FUTURE RESILIENCY:
A POSSIBLE NEW AES EVALUATION CRITERION

Don B. Johnson, Certicom (djohnson@certicom.com)

January 29, 1999

Keywords

cryptography, symmetric key, encryption, AES, future resiliency, crypto-diversity, crypto toolbox, super AES, target diffusion

AES Topics

AES Evaluation Criteria
Cross-Cutting Analysis

Introduction

As announced by NIST, AES candidates will be evaluated on the criteria of security, cost (including performance and code size), and implementation characteristics such as flexibility and simplicity. This paper introduces the new term of "future resiliency" and proposes that NIST strive to ensure that the initial culling of the fifteen initial AES candidates to "about five" second round finalist candidates meets the future resiliency goal.

An Unfair Battle

Cryptographic algorithms, such as the AES candidates, are first designed and scrutinized; then some time later cryptosystems based on these algorithms are designed and deployed. At some point even later in the future, an adversary decides to attempt an attack on a deployed cryptosystem. However, the adversary has an advantage, he gets to design and execute his attack in the future, when there may be more knowledge available than when the algorithm was first designed. This is an unfair battle, in favor of an adversary. But some battles, even if unfair, must still be fought.

The algorithm designer must make some reasonable assumptions about what will be possible in the future, but there is no way that he will be able to know for certain what will be possible in the future. Sometimes there are breakthroughs that cannot be foreseen.

Future Resiliency

This paper introduces the term “future resiliency” as a goal for system designers to try to design systems that attempt to allow for the vagaries and uncertainties of the unknown future. For example, the Internet was designed to be able to route around damage automatically, it does not assume that any particular path will always remain useable. This is because any particular path might go down due to weather, accident, etc. So, in some sense, the Internet is future resilient and this feature is one of the reasons it is so popular.

While we cannot know for sure what the future will bring, we can make educated guesses. We can look to the past to see what kind of changes occurred and how they were addressed.

Crypto-diversity

We know from biology that ecosystems that are diverse and support many different life forms are better able to weather the changes the future brings. As a counterexample of bio-diversity, in the Nineteenth century, people in Ireland depended almost entirely on potatoes for food, as growing potatoes resulted in the most calories per acre. When a potato blight struck, the results were devastating; this is the reason there are today more people of Irish ancestry in the USA than in Ireland. In other words, bio-diversity is a form of future resiliency.

Similarly, crypto-diversity is a form of future resiliency. Information infrastructures that support only a single cryptographic algorithm are subject to the catastrophic failure of that algorithm. One way to address future resiliency in regards to cryptosystems is to ensure that the cryptosystems are designed to be algorithm-independent. This way, if a particular algorithm is later shown to have a fatal flaw, then a different (unbroken) algorithm could be “dropped” in as a replacement. However, to allow for this possibility, there obviously needs to be some number of cryptographic algorithms to choose from and not just one algorithm.

Future Resiliency and Asymmetric Cryptographic Algorithms

For example, the ANSI X9.F.1 workgroup has decided to have asymmetric cryptographic algorithm standards based on the presumed difficulty of three different hard mathematical problems, as follows:

1. the integer factorization problem (RSA, Rabin-Williams),
2. the (normal) discrete logarithm problem (DSA, Diffie-Hellman, MQV, etc.), and
3. the elliptic curve discrete logarithm problem (ECDSA, ECDH, ECMQV, etc.).

This is also the policy being followed by other cryptographic standards bodies such as IEEE P1363, ISO SC27, and the IETF. Having multiple algorithms based on different hard problems allows for the possibility of "drop-in" replacement, if such should ever be required; for example, if a particular hard problem turned out to be easier than believed today. Having such crypto-diversity implies better future resilience than sole reliance on a single algorithm.

Future Resiliency and AES

NIST's stated goal of the AES process is to ideally find the "best candidate" given the criteria of security, cost, flexibility, and simplicity first by narrowing down the candidates to "about five" second round finalists and then to select the overall winner in a third round. This current paradigm will be called the "single winner" perspective in this paper.

Given considerations of future resiliency, this paper suggests that it would be better for NIST to select a small number of winners rather than seeking to find only one, where ideally each of the winners is based (as much as possible) on different design principles, that is, the winners are (hopefully) based on different hard problems. This different paradigm will be called the "future resiliency" perspective.

For example, a number of the AES candidates are based on the Feistel network approach. It would be unfortunate if all the second round finalists were based on the Feistel network approach; what if a fundamental flaw was discovered with this approach in the future?

