \cdot 2^{256}$ one-round computations. The total complexity of the attack is $2^{p+129} + 7 \cdot e^{-2^{2 \cdot p-191}} \cdot 2^{256}$ one-round computations. In the improved analysis, the total complexity of the attack is $2^{p+129} + 7 \cdot e^{-7 \cdot 2^{2 \cdot p-193}} \cdot 2^{256}$ one-round computations. In the attack that uses only the 5-round impossible differential the total complexity is $2^{p+129} + 7 \cdot e^{-2^{2 \cdot p-193}} \cdot 2^{256}$ one-round computations. Note that exhaustive search takes $7 \cdot 2^{256} = 2^{258.81}$ one-round computations.

Figure 8 shows the total complexity of all these cases, and its best result are shown in the Table 2.

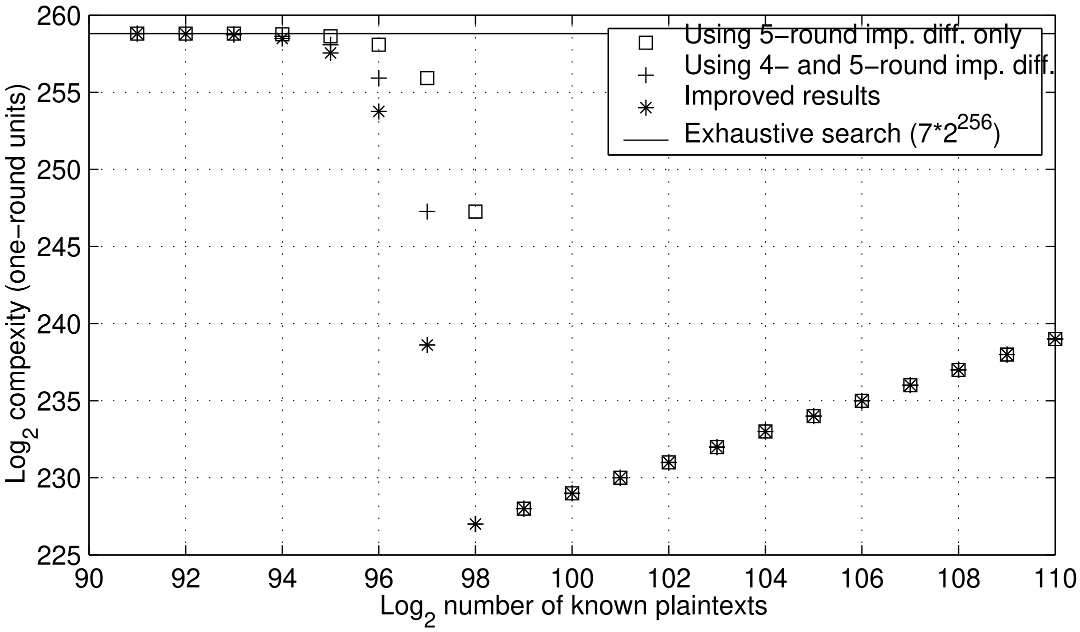

Figure 8: *The total attack complexity for 7-round Twofish with 256-bit keys*

	one-round computations	known plaintexts
5-round imp. diff. only	2^{228}	2^{99}
5-round and 4-round imp. diff.	$2^{226.88}$	$2^{97.85}$
Improved using the 5-round and 4-round imp. diff.	$2^{226.48}$	$2^{97.46}$

Table 2: *The best results for the attack on 7-round Twofish with 256-bit keys*

References

[1] E. Biham, A. Biryukov, A. Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials*, LNCS 1592, Advanced in Cryptology - Proceeding of EUROCRYPT'99, pp. 12-23, Springer-Verlag 1999.

[2] N. Ferguson, *Impossible Differentials in Twofish*, Twofish Technical Report 5, Counterpane Systems, October 1999.

[3] L. Knudsen, *DEAL - A 128-bit Block Cipher*, NIST AES Proposal, June 1998.

[4] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson, *Twofish: A 128-bit Block Cipher*, NIST AES Proposal, June 1998.

