MARS and the AES Selection Criteria

IBM MARS Team

May 15, 2000

Abstract

As the AES selection process enters its final days, it sometimes seems that the discussion has been reduced to a “beauty contest”, with various irrelevant or red herring issues presented as *the differentiating factors* between the five finalists. In this note, we discuss the criteria that should (or should not) serve as the basis for selecting an AES winner, and we compare MARS to the other finalists based on these criteria. Also, we examine several of these “beauty contest” issues that were raised, and demonstrate that when subjected to closer scrutiny they turn out to be meaningless.

1 Security and Robustness

Although everyone seems to agree that security should be the main criterion for selection, what different people see as the implications of this statement vary widely. It is generally agreed that *barring a substantial breakthrough in cryptanalysis, all the finalists are secure*. Therefore, some would argue that we should view all ciphers as secure and concentrate on performance and flexibility issues as the selection criteria. This argument is flawed and we strongly disagree with it. With two substantial cryptanalytic breakthroughs in the last ten years (differential and linear cryptanalysis), betting “The Store” (i.e., the security of the AES) on the assumption that no further cryptanalytic breakthroughs will occur, is risky and possibly even dangerous.

We postulate that the main criterion for the choice of an AES winner is and should be *robustness* against future advances in cryptanalysis. With 128 bit blocks and key lengths up to 256 bits, there is no technological reason why the AES cannot withstand brute force key exhaustion attacks for a very long time (25 years, 50 years, perhaps longer). All five finalists have defended adequately against the known powerful cryptanalytic attacks (including differential and linear cryptanalysis). The single remaining threat is from attacks that have not yet been developed or discovered. NIST did not levy a robustness requirement on the candidate algorithms. It was up to the designers to balance robustness against other algorithm design points (performance, flexibility, complexity, etc.).

We believe that the wisest, most responsible, and most defensible course of action is for NIST to select a cipher that is well-positioned to withstand future advances in cryptanalysis. Such a course will minimize the risk that a new cryptanalytical technique will show a weakness in the AES or render it insecure. It should be stressed that even a perception of a break in the AES can potentially cost billions of dollars and the consequences of a real break would be disastrous.

There are several different views regarding what constitutes robustness. A popular view is to compare the number of rounds in the cipher to the smallest number of rounds for which any attack is known. We feel that this view too is slightly off the mark. For one thing, it stands to benefit ciphers that are less understood, or have a steeper learning curve (since for such ciphers, it may take longer to devise attacks against a large number of rounds). Even worse, it does not take into account the notion of *minimizing the trust in any single component*, which is a central pillar in any security design.

To illustrate the last point, consider the RC6 cipher. It is a very simple and fast cipher, whose cryptographic strength is based on the power of data-dependent rotations. A common criticism against it is that twenty rounds may not be sufficient, as theoretical attacks exist against RC6 with 15 rounds. It should be noted, however, that more than five years after RC5 brought data-dependent rotation to center stage, we still do not have a good "handle" on analyzing it. Essentially, all we can currently use in its analysis is the trivial observation that with some probability no rotation takes place (so we can sometimes ignore this operation altogether).

Moreover, it seems clear that RC6 will remain secure as long as this is the only tool available for cryptanalysis. On the other hand, if a significantly better tool is discovered, then there is essentially no way of gauging the number of rounds that will be needed to protect RC6 against it. Viewed in this light, a 24-round RC6 seems just as vulnerable as a 20-round RC6. This can be compared to MARS, whose cryptographic strength is also based on the power of data-dependent rotations. However, due to the additional mechanisms in MARS, a major advance in the analysis of that operation is likely to be much less devastating to MARS than to RC6.

Everyone's expectation is that the AES will be implemented in many different applications and products, both current and future. New national and international standards will be based on the AES and elsewhere it will become a de facto standard. In large measure, the security of the Internet will be predicated on the assumed strength of the AES.

Undoubtedly, NIST will receive criticism no matter which AES candidate algorithm is selected – "it's too slow" or "it's too complex." or "it's not optimal for my application or environment." But, these minor complaints and criticisms will be forgotten provided that the AES is secure and it "gets the job done." However, quite the opposite would be true if the AES should happen to be crippled or broken. The cost to industry and governments (and the damage to NIST's credibility) would be immeasurable. Thus, algorithm robustness must be the first criterion for selecting the AES winner. To this end, not only should NIST document the rationale for selecting the AES winner (over the other candidates), but it should be prepared to defend this rationale in the event that new attacks are found against the AES.

