AES and Future Resiliency: More Thoughts And Questions

By Don B. Johnson
djohnson@certicom.com
March 10, 2000

Introduction

In a paper submitted previously, the author argued that a new AES evaluation criterion of future resiliency be added, defined as the ability to respond to the uncertain future and that this criterion could best be met by NIST selecting multiple disparate AES winners. This paper continues that discussion by providing more rationale. It also asks some questions and explores possible outcomes of the AES process.

Summary From Previous Paper

The author's previous paper closed with this summary:
"NIST should carefully examine the various classification schemes that have been made and endeavor to choose the AES second round finalist candidates considering that it is a worthwhile goal to try to ensure that differing design approaches are included. This is because of reasons of future resiliency, extending cryptographic knowledge, Super AES, crypto toolbox philosophy, possible patent complications, target diffusion, avoidance of artificial tiebreakers, recognition of the problem being multidimensional with imperfect information, and the constraints of other standards organizations. That is, in selecting the handful of AES second round finalists, disparity of design approaches is to be desired over conformity."

For further explanation of these rationales, see the paper at the NIST AES website at www.nist.gov/aes. These rationales continue to be valid in the discussion regarding whether there should be multiple AES winners or not.

NIST is to be congratulated for their selection of AES finalists as they do represent a disparate selection from among the submitted AES candidates. Furthermore, the inventors of each finalist algorithm represent a significant portion of the skills in the cryptographic community. As the AES winner(s) must give a royalty-free license (if the algorithm is patented), perhaps the main rationale to participate in

the AES process is the recognition one receives, with the winner(s) getting major “bragging rights.” Another way to look at the NIST AES finalist selection is that NIST has put “five cats in a bag” to see who survives as each submitting group is highly motivated to find chinks in the armor of the other AES finalist algorithms. Better to find out now rather than later.

I) Additional Rationale for Multiple Algorithms

Space Probe Scenario

A reason to consider multiple winners is that sometimes one needs to use hardware for performance reasons, but the hardware is difficult or impossible to change once deployed. Consider a commercial space probe [JC]. Once it arrives at its destination, it must be essentially self-sufficient. Calling it back is out of the question. However, backup circuitry is a normal part of its design and this flexibility could be extended to include a backup symmetric cryptographic algorithm. As these types of projects might take years or decades, such an algorithm backup is simply prudent.

AES Selection Time

Another factor that should be considered by NIST is the amount of time that was taken by the AES process. If a sole AES winner were to prove unfortunate for some reason, then it could take many years to determine a substitute. It has been said that three months is considered an Internet year. The time needed to do another AES process may not meet the requirements of the market.

Infrastructure Overoptimization

As we saw with the deployment of DES, the selection of one algorithm by NIST meant that best-practices resulted in the use of that **one** algorithm. For much of the life of DES, there was no pressing need for vendors to try to design systems to support multiple symmetric cryptographic algorithms, DES was it. With DES the only choice, this simplified things for a vendor. However, we see today that this simplification resulted in a deployed infrastructure where there are concerns that some portions are now vulnerable to a determined attack.

Einstein is reputed to have said, “One should try to make things as simple as possible, but no simpler.” Even the selection by NIST of as few as two winners will mean that vendors will need to design in flexibility of algorithm choice in some products and provide for the possibility of algorithm replacement in others, rather than overoptimize as was done in the case of DES.

NIST as AES Architect

NIST is overseeing the AES process. As such, NIST is the architect of the AES process, that is, it is creating the AES design architecture. There are two fundamental responsibilities of an architect, as follows:

1) Specify **enough** detail to allow others to proceed.
2) Know what **not to specify** to allow creativity and flexibility in others.

In this AES architect role, NIST should follow the general principle of “If in doubt, don’t.” NIST/NSA can and should make “apparent health” statements on the security of the AES finalists. NIST can and should make decisions about which AES finalist algorithms are suitable for government use, using whatever additional criteria (if any) besides security that NIST deems appropriate. This is ALL that NIST should try to do. NIST should resist the temptation to try to solve potential challenges resulting from the existence of multiple algorithms, such as the need to negotiate algorithms or the need for a vendor or market segment to select the most appropriate algorithm.

NIST or the Marketplace?

Asking NIST to select a sole AES winner means that one believes this decision is appropriate for top-down decision-making, as in a command economy or an army. A top-down methodology is appropriate when any decision is better than no decision (e.g., traffic lights) or when a decision must be made quickly (e.g., a battle). However, simply as a matter of information flow, all the relevant information cannot be expected to be available to the responsible top-level decisionmaker. The marketplace (bottom-up decision-making) has been shown to be much more responsive and adaptable than a command economy. This is because each economic entity or group of entities makes decisions based on its own information and needs.

