The Security of MacGuffin

David A. Wagner

Abstract. This paper examines MacGuffin, a recently proposed block cipher based on a new DES-like architecture. The strength of MacGuffin against differential cryptanalysis is investigated: DES is more resistant to these attacks. A differential attack requiring $2^{51.5}$ chosen plaintexts is described. In addition, the internal components of MacGuffin are shown to be sub-optimal with respect to differential cryptanalysis. Finally, a weakness in the key schedule is discussed.

1 MacGuffin

DES is a balanced Feistel network which divides the input into two equal-sized registers, modifying one half every round. Blaze and Schneier have proposed a variation on the basic DES design. Their generalized unbalanced Feistel network (GUFN) splits the block unevenly. To illustrate the power of this model and incite debate on its merits, Blaze and Schneier have offered an example design called MacGuffin [1].

The most conspicuous disadvantage of a GUFN is that each round modifies less than half of the block. One MacGuffin round affects just 16 bits, yet DES alters 32 bits of the block. For these reasons, Blaze and Schneier hypothesize that two rounds of MacGuffin should yield at least the strength of one DES round [1]. Their hypothesis is central to the security of MacGuffin.

This paper presents an overview of several attacks on MacGuffin, rather than specifying the methods in great detail. See [2] for a more comprehensive discussion of these topics.

2 Preliminary analysis

To compare the strength of DES and MacGuffin, one can first examine reduced-round variants of the two block ciphers with a variety of standard statistical tests. Table 1 summarizes the results of this investigation. No weaknesses were found, and MacGuffin behaved about as well as DES in these simple tests.

Another way to compare the reduced-round variants of DES and MacGuffin involves examining several elementary attacks on these simplified ciphers. The cryptographic literature documents a great variety of elementary attacks on DES, so there is no lack of ideas to try. Notably, MacGuffin has no obvious complementation properties to speed up exhaustive search, and is immune from Chaum and Evertse's meet-in-the-middle attack [5] after more than 6 rounds. MacGuffin is about as resistant as DES against most other elementary attacks.

Still, two slightly more sophisticated statistical attacks penetrated reduced-round variants of MacGuffin successfully enough to be worrisome. The first uses

Table 1. Statistical tests on DES and MacGuffin

Test	Rounds to pass DES	Rounds to pass MacGuffin	Comments
avalanche	6	14	
strict avalanche	6	13	consistent with above test up to limits of statistical significance
key-dependence avalanche	6	5	MacGuffin passes so quickly because of its non-linear key schedule
sequence complexity	5	10	
binary derivatives	5	10	
bit dependencies	5 [3]	9	
cycle lengths	†	†	complexity of exhaustive testing too large to be practical
small subgroup	‡	‡	complexity of testing MacGuffin too large to be practical

Notes:

† DES passes limited cycling tests [3]; MacGuffin passed a limited cycling test too.
‡ DES does not generate a small subgroup [4]; MacGuffin's status is unknown.

statistical regularities detected by the strict avalanche test to break MacGuffin-16. The second attack takes advantage of correlations caused when S-boxes in nearby rounds share input bits. (This approach appears to be similar to a recently published attack on DES which exploits duplicated input bits at neighboring S-boxes in one round [6].) In both cases, the statistical regularities seem more pronounced in MacGuffin than DES, so some reduced-round variants of MacGuffin are more vulnerable than those of DES. On the other hand, none of these attacks penetrate more than 24 rounds of MacGuffin, so they will not be discussed any further.

3 Differential cryptanalysis

A variety of mathematically plausible heuristics were applied in search of a high probability iterative characteristic for MacGuffin. These experiments found that MacGuffin admits a 4 round iterative characteristic with probability 1/148.9. This characteristic is shown in full detail in Figure 1. In comparison, the best DES iterative characteristic has 2 rounds and probability 1/234 [7]. Thus, differential cryptanalysis penetrates MacGuffin more successfully than DES.

