Comments on Proposed AES Minimum Acceptability Requirements And Evaluation Criteria

(in response to Jan. 2, 1997 FR announcement)

Return-Path: <stewarts@ix.netcom.com>
X-Sender: stewarts@popd.ix.netcom.com (Unverified)
Date: Thu, 02 Jan 1997 17:31:20 -0800
To: AES@nist.gov
From: Bill Stewart <stewarts@ix.netcom.com>
Subject: Attn: FIPS for AES Comments
Cc: cryptography@c2.net

To: Director, Computer Systems Laboratory,
Attn: FIPS for AES Comments,
Technology Building, Room A231,
National Institute of Standards and Technology,
Gaithersburg, MD 20899.

NIST[Docket No. 960924272-6272-01] RIN 0693-ZA13 is a request for comments on draft minimum acceptability requirements and draft criteria to evaluate candidate algorithms for a new Advanced Encryption Algorithm.

I am commenting on the criteria as an individual, and not as a representative of my employer. I work in the telecommunications/ computer industry, including security analysis and some cryptography.

Overall, the drafts and process look good, and I'm quite pleased to see a commitment to an open process from NIST, as opposed to another closed process such as the CCEP and Clipper projects.

Unfortunately, there is also one very serious process problem, which may make the proposed selection approach unworkable and illegal unless addressed carefully by NIST and the Administration. The problem is the conflict between an open process, with submission requirements (B.1 and B.2) for complete algorithm specification, security analyses, and working source code, vs. the International Trafficking in Arms Regulations and other existing, announced, and proposed export policies which prohibit or require licensing or prior jurisdictional determination for "export" of source code, technical data, and cryptographic components, including open publishing on the Internet, discussion with foreigners, export of machine-readable media, and possibly even of paper documentation. While the NIST and NSA can reasonably operate in this environment, industry, academia, and non-US cryptographic experts cannot adequately participate in open discussions without some assurance of legal protection and the ability to exchange information with each other.

How does NIST propose to address this issue? International participation is a particularly important issue, given the expertise of people such as Biham, Shamir, Lai, and other non-US academic cryptographers and the need for interoperation and efficiency for telecommunications and finance

implementations.

An important part of an open process is positioning AEA/AES as a recognized
symmetric algorithm for non-military applications, since the military
generally uses closed standards, while the commercial world generally
prefers a negotiation among a family of encryption algorithms, including
- export-approved trivial and near-trivial algorithms,
    such as RC4/40 and RC4/128 with 88-bit exposed salts
    and the US and GSM cell-phone encryption algorithms,
    plus algorithms that may be approved in the future,
- fallback DES support
- slow but secure algorithms like Triple-DES
- fast newer algorithms such as RC4, RC5, and Blowfish
- hardware implementations, including proprietary systems and
    accelerators for (Triple) DES
- fast-setup algorithms for some applications and
    slower-setup high-throughput algorithms for others.
- block vs stream cyphers depending on application

There are three of the design criteria that have problems,
some technical and some organizational/political.

A.3 AES shall be designed so that the key length may be increased as needed.

The straightforward technical problem is with criterion A.3:
It's a good goal, but it unfortunately excludes the most important
existing candidate symmetric cypher algorithm, Triple-DES.
Triple-DES may be slow and clumsy to implement in software,
but it's very well understood, allows reuse of existing designs,
and is secure enough for probably the next 50 years of computer speed growth.
It's possible to accommodate Triple-DES into the criterion by
treating it as part of a family of DES, 3-DES, 5-DES, 7-DES, etc.,
but it's inelegant and stretches the wording of the criterion.