So one question that NIST needs to ask itself is does it see the AES process (that is, the development of commercially-appropriate symmetric cipher or ciphers) as needing a top-down decision to be made or does it believe that the marketplace is the most appropriate place for this decision to be made. The marketplace has a way of determining what is appropriate; if there is truly one finalist that is superior in many ways, it does not need NIST's selection of it as the winner to emerge as the winner in the marketplace. However, there is a real concern that NIST could make a suboptimal choice due to insufficient information. In this case, "hands off" is the wisest course of action.

NIST needs to resist the temptation to make a decision in an area beyond their (or anyone's) competence. The round 2 discussion issues asking questions about how to assess speed versus security margin, need for low-end flexibility, and hardware versus software performance indicate that NIST recognizes its lack of certainty in these areas. This is fundamentally because there are no obviously single correct answers to these questions. Different applications may require different answers to these questions. NIST should make a virtue of its (really, everyone's) ignorance and not attempt to decide these unanswerable questions one way or the other, but let others make each decision that is most appropriate for them.

Bias?

It may not be politically correct to say so, but NIST should understand that any counsel given it might be biased; this might be especially true of counsel from submitters of algorithms. This is not necessarily a bad thing, the submitters of the AES finalists have very high cryptographic skills and it is certain that the submitters made their decisions after thinking long and hard about the problem. It is just that each submitter naturally thinks their beliefs are correct.

For example, it would be no surprise that a designer of a very flexible algorithm might think that flexibility is an important AES criterion. That is likely one of the reasons the submitted algorithm was made flexible in the first place, so that it would have an advantage when compared with other AES candidates.

The point is that if NIST were to announce they are seeking a single winner then this (in turn) results in a ranking of finalists, just as identifying any other AES criterion as critically important would also potentially rank the finalists. However, note that if an algorithm is truly more flexible than another, it still stands a greater chance of being used in the "marketplace" selection process mentioned above. That is, any advantages of an algorithm remain advantages; by selecting multiple winners and relying on the marketplace, NIST is not required to try to determine which advantages are more important than others.

II) Some Questions

Quantum Computers

One big question regarding the future is whether or not quantum computers are feasible and if they are, what effect they will have on cryptography. An arbitrary bitsize quantum computer (assuming it can be built) allows a square root attack on a symmetric cipher. The possibility of this provides some justification for the larger AES keysizes; a 256-bit symmetric cipher would take 2**128 quantum operations to exhaust the key space.

However, an interesting question is whether there is some limit in practice to the number of bits of a quantum computer. Many researchers suspect this is the case, that quantum decoherence will prove insurmountable for some number of quantum bits.

In terms of AES this question becomes: if one can only build an x-bit quantum computer, how much does this help in attacking each AES finalist? As all block ciphers are composed of smaller chunks, how might these chunks interact with a quantum computer? This possibility can be termed a partial quantum attack. And of course, an adversary could construct many quantum computers to run the attack in parallel, assuming this would help. So the question is: "How does a parallel partial quantum computer affect the ability to attack the AES finalists?"

As an example, DES is composed of a 56-bit key. A 56-bit quantum computer should be able to attack the DES. However, the DES design is such that each of the sixteen rounds uses a 48-bit key. This

suggests the possibility that a 48-bit quantum computer might somehow be able to be used to successfully attack the DES. The question of how the AES finalists stack up in relation to parallel partial quantum computers is a critical question to be answered. NIST should step up to this analysis if it is not forthcoming from the research community. No final AES decision should be made without some exploration of the expected effects of this possibility.

Random Cipher

It is clear that a random cipher for a certain blocksize is the unrealizable ideal. This is a cipher that selects a random choice for the output block for each input block, the key providing an index into a set of random selections. There is no structure that is able to be attacked by an adversary. The best attack is key exhaustion, which is the goal of any symmetric cipher. It is also clear that such a ideal block cipher is totally impractical as the space needed is totally infeasible. However, one would like any particular block cipher to "appear" to be ideal to an adversary. That is, even though the structure is known to an adversary, this structure does not allow any shortcuts to be made. A critical question is whether a AES finalist appears "random." There are many established randomness tests. Any deviation from random is a cause for concern.

Another related important question is at what point do degenerate forms of a finalist not appear random. For example, a finalist may have 20 rounds. It is important to know if the output after 4 rounds appears random or if it takes 8 or even 16 rounds. This is important as it gives an indication of the margin of safety built into the cipher. It is obvious that a round of cipher A cannot be considered equivalent to a round of cipher B but this type of analysis allows one to at least map some internals of one algorithm to another for comparison purposes.