3.1 Experimental cryptanalysis

A straightforward $4R$ differential attack based on the clique method was implemented to test the behavior of the differential attack in practice, breaking

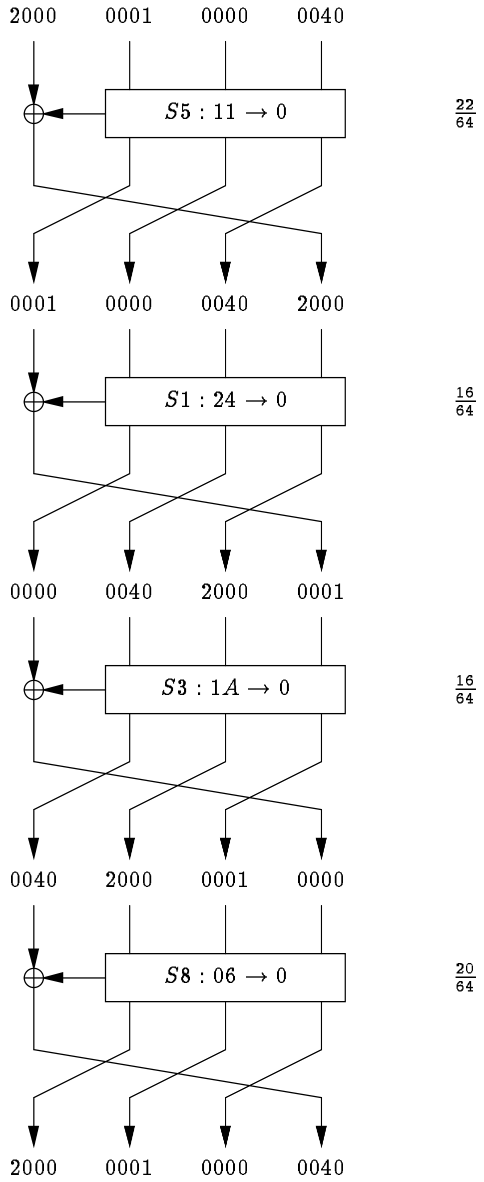

Fig. 1. An iterative characteristic for MacGuffin

MacGuffin-16 with great success. Though the experimental implementation undoubtedly could have been improved with several sophisticated techniques, the point was to test the essential features of a differential attack. This goal was achieved successfully without any fancy optimizations.

In general, 10–20 right pairs assured a significant success rate. This rather large requirement is explained by the need to repeat the attack for every round subkey. With fewer right pairs, the attack often failed to find a unique value for all the round subkeys; yet even then, usually only a few possible values remained. The observed graceful degradation is reassuring (to the attacker!).

3.2 Attacking all 32 rounds

To attack all 32 rounds of MacGuffin, bypass the first three rounds with the techniques of Biham and Shamir [8], use a $3R$ attack, and cover the middle 26 rounds with repetitions of the 4 round iterative characteristic from Figure 1. The characteristic has probability

$$\frac{20}{64} \cdot \frac{1}{148.9^6} \cdot \frac{22}{64} \approx 2^{-46.5}.$$

Count on 54 round subkey bits. Since each right pair must have zeros in 50 bits of the ciphertext xor, we expect $S/N \approx 12$. (Note that the S/N ratio degrades too much with a $4R$ attack.) The attacker will need approximately 10–15 right pairs for this first stage, so $2^{50.5}$ chosen pairs should suffice to find much of the last round subkey.

The cryptanalyst may again repeat the attack with rotated versions of the iterative characteristic to find the rest of the final round subkey. Partial information gained in the first pass can be used to greatly improve the S/N ratio for later passes. Finally the last round subkey may be peeled off and the attack repeated until all round subkeys are known.

Combining four characteristics into one structure, this attack will need $2^{51.5}$ chosen plaintexts. The attack uses 2^{54} counters, and all ciphertext blocks must be saved for each pass, so the cryptanalyst will need approximately $2^{54} + 8 \cdot 2^{51.5} \approx 2^{55}$ bytes of storage. In comparison, the best differential attack on DES with independent subkeys requires approximately 2^{60} chosen plaintexts and very little memory [7, p. 68]. (There is an improved attack on vanilla DES with dependent round subkeys [8], but it succeeds only by taking advantage of the linear DES key schedule, so MacGuffin is best compared to DES with independent round subkeys.) In short, MacGuffin is no stronger than DES against a differential attack.

