MARS Attacks! Preliminary Cryptanalysis of Reduced-Round MARS Variants

John Kelsey and Bruce Schneier

Counterpane Internet Security, Inc., 3031 Tisch Way, San Jose, CA 95128
{kelsey,schneier}@counterpane.com

Abstract. In this paper, we discuss ways to attack various reduced-round variants of MARS. We consider cryptanalysis of two reduced-round variants of MARS: MARS with the full mixing layers but fewer core rounds, and MARS with each of the four kinds of rounds reduced by the same amount. We develop some new techniques for attacking both of these MARS variants. Our best attacks break MARS with full mixing and five core rounds (21 rounds total), and MARS symmetrically reduced to twelve rounds (3 of each kind of round).

1 Introduction

MARS [BCD+98] is a block cipher submitted by IBM to the AES [NIST97a] [NIST97b], and one of the five finalists for AES. The cipher has an unconventional structure, consisting of a cryptographic "core" in the middle, and a "wrapper" surrounding the core to protect it from various kinds of attack. As with all ciphers, the only way we know to determine the strength of MARS is to try to cryptanalyze various weakened versions of it.

In this paper, we discuss attacks on reduced-round variants of MARS. Because of MARS' unconventional structure, there are many different reduced-round variants worth considering. Here, we focus on two: A variant with the full "wrapper" but fewer rounds of cryptographic core, and a variant with both the core and wrapper reduced by the same number of rounds. In other work [KS00], we have considered the cryptographic core without the wrapper. In view of the stated purpose of the "core" and "wrapper" rounds, we believe the two variants in this paper have a great deal to teach us about the ultimate strength of MARS.

1.1 Current Results

Our results are as follows:

Attacks on Reduced-Round MARS Variants

Reduced-Round Version	Work	Memory	Text Requirements
Full Mixing + 5 Core	2^{232} half encs	2^{236} bytes	8 known plain
Full Mixing + 5 Core	2^{247} partial encs	2^{197} bytes	2^{50} known plain
6 Mixing + 6 Core	2^{197} partial decs	2^{73} bytes	2^{69} chosen plain
0 Mixing + 11 Core	2^{229} partial encs	2^{69} bytes	2^{65} chosen plain

For reasons of both space and clarity of presentation, the attacks against the core rounds only are not included in this paper, and can be found in [KS00].

1.2 Implications of the Results

None of our current results on MARS come close to breaking the full cipher; our best results to date break only 21 out of 32 rounds (and this counts attacking 16 mixing rounds, which are far weaker than the core rounds). However, the attacks in this paper demonstrate ways to attack the MARS structure, and so highlight potential weaknesses of that structure. They also help us to understand how the components of this complex cipher interact to resist attack.

We also introduce a new kind of meet-in-the-middle attack, which may be of independent interest. Although we demonstrate its use initially on MARS, it may be useful against other ciphers, especially other ciphers with heterogenous structures.

1.3 Guide to the Rest of the Paper

The remainder of the paper is arranged as follows: First, we discuss the structure of MARS, and introduce notation and terminology to describe its inner workings. Next, we discuss a set of attacks on MARS with only the number of core rounds reduced. Next, we develop attacks on MARS variants with the same number of each kind of round taken out. We conclude the paper with a discussion of the new techniques we have developed for MARS, the implications of our results, and some open questions.

2 The MARS Structure

The MARS structure can be considered as six different layers through which a plaintext block must pass to become a ciphertext block:

1. Pre-Whitening Layer: The plaintext has 128 bits of key material added to its words modulo 2^{32}.
2. Forward Mixing Layer: Eight rounds of unkeyed mixing operations making extensive use of the MARS S-box.
3. Forward Core Layer: Eight rounds of keyed unbalanced Feistel cipher, using a combination of S-box lookups, multiplications, data-dependent rotations, additions, and XORs to resist cryptanalytic attack.
4. Backward Core Layer: Eight rounds of keyed unbalanced Feistel cipher, using a combination of S-box lookups, multiplications, data-dependent rotations, additions, and XORs to resist cryptanalytic attack.
5. Backward Mixing Layer: Eight rounds of unkeyed mixing operations making extensive use of the MARS S-box.
6. Post-Whitening Layer: The block has 128 bits of key material subtracted from its words modulo 2^{32}.

In this paper, we typically discuss MARS in terms of two different components. The forward and backward core layers together make up the "cryptographic core"; this core looks like a relatively conventional block cipher, and appears to be reasonably resistant to attack. The pre-whitening, forward mixing layer, backward mixing layer, and post-whitening layer together make up the "wrapper," which protects the cryptographic core from various kinds of attack by requiring a large key guess or some clever cryptanalysis to gain access to inputs and outputs of the core. This is a very different block cipher design than is used in the other AES candidates. Among other things, this new design makes it relatively difficult to determine how to come up with reduced-round variants of the cipher to attack.