A more complex problem with criterion A.3 (and thus A.6) is that the
relationships between strength and key length are not simple:
An algorithm that performs very securely for longer keys may be very weak
with shorter keys which permit optimized attack methods,
and encryption speed may or may not differ significantly with key length.
    (For instance, with DES, the key schedule is relatively slow for
    single keys, but recent work has shown that a brute-force keyspace
    search in Grey-code order can reduce the key-schedule work for
    additional keys to a small fraction of the single-key time.
    Pre-computation attacks work quite well on algorithms like RC4/40,
    but fail on variants like RC4/128 with revealed 88-bit salts,
    even though both have the same size secret key and similar speed.)
This means that keylengths chosen for political reasons, e.g. 56 bit
limits for exportable algorithms, may affect different candidate
algorithms to very different extents. In particular, an algorithm that's
as strong as possible for short key lengths may be slow with longer keys,
or may require a very long setup time (e.g. Blowfish), and an algorithm
that's a very good choice for realistic commercial-strength key lengths
maybe too weak at exportable lengths.

A.2 AES shall be a symmetric block cipher.

Block cyphers are probably more important than stream cyphers,
and this is probably a good choice. However, the issues of streaming
and block chaining need to be addressed - some algorithms like DES
and Triple-DES can work well in either block chaining or codebook modes,
while others such as RC4 require more care for some environments.

The security of some applications is also quite sensitive to block sizes.
For instance, known plaintext attacks may be more effective with
shorter block sizes because of short standard file/data headers.

A.1 AES shall be publicly defined.

"Publicly defined" needs to be defined carefully, and publicly.
DES suffered reputation problems for years because of the
"What does the NSA _really_ know about the S-Box Structure?" uncertainties,
which were increased when people discovered efficiencies due to
group structures in the S-boxes, and really only abated after the
discovery of differential cryptanalysis by Biham and Shamir and
the confirmation that the NSA had used those techniques to strengthen DES.

It's especially important to have open public discussion of the
tradeoffs and criteria for selecting between algorithms. For instance,
the comparisons between Digital Signature Algorithm vs. RSA signatures
depend on the relative importance of signature speed vs. verification speed,
and industry generally viewed both NIST's and PKP's positions on
that issue to be motivated more by ownership concerns than technical ones.

Thanks!

Bill Stewart
stewarts@ix.netcom.com
Mountain View, CA.

References: [Federal Register: January 2, 1997 (Volume 62, Number 1)]
[Notices] [Page 93-94]
Federal Register Online via GPO Access [wais.access.gpo.gov]

http://jya.com/aes010297.txt

==========

Return-Path: <dthorn@gte.net>
Date: Thu, 02 Jan 97 21:56:13 -0800
From: Dale Thorn <dthorn@gte.net>
To: AES@nist.gov
Subject: New Encryption Standard

I'd suggest you use what I use, and perhaps redevelop it to suit your
needs. For security, simplicity, ease of use and rewrite:

Take a bitstream and:

1. Pre-normalize the relative number of zero- and one-bits to whatever ratio you prefer. Techniques include padding unused bits, and/or adding new bits to the stream.

2. Rearrange the bits (don't change any except in step #1) using a pseudo-random scheme of some kind:

a. Use the PRN stream in several ways simultaneously, i.e., to specify temporary block size, to specify move-to locations**, to pad the filestream, etc.

** Note that move-to's are not specified directly by output from PRN's, only by the relative amplitude of the PRN's.

b. Use different PRN sources for each of several encryption passes, to break continuity and patterning.

Note that by shifting the emphasis from changing bits to moving bits, you can get as close to the ideal of a true random distribution as you wish (or have time for), as if you were to put the bits into a lottery tumbler and mix them by turning the crank a few hundred times.

3. Fragment the encrypted data and store the fragments mixed with fragments from numerous other streams. This should help keep control of encrypted data more centralized.

I have sample (compilable) PC code in ANSI 'C', fully commented, plus a FAQ which answers most common questions.

