Twofish Technical Report #4

Further Observations on the Key Schedule of Twofish

Doug Whiting* John Kelsey† Bruce Schneier‡ David Wagner§
Niels Ferguson ¶ Chris Hall‖

March 16, 1999

Abstract

Twofish is a 128-bit block cipher submitted as an AES candidate [SKW+98]. Mirza and Murphy [MM99] recently noted two interesting properties in the Twofish key schedule for 128-bit keys: there is a non-uniform distribution of 128-bit whitening keys, and the 64-bit round subkeys are non-uniformly distributed over each subset of keys that fixes the S-boxes. This paper extends these results and explains why they do not affect the security of Twofish. First, it is shown that pairs of 64-bit subkeys in Twofish, including the whitening keys, actually have less than 117 bits of entropy, considerably less than predicted by [MM99], but that this fact is not at odds with the goal of the whitening keys. Second, it is shown that other block ciphers, notably DES and Triple DES, achieve far less uniform subkey distributions than Twofish over simiarly constructed subsets of keys, but this fact has never led to a known attack on these ciphers.

1 Background

The Twofish algorithm was submitted as an AES candidate in 1998 [SKW+98]. It is a 128-bit block cipher, using sixteen Feistel rounds and key-dependent S-boxes. A total of forty 32-bit expanded key words (K_i) are computed from the key material. Four of these expanded keys words (K_0, K_1, K_2, K_3) are used as input whitening before the initial round, and four words (K_4, K_5, K_6, K_7) are used for output whitening after the final round. In each round $r = 0..15$, two expanded key words (K_{2r+8}, K_{2r+9}) are merged into the 64-bit value computed from the left half via the S-boxes, and the resulting 64-bit result is XORed into the right half of the block.

Mirza and Murphy cleverly noted some non-uniform distributions in the Twofish subkeys [MM99].[1] Unfortunately, the original Twofish paper and the Mirza/Murphy paper use somewhat different mathematical notation; we will attempt to be consistent with the notation of [SKW+98] in this paper. Since the details of the Twofish algorithm and subkeys have been completely described in both places, we will not repeat the algorithm specification here.

Some terminology is useful for the discussion below. See Section 4.3.4 of [SKW+98] for the mathematical details behind these terms. An expanded key word is one of the forty 32-bit values K_i, for $i = 0..39$. A subkey V_j for $j = 0..19$ is comprised of

*Hi/fn, Inc., 5973 Avenida Encinas Suite 110, Carlsbad, CA 92008, USA; dwhiting@hifn.com.
†Counterpane Systems; 101 E Minnehaha Parkway, Minneapolis, MN 55419, USA; kelsey@counterpane.com.
‡Counterpane Systems; schneier@counterpane.com.
§University of California Berkeley, Soda Hall, Berkeley, CA 94720, USA; daw@cs.berkeley.edu.
¶Counterpane Systems; niels@counterpane.com.
‖Counterpane Systems; hall@counterpane.com.

[1] At this point, they are the leading contenders for the $10,000 (US) prize offered by the Twofish design team for the best analysis on the algorithm.

64 bits from consecutive expanded key words:

$$V_j = (K_{2j}, K_{2j+1}).$$

Note that each subkey V_j is computed from a simple combination of the two 32-bit terms A_j and B_j, where the A_j values depend on half of the key bits and the B_j depend on the other half of the key bits. A 128-bit whitening key is a pair of subkeys. The input whitening key consists of V_0 and V_1, while the output whitening key consists of V_2 and V_3. The subkeys V_{r+4} for $r = 0..15$ are used as the 64-bit round keys.

All entropy values in this paper use the Shannon definition of entropy:

$$H = \sum_i p_i \log_2 p_i.$$

2 Entropy of Subkey Pairs

Mirza and Murphy first note that not all values of the 128-bit whitening keys are possible. They give a very brief argument that the actual distribution of whitening keys only achieves about $1 - e^{-1} = 0.632$ of all possible values. Their general observation is correct, but the actual distribution is considerably less uniform than their estimate.