2.1 The Cryptographic Core

The strength of MARS resides fundamentally in the strength of the core rounds. Both forward and backward core rounds use the same E function, which takes one 32-bit input and two subkey words, and provides three 32-bit words. Each output is combined into one of the three other words. The only difference between forward and backward rounds is the order in which the outputs are combined with the words. The core rounds' strength is based primarily on mixing incompatible operations in the E function, and in their target-heavy Feistel structure, which causes both linear and differential characteristics to quickly spread out into every word. A full description of the MARS core rounds appears in [BCD+98].

The cryptographic core, with a few additional rounds, could stand alone as a cipher; indeed, this would have been a fairly conventional design. Instead, the MARS design team chose to use a smaller number of core rounds,[1] but to surround the core with a "wrapper."

2.2 The Wrapper

The key addition/subtraction and mixing layers surround the core rounds, preventing direct access to the core rounds from either the plaintext or the ciphertext side. While the wrapper itself isn't particularly resistant to cryptanalysis, it is quite different in structure than the core, and it is designed to require guessing of key material before an attacker can learn or control either inputs or outputs to the core.

The mixing layers, like the core, have an unbalanced (target-heavy) Feistel structure, but use only S-boxes and mixing of addition and XOR.

We are a little puzzled by the decision to involve only 128 bits of key material on each side of the core. This leaves the possibility of an attacker guessing his way past either half of the wrapper, and thus seeing either input or output, with a guess of only half the maximum key length. A small change to the design would have involved 256 bits of key material on each side, and thus made partial key

[1] Assuming the same level of performance, adding the wrapper requires reducing the number of core rounds.

guessing worthless as a method of bypassing the wrapper. Below, we consider some attacks that simply guess key material to bypass the wrapper entirely. Inclusion of additional key material would apparently have stopped such attacks at very little cost. While we understand the role of the "wrapper" in helping the core resist attacks, we don't understand why it couldn't fulfill this role just as well with another 128 bits of key material being combined in on each side. Such a change to the design would have rendered many of the attacks we describe in this paper impossible, at a low performance cost.

2.3 Reduced-Round MARS Variants

In a conventional cipher design, the rounds are all more-or-less the same except for subkeys (and sometimes round constants). There is an obvious way to develop weakened versions of such ciphers: simply reduce the number of rounds. Because of the very different roles of the different kinds of rounds in MARS, however, there are a number of reduced-round MARS variants that can teach us valuable lessons about the ultimate strength or weakness of MARS.

Some reduced-round variants we have considered include:

Chopping Off the Beginning or End We evaluate the strength of most ciphers by considering versions with several of the first or last rounds omitted: first the whitening layers and then several rounds of the mixing layers. This isn't a terribly rewarding way to look at MARS, since it omits important parts of the cipher's structure.

Core Rounds Only Because most of the cryptographic strength of MARS apparently resides in the core rounds, it is reasonable to consider the strength of these rounds independently. By developing such attacks, we learn how to attack a fundamental component of the cipher, which may be of use in mounting attacks on the full cipher in the future. For space reasons, most of our analysis of the MARS core is described in another paper [KS00].

Full Cipher with Reduced Core Rounds An alternative way to evaluate the strength of MARS is to consider the full cipher, but with fewer core rounds. This allows us to see how the core rounds might be attacked, even through the whitening and mixing layers that wrap the core rounds of the cipher. It also gives us insights into how strong the core needs to be to allow MARS to resist cryptanalysis.

Symmetric Reductions of the Cipher In the full MARS, there are four different types of rounds, each repeated eight times, for a total of 32 total rounds. It is reasonable to consider symmetric reductions of this; for example, we can consider a MARS variant with only three or four or six of each kind of round. In some sense, this probably provides more information about attacking the full MARS cipher than other kinds of weakened variant, because all the components of the cipher are present.

We believe the last three can teach us many lessons about the ultimate strength of MARS, both in terms of developing tools for attacking the full cipher, and in terms of evaluating how close the best current attacks come to breaking the full MARS.

3 Full Mixing with Reduced Core Rounds

In this section, we consider attacks on a MARS variant with the full "wrapper," but a reduced "cryptographic core." These attacks demonstrate how it is possible to mount attacks on a cryptographic core, even through the full wrapper, albeit against a much-weakened core. These attacks penetrate by far the largest number of rounds of the cipher, because they focus on the relatively weak mixing rounds, rather than the much stronger core rounds.

Our attacks in this section are meet-in-the-middle attacks, requiring enormous memory resources to implement, and thus purely academic. In the remainder of this section, we will assume that one memory access to these huge memory devices costs about the same amount of work as a partial encryption. There are ways to trade off time for memory in these attacks, but they generally aren't useful in the context of these attacks.