==========

Return-Path: <whmurray@dttus.com>
To: "AES@nist.gov" <aes.nist.gov>
Subject: Comment
Date: Fri, 10 Jan 97 11:36:49 -0500
From: William Hugh Murray <whmurray@dttus.com>

-- [From: William Hugh Murray * EMC.Ver #3.0] --

I understand that the intent of the initiative is to have a high-performance cipher that advances the state of the art beyond DES. However, unless the intent is also to exclude a stream cipher or asymmetric key cipher that performed as well as a traditional cipher, then the bullet that requires a symmetric block cipher is inappropriate. It dictates a solution rather than a desirable property of the solution.

I suggest that the bullet be re-written to stress the property, i.e., high-performance, rather than the means for achieving it. In the absence of new invention, the outcome might be the same but why pre-judge.

In 1996 we could do 10,000 DES operation for the cost of one in 1977.

==========

Return-Path: <seward@netcom.ca>
From: seward@netcom.ca (John Savard)
To: AES@nist.gov
Subject: Comments respecting the Advanced Encryption Standard
Date: Sat, 11 Jan 1997 19:38:59 GMT
X-Newsreader: Forte Free Agent 1.0.82

John Savard
10245 - 151st Street
Edmonton, Alberta
Canada
T5P 1T6

January 11, 1997

The draft criteria and procedures for submissions for the Advanced Encryption Standard are appropriate, reasonable, and well thought out, as well as largely noncontroversial. However, I feel that a few comments respecting some of the criteria are still in order.

A.1 AES shall be publicly defined.

I personally approve of this criterion. However, given the following facts:

- a classified algorithm, known as "Skipjack", to be implemented only with a key escrow feature, has been put forward as the next encryption standard,

- that algorithm can reasonably be presumed to meet the security requirements of government communications in the time frame to be covered by the Advanced Encryption Standard, and

- the alternative of a publicly defined encryption standard significantly more advanced than DES has been claimed to have an adverse impact on national security, and this claim has been accepted and endorsed by the Administration,

entities wishing to submit algorithms may harbor reservations as to whether or not the process will be permitted to proceed normally to its conclusion.

A.2 AES shall be a symmetric block cipher.

That the AES should be a symmetric-key cipher, and not a public-key cipher, is clearly correct. For reasons of computational efficiency, a symmetric-key cipher would be an essential component of a practical cryptographic system, and so one would be required as part of any standard.

As research into new public key ciphers is continuing, and the design of new ones involves making advancements in mathematical theory, not only would standardization in that aspect be premature, but also a standards process of the current form would likely not succeed in

eliciting the submission of very many new public-key ciphers. Conversely, however, it could be argued that a standard without a public-key component is incomplete, and that the development of new symmetric-key ciphers is sufficiently trivial as to be unworthy of an extensive standards process.

There are several factors which support the criterion that the AES be a block cipher rather than another form of symmetric-key cipher:

Block ciphers have been subject to much public study, and there is a body of applicable theory for their design.

As there are a number of modes of operation for block ciphers, these ciphers have a greater flexibility than stream ciphers.

But the modes of operation of a block cipher which are applicable to the circumstances in which a stream cipher might also be used, specifically Output Feedback mode and Cipher Block Chaining mode, are somewhat unsatisfying in respect of the complexity of the part of the cipher that varies with each block. The security of a block cipher in such cases could, I believe, be augmented significantly by provision for a simple stream cipher for use in modes similar to Counter mode to also be available without exceeding the terms of the standard.

Also, a block cipher could be made amenable to use with an accompanying stream cipher by so designing it that part of the key could be modified with a much shorter set-up time than is normally required for loading in a new complete key. I envisage the reduced set-up time as being comparable to the time to encipher one block.

However, such a modified block cipher design could be dangerous, as careful design would be required to ensure that, were this rapidly changing key not kept completely secure, knowledge of it would not significantly augment attacks such as differential and linear cryptanalysis.

A.3 AES shall be designed so that the key length may be increased as needed.

In connection with this, I shall hazard the following conjecture without proof: if the full security-related benefits of an increased key length are to be realized, for a key of length n, the block size should increase proportional to n, and the computational time of applying the block cipher should increase proportionally to n squared.