2 The MARS design philosophy

The MARS design philosophy was to set the highest security and robustness goals, while maintaining a fast and flexible cipher. The principles behind our design were as follows:

- *Do not trust any single component of the cipher*, not even the components that we believe to be strong. For example, at one point we had a design for an E-function that was based only on the data-dependent rotation. Although we could see no weaknesses in that design, we felt that it was prudent to add the S-box, so as not to rely on a single operation.

  Similarly, although the MARS core is a very strong design, we added also the mixing layers to give the cipher extra protection against future advances in cryptanalysis. This way, the robustness of the cipher is based on many "fail-stop" mechanisms, not just on the number of rounds.

- *Design an easy-to-analyze cipher*. An important goal of the design was to be able to analyze the cipher. For example, this was the reason that we chose the "target heavy" unbalanced Feistel structure: since the components (eg., the E-function) are relatively small, it is easier to analyze them. This choice is what allowed us to present not only a concrete analysis of the cipher, but also a few "lower bounds", eliminating some classes of potential attacks.

Another consequence of this "design for analysis" is that we deliberately avoided using "nice little tricks" whose security consequences are not immediately clear. For example, the S-box was generated via a "pseudorandom process", rather than building it as a combination of smaller boxes (as it is done in Twofish, for example), because we felt that using a "more sophisticated" process may introduce some weaknesses.[1]

- *Produce a fast cipher* (within the design parameters outlined above). With all its additional "fail stop" mechanisms, MARS is still a very fast cipher. When we designed the additional mechanisms, we made sure that it is still possible to get a very fast implementation of the cipher in the most common use (i.e., software) environment, and an additional 10x speedup when using dedicated hardware.

2.1 Security of MARS

We think that our design choices in MARS [2] resulted in a cipher that is not only secure by today's standards, but is extremely unlikely ever to be broken. With respect to cryptanalysis of MARS that was done to date, we have the following remarks:

- When we designed MARS, we knew that for this type of unbalanced Feistel network, up to nine rounds could be distinguished from a random permutation, even when the "E-function" is an ideal random function. (This was described in Jutla's paper in Crypto'98 [8].) This was one of the reasons behind our decision on 16 rounds of core.

  The recent works of Biham and Furman [1] and Kelsey et al. [9] show more efficient ways of distinguishing 8 to 8½ rounds of the MARS core from a random permutation (and then guessing the keys in subsequent rounds to get an attack against 10-11 core rounds). Jutla's original techniques as well as the newer techniques, do not need to send differentials through many rounds of the cipher, and instead use some "border conditions". It is rather clear that these techniques "run out of steam" after about 9 rounds. Any further advancement would have to stem either from a major flaw in the E-function itself, or from an entirely new attack technique.

- In addition to specific attempts of attacks, the modular structure of MARS allows us to show some crude lower bounds on several classes of attacks. It is important to understand that although such bounds cannot prove that attacks are impossible, they are very useful in clarifying which lines of attacks may potentially be useful against the cipher and which lines are "doomed".

  Recently, Robshaw and Yin presented some comments on the linear analysis lower bounds in the MARS submission document [12]. In particular, they claimed that the arguments in that analysis can only show a bound of 2^{-49} (rather than the claimed 2^{-69}) on the bias of any "straightforward linear approximation" of the MARS core. Some of their criticism is the result of poor wording of the argument in our original paper, leading to their misinterpretation of our intent. Other parts represent some "slightly more sophisticated" approximations that were not covered by the original bound. In a separate note [3], we give a more detailed description of the argument, and show that even with the "more sophisticated" approximations, we can still show a bound of 2^{-61}, which means that any such analysis must essentially use the entire plaintext-ciphertext codebook. In other words, such attacks are not feasible.

[1] The Twofish team chose to use such "nice little tricks" and rely on "over one thousand man hours" of cryptanalysis to ensure that this does not introduce weaknesses. Given the relatively short review process for the AES, we felt that it is more prudent to avoid this altogether.