The categorizations and taxonomy classifications done by others at this conference provide a very useful service when future resiliency is considered as a goal. Given NIST's stated goal of selecting “about five” second round finalists, the future resiliency perspective suggests that no more than three finalists be based on the Feistel network approach, or on any other particular approach.

This means that all Feistel network algorithms are in competition with each other during the first round, so that there is some selection and pruning process occurring in the first round between similar designs. This also implies that different designs that are not shown to have a fatal weakness in the first round have an increased possibility of making it into the second round, just for the reason of their being different.

Extending Cryptographic Knowledge

Trying to ensure that disparate designs comprise the second round finalists has other advantages. Trying to ensure that different approaches are given the greater scrutiny expected to be done with the AES second round finalists will benefit the cryptographic community with the possibility of a wider-based increase in knowledge in symmetric block cipher design. If all the second round finalists are similar in design, then this opportunity will be missed.

Strength Considerations and Super AES

Given the required AES key sizes of 128-, 192-, and 256-bits, there seems little chance of using traditional brute-force key exhaustion techniques to attack any AES candidate. Of course, there might be a breakthrough with quantum computers or similar “wild” ideas, but it seems likely that the growth of quantum computing capability will be a bit or a few bits at a time. That is, even if quantum computing were feasible, the inherent complexity of a quantum supposition of states would likely mean that a limited number of bits in a quantum computer would actually be able to be realized in practice at a certain time. It seems clear that a 2-bit quantum computer would be built before a 256-bit one and the progress over time of achieving greater numbers of quantum bits could be tracked.

Given the required AES block size of 128-bits, there also seems little chance of using text attacks to recover information about the plaintext without recovering the key, assuming, for example, the CBC (cipher block chaining) mode of encryption is used. Recall that a text attack on a 128-bit block cipher used in CBC mode is expected to become possible (due to the birthday phenomenon) after about 2**64 blocks have been encrypted. To use a technical term, this is a "whopping" amount of text.

Given the AES key size and block size, the most likely way for an AES candidate to fail is via a security flaw. For many applications, a 128-bit keysize and a 128-bit blocksize will be sufficient. However, there is also the possibility of an application dealing with "million dollar messages" where extraordinary precaution may be warranted. For such applications, security is paramount and superencipherment using multiple AES candidates (that is, Super AES) may be desired to be used. It seems clear that if Super AES were to be used, it would be most advantageous if the algorithms used were of different fundamental design so that they would (hopefully) be based on different hard problems, so that cracking one does not result in cracking another.

The AES Beauty Contest

NIST has the authority to select a symmetric block cipher for the USA Federal government. Historically, when the IBM-designed algorithm now known as DES was endorsed by having it made into FIPS 46 (Federal Information Processing Standard 46 Data Encryption Standard), then it was quickly endorsed by ANSI as X3.92 (following the policy of "what is good enough for the government is good enough for me") and used as the cryptographic algorithm in a suite of ANSI X9 standards for financial institution use. This was largely due to the NSA “seal of approval” that stated that DES was suitable for its intended use.

The AES process can be seen as a beauty contest. However, everyone knows that beauty is in the eye of the beholder. Each standards organization deals with its own particular sets of constraints and determines for itself what are the important “beauty” criteria. Examples of possible constraints include financial cost, code size, performance, energy use, ability to execute in constant time, tradeoffs between setup costs and throughput costs, etc. A constraint that is absolutely crucial to one standards organization may be of little or no concern to another. Certainly, no one wants to proliferate cryptographic algorithms for the sake of proliferation, but a “one-size fits all” approach may simply not meet everyone’s needs.

From the point of view of other standards bodies, it might be most valuable for NIST to have a “small handful” of algorithms where each had no known weaknesses but where each had different designs and different implementation attributes and tradeoffs.

Drawing examples from the asymmetric cryptographic algorithm situation today, one might choose to use low public exponent RSA to achieve faster signature verification or one might choose to use EC to achieve smaller keys, smaller certificates, faster signature generation, faster key agreement, and/or more esoteric advantages such as the ability to validate a public key for arithmetic conformance with the specification without needing to query the associated private key as an oracle which leaks some information about the private key.

Crypto Toolkit Philosophy

Consider a carpenter’s toolkit, it is typical to include a hammer, screwdriver, wrench, and saw; each tool has different attributes. One might be able to substitute one tool for another in a pinch, but using the right tool for the right job is preferred. As a cryptographer, having a crypto toolkit with a selected number of algorithms with different attributes can allow for the best algorithm to be chosen for the job at hand, that is, the algorithm which best meets the goals and constraints of the task.