3.3 An attack based on a key schedule weakness

If memory is at a premium, a simpler differential attack may be of interest. Reverse the normal differential attack, requesting chosen ciphertexts in hopes of recovering the first round subkey. Use a $2R$ attack, and cover the remaining 30 rounds with a characteristic of probability

$$\frac{20}{64} \cdot \frac{1}{148.9^7} \cdot \frac{22}{64} \approx 2^{-53.6}.$$

After 2^{58} chosen ciphertext pairs, we expect to find 20 right pairs and one wrong pair. Because of the excellent recognition of right pairs, we can use the clique method and avoid the memory requirements of the normal counting method. Repeating the attack with rotated versions of the characteristic will recover the entire last round subkey; by taking advantage of partial knowledge of the last round subkey gained by earlier characteristics, we can achieve nearly perfect recognition of all right pairs. Experiments with MacGuffin-16 indicate that one can find the first round subkey with about 2^{59} chosen ciphertexts and very little memory.

After the first round subkey is known, the search method described in Section 5.2 can derive the full 128 bit key with 2^{80} off-line trial encryptions. If 2^{22} related-key chosen plaintexts are available, the full 128 bit key can be found with just 2^{58} off-line trial encryptions. In this case the total complexity of the attack will be reduced to approximately 2^{59} operations, with negligible storage requirements.

3.4 Other issues

If partial information about the first round subkey is available, we can often improve the probability of the first round of a characteristic [7, pp. 46–47]. For MacGuffin, the probability and S/N ratio of the iterative characteristic can be improved by a factor of about 1.4 if one bit of the first round subkey is known. On the other hand, there is no way to take advantage of this enhancement if we bypass more than two rounds.

The probability of a characteristic is often dependent on subkey values in later rounds as well [7, pp. 54–55]. Here, though, the MacGuffin characteristic has only a very weak key dependence. In the most extreme possible situation, the 4 round iterative characteristic could be improved by a factor of $1.02 \cdot 1.05 \cdot 1.05$. This leads to a total improvement in the probability and the S/N ratio by at most a factor of 2.0 even if all round subkeys are favorable. Since this event occurs with probability 2^{-18}, the key dependence is essentially negligible.

Lai, Massey, and Murphy [9] have studied the propagation of differential characteristics, modeling them as Markov chains. MacGuffin is a Markov cipher, so their results prove that MacGuffin is immune from differential cryptanalysis after sufficiently many rounds under the hypothesis of stochastic equivalence and the assumption that the round subkeys are uniformly and independently distributed (two plausible mathematical hypotheses). Lai, Massey, and Murphy also propose a design principle for block ciphers:

> The transition probability matrix of a Markov cipher should not be symmetric.

MacGuffin passes this design criterion after 11 rounds.

3.5 Related research

Rijmen and Preneel [11] independently found the iterative characteristic depicted in Figure 1 and showed that MacGuffin is no stronger than DES. They also demonstrated that MacGuffin is susceptible to linear cryptanalysis, though not quite so vulnerable as DES. Their work is especially impressive, for they cryptanalyzed MacGuffin at the workshop where it was first formally presented.

4 MacGuffin design choices

Blaze and Schneier do not report taking any special precautions during the design process to protect against differential cryptanalysis. In contrast, DES was apparently designed with immunity to differential cryptanalysis in mind at every stage of the design [12]. It is interesting to examine whether other MacGuffin-like designs might improve its resistance to differential cryptanalysis.

One possibility is to investigate different choices of the input permutation. Blaze and Schneier list five design criteria used to direct f inputs to the S-box inputs [1]; most are straightforward adoptions of the DES design rules or logical new principles specific to GUFNs. Examination of the MacGuffin input permutation suggests that Blaze and Schneier probably followed another design rule (though [1] never mentions it):

P6 No S-box in any round sends both its output bits to any one S-box in any other round.

One omission is noteworthy: Blaze and Schneier avoided an explicit expansion box. This has dubious cryptographic merits—for example, Biham and Shamir found that removing the E box weakens DES by about 9 orders of magnitude [7]. On the other hand, the GUFN structure may fulfill the same cryptographic function as the E box.