A The Subkey Computation in a Unit Time

From Figure 3 we can see that the $\hat{K}^{6,1}$ and $\hat{K}^{6,2}$ have independent computations. Thus, we can compute $\hat{K}^{6,1}$ and in parallel $\hat{K}^{6,2}$. Without loss of generality, we show how to find the supposed values of $\hat{K}^{6,1}$. $\hat{K}^{6,2}$ can be found similarly.

As we discussed earlier the 32-bit MDS output difference may be either 0 or 2^{31}. In the first case (a difference 0), the 32-bit MDS input difference is 0 as well. It means that the 8-bit output differences of all the S-boxes are zeroes, and thus the 8-bit input differences of all the S-boxes are zeroes as well. So we get back to the situation described in Figure 2 for one of the words (rather than for both words as shown there). This information suffices to find all the matching subkeys directly.

In the other case the 32-bit MDS output difference is 2^{31}. The 32-bit MDS input difference is some fixed value that may be calculated by multiplying MDS^{-1} with 2^{31}. We denote the received 8-bit output differences of the S-boxes by Δ_{S0}, Δ_{S1}, Δ_{S2} and Δ_{S3}. During the attack, when we are going to find $\hat{K}^{6,1}$, we know $\hat{v}_i^{6,1}$, $\hat{v}_j^{6,1}$, a_i^6, a_j^6, Δ_{S0}, Δ_{S1}, Δ_{S2}, Δ_{S3} and S-box key S, but it still takes 2^{64} guesses to find the subkeys matching the condition. A more efficient method is used a precomputed table, that reduces the time need during the attack.

We are looking on the part of condition laying in right side of the $\hat{K}^{6,1}$ (XOR with a^6 and S-boxes), as shown on Figure 3. Before the attack we don't know the a_i^6, a_j^6 and S-box key S, but we know the Δ_{S0}, Δ_{S1}, Δ_{S2} and Δ_{S3}. We want to find all possible pairs of 32-bit values that may lead to the difference (Δ_{S0}, Δ_{S1}, Δ_{S2}, Δ_{S3}) for some a_i^6, a_j^6 and S-box key S. For each 32-bit difference there exist 2^{32} pairs satisfying the difference. We guess a S-box key S and values a_i^6, a_j^6, and build a table that for any such guessing give all possible input pairs, that together with guessed values lead to required output difference (Δ_{S0}, Δ_{S1}, Δ_{S2}, Δ_{S3}). If we do it in such way, it takes 2^{32} (possible output pairs) $\times$ 2^{64} (possible S-box key S guessing) $\times$ 2^{64} (possible a_i^6, a_j^6 values) $= 2^{160}$ computations, that is too much for 128-bit keys case. But the S-boxes are independent and the XOR operation of 32-bit values may be represented as four independent XOR operations of 8-bit values, so we can make the above computations in four independent computations, where each of them is going in described earlier way but work only on 8-bit values. Thus we build four tables, and each of them takes 2^8 (possible output pairs) $\times$ 2^{16} (possible S-box key S guessing) $\times$ 2^{16} (possible a_i^6, a_j^6 values) $= 2^{40}$ computations, that takes in total $4 \cdot 2^{40} = 2^{42}$ computations. In additional, we hash the received input pairs by their additive difference. This reduces the time, taken during the attack to find the $\hat{K}^{6,1}$, because when we know the $\hat{v}_i^{6,1}$ and $\hat{v}_j^{6,1}$, we may to calculate their additive difference and it preserves when it passes through addition operation.

During the attack we know values: a_i^6, a_j^6, S-box key S and additive difference of the $\hat{v}_i^{6,1}$ and $\hat{v}_j^{6,1}$, so we get the possible pairs from the four tables, and starting from the least significant 8 bits to the most significant 8 bits we are getting $\hat{K}^{6,1}$. There are one such subkey in average, so it takes one operation in average.

Note, that the fact that in additive difference may have a carry does not disturb us. If there is an input carry to some 8-bit additive difference, we just consult the table with the original additive difference minus one.