Therefore, both our analysis and the analyses done by others confirm that with today's technology, *even the MARS core by itself is a secure cipher*. The robustness of MARS is evident in that there is an entire component (i.e., the mixing part), which is meant primarily to "future proof" the cipher against unforeseen new attacks.

3 Common Misconceptions

Throughout the AES process, and especially in the last few weeks, there were several claims that MARS (and RC6) are "not suitable for implementation in environment X". These claims are sometimes so ridiculous, that it is hard to understand how they can be taken seriously. Nonetheless, they are repeated quite often these days, and it seems that at least some people are "buying them". Below we therefore take the time to examine these claims, and show that they have very little to do with reality.

3.1 Misconception 1: MARS is not suitable for Hardware implementation

Recent presentations have given the impression that MARS (and RC6) are hard to implement in hardware. Recent commentary in some newsletter went as far as proclaiming that "MARS is so bad in hardware that it would be a disaster for Internet applications, and RC6 is close" [13]. Others claimed that MARS uses too many circuits, or too much die area.

Such claims are highly misleading. Certainly MARS requires more hardware than the other candidates, but at AES3, a hardware design for MARS was presented (using today's shipping technology) that is certainly fast enough even for high-end applications [11]. This design uses only 13.8K gates to achieve up to 1.28 Gbit/sec throughput. This amount of circuitry is so small, that even the smallest chip that can be manufactured using IBM's existing manufacturing lines would have to include at least ten copies of this design (!), for a total throughput over 12 Gbit/sec., at a quantity cost of roughly $13 per chip. One dollar per Gbit/sec is simply not an issue to consider, particularly since Gbit routers or switches that would use these chips typically cost in the $50,000 range. This implementation is with ***today's*** technology; tomorrow's technologies will only make the issue even more laughable. Moreover, in many applications a custom chip would not be built, rather, this logic would be added to existing chips, making the added cost negligible, it's effectively free.

We conclude that statements such as above (about MARS not being suitable for hardware implementation) are completely false. They represent either ignorance regarding hardware costs and sizes, or an attempt to move the focus from where it belongs: security of the cipher.

3.2 Misconception 2: MARS is not suitable for low-end smart cards

Throughout the AES process, the issue of low-end smart cards has been used as a "moving target", in an attempt to disqualify MARS (and RC6) as needing too much RAM for the expanded keys. (First it was claimed that a cipher has to fit in 128 bytes of memory, then the threshold was lowered to about 64 bytes, etc.) The original MARS design included expanded keys that took 160 bytes to store, and an accepted "tweak" to the MARS key setup makes it possible to store only 40 bytes of expanded keys at a time. Even the smallest smart cards can support MARS in this mode. Still, even at AES3, one presentation [10] claimed that MARS requires 512 bytes RAM for key storage, which is just plain wrong.

In addition, no serious system security design will place the AES on a very low-end smart card, since these are extremely vulnerable to attacks (e.g., power attacks), as was demonstrated in [4]. Security conscious applications are rapidly migrating to public key based cards to avoid the serious system vulnerabilities inherent in symmetric key based designs. This trend will clearly accelerate as chip costs inevitably decline.

For these reasons, smart card suitability is simply not an issue for any of the AES finalists.

3.3 Misconception 3: MARS is not key-agile

Several arguments were put forth that key agility is an important criterion in the selection of the AES algorithm. Some presenters claimed that MARS (and RC6) have terrible key agility in hardware implementations. One zealous comment went as far as declaring that "MARS subkey generation is atrocious from an agility viewpoint". Once again, this is a ridiculous assertion. It was suggested that in extreme cases, a high-speed network switch will have to maintain about a half million contexts, and to switch between them every four block encryptions. But even for this extreme environment the key agility issue does not pose any problem for MARS. To see this, note the following:

- Even with a half million contexts, keeping the entire MARS expanded key in memory only adds about 70Mbyte of memory above what is needed for keeping 3DES keys. Hence, one additional standard 128Mbyte memory-card on the server is more than enough to do the job. We doubt very much that adding this one card would have a noticeable effect on the price of a configuration for a high-speed network switch.
- One should notice that the major requirement from architectures that need key-agility is high throughput, not low latency. Hence, one can use additional hardware to completely hide the cost of the key setup procedure. For example, instead of putting 10 MARS engines on the $13 chip from above, you can put 10 key-setup engines for each encryption engine. Hence you could still have the raw encryption throughput, even if the hardware key-setup takes 10 times longer than block encryption. With the silicon real estate being extremely small (for all the finalists), adding extra silicon to eliminate cycles is practically free.