One can also study other choices of S-boxes. The MacGuffin S-boxes are just truncated DES S-boxes; much literature is available on design rules for creating DES-like S-boxes, so it is convenient to generate new DES-like S-boxes and truncate them. These experiments followed S-1, S-2, S-3, S-4, and S-7 from Coppersmith's account of the DES design process [12]; his other criteria were relevant only to the carefully balanced machinery in DES and appear to be of limited value to MacGuffin. Possibly with great care one could generate better S-boxes specifically tailored towards use in MacGuffin without enlarging their dimensions; on the other hand, the small size of the MacGuffin S-boxes may be a fatal flaw.

Many new input permutations and new S-boxes were tested for resistance to differential cryptanalysis. No attempt was made to evaluate the effectiveness of non-differential attacks on the new variations, so there is no guarantee that they would actually improve MacGuffin. The results of these experiments are reported in Table 2, with p denoting the probability of a 4 round iterative characteristic. (No useful iterative characteristics with more than 4 rounds were found.)

Recall that the MacGuffin S-boxes are obtained by discarding the middle two output bits of the DES S-boxes. Rijmen and Preneel have examined other subsets of output bits from the DES S-boxes, and found that the MacGuffin choices are sub-optimal [11].

Table 2. Other design choices for MacGuffin

Category	Permutations		S-boxes
	P1–P5	P1–P6	
Best p	1/655.4	1/546.1	1/336.1
Median p	1/256.0	1/240.1	1/132.4
Worst p	1/60.7	1/93.1	1/47.0
% with $p < 1/148.9$	75%	90%	50%
% with $p < 1/243$	58%	55%	33%

5 The key schedule

The MacGuffin key scheduling algorithm (shown in Figure 2) departs drastically from its DES origins, in an attempt to approximate a one-way function. (One-way key schedules can help foil related-key attacks, prevent recovery of the master key from the round subkeys, partially discourage differential and linear cryptanalysis, and avoid weak keys.)

KeySched(K)
1. $subkey[r][i] \leftarrow 0 \quad \forall i, r$
2. UpdateSubkey($subkey$, $K_{0\ldots63}$)
3. UpdateSubkey($subkey$, $K_{64\ldots127}$)

UpdateSubkey($subkey$, $block$)
1. $(l, a, b, c) \leftarrow block$
2. for $r \leftarrow 0$ to 31
3. $\quad (l, a, b, c) \leftarrow$ MacGuffinEncrypt($subkey$, (l, a, b, c))
4. $\quad subkey[r][0] \leftarrow subkey[r][0] \oplus l$
5. $\quad subkey[r][1] \leftarrow subkey[r][1] \oplus a$
6. $\quad subkey[r][2] \leftarrow subkey[r][2] \oplus b$

Fig. 2. The MacGuffin key scheduling algorithm

5.1 Weak keys

Because of its linear key schedule, DES suffers from several weak keys; it is natural to wonder whether MacGuffin's revised key schedule banishes all those difficulties. The complexity of MacGuffin's key schedule eliminates all the obvious candidates, but there still might be subtle interactions. A search was mounted for special fixed points which could cause a weak key. No weak keys were found.

5.2 Inverting the key schedule: a special case

Ideally the MacGuffin key scheduling algorithm should behave roughly like a one-way function. For example, inverting the key schedule (determining the 128 bit key given some of the round subkeys) should require about 2^{128} operations. In fact, the MacGuffin key schedule can be inverted much more efficiently.