Another reason to desire multiple symmetric algorithms with different attributes is that this allows for the possibility of a hybrid solution, if that is the best way to solve a particular problem. For example, in the asymmetric cryptographic algorithm case, if one needs a solution where code size and key size are not significant limitations, but amount of computation is a critical limiting factor for one communicating party but not the other, then there are various solutions. Given current knowledge about security and performance, one might choose (A) low public exponent RSA if signature verifications or encryptions dominate the work to be done or (B) EC if signature generations or decryptions dominate or if all functions are called about the same amount. However, a hybrid solution (C) using RSA for signature verification and EC for key agreement and signature generation might minimize total computation time for the constrained device and be the preferred solution.

Patent Considerations

NIST has required that every algorithm submitter agree to a royalty-free license for use of the algorithm, if it is selected as the winner of the AES competition. Some AES candidate designers have gone further and deliberately did not patent their algorithm, that is, the algorithm is in the public domain, regardless whether it is selected as the winner or not.

However, there is always the possibility of a “ringer,” that is, a patent not by the algorithm submitter, but that is needed to be licensed to use the algorithm. NIST can attempt to use due process, and the call for papers for the Second AES conference specifically mentions presentations on patents, but the unfortunate reality is that it might be that the final decision will be left up to some court.

Even if a particular algorithm was a clear winner from all stated security and performance criteria, there is always the chance of patent complications. The more that one algorithm is similar to another, the more the chance that a particular patent would apply to both algorithms. Therefore, the suggested future resiliency approach of choosing a small disparate handful of algorithms seems warranted when considering possible patent complications.

Target Diffusion

When a cryptanalyst writes a paper attacking an algorithm, he or she is doing it to increase cryptographic knowledge and earn the respect of his or her peers. An attack may be theoretical and may never need to actually be executed. Contrast this with the actions of an adversary.

When an adversary tries to attack an algorithm, he or she is doing it to try to commit a crime, often to obtain resources (for example, money) illicitly. Cost factors are likely a concern, one may be willing to spend $5 million to make $10 million, but probably will not spend $5 million to make $1 million. If there is one and only one AES winner, then all adversarial resources might be concentrated on breaking that algorithm, for example, on an AES cracker machine.

If an adversary knows that another algorithm is waiting in the wings (so to speak) as a replacement, it may not be worth the adversary’s time and money to attempt an attack in the first place. This is because the adversary will likely want to amortize the costs of building an AES cracker by attacking many different applications. That is, while it may seem paradoxical at first thought, the publicly known existence of a back-up algorithm may mean that the back-up is never needed as the cost/benefit equation for the adversary is altered to reduce the potential benefit. Alternately, the deployment of multiple symmetric algorithms may mean that it is not worth the cost to an adversary to attack any one, as the payoff for each is reduced.

Analysis and Specific Recommendations

In order to try to be as helpful as possible, the following analysis with recommendations is given. A major caveat is that this analysis is based ONLY on the results presented by Miles Smid at the RSA ’99 conference. It was decided to do it this way, as trying to incorporate the latest results seemed like a never-ending task and is a purpose of the Second AES conference anyway. Subsequent results should be expected to modify these recommendations dramatically. As the results of the Second AES conference are not known at the time of writing this paper, these recommendations must be seen at tentative and are most useful when seen as examples of using the principles discussed herein. Also note that the order of candidates in any list below is always given alphabetically.

Below find a summary chart derived from Miles Smid's presentation.

Name	Type	Cryptanalysis	E/D Speed
CAST-256	MFN	-	Medium
CRYPTON	SPN	V (weak keys)	Fast
DEAL	FN	L,Kn,L,Kn (weak)	Medium
DFC	FN	C (weak keys)	Medium
E2	FN	-	Fast
FROG	Interp.	WFS (weak)	Medium
HPC	OMNI	-	Slow
LOKI97	FN	RK (weak)	Medium
MAGENTA	FN	BBFKSS (weak)	Slow
MARS	MFN	-	Fast
RC6	MFN	-	Fast
RIJNDAEL	SPN	-	Fast
SAFER+	SPN	Ke (weak 256 key)	Medium
SERPENT	SPN	-	Medium
TWOFISH	FN	-	Fast

Legend
MFN – Modified Feistel Network
SPN – Substitution Permutation Network
FN – Feistel Network
Interp. – Interpreter design
OMNI – different design (name by designer)
Cryptanalysis results are by initials of authors.
V – Vaudenay, et al.
L – Lucks
Kn- Knudsen
C - Coppersmith
WFS – Wagner, Ferguson, Schneier
RK – Rijmen and Knudsen
BBFKSS – Biham, Biryukov, Ferguson, Knudsen, Schneier, Shamir

Ke - Kelsey
Fast is about 25M/s.
Medium is about 8M/s.
Slow is about 2M/s.