3.1 A Straightforward Meet-in-the-Middle Attack on Five Core Rounds

Consider MARS with full mixing and whitening layers, but with the core reduced to three forward and two backward core rounds. This is vulnerable to a meet-in-the-middle attack as follows:

1. Request eight plaintext/ciphertext pairs.
2. From the plaintext side, guess:
   (a) The 128-bit pre-whitening key.
   (b) The 62-bit first round key.
   (c) $K^{\times}$ and the low nine bits of K^{+} for the second round.
   (d) This yields knowledge of $A_2 = D_3 >>> 13$. Compute this value for all eight plaintexts, and put the resulting 256-bit value in a sorted list.
3. From the ciphertext side, guess:
   (a) The 128-bit post-whitening key.
   (b) The 62-bit last round key.
   (c) $K^{\times}$ and the low nine bits of K^{+} for the next-to-last round.
   (d) This yields knowledge of $A_2 = D_3 >>> 13$. Compute this value for all eight ciphertexts, and search the sorted list from the plaintext guesses for a match on this 256-bit value.

This attack passes through 16 mixing rounds and 5 core rounds (thus 21 rounds total), at a cost of about 2^{232} half encryptions' work (that is, 2^{229} work for each of the eight texts), and about 2^{236} bytes of memory. The memory requirements are totally unreasonable in practice, so this attack is purely academic.

Summary of Results

Attack On:	Full Mixing Plus Five Core Rounds (21 total rounds)
Attack Type:	Meet-in-the-Middle
Work:	2^{232} half-encryptions
Memory:	2^{236} bytes
Texts:	Eight known plaintexts

3.2 The Differential Meet-in-the-Middle Attack

Here, we introduce the concept of a differential meet-in-the-middle attack. This attack is related to the attack on the Mansour-Even construction by Daemen [Dae95], the attack on DESX by Kilian and Rogaway [KR96], and the inside-out attack of Wagner [Wag99].

In a standard meet-in-the-middle attack, we guess some key from the first and second halves of the cipher, and then match on some middle value. For example, an attack on double-DES starts by getting two plaintexts and their corresponding ciphertexts. We then guess the key for the first DES encryption, and for each such key guess, we compute the middle value from the two plaintexts if they were encrypted under that key. This is stored in a sorted list. We then guess the second DES key, and compute, for each guess, the middle value from decrypting the two ciphertexts. These values are searched for in the sorted list. When we find a matching value, it is very likely that this corresponds to the right key.

This attack can be generalized. For example, it is not necessary that the whole intermediate value to an encryption be computed; we can compute a single bit from each direction, and then examine more plaintext/ciphertext pairs. Similarly, if we can compute some checksum from intermediate values we reach by key guesses from the plaintext and ciphertext sides, then we need never have any knowledge of actual intermediate text values, as in [KSW99].

An extension to this idea allows the use of probability one differentials through some intermediate part of the cipher. Consider the truncated differential $(0, 0, 0, \delta_0) \rightarrow (\delta_1, 0, 0, 0)$, which goes through three MARS core rounds with probability one. The truncated differential works the same way in reverse, naturally. This means that if we see a right input pair (a pair with difference $(0, 0, 0, \delta_0)$), we will also see a right output pair (a pair with difference $(\delta_1, 0, 0, 0)$).

In a meet-in-the-middle attack, we must find some value that can be computed from both the top (input) and the bottom (output) of the cipher with a key guess, build a sorted list of these values, and look for pairs of keys that match on these values.

With these differentials, we can compute such a value as follows:

1. Get about 2^{50} known plaintexts and their corresponding ciphertexts. Label each plaintext/ciphertext pair with an index number, $0..2^{50} - 1$.
2. Guess part of the key from the top, and compute intermediate states for each plaintext given that key guess.
3. Sort the plaintext-intermediate values on their first three words.
4. Go through these values, and note each pair of texts that matches on their first three words by their index numbers. List these in sorted order, lower index number first in each pair. We expect about eight of these pairs.
5. Guess part of the key from the bottom, and compute intermediate states for each ciphertext from that key guess. Sort the ciphertext-intermediate values on their last three words.

6. Go through these values, and note each pair of texts that matches on their last three words by their index numbers. List these in sorted order, lower index number first in each pair. We expect about eight of these pairs.
7. Because the differential has probability one in both directions, there must be the same number of these pairs, and the pairs must be identical, from both plaintext and ciphertext. All we've done here is to list which pairs have the right input and output XOR differences to fit this truncated differential.

From this, we now have a "checksum" (the right pair indices) that we can compute across three MARS core rounds. (We can easily restrict the checksum's size to four or eight matching pairs. The indices of the pairs must be put in some standard order; for example, note each right input pair of indices in sorted order, and then sort the pairs in order of each pair's lowest index number.) This checksum costs about $50 \times 2^{50} \approx 2^{56}$ work to find for any block of 2^{50} texts. We can thus do the following differential meet-in-the-middle attack on the full MARS mixing layers plus five rounds of core:

1. Get 2^{50} known plaintext/ciphertext pairs, and label each by an index number as described above.
2. From the plaintext side, guess the pre-addition key and the first core round key, a total of 2^{190} different key guesses.
3. For each key guess, take the predicted inputs to the second core round, and compute the input right pair indices as described above. This takes about 2^{56} work per key guess. Write the input right pair indices to a huge list, one entry per key guess with the first eight right input pair indices in sorted order.
4. Do the same thing from the bottom, continuing to add entries to the huge list.
5. Sort the huge list, which will now have 2^{191} entries in it, and should thus take about $191 \times 2^{191} \approx 2^{199}$ work to sort.
6. Find the match between key guesses from the plaintext and ciphertext sides.