We also note that the MARS key setup offers a tradeoff between memory and speed. Specifically, it is shown in [7] that with additional storage of only about 25 bytes per key, one can reduce the key setup time to only 2-3 block encryptions, and with additional 60 bytes per key you can get it down to one block encryption or less.

3.4 Misconception 4: MARS is not suitable for FPGA implementation

Given the low cost of custom high-performance MARS chips, we are curious as to why FPGA implementations would be considered. Performance and price are not issues, given the $13 custom chip mentioned earlier. (One argument made is that FPGA's provide the ability to modify the algorithm. This is a terrible "feature" from a security perspective, as it introduces the ability of the hacker to make malicious alterations to the crypto engine.) Given the performance of software and hardware alternatives, and the security issues associated with "agile" crypto implementations, we feel that the FPGA issues should not be considered for AES selection.

Regarding the "suitability" of MARS, even from the presentations in AES3 it is clear that MARS is implementable in FPGA. Moreover, one of these presentations frankly states that "Further optimizations of the Mars implementation are certainly possible, but would require the higher development effort …" [6]. As is the case for ASIC, it is certainly true that a MARS FPGA implementation would be more involved than the other candidates. However, as for ASIC, it is clear that a careful design would result in an implementation that is more than good enough for all practical purposes.

3.5 Summary

In this section we have demonstrated that MARS is in fact suitable for various implementations, and has reasonable price-performance characteristics, in any realistic setting. It is clear MARS is more expensive than other finalists in some environments, but it is workable in any setting, and for the "price" of added complexity you get a cipher with unique robustness properties.

It is our feeling that the common misconceptions about MARS's "suitability" are the result of several misleading statements (such as the ones quoted above), which for some reason were accepted by many without questioning their validity. Moreover, it seems that these statements produced an attitude, that "MARS is so complex that we shouldn't even invest efforts in looking at it carefully and implementing it". Such an attitude is evident in several evaluations that were presented at AES3. (For example, the FPGA report in [5] did not implement MARS, the assertion about the 512 byte RAM requirement in [10], and others.) This misconception is partly our fault, for not responding earlier to the "negative campaigning" against MARS. We hope to correct some of it with this report.

4 Comparison of the five AES candidates

In the previous section we demonstrated that the claims that MARS "is not suitable" for some real-life environments are baseless. In fact, from the point of view of speed and flexibility, each of the finalists is suitable for every environment. The criteria for selection should first and foremost be the security and robustness of the ciphers. Below we examine the five finalists, pointing out what we see as their strengths and limitations.

MARS. The main strength of MARS is its robustness. This was the main design goal, and MARS contains more "fail stop" mechanisms than any of the other finalists. Due to the heterogeneous structure and the large variety of "strong operations" in MARS, even a major advance in the cryptanalysis of any one of its components is very unlikely to lead to a significant attack against the overall cipher.

MARS is also a very fast cipher in common use environments (i.e., in software). The large number of fail-stop mechanisms in MARS makes its hardware implementation more involved than the other finalists, but as we explained above, it is still very small and cheap to implement in hardware, and is suitable to any real-life environment.

RC6. The main advantage of RC6 is its simplicity and speed. Its author, Ron Rivest, enjoys a well-deserved reputation in the cryptographic community, based on carefully crafted ciphers such as RSA, RC2, RC4, and RC5, which may serve as an indication for the suitability of the current design as well. The main argument against RC6 is "single point of failure" design. There are also lingering concerns regarding the number of rounds used in RC6.

Rijndael. It is a fast, flexible and elegant cipher. Rijndael is somewhat similar to SQUARE, and the lessons from SQUARE are incorporated in its design. The main worry about Rijndael is that it may not be conservative enough. Moreover, this style of design (and its analysis) has been around for less than five years - so a major advance in its analysis may be more likely than for the other ciphers.