Before exploring the actual distribution of the whitening keys, some comments on this phenomemon are useful. First, this observed non-uniformity applies to all pairs of subkeys, not just the whitening keys, so our discussion here will be apply to all subkey pairs. Also, it is immediately obvious from the definitions of the subkeys that no two distinct subkeys can be equal, so the pairwise distribution clearly cannot be uniform. In particular, V_i cannot be equal to V_j for $i \neq j$, because that would imply that $A_i = A_j$ and $B_i = B_j$ (since the function to compute V_i from A_i and B_i is reversible). In turn, this would imply that $\text{MDS}^{-1}(A_i) = \text{MDS}^{-1}(A_j)$ which cannot occur because these values are derived from the same bijective S-box with different input values.

In fact, there is a trivial stronger result. Each of the eight input bytes to the MDS matrixes used to compute V_i must be different from the corresponding byte in the computation of V_j. Again, this is because they are constructed from the same bijective S-box using different input values. As a result, the total number of unique subkey pairs cannot exceed $(256 \times 255)^8$, or about 127.95 bits of entropy. Thus, although it was never explicitly stated in the original Twofish paper [SKW+98], it was always understood that the distribution of subkey pairs was non-uniform. The magnitude of the actual non-unformity was certainly not fully anticipated, but there was never an "implicit claim" to the contrary, as Mirza and Murphy assert.

2.1 Results

The actual amount of non-uniformity in subkey pairs for 128-bit keys can easily be computed empirically. The key insight is that this is not truly a problem of two 64-bit quantities, but instead a collection of eight independent problems, each involving two 8-bit quantities. Instead of looking at A_i and B_i, we deal with the two sets of four bytes which produce them; i.e., the bytes that feed into the MDS matrix to produce A_i and B_i. If we examine the entropy achieved by associated byte pairs over all 128-bit keys for a given MDS input, the total entropy of subkey pairs is merely the sum of the entropies of the individual byte positions, since the MDS operation is bijective. The 8-bit problem can easily be analysed empirically, but much of the results can be understood as follows. Take a given byte path in a subkey computation, such as

$$a_i = q_1[q_0[q_0[i] \oplus k_0] \oplus k_1]]$$

where k_0 and k_1 are two key bytes. Consider the two values a_i and a_j for different subkeys; i.e., $i \neq j$. If we consider the pair (a_i, a_j), how many possible values does this byte pair achieve over all 2^{16} possible key byte pairs (k_0, k_1)? For a uniform distribution, all 2^{16} values should occur. Although it is obvious that a_i (and a_j) will individually take on all possible 256 values, the total number of unique values of the pair is certainly no greater than 256×255, since $a_i \neq a_j$, as noted previously.

Now, define

$$c_i = q_0[q_0[i] \oplus k_0] \oplus k_1$$

so that $a_i = q_1[c_i]$. Since q_1 is a permutation, the distribution of (c_i, c_j) over all keys is merely a permutation of the distribution of (a_i, a_j), and therefore has the same entropy. It is now simple to note that the pair (c_i, c_j) can occur if and only if the XOR difference

$$d_{ij} = c_i \oplus c_j = q_0[q_0[i] \oplus k_0] \oplus q_0[q_0[j] \oplus k_0]$$

can occur for some k_0. Observe that k_1 does not enter here, and the equation above is very similar to that used in the computation of $\text{DP}_{\max}$ [Mat96]. If we define $\Delta_{ij} := q_0[i] \oplus q_0[j]$, then we note that, if a given value of k_0 is a solution for the given value

of d_{ij}, then so is the key value $k_0 \oplus \Delta_{ij}$. Thus, for every allowed pair (c_i, c_j), there are at least two distinct key byte pairs that map to that value. In other words, no more than half of the possible 2^{16} values are possible! This result is similar to the fact that $2^n \times \mathrm{DP}_{\max}$ is always an even number for an n-bit function. So, instead of losing a fraction of a bit of entropy in subkey pairs, we actually lose at least eight bits of entropy, one bit for each of the eight byte paths.