The total work done is thus $2^{190} \times 2^{56} \times 2 + 2^{199} \approx 2^{247}$. The attack recovers all key material used in the pre- and post-addition/subtraction keys, and the first and last core rounds' values, as well. The total memory taken is 56×2^{191} bytes.

Summary of Results

Attack On:	Full Mixing Plus Five Core Rounds (21 total rounds)
Attack Type:	Differential Meet-in-the-Middle
Work:	2^{247} partial encryptions
Memory:	2^{197} bytes
Texts:	2^{50} known plaintexts

3.3 Tradeoffs Between Differential and Conventional Meet-in-the-Middle Attacks

Note that the differential meet-in-the-middle attack requires slightly more work but considerably less memory than the conventional meet-in-the-middle attack.

The advantage of the differential meet-in-the-middle attack is that it allows us to pass through three core rounds for free; the disadvantage is in the cost of detecting the property that passes through those three core rounds for free, and the far larger number of known plaintexts required. This tradeoff determines which attack is best-suited for a given cipher and attack model.

For reference, we will point out that the differential meet-in-the-middle attacks can be used with less memory against smaller numbers of core rounds. For example, we can use the same truncated differential and filtering process against three rounds of core, dropping the total memory required to about 2^{133} bytes of memory, at a work factor of about 2^{185} partial encryptions.

We have considered ways of extending the differential meet-in-the-middle attack another round. Unfortunately, there are complications involved in using either differentials with probability substantially lower than one, or in using differentials that don't run both directions with approximately equal probability.

3.4 Using Lower Probability Differentials

Consider a differential with probability 1/2 through several rounds of some cipher, and assume we must find four input right pairs that are also output right pairs. The problem is that we must have an exact match for the final sorting and searching phase of the meet-in-the-middle attack to work. The only way we can see to mount the attack in this situation is to generate and store many different input right pairs, in hopes that one will consist of all successful differentials, and thus, right output pairs.

The most efficient way to do this will probably be to find R right input pairs, and add to the sorted list of input right pairs every possible 4-tuple of the pairs, and then to do the same with the right output pairs. That will involve R choose 4 entries in the list, and we can expect it to work if we expect at least 4 of the R input right pairs to result in output right pairs. The number of expected right output pairs from R right input pairs is binomially distributed; for reference, with nine right input pairs, we expect four right output pairs with probability 1/2. With twenty right input pairs, we have about a 0.94 probability of getting some subset of four right input pairs.

This implies an unpleasant tradeoff between probability of the differential used, and memory required for the attack. Consider the following numbers, which describe the impact of using lower-probability differentials on the difficulty of the attack. These numbers are for parameters that give the attack an approximate probability of success of 1/2.

Memory vs. Probability Tradeoff

Prob. of Characteristic	Num. Right Input Pairs Required	Num. Entries in Sorted List per Key Guess
0.9	5	5
0.5	9	126
0.1	47	178365
0.01	467	1.96×10^9
0.001	≈ 5000	2.60×10^{13}

As a rule, multiplying the number of entries in the sorted list per key guess by N multiplies the size of that list by N, which multiplies the work involved in handling it by $N \log N$. We thus have great difficulty in using differentials with very low probabilities.

Truncated Differentials with Substantially Lower Probabilities in the Decryption Direction Normal differentials must have the same probability in both directions. (This can be established by a simple counting argument.) However, truncated differentials, which don't specify the whole difference, can have different probabilities in different directions. For example, in the MARS forward core rounds, the following four-round differential has probability one:

$$(0, 0, 0, 2^{31}) \rightarrow (?, ?, (\text{low 12 bits} = 0x1000), 2^{12})$$

However, this truncated differential cannot be run backwards with reasonable probability. There are about $2^{211} + 2^{127}$ pairs of inputs that will yield this output difference; of these pairs, only about 2^{-84} have input difference $(0, 0, 0, 2^{31})$. This makes the attack much more costly; in fact, our best methods to mount the attack in this case allow us to attack the full mixing and four rounds of core, but not five rounds of core.

3.5 Boomerang Meet-in-the-Middle Attacks

We have also considered using the same kind of technique, but with boomerangs [Wag99] (basically, 4-tuples with a differential relationship between all four texts in the middle of the cipher) instead of individual ciphertexts. The problem of detecting when we have the expected boomerangs is difficult; thus far, we have been unable to find a way to do this that isn't far costlier than the rest of the attack can afford.

3.6 Other Techniques

In this section, we have focused on meet-in-the-middle attacks, because these are the most obvious kinds of attacks to consider. However, there are other attacks that might be useful against this kind of reduced-round MARS version. For example, we might guess our way past the pre-addition key and forward mixing layers, and look for a set of text pairs whose properties will show through eight rounds of backward mixing layer. We haven't yet found an effective way to do this for all eight backward mixing rounds, but research is ongoing.