Knowing what to do with this analysis is more problematical. Regardless, this is an important data point. If I know that cipher A is essentially as fast as cipher B, but that cipher A results in random-appearing output after 5 of 16 rounds and cipher B results in random-appearing output after 8 of 12 rounds, then cipher A may be the more conservative choice in some sense. But NIST should be wary of this analysis, one can simply add more rounds at a performance cost.

Should a cipher be rewarded (or penalized) for minimizing overhead? Should a cipher be rewarded (or penalized) if it has “more” rounds? This means that (apparent) security and performance are very closely tied together.

Combined Attacks

In the real world, the adversary is able to combine the effects of various attacks. Even if each attack results in only a relatively small advantage that is not relevant when considered by itself, a combination of attacks may accumulate to result in a feasible attack. For this reason, any discovered theoretical advantage for an adversary attacking an AES finalist (no matter how apparently small) is a concern.

III) Thoughts on the AES Finalists

Following are some thoughts on the AES finalists. It should be recognized that these ideas are tentative and subject to improvement and correction. Of course, the detection of any security flaw in a finalist would have a major impact. Each finalist algorithm can be seen as a statement by the designers regarding not only one way to solve the various tradeoffs of the AES puzzle, but also as how the designers see the future. It is hoped that these thoughts on the finalists are used by NIST in the spirit in which they are given, as food for thought.

MARS

MARS was designed with some thought to try to avoid potential future attacks, especially in its heterogeneous structure, a keyed-core surrounded by unkeyed forwards- and backwards-mixing functions. The unkeyed mixing functions cost time and space, but their inclusion seemed prudent to the designers and worth the cost. The core “mixing” function uses addition, multiplication, fixed and data-dependent rotations, and an S-Box (straightforward substitution cipher). The designers responded to criticism to improve the performance of MARS by using the “tweak” allowed by NIST.

From a perspective on the future, the designers of MARS believed the best way to handle uncertainty was to use many different techniques using a cost/benefit analysis. The MARS design is the

most different of the Feistel cipher finalists. Another way to look at MARS is that IBM is a large organization which had many people with good ideas trying to get them incorporated into the IBM submission. This can be seen in the number of authors of the MARS paper.

From a perspective of future resiliency, the inventors of MARS thought that a heterogeneous structure was important.

RC6

RC6 was built from a heritage of RC5 and was designed to be fast and simple to describe. The core ideas of RC6 came from RC5, which was designed by one person, as such it represents a unity of design approach. In many scenarios, RC6 is the fastest AES finalist. The pseudocode for RC6 is very straightforward with basic operations defined on 32-bit words; the RC6 pseudocode is the shortest of all finalists. It uses addition, multiplication, data-dependent rotations and substitution to do the cryptographic "mixing." RC6 can be seen as an example of building a performance-optimized cipher on the idea of data-dependent rotations. The challenge for the designers of RC6 is to show that their design is not **too simple**. For example, comparing RC6 to MARS, MARS adds more complexity to its specification to try to provide more mixing.

Indeed, the "Correlations in RC6" paper by Knudsen and Meier (available at www.nist.gov/aes) indicate that reduced rounds of RC6 do not appear random. The observation by Saarinen in the NIST RC6 forum on finding "almost equivalent" keys in RC6 suggests other possible concerns. These ideas hint that RC6 may be on the edge of security.

From a future resiliency perspective, the designers of RC6 believed that parameterization was paramount. In this way, if a certain number of rounds was found to be weak, this number could be adjusted upwards.

Rijndael

Rijndael does not use a Feistel structure, rather it uses a matrix structure where the cryptographic mixing involves byte substitution, row shifting and column multiplication. Rijndael has the most

different structure when compared with the other AES finalists. It can be implemented using byte operations and is therefore very flexible.

From a future resiliency perspective, the designers of Rijndael were willing to go in new directions and wanted high flexibility in implementation.

Serpent

Serpent is a conservative design and deliberately tries to build on the vast amount of information relating to DES. Serpent is also the **slowest** of the five AES finalists on most platforms. Being the slowest, the challenge for the designers of Serpent is to try to show how the other finalist algorithms cut corners in ways that Serpent did not (that is, the additional performance cost should be justified). For example, suppose that NIST gave all five AES finalists "certificates of apparent security," it is not clear what symmetric algorithm niche would best be filled by use of Serpent, as opposed to one of the other finalists. Of course, a specific implementation might find that Serpent is the fastest method, if the instructions it uses are fast and the instructions that other methods use are slow.

The designer's of Serpent have presented an "equivalent rounds" analysis of the AES candidates and tried to show how Serpent uses more rounds than might be thought needed as a safety margin. Yet the designers did not officially change the specification of Serpent (even though they knew that there were many other faster AES candidates) so they must believe they have good reasons for designing it as they did. Serpent and RC6 appear to have opposite design philosophies in this area of tradeoff between security margin and performance.