There is also some further loss due to the additional "collisions." Empirically, the total loss of entropy is almost exactly 11 bits, so there are only about 2^{117} distinct values of subkey pairs (V_i, V_j) for any particular (i, j). This is certainly an unexpectedly large deviation from a uniform distribution. Further testing along these lines found that subkey triples (V_i, V_j, V_k) for 128-bit keys achieve the full possible 128 bits of entropy. In other words, after three subkeys have been used, the full key entropy has been "injected" solely via the subkey mechanism, ignoring the contribution of the key-dependent S-boxes.

For key sizes larger than 128 bits, the subkey pair distribution empirically achieves entropy extremely close to the theoretical maximum of 127.95 bits noted above.

2.2 Implications

With respect to the whitening keys in particular, the Twofish design team believed that only the subkeys V_0 (which whitens the inputs to the S-boxes in the first round) and V_2 (which obscures the inputs to the S-boxes of the last round) were necessary. For example, RC6 uses this type of "half" whitening [RRS+98]. The design goal for whitening was that both of the 64-bit subkeys V_0 and V_2 independently achieve all possible 2^{64} values across keys in order to obscure the S-box inputs in the initial and final rounds. The round subkeys already obscure the S-box outputs. During the design the Twofish team discussed the choice of "half" vs. "full" whitening. While we believed that the whitening bits of V_1 and V_3 were not necessary from a cryptographic standpoint, we decided to include the full whitening mainly because it has some nice theoretical basis [KR96], it is quite inexpensive computationally, and we felt that the additional whitening would not hurt security. Therefore, we believe that the non-uniform distribution of the whitening keys, although inelegant, has no effect on the security of Twofish.

The remaining relevant question here is whether the non-uniformity of round subkey pairs affects the security of Twofish in any way. Recall that the key-dependent S-boxes also inject key entropy into the cipher on each round in an "orthogonal" way. In the following section will show that effectively more than 127 bits of key entropy are involved per round. From a point of view of entropy, the round key in the next round only has to contribute less than a bit to achieve full key-entropy mixing in two rounds.

For keys larger than 128 bits the round keys do not exhibit a significant loss of pairwise entropy, so the whitening keys and the round keys of any two rounds have nearly full entropy. We have not yet investigated whether 192-bit keys exhibit a loss of entropy for subkey triples, or 256-bit keys exhibit a loss of entropy for subkey quadruples. Even if this were the case, we do not see how this could possibly affect the security of Twofish.

3 Entropy of Subkeys with S-boxes Fixed

The second result of Mirza and Murphy is slightly more subtle. Essentially, it consists of a clever observation about a guessing attack, similar to that discussed briefly in section 8.6 of [SKW+98]. The idea is to guess the key material (S_i) controlling the S-boxes, which is a 64-bit guess in the case of the 128-bit key. Mirza and Murphy point out that the assertion in [SKW+98] that such a guess provides "no information about the round keys K_i" is incorrect, and they prove their point by constructing a case where two keys that produce the same key-dependent S-boxes result in the same round subkey for the first round. They also conjecture that, given the guess, the resulting distribution of 64-bit round subkeys behaves as a random function, with only $1 - e^{-1} = 0.632$ of all possible values occurring.

Actually, the "incorrect" statement in the Twofish paper is somewhat ambiguous. In fact, it is true that the guess provides no information about *individual* expanded key words K_i, since all 2^{32} values are possible and equally likely for a given value of $i = 0..39$. However, the guess does provide a fraction of one bit of information about *pairs* of expanded key words (K_i, K_j), where $i \neq j$. Formally, the statement is indeed incorrect, and Mirza and Murphy deserve credit for noting the inconsistency, but the noted property hardly seems to affect the security of Twofish, as we shall see below.

3.1 Results

The general problem of determining the entropy of the expanded key word pairs is very computationally intensive. Unlike the property discussed in the previous section, there seems to be no way to break it down into a set of smaller, more tractable problems. Gathering the full statistics required to verify the conjecture of Mirza and Murphy requires either a very large amount of memory, or some optimized multi-pass code that would take a long time to run. The effective space-time product of the complexity level is roughly 2^{64}. No attempt has been made to model the entire problem, but a smaller version of the problem, corresponding to a version of Twofish with a 64-bit block, has been tested extensively. Instead of using a four-by-four MDS matrix, the simplified model uses a two-by-two MDS matrix, with a Reed-Solomon code having only four information bytes and two parity bytes. We believe that this model accurately represents what would happen in the "full" case, although we have not formally proven that.