4 Symmetric Reductions of the Cipher

MARS consists of eight rounds each of four kinds of round functions. A natural way to derive a reduced-round version of MARS to analyze is to consider k rounds of each kind. For example, when $k = 2$, we have eight total rounds; when

$k = 3$, twelve; and when $k = 4$, sixteen. Cryptanalysis of such reduced-round versions of the cipher allows us to learn important lessons about how to attack a the general MARS structure.

Our attacks typically work as follows:

1. First, we choose N batches of input pairs, so that one such batch is likely to consist of many pairs that have some differential after the mixing layer.
2. We then exploit some differential property that passes through the core rounds, leaving a detectable differential property somewhere near the end of the cipher.
3. Finally, we guess enough key material at the end to detect the detectable property; the partial key guess that allows us to detect the differential property is the correct one.

4.1 Attacking MARS Symmetrically Reduced to Eight Rounds

When $k = 2$ (eight rounds total), we have a cipher that is obviously not very strong. It is still worthwhile to consider how this might be attacked, in part to help develop techniques for attacking stronger versions. Recall that this cipher consists of the key addition, the first two forward mixing rounds, two forward core rounds, two backward core rounds, the last two backward mixing rounds, and the key subtraction.

For this version, we can simply use one of the meet-in-the-middle attacks discussed in the previous section, since there are only four rounds. However, we can do much better than that.

Our attack works as follows:

1. We choose N batches of eight pairs each, where $N \leq 2^{40}$. The batches will be described below; one batch will have all eight pairs with difference $(0, 0, 0, 2^{31})$ after the forward mixing layer.
2. In the right batch, this passes through the four core rounds with probability one, leaving $(?, ?, ?, 2^{12})$.
3. We guess a few bits of subtraction key at the end of the cipher, and thus distinguish the right batch from all the wrong batches. If we guess m bits of effective key, we will need about 2^{m+44} partial decryptions to distinguish the right batch from the wrong batches.

Choosing the Batches The first step to this attack is to get pairs of texts through two forward mixing rounds, so that we have pairs with the difference $(0, 0, 0, 2^{31})$ in the input to the first core round.

Our plan is to put a 2^7 difference in A, and an offsetting difference T in B, and finally a difference to cancel A's difference in D. To simplify the filtering problem at the end, we will choose *batches* of eight pairs of texts, so that one of the batches will give us eight right pairs through the forward mixing layer.

Choosing A, A^* We show the first difference as being 2^7, in A, which passes through the key addition with approximate probability $1/2$, and then generates expected difference T in the output to the first use of S-box s_0 with probability 2^{-7}. This then has probability of 2^{-8}. This is based on simply looking for a pair of S-box inputs, $(u, u \oplus 2^7)$, such that $s_0[u] \oplus s_0[u \oplus 2^7] = T$. We look at all 128 such pairs, and use the difference with the lowest weight in its low 31 bits, for reasons that will become clear momentarily.

For each batch, we hold the low eight bits of A constant. For one such value, $A, A^* = A \oplus 2^7$ will leave a 2^7 difference in A, and a T difference in the output from the first s_0.

Choosing B, B^* The second difference is shown as being T, in B. This passes through the key addition with approximate probability $2^{-w(T)}$, where $w(T)$ is the Hamming weight of the low 31 bits of T. If we get T as the XOR differences in both line B and the s_0 output from A, they cancel out with probability one. (We can also consider a mod 2^{32} difference T that passes through the key addition with probability one, and cancels out the T additive difference in the s_0 output with probability about 2^{-w}; naturally, there is no difference in the probabilities involved.) Recall that we chose $u, u \oplus 2^7$ in A to minimize $w(T)$. Let us assume that the minimum value for $w(T)$ is 12. Then, we have about 2^{-12} probability of finding a pair B, B^* such that their difference after the key addition is T, simply by using the rule that $B^* = B \oplus T$. We can actually do somewhat better than this in our selection of batches.

Building the Batches The third difference is shown as being 2^{31}, in D. This difference passes through all XORs and additions with probability one.

We can thus build batches of $(A, B, C, D), (A^*, B^*, C, D^*)$ pairs. Each batch of eight pairs contains the same low-order eight bits for A and all the same bits for B. There are thus $2^{24} \times 2^{32} \times 2^{32} = 2^{88}$ possible pairs for each batch, and there are 2^{40} batches possible, and about $2^{12} \times 2^8 = 2^{20}$ expected to be necessary.

We build 2^{20} batches of eight pairs, for a total of 2^{24} chosen plaintexts.

Guessing Key at the End After the core rounds, input pairs with difference $(0, 0, 0, 2^{31})$ must have output difference $(?, ?, ?, 2^{12})$. We must thus learn the value of D in the output from the core rounds. To do this, we must guess about 12 bits of the subtractive key for C, and all of the subtractive key for D. Using these guesses, we can derive the values for D after the core rounds. We must do this for all 2^{24} texts. We thus have total work of about $2^{24} \times 2^{44} = 2^{68}$ partial decryptions. (We suspect that there are better attacks for $k = 2$, but that these attacks don't generalize for larger k values.)