The first thing to notice from the above chart is that there are eight AES candidate algorithms that have no known weaknesses (as of January 1999). These eight are CAST-256, E2, HPC, MARS, RC6, RIJNDAEL, SERPENT, and TWOFISH. As security is the paramount consideration, one valid method for NIST to use to select the second round finalists would be to simply take these eight candidates. However, NIST's stated goal was to select "about five" candidates as second round finalists.

If eight candidates cannot be selected for consideration as second round finalists, then the following analysis is offered as a way to narrow things down some more.

Clear losing candidates due to cryptanalysis that shows that the candidate algorithm strength is weaker than it should be are DEAL, FROG, LOKI97 and MAGENTA. These are the bottom-tier candidates in both of the following perspectives, as the security criterion is paramount.

From a single winner perspective (that is, for NIST to select the one ideal all-around-best AES candidate as the winner), the above data suggests that the top AES candidates for finalist consideration are E2, MARS, RC6, RIJNDAEL and TWOFISH as they all have no known weaknesses (as of January 1999) and they are all fast in encryption and decryption performance on NIST's reference platform. Selection of these five exactly meets NIST's "about five" goal for second round finalist consideration.

Second tier candidates are CAST-256 and SERPENT as there are no known weaknesses and both are medium fast. If more than five candidates would be selected to be second round finalists, these would be the next choices. Third tier candidates are HPC (slow, probably due to being a different design using 64-bit instructions), CRYPTON (weak keys), DFC (weak keys), and SAFER+ (weakness with 256-bit keys).

From a future resiliency perspective (that is, diversity of design is a goal to have among the second round finalists) then one way to proceed would be to attempt to use the AES first round results to answer the following questions:

1. Which non-broken Feistel Network designs (DFC, E2, or TWOFISH) should advance?
2. Which modified Feistel Network designs (CAST-256, MARS, or RC6) should advance?
3. Which Substitution-Permutation Network designs (CRYPTON, RIJNDAEL, SAFER+, or SERPENT) should advance?
4. Which non-broken “other” design (HPC) should advance?

Given the results known as of January 1999, this suggests that to address future resiliency considerations in a balanced manner that E2, HPC, MARS, RC6, RIJNDAEL, SERPENT, TWOFISH should be second round finalists. This would mean there are seven finalists, somewhat above NIST’s stated goal of “about five.” However, this means that the finalists consist of two Feistel Network designs (E2 and TWOFISH), two modified Feistel Network designs (MARS and RC6), two Substitution-Permutation Network designs (RIJNDAEL and SERPENT) and one “other” design (HPC).

HPC advances as it has no known weaknesses and is a very different design (even though it is slow). CAST-256 is not a finalist as there are two faster modified Feistel designs (MARS, RC6) among the finalists; if there were room for eight finalists, it would be included. DFC is not a finalist as there are Feistel Network designs (E2, TWOFISH) among the finalists that are not known to have weak keys (admittedly, this is a weak reason). SAFER+ is not a finalist as there are Substitution-Permutation Network designs (RIJNDAEL, SERPENT) among the finalists that are not known to have a weakness with 256-bit keys. CRYPTON is not a finalist as there are Substitution-Permutation Network designs (RIJNDAEL, SERPENT) among the finalists that are not known to have weak keys (again, admittedly a weak reason).

The most significant result is that both perspectives lead to approximately the same results, in that a common core of the same five candidates appears in both. This suggests that adding the future resiliency criterion may not be as difficult as might be feared.

The most notable changes that take place when moving from a single-winner perspective to a future resiliency perspective in this example analysis are as follows:

1. SERPENT moves from being tied-for-sixth place to being added to the finalists so that there are two Substitution-Permutation Network designs among the finalists.
2. HPC moves from tied-for-eighth place to being added to the finalists (even though it is slow) as it is a radically different design that has no known weaknesses.
3. CAST-256 drops from tied-for-sixth place to eighth place (even though it has no known weaknesses), as there are already two Modified Feistel designs, both of which are faster. As noted before, if the first-round criterion is simply that there are no known weaknesses to the design, then it should be included among the finalists in both perspectives.