Serpent. The main selling point of Serpent is its very conservative number of rounds. Serpent does not have "fail-stop" mechanisms as in MARS, so in principle it is possible that a single major advance in cryptanalysis would yield a damaging attack. However, the large number of rounds makes such a possibility extremely remote. The authors of Serpent include Eli Biham and

Lars Knudsen, two of the leading experts in cryptanalysis, which can be viewed as an indication to its strength.

Serpent’s main drawback is that it is slower than the other finalists in software. On the other hand, it is very fast in hardware.

Twofish. This cipher was designed for flexibility, and indeed it offers a wide variety of implementation tradeoffs. It is also a very fast cipher. However, the same design for flexibility also resulted in a cipher which is very hard to analyze. To obtain flexibility, the designers used many “tricks”, whose security implications are not clear. The result is that among the five finalists, Twofish, by far, has the steepest learning curve.[2] (In fact, even the designers’ analysis turned out to be incorrect, despite “over one thousand man hours” of cryptanalysis.)

Given the relatively short selection process for the AES and the steep learning curve of Twofish, it seems that the actual security of this cipher is still a big question mark.

4.1 Conclusions

It is our opinion that robustness is the most important selection criterion of the AES. From this perspective, we believe that only MARS and Serpent are sufficiently robust against future analysis. Of these, only MARS has robustness both in the number of rounds and in the redundancy of structure and operations. Hence, we think that MARS is the best choice for the AES.

For reasons of simplicity and interoperability, we do not believe implementors should be forced to include multiple winning AES algorithms. Should NIST decide to name two algorithms, we believe that implementation of both algorithms should be optional. The selection criteria for a second algorithm should also be robustness. Serpent would make a good second choice, as its large number of rounds also appears to give more robustness than the remaining three candidates.

References

[1] E. Biham, and V. Furman. “Impossible Differential on 8-Round MARS' Core”. Presented in the 3rd AES conference, NY, USA, April 2000.

[2] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas Jr., L. O'Connor, M. Peyravian, D. Safford and N. Zunic, “MARS - a candidate cipher for AES”. Presented in the 1st AES conference, CA, USA, August 1998.

[3] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas Jr., M. Peyravian, D. Safford and N. Zunic, “Comments on MARS’s linear analysis”. Submitted as a formal comment to NIST.

[4] S. Chari, C. Jutla, J.R. Rao, and P. Rohatgi, “A Cautionary Note Regarding Evaluation of AES Candidates on Smart Cards”. Presented in the 2nd AES conference, Rome, Italy, March 1999.

[2] Like MARS, twofish is also built using a modular design. But in MARS, the components were combined in a way that facilitates analysis (since ease of analysis was one of the design goals). In Twofish, on the other hand, the components were combined to achieve implementation flexibility. As a result, the modular design of Twofish hinders analysis, rather than facilitate it.

[5] A.J. Elbirt, W. Yip, B. Chetwynd, and C. Paar, "An FPGA implementation and performance of the AES block cipher candidate algorithm finalists". Presented in the 3rd AES conference, NY, USA, April 2000.

[6] K. Gaj, and P. Chodowiec, "Comparison of the hardware performance of the AES candidates using configurable hardware". Presented in the 3rd AES conference, NY, USA, April 2000.

[7] S. Halevi. "Key agility in MARS". Submitted as a formal comment to NIST.

[8] C. Jutla, "Generalized Birthday Attacks on Unbalanced Feistel Networks", in proceedings of Crypto'98, LNCS vol. 1462 pp. 186-199. Springer-Verlag, 1998.

[9] J. Kelsey, T. Kohno, and B. Schneier. "Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent". Presented in the Fast Software Encryption Workshop, NY, USA, April 2000.

[10] F. Sano, M. Koike, S. Kawamura, and M. Shiba, "Performance Evaluation of AES Finalists on the High-End Smart Card", Presented in the 3rd AES conference, NY, USA, April 2000.

[11] A. Satoh, N. Ooba, K. Takano, and E. D'Avignon, "High-Speed MARS Hardware", Presented in the 3rd AES conference, NY, USA, April 2000.

[12] M. Robshaw and Y.L. Yin. Potential flaws in the conjectured resistance of MARS to linear cryptanalysis. Submitted as a formal comment to NIST.

[13] B. Schneier, 15 April 2000 Cryptogram (`http://www.counterpane.com/crypto-gram-0004.html`)