Summary of Results

Attack On:	MARS Symmetrically Reduced to Eight Rounds
Attack Type:	Differential
Work:	2^{68} partial decryptions
Memory:	2^{29} bytes
Texts:	2^{25} chosen plaintexts

4.2 Extending the Attack to $k = 3$ (Twelve Rounds)

We now consider an attack on MARS symmetrically reduced to 12 rounds. Again, the cipher is obviously not very strong. However, the structure is beginning to add difficulties to the attack. We use a boomerang-amplifier to cover the six core rounds with probability 2^{-96} for each pair of pairs with difference $(0, 0, 0, 2^{31})$ into the core rounds. With about 2^{50} right pairs into the core rounds' input, we expect to see about four right pairs of pairs. These pairs will then pass through six more rounds, and can be detected by examining the whole output blocks from all the texts.

Requesting Inputs We use the same input structure as before, but instead of requesting eight pairs for each batch, we request 2^{50} pairs for each batch.

The Boomerang Amplifier The boomerang-amplifier attack is introduced in [KS00], and is based on the concept of "boomerangs," as described in [Wag99]. The basic idea of the attack involves a property occurring in pairs of pairs of texts.

For the MARS core, we use the batches of input pairs described above to try to find a batch of 2^{50} pairs of texts, all of which will have difference $(0, 0, 0, 2^{31})$ in the input to the first core round. Since there is a probability one differential for the core rounds, $(0, 0, 0, 2^{31}) \rightarrow (2^{31}, 0, 0, 0)$, this means that all 2^{50} pairs of the batch will have difference $(2^{31}, 0, 0, 0)$ after three core rounds.

Consider the set of 2^{50} of these pairs in the batch. We will refer to these pairs as $((W_0, W_0^*), (W_1, W_1^*), ..., (W_i, W_i^*))$ in input to the core rounds, and as $((X_0, X_0^*), (X_1, X_1^*), ..., (X_i, X_i^*))$ after round three. There are about 2^{99} *pairs of pairs*. That is, there are about 2^{99} different ways to choose two of these pairs out of this batch and look at them together; for example, $((X_i, X_i^*), (X_j, X_j^*))$. Now, consider the difference $(0, 0, 0, a)$, where a is unknown. For any pair of texts to have such a difference, they must collide in 96 bits; the difference thus is expected to occur in 2^{-96} of all random pairs of texts. Thus, if X_i, X_j can be considered as a more-or-less random pair of texts (and they apparently can), then the probability that each i, j pair will have this difference is 2^{-96}, and since we have 2^{99} such pairs, we expect about eight pairs X_i, X_j with this difference. However, we know that $X_i \oplus X_i^* = (2^{31}, 0, 0, 0)$ for all i. This lets us algebraically show that when $X_i \oplus X_j = (0, 0, 0, a)$, $X_i^* \oplus X_j^*$ must also equal $(0, 0, 0, a)$. This boomerang structure thus amplifies the effect of the low-probability event, making it detectable, since when this happens we get *two* pairs of texts that follow the truncated differential $(0, 0, 0, a) \rightarrow (b, 0, 0, 0)$ over three rounds.

Distinguishing the Right Key Guess To distinguish the right key guess, we examine the result of trial partial decryption of a whole batch of pairs at a time. Let Y_i, Y_i^* be the results of encrypting input pair W_i, W_i^* through the whole cryptographic core in some batch. We build a list of all the Y_i and Y_i^* values. We then sort this list on its low-order 96 bits. Next, we go through the

list, and for each pair Y_i, Y_j or Y_i, Y_j^* that matches in those last 96 bits, the pair i, j is added to a sorted list of pairs that collided. Finally, we *count* the number of times each i, j appears in the list. When we see two or more instances of the same i, j occurring twice, we are extremely likely to have a correct key guess.

Recall that we expect eight pairs i, j such that $X_i \oplus X_j = X_i^* \oplus X_j^* = (0, 0, 0, a)$ These will inevitably lead to eight pairs i, j such that $Y_i \oplus Y_j = (b, 0, 0, 0)$ and $Y_i^* \oplus Y_j^* = (b', 0, 0, 0)$. (The property works just as well if the collision occurs between X_i and X_j^*, naturally.)

The probability of any given i, j pair having this property after a random permutation has been applied to it is 2^{-192} Since there are 2^{99} pairs in each batch, we expect no such i, j pairs. The probability of seeing two such pairs in a batch (that is, among 2^{99} potential pairs) is about 2^{-186}. We will be examining 2^{20} different batches, each under 2^{128} different keys, so we'll have 2^{148} total batches to examine in this way. So with overwhelming probability, there will be only one partial key guess that will give us two or more such i, j pairs.