Note that there likely are other ways to categorize the AES candidates than by grouping them as Feistel network, modified Feistel network, Substitution-Permutation network and "other" as Miles Smid did. Papers presented at the Second AES conference on the taxonomy of the candidates should be examined for alternative categorizations that might be useful to consider from a future resiliency perspective.

Artificial Tiebreakers

There is a temptation that NIST may fall into if it follows the single winner perspective. The temptation is that "hard" data such as performance times, code size, etc. may overshadow "soft" data such as confidence in security and freedom from patents. While we may never be able to say that an algorithm is secure, there is a big difference between saying there are no known attacks after an algorithm has been scrutinized for a year and after it has been scrutinized for ten years. Of a necessity, decisions must be made in the AES process in a reasonable time frame. However, recall the experience with DES; it took many years for differential cryptanalysis to become publicly known.

The concern is that criteria that are able to be measured in a straightforward manner (such as 2% better performance, etc.) might be used as a somewhat artificial tiebreaker, due to lack of knowledge about criteria that are harder to measure but that are more important in the long run, such as security.

A future resiliency perspective can be thought of as a way for NIST to avoid needing to come up with a somewhat artificial way to break ties to select a single winner. If there are a small handful of good algorithms with different attributes, then NIST should sanction those algorithms and let the market decide.

Potential Criticism of Future Resiliency Perspective

Some might say that this is the wrong time to add new a new goal to NIST's stated evaluation criteria. Some may even say it is unfair to change the criteria. However, the call for papers for the Second AES Conference specifically mentions a possible topic as being comments on the AES Evaluation Criteria. The author's belief is that NIST did the right thing at the start of the AES process by asking for submissions under the single winner perspective. There was the possibility of either too few or too many candidates. By asking for submissions as they did, NIST obtained fifteen candidates, which is neither too high to handle nor too low about which to feel comfortable.

However, now that there are fifteen candidates, it is time to step back and try to obtain the most value from those candidates. As Ralph Waldo Emerson said, "A foolish consistency is the hobgoblin of little minds, adored by little statesmen and philosophers and divines." In other words, do not be consistent just to be consistent, be consistent with past beliefs if they are still correct, but be willing to change. NIST has demonstrated its wisdom by asking for feedback on the AES Evaluation Criteria. Certainly, it is a sign of intelligence to be able to change one's mind as new information becomes available.

Another criticism is that the submitters might now not work as hard to show that their candidate is superior. The author believes that exactly the opposite is the case. Now the submitters of the “final disparate handful” of algorithms will need to demonstrate for each and every application why a designer should choose their algorithm. That is, the evaluation process will be ongoing and therefore of constantly increasing quality.

Saving NIST from Itself

Here is a final parting thought. The AES process is faced with solving a multidimensional problem with imperfect information. It is no surprise that there may not be a single clear winner. No one is surprised in the two-dimensional case with perfect information that there is no single largest number in the set {(2,4), (4,2)}. Such complex processes are often handled as a political process, with power blocs, vote trading and the like, but no one wants the AES process to resemble a political convention!

Considering the complexity of the AES evaluation process and the fact that additional information could arrive at any time to modify or even invalidate a decision, it would take almost omniscience on the part of NIST to pick a single winner. NIST will get assistance from open public review and the NSA, but neither of these is omniscient either. Miles Smid and all the NIST participants are to be commended heartily for being willing to act as facilitators in the AES process. But another way of viewing the future resiliency perspective is that it is a way to allow the dethroning of NIST from trying to accomplish an impossible job, namely, the job of trying to act as final arbiter in picking a single winner.

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.

Acknowledgements

The author would like to thank Certicom for providing an environment that encouraged him to write this paper, insofar as this paper met the “Certicom rule” of finding a way to include mentioning some advantages of using elliptic curves. (Smile.)

References

This paper analyzes algorithms from a high level perspective, detailed specific analyses of candidates are left to other papers and extensive formal references were not seen as needed by the author.

1. The NIST AES website at http://www.nist.gov/aes/ is the official location for AES information, including specifications of the selection criteria and the candidate algorithms, and AES conference information.

2. Miles Smid’s AES presentation at the RSA ’99 Conference is available in the conference proceedings or from him.

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 and is an inventor of the unified model of key agreement found in ANSI X9.42 and IEEE P1363, methods for public key validation, the CDMF (40-bit DES) algorithm, and the reverse signature concept found in ANSI X9.44 and IEEE P1363. He was the editor of the X9.62 Elliptic Curve Digital Signature Algorithm (ECDSA) standard.