From the testing, it appears that the conjecture of Mirza and Murphy is correct. The expanded key word pairs behave effectively as random functions, implying that only about 0.632 of all possible pairs of expanded key words (K_i, K_j) can occur, given the guess. The resulting entropy is empirically about 63.2 bits for each pair of expanded key words, so the loss is less than one bit. This loss of about 0.8 bits matches the theoretically expected result for such a Poisson distribution. For single expanded key words K_i, all 2^{32} possible values occur and are equally likely. For key sizes larger than 128-bits, our tests show not only that all expanded key word pairs are possible but that the distribution is extremely uniform across keys (the empirical entropy was 128.000 bits to three significant places). No tests were performed on expanded key word triples for the larger key sizes.

3.2 Implications

The Twofish S-box guessing attack is somewhat interesting because, with known S-boxes, Twofish looks much more like a conventional Feistel cipher. Basically, the attack consists of imposing a set of $N/2$ linearly independent linear relations on the key bits, where N is the key size in bits. By carefully choosing these linear relations to correspond to the Reed-Solomon code used to generate the S-box key material, this guess effectively turns the key-dependent S-boxes into known S-boxes.

However, this advance is made at a considerable cost due to the size of the guess, and it has not been shown that any advantage has been gained. Indeed, it is not at all clear why the loss of less than one bit of entropy in a round subkey is of any concern. The argument here is perhaps best illustrated by noting that DES and Triple DES have a *much* more severe loss of subkey entropy under an analogous attack. Suppose that in DES we impose n linearly independent linear relationships among the 56 key bits. For example, we could simply guess n bits of the key, but the linear relationship method allows a more general type of guess. Having made this n-bit guess, now look at the 48 subkey bits in any particular round. If n is larger than 8, we are guaranteed that each round subkey has fewer than 2^{48} possible values. In fact, the loss of entropy is guaranteed to be at least $48 - (56 - n) = n - 8$ bits. If $n = 28$, then the 48-bit round subkey has no more than 28 bits of entropy. With a block size of 64 bits, this means that considerably fewer than 32 bits of key entropy are being used in each round. A similar argument applies to triple-DES, where (for the two-key version) you could guess 56 bits and guarantee that no round has more than 28 bits of subkey entropy. To our knowledge, no one has ever suggested that this fairly obvious property constitutes a weakness in DES or Triple DES.

Contrast the DES case with Twofish, where there is never more than a loss of a fractional bit of key material entropy per round. Further, in Twofish the loss of entropy is a very nonlinear relationship, requiring complexity on the order of 2^{64} to quantify and store. Thus, it seems likely that any attack on Twofish based on this approach could probably be readily applied to DES and Triple DES.

4 Conclusion

Mirza and Murphy pointed out two interesting properties of the Twofish key schedule, including one that was incorrectly characterized in the original Twofish paper. We have shown that the actual distribution in one case was considerably less uniform than they had speculated, but that their other statistical conjecture seems to be correct. We have also shown that these statistical properties do not apply to key sizes larger than 128 bits, and we have also argued that no cryptographic weaknesses in Twofish result from these properties in the 128-bit key case.

References

[KR96] J. Kilian and P. Rogaway, "How to Protect DES Against Exhaustive Key Search," *Advances in Cryptology — CRYPTO '96 Proceedings*, Springer-Verlag, 1996, pp. 252–267.

[Mat96] M. Matsui, "New Structure of Block Ciphers with Provable Security Against Differential and Linear Cryptanalysis," *Fast Software Encryption, 3rd International Workshop Proceedings*, Springer-Verlag, 1996, pp. 205–218.

[MM99] Fauzan Mirza and Sean Murphy, "An Observation on the Key Schedule of Twofish," *Second AES Candidate Conference*, March 1999, to appear.

[RRS+98] R. Rivest, M. Robshaw, R. Sidney, and Y.L. Yin, "The RC6 Block Cipher," NIST AES Proposal, Jun 98.

[SKW+98] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-Bit Block Cipher," NIST AES Proposal, Jun 98. `http://www.counterpane.com/twofish.html`