Summary of the Attack The attack on 12 rounds ($k = 3$) makes use of a boomerang amplifier. It requires about $2^{20} \times 2^{48} \times 2 = 2^{69}$ texts, about 2^{25} bytes of random-access memory (to hold a batch of texts at a time), and about 2^{73} bytes of sequential memory to store all the ciphertexts so we can apply our guesses to them. The attack also requires about $2^{128} \times 2^{69} = 2^{197}$ partial decryptions, each consisting of about one quarter of the cipher.

Summary of Results

Attack On:	MARS Symmetrically Reduced to 12 Rounds
Attack Type:	Boomerang Amplifier
Work:	2^{197} partial decryptions
Memory:	2^{73} bytes
Texts:	2^{69} chosen plaintexts

5 Conclusions

In this paper, we have developed several new attacks on reduced-round versions of MARS. While none of these attacks is able to break the full cipher, we feel that these results provide valuable insights into the security of MARS. We regard these results as preliminary, and would be unsurprised to see moderate improvements in any of our attacks. However, if major improvements in the results are possible, we expect that they will require new techniques. Below, we describe some ideas for additional attacks on reduced-round MARS variants.

5.1 Lessons from the Analysis

The results in this paper show the overwhelming importance of the strength of the MARS cryptographic core; we can attack the full mixing layers with only five core rounds, a total of 21 rounds, but can currently attack no more than 11 core rounds.

Our results also show how the "wrapper" layers protect the core rounds from attacks that require large numbers of chosen plaintexts or chosen ciphertexts.

Finally, our results demonstrate that, when evaluating a fundamentally new cipher design, it is important to be able to innovate—to develop new techniques to attack the cipher, rather than merely reusing the standard differential and linear attacks. Because MARS is such an unconventional block cipher, we needed to develop new attacks to get very far in our analysis.

5.2 Why This Is Important

The only way we know of actually determining the strength of a cipher is to try to attack it, including reduced-round versions. Proofs of security have proven unreliable; security arguments based on estimates of the best differential and linear characteristics tell us little about what other attacks may be done; design principles that protect against some attacks sometimes allow new attacks in their place; as in Square, where the use of the MDS matrix made differential attacks extremely difficult, while allowing Knudsen's dedicated attack. The history of cryptography is littered with ciphers whose designers were convinced of their security, but whose attackers were not. Without a solid understanding of the security of each of the AES finalists, NIST and the cryptographic community will likely make a final decision on AES based only on performance.

In this paper, we have done some very preliminary analysis of two versions of MARS with reduced rounds. MARS is a complicated enough design that beginning to analyze it involves a significant investment of time (though even conceptually very simple ciphers seem to have much the same property). We hope to see our work spur others to go beyond the very preliminary results in this paper.

5.3 Ideas for Future Attacks

We have spent considerable time trying to get boomerangs to work within meet-in-the-middle attacks. A "boomerang-in-the-middle" attack would go through six rounds for free, and thus would be quite powerful. Similarly, there is a seven-round impossible differential through the core rounds; we are as yet unable to find a way to use either of these ideas in a meet-in-the-middle attack. The underlying problem in the case of the boomerang-in-the-middle attack is that a boomerang 4-tuple can be identified only by considering both input and output simultaneously. We have not been able to find a way around this problem so far. The underlying problem with the impossible differential meet-in-the-middle attack is that we can rule out candidate key guesses only by (again) examining right pairs for both input and output simultaneously. We are still looking for a way around this problem, or for a proof than none exists.

In our differential meet-in-the-middle attacks, we dealt with the mixing layers by simply guessing our way past them. We expect significant improvements to the attacks are possible with more analysis of the mixing layers, particularly in terms of partial guessing of key material. We have spent far more time analyzing

the core rounds than the unkeyed mixing layers, and so this is a good area for further research.

Attacking the symmetrically reduced version of the cipher with $k = 4$ apparently requires a better way of choosing inputs than the input structure we discuss above. We hope to find a better input structure, or a better property to push through all eight core rounds. Previous attempts to attack $k = 4$ have exceeded 2^{256} work, usually due to the huge plaintext requirements.

It may also be worthwhile to attack variants of MARS that cut off in the middle (after 12 or 16 rounds total); we have some ideas in this direction.

Finally, in future work, we hope to examine how the MARS key schedule functions with various reduced-round variants.

6 Acknowledgements

The "extended Twofish team" met for two week-long cryptanalysis retreats during Fall 1999, once in San Jose and again in San Diego. This paper is a result of those collaborations. Our analysis of MARS and Serpent has very much been a team effort, with everybody commenting on all aspects. The authors would like to thank Niels Ferguson, Mike Stay, David Wagner, and Doug Whiting for useful conversations and comments on these attacks, and for the great time we had together. The authors would also like to thank Susan Langford for helpful suggestions on one of the attacks, and Beth Friedman for proofreading the final paper.

References

[ABK98] R. Anderson, E. Biham, and L. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard," NIST AES Proposal, Jun 98.

[Ada98] C. Adams, "The CAST-256 Encryption Algorithm," NIST AES Proposal, Jun 98.

[BBS99] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials," *Advances in Cryptology — EUROCRYPT '99 Proceedings*, Springer-Verlag, 1999, pp. 12–23.