Inverting the key schedule is simplest when the first round subkey is known. If the first round subkey is (x, y, z), the algorithm shown in Figure 3 will derive the 128 bit encryption key. It requires about 2^{81} off-line encryptions in the worst case, or half that on average. This is many orders of magnitude faster than exhaustive search.

```
InvertKeySchedule(x,y,z)
for each possible value of K_{0...63}
    temp[r][i] ← 0   ∀r, i
    UpdateSubkey(temp, K_{0...63})
    l ← x ⊕ temp[0][0]
    a ← y ⊕ temp[0][1]
    b ← z ⊕ temp[0][2]
    for each possible 16 bit value c
        K_{64...127} ← MacGuffinDecrypt(temp, (l, a, b, c))
        check the guessed value K on a few known plaintexts
        if it passes them all, return K as the true key
```

Fig. 3. Inversion given the first round subkey

5.3 Generalized inversion of the key schedule

An extension of this method can be used when the second round subkey is known instead. For fixed a, b, c define the map $H_{a,b,c}$ by

$$H_{a,b,c}(l) = l \oplus f(l \oplus a, a \oplus b, b \oplus c).$$

Note that the expected cardinality of the inverse image $H^{-1}_{a,b,c}(y)$ is exactly 1. To invert the key schedule given a second round subkey, the straightforward extension of Figure 3's InvertKeySched algorithm must invert $H_{a,b,c}$. Clearly an inverse image $l = H^{-1}_{a,b,c}(y)$ can be found by simply trying all 2^{16} possibilities for l. This leads to an algorithm for inverting the MacGuffin key schedule given the second round subkey requiring roughly 2^{96} operations.

By taking advantage the special structure of the f function, one can in fact invert $H_{a,b,c}$ more efficiently. First guess the 6 bits $l_{0...5}$, the $S1, S2$ inputs and outputs. Discard the guesses which are inconsistent with a, b, c, y, leaving about

2^2 possibilities. Guess the bit l_6, so that the $S3$ inputs and outputs are known, and again discard impossible guesses. Continue guessing bits from the sequence

$$l_{0\ldots5},\ l_6,\ l_{7,12,14},\ l_{11,13,15},\ l_9,\ l_{8,10}$$

discarding inconsistent guesses at each stage. This approach requires about

$$2^6 + 2^2(2 + 2^{-1}(2^3 + 2(2^3 + 2(2 + 2^{-1}(2^2))))) \approx 2^7$$

table lookups, so the map $H_{a,b,c}$ can be inverted about as quickly as one trial encryption. A careful counting shows that this method enables an attacker to derive the 128 bit encryption key from knowledge of the second round subkey with expected complexity of 2^{81} encryptions.

When a later round subkey is known, a similar analysis can be carried through by a fairly straightforward application of these two ideas. The complexity of these inversion attacks on the MacGuffin key schedule is presented in Table 3. These results show that the MacGuffin key schedule falls short of achieving ideal one-wayness.

Table 3. Average-case complexity of deducing K given a round subkey

Round	Encryptions
1	2^{80}
2	2^{81}
3	2^{97}
4	1.5×2^{113}
5–32	2^{127}

5.4 Inverting the key schedule with related keys

Several authors have introduced the idea of related key cryptanalysis [13, 14]. Suppose we are allowed 2^t $(t < 64)$ related key chosen plaintext queries, and we wish to speed up inversion of the key schedule. Choose one plaintext P and obtain the encryption of P under the related keys

$$\mathcal{K} = \{k_{\text{orig}} \oplus \chi : \chi_{t\ldots127} = 0\}.$$

Replace the loop in line 1 of Figure 3's InvertKeySchedule by a loop which only examines keys satisfying $K_{0\ldots t-1} = 0$. Next change line 9 so that the result of the trial encryption $E_K(P)$ is compared against the set of ciphertexts $\{E_k(P) : k \in \mathcal{K}\}$. Upon finding a match $E_K(P) = E_k(P)$ for $k = k_{\text{orig}} \oplus \chi$ with χ, K known, we may guess $K = k$ and thereby guess $k_{\text{orig}} = K \oplus \chi$. By testing this against a few known plaintexts, we can filter out all incorrect guesses with very high probability. Observe that there must be some $K \in \mathcal{K}$ with $K_{0\ldots t-1} = 0$, so the cryptanalyst is guaranteed to find the correct value of k_{orig}.

This modification can also be applied in a straightforward way to the more general algorithms described earlier for inverting the key schedule given the r-th round subkey. The complexity of the inversion attacks (see Table 3) is decreased by a factor of 2^t when 2^t related key chosen plaintext queries are feasible.