From a future resiliency perspective, the designers of Serpent decided to use more rounds and affect performance to try to achieve a higher security margin of safety. This means Serpent may have some performance concerns, at least when compared with the alternatives.

Twofish

Twofish is a byte-oriented Feistel cipher with great flexibility of implementation, allowing a wide range of time/space tradeoffs. Many research reports have been written on various aspects of Twofish, which give confidence in its security. There was also a cost/benefit analysis done by the designers to decide which operations to use.

From a future resiliency perspective, Twofish's goals were security and implementation flexibility.

IV) Possible Outcomes

Does NIST want the fastest cipher? ... the cipher with the largest safety margin? ... the cipher with the most flexibility? ... the cipher with the most disparate instructions? ... the most Feistel-like cipher? ... the cipher with the most disparate design? Single or multiple winners? ... some other criteria? The point is that different answers to each question can lead to a different ordering of the AES finalists. Furthermore, any selection by NIST indicates in a backwards fashion which criteria they decided was more important than others. As one example, comparing MARS and Serpent, are more rounds or different rounds the better way to address having a sufficient safety margin? As another example, comparing MARS and RC6, are many different ideas or unity of design the better way to design a cipher?

The problem for NIST is not that there are no answers, it is that there are **too many** rational answers. Barring a security flaw, any of the AES finalists could be justified as being the sole winner simply by NIST adopting the corresponding design philosophy behind the winner as its own. NIST should resist any temptation to do this. Rather, as each submission has a different design philosophy, NIST should accept the implication that there was no obvious single all-around best solution. NIST should accept this implicit "higher-level" statement from the submitters and agree with them (as a group) that there is no single all-around best answer.

Strictly speaking, NIST's AES mandate is to select a winner or winners that is/are suitable for use by the US Federal government to protect sensitive non-classified data. Following the historical pattern of DES, it is also expected that NIST/NSA will issue a statement that the winner(s) is/are suitable for the intended purpose. Historically, it

was this endorsement that gave confidence to other groups, such as the American Bankers Association, to also endorse DES, which in turn led to DES becoming the most-deployed commercial cryptographic algorithm.

Now, some 25 years after DES, we see the endorsement by NIST of 3 families of asymmetric cryptographic algorithms in the revision of FIPS 186; namely, those based on the difficulty of integer factorization, the normal discrete logarithm, and the elliptic curve discrete logarithm. This allows the advantages of each method to determine the way asymmetric cryptography rolls out in the future. That is, NIST recognizes that there are multiple answers to the asymmetric cryptography question.

This author hopes that similar rationale will prevail among the NIST AES selection team regarding the symmetric cryptography question. While this author believes that the best outcome of the AES process is a handful of winners which lets the marketplace determine each algorithm's niche, it is realized that not all others share this opinion.

Ranking?

NIST should realize its decision is not restricted between having one AES winner and having multiple winners, it could also decide to have a ranking among multiple winners. As an example, NIST might specify that algorithm A is the primary winner and algorithm B is the backup. In this example, an implementation would be expected to either implement algorithm A (if resources are constrained) or both algorithms A and B (if resources are available). This seems much preferable to declaring a single AES winner, although inferior to selecting multiple co-equal winners.

Multiple Endorsement?

Another alternative is that regardless whether one or multiple winners (ranked or not) are selected by NIST for use by the US Federal government, NIST/NSA could issue health statements that certain finalists meet their intended security goals. This would at least allow other standards bodies to negotiate with increased confidence for the rights to an endorsed algorithm, if that algorithm better met their needs. For example, NIST might say that algorithm A wins (for US

Federal government use), but also issue a NIST/NSA report that algorithms A, B, and C meet their intended security goals.

Just to be clear on this point, if all five AES finalists have no known security weaknesses, then all five finalists should be giving a "certificate of health" regardless of the decision regarding the number or specific selection of AES winner(s) for approval for US Federal government use.

Acknowledgements

The author wishes to thank Certicom for providing the environment in which to write this paper.

Reference

[JC] Jerry Coffin in a post on sci.crypt on AES mentioned that satellites were an example where hardware was infeasible to change once deployed and saw this as a reason to have multiple winners.

Biography

Don B. Johnson is Director of Cryptographic Standards for Certicom, is a member of Certicom Research, and sits on the Advisory Board of the Standards for Efficient Cryptography Group (SECG). He participates in ISO SC27, ANSI X9, IEEE P1363 and other standards bodies. He has over 40 patents and patent applications in the area of cryptography. He was the editor of the X9.62 Elliptic Curve Digital Signature Algorithm (ECDSA) standard.