[BCD+98] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas, L. O'Connor, M. Peyravian, D. Safford, and N. Zunic, "MARS — A Candidate Cipher for AES," NIST AES Proposal, Jun 98.

[Bih94] E. Biham, "New Types of Cryptanalytic Attacks Using Related Keys," *Journal of Cryptology*, v. 7, n. 4, 1994, pp. 229–246.

[Bih95] E. Biham, "On Matsui's Linear Cryptanalysis," *Advances in Cryptology — EUROCRYPT '94 Proceedings*, Springer-Verlag, 1995, pp. 398–412.

[BS93] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.

[BW99] A. Biryukov and D. Wagner, "Slide Attacks," *Fast Software Encryption, 6th International Workshop*, Springer-Verlag, 1999.

[Dae95] J. Daemen, "Cipher and Hash Function Design," Ph.D. thesis, Katholieke Universiteit Leuven, Mar 95.

[DR98] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," NIST AES Proposal, Jun 98.

[HKM95] C. Harpes, G. Kramer, and J. Massey, "A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-up Lemma," *Advances in Cryptology — EUROCRYPT '95 Proceedings*, Springer-Verlag, 1995, pp. 24–38.

[HKR+98] C. Hall, J. Kelsey, V. Rijmen, B. Schneier, and D. Wagner, "Cryptanalysis of SPEED," *Selected Areas in Cryptography*, Springer-Verlag, 1998, to appear.

[HM97] C. Harpes and J. Massey, "Partitioning Cryptanalysis," *Fast Software Encryption, 4th International Workshop Proceedings*, Springer-Verlag, 1997, pp. 13–27.

[KM96] L.R. Knudsen and W. Meier, "Improved Differential Attacks on RC5," *Advances in Cryptology — CRYPTO '96*, Springer-Verlag, 1996, pp. 216–228.

[Knu94a] L.R. Knudsen, "Block Ciphers — Analysis, Design, Applications," Ph.D. dissertation, Aarhus University, Nov 1994.

[Knu95b] L.R. Knudsen, "Truncated and Higher Order Differentials," *Fast Software Encryption, 2nd International Workshop Proceedings*, Springer-Verlag, 1995, pp. 196–211.

[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.

[KRR+98] L.R. Knudsen, V. Rijmen, R. Rivest, and M. Robshaw, "On the Design and Security of RC2," *Fast Software Encryption, 5th International Workshop Proceedings*, Springer-Verlag, 1998, pp. 206–221.

[KRW99] L.R. Knudsen, M.B.J. Robshaw, and D. Wagner, "Truncated Differentials in Skipjack," *Advances in Cryptology — CRYPTO '99 Proceedings*, Springer-Verlag, 1999, 99. 165–180.

[KS00] J. Kelsey and B. Schneier, "Initial Cryptanalysis of the MARS Core," draft.

[KSW99] J. Kelsey, B. Schneier, and D. Wagner, "Key Schedule Weaknesses in SAFER+," *The Second Advanced Encryption Standard Candidate Conference*, 1999, pp. 155–167.

[LMM91] X. Lai, J. Massey, and S. Murphy, "Markov Ciphers and Differential Cryptanalysis," *Advances in Cryptology — CRYPTO '91 Proceedings*, Springer-Verlag, 1991, pp. 17–38.

[LH94] S. Langford and M. Hellman, "Differential-Linear Cryptanalysis," *Advances in Cryptology — CRYPTO '94*, Springer-Verlag, 1994.

[Mat94] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *Advances in Cryptology — EUROCRYPT '93 Proceedings*, Springer-Verlag, 1994, pp. 386–397.

[NIST97a] National Institute of Standards and Technology, "Announcing Development of a Federal Information Standard for Advanced Encryption Standard," *Federal Register*, v. 62, n. 1, 2 Jan 1997, pp. 93–94.

[NIST97b] National Institute of Standards and Technology, "Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard (AES)," *Federal Register*, v. 62, n. 117, 12 Sep 1997, pp. 48051–48058.

[NSA98] NSA, "Skipjack and KEA Algorithm Specifications," Version 2.0, National Security Agency, 29 May 1998.

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

[Saa99] M. Saarinen, "A Note Regarding the Hash Function Use of MARS and RC6," available online from `http://www.jyu.fi/ mjos/`, 1999.

[SK96] B. Schneier and J. Kelsey, "Unbalanced Feistel Networks and Block Cipher Design," *Fast Software Encryption, 3rd International Workshop Proceedings*, Springer-Verlag, 1996, pp. 121–144.

[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.

[SKW+99] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *The Twofish Encryption Algorithm: A 128-bit Block Cipher*, John Wiley & Sons, 1999.

[Vau96] S. Vaudenay, "An Experiment on DES Statistical Cryptanalysis," *3rd ACM Conference on Computer and Communications Security*, ACM Press, 1996, pp. 139–147.

[Wag99] D. Wagner, "The Boomerang Attack," *Fast Software Encryption, 6th International Workshop*, Springer-Verlag, 1999, pp. 156–170.