5.5 Repairing the key schedule

It may be possible to fix these deficiencies by appending the following line to the KeySched procedure in Figure 2:

4. UpdateSubkey(*subkey*, $K_{0\ldots 63}$)

The proposed change foils the attack described in this paper. More research is needed to determine if the modified version is a secure one-way key schedule.

6 Conclusion

MacGuffin was tested for weaknesses with a variety of standard statistical tests. A number of elementary cryptanalytic methods were also applied to MacGuffin. No fatal weaknesses were found, and MacGuffin behaved about as well as DES in most of this examination.

Several aspects of the MacGuffin key schedule were examined for weaknesses. No weak keys were found. However, the key scheduling algorithm can be inverted in certain circumstances much faster than would be possible if it were a one-way function. The complexity of inversion is reduced even further when related key chosen plaintexts are available. A modification was suggested to fix this defect.

MacGuffin was found to be weaker than DES against differential cryptanalysis. One attack breaks MacGuffin with about $2^{51.5}$ chosen plaintexts. A differential attack was implemented on a workstation, cryptanalyzing MacGuffin-16 with great success.

The MacGuffin internal design elements were compared to some other possible choices. The input permutation and S-boxes were found to be sub-optimal with respect to differential cryptanalysis.

Because of these problematic weaknesses, long-term use of MacGuffin is not recommended. On the other hand, MacGuffin is just one particular incarnation of the more general GUFN structure. DES was carefully crafted so all components would mesh together to provide the maximum security possible [12]; perhaps the GUFN architecture could reap substantial benefits from this kind of meticulous attention to detail.

7 Acknowledgements

I would like to extend my gratitude to Steve Bellovin for proposing this research topic and offering many helpful suggestions; his kindness is appreciated. Thanks also to Vincent Rijmen for several useful discussions.

References

1. Matt Blaze and Bruce Schneier. The MacGuffin block cipher algorithm. In Bart Preneel, editor, *Fast Software Encryption: Proceedings of K. U. Leuven Workshop on Algorithms*. Springer-Verlag, 1995. (to appear).
2. David Wagner. The security of MacGuffin. Senior thesis, Princeton University, 1995. Available for anonymous `ftp` on the Internet from `ftp.princeton.edu` as `/pub/dawagner/crypto/me/senior-thesis.ps`.
3. Alan G. Konheim. *Cryptography: A Primer*. John Wiley and Sons, New York, 1981.
4. Keith W. Campbell and Michael J. Wiener. DES is not a group. In *Advances in Cryptology: CRYPTO '92*, pages 512–520. Springer-Verlag, 1992.
5. David Chaum and Jan-Hendrik Evertse. Cryptanalysis of DES with a reduced number of rounds. In *Advances in Cryptology: CRYPTO '85*, pages 192–211. Springer-Verlag, 1986.
6. D. Davies and S. Murphy. Pairs and triplets of DES S-boxes. *Journal of Cryptology*, 8(1):1–25, Winter 1995.
7. Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, New York, 1993.
8. Eli Biham and Adi Shamir. Differential cryptanalysis of the full 16-round DES. In *Advances in Cryptology: CRYPTO '92*. Springer-Verlag, 1993.
9. Xiejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In *Advances in Cryptology: EUROCRYPT '91*, pages 17–38. Springer-Verlag, 1991.
10. Vincent Rijmen and Bart Preneel. Cryptanalysis of McGuffin. In Bart Preneel, editor, *Fast Software Encryption: Proceedings of K. U. Leuven Workshop on Algorithms*. Springer-Verlag, 1995. (to appear).
11. Vincent Rijmen and Bart Preneel. Cryptanalysis of McGuffin, 1995. (a corrected version of [10]).
12. D. Coppersmith. The Data Encryption Standard (DES) and its strength against attacks. *IBM J. Res. Develop.*, 38(3):243–250, May 1994.
13. Robert Winternitz and Martin Hellman. Chosen-key attacks on a block cipher. *Cryptologia*, XI(1):16–20, January 1987.
14. Eli Biham. New types of cryptanalytic attacks using related keys. In *Advances in Cryptology: EUROCRYPT '93*, pages 398–409. Springer-Verlag, 1994.