Although it may be intentional that a block size is not recommended or suggested, it should be noted that if the security of a block cipher is assessed based on the complexity of defeating the cipher under theoretical circumstances requiring large amounts of known plaintext, then, as pointed out by one Terry Ritter, an independent developer of cryptographic software, any block cipher with an n-bit block size is subject to an attack of complexity 2^n requiring O(2^n) known plaintexts or exactly 2^n chosen plaintexts: accumulate a complete table of the block cipher's inputs and outputs.

In any event, while typical block cipher designs can provide for their keys to be increased in size up to a certain limit (that of independent subkeys) without otherwise modifying the operation of the cipher, if indefinite increase of key length is a requirement, at least the number of rounds will have to be subject to increase.

It seems that, in the absence of more specific guidance, the proposer of a standard would need to offer more than one option for increasing the key length, with various rates of increase in the time required to apply the cipher as the key length increases.

A.5 AES shall either be a) freely available or b) available under terms consistent with the American National Standards Institute (ANSI) patent policy.

It may be noted that, as patents relating to the original Data Encryption Standard and its experimental predecessor have expired, the construction of a secure block cipher need not depend on proprietary technologies, making this standard requirement particularly reasonable in this case.

John Savard

==========

Response to NIST request for comments on requirements for AES
By Don B. Johnson, Certicom Jan 17, 1997

I have 3 comments:
1. I believe the AES request should be reformulated to expand its scope. Let me explain.

DES encryption can be characterized as a non-linear permutation mapping a 64-bit input to a 64-bit output under control of a 56-bit key. DES decryption is the inverse operation under control of the same key. There are an additional 8 bits in the key that may be used for parity. The group generated by the set of DES transformations is very large.

The experience we have had with DES has seen it used in many ways. For example, review the following non-exhaustive list:

1. MAC calculation (as in ANSI X9.9 and X9.19)
2. Block data encryption (as in ANSI X3.106 CBC mode)
3. Stream data encryption (as in ANSI X3.106 OFB mode)
4. Symmetric key encryption (as in ANSI X9.17)
5. Random number generation (as in ANSI X9.17 or ANSI X9.30)
6. Nonce generation

I am sure others can easily add to this list.

This request to expand the question to reformulate its scope is based on the analysis that each of the above uses has differing requirements. These requirements sometimes come into conflict and cannot all be satisfied, at least in the ideal scenarios. For example, an ideal n-bit random number generator is expected to repeat after about n/2 samples (due to the birthday phenomenon), while an ideal n-bit nonce generator would not repeat until around n samples, that is, it would have a very long cycle length.

This request to expand the scope is as follows:
1. Identify the uses of symmetric key algorithms.
2. For each use, identify the exact ideal security requirements.

3. Accept proposals identified as meeting the requirements of each use. It is obviously best if one solution can be transformed into another via a relatively minor tweak, but this should not be assumed beforehand.

The goal is to obtain excellent solutions for each identified use. This will allow solution design and architecture to be easier, as each building block will be accepted in its use. Also, accepting this suggestion may help avoid arguments over deciding the requirements for a single AES algorithm.

2. I note that the prevailing philosophy today is to design an algorithm so that its security is associated with a variable that may be increased. Asymmetric algorithms can have key size increases to address increases in computation power. Similarly, I note that the call for requirements identified variable key and block size as a likely requirement for a symmetric algorithm. I believe there should be a similar call for a variable length output one-way hash function. This way, a system designer will be able to tailor all relevant security sizes to the expected attack computation capability. Any solution will be able to be balanced against all computation-based attacks. For example, today SHA-1 has an output of 160 bits. This means that many uses of SHA-1 depend on the unfeasibility of finding a collision in about 2**80 trials (due to the birthday phenomenon). 2**80 trials is considered infeasible in today's computing environment but may not be in the future. There !
is no
need to arbitrarily limit this number.

3. I believe the idea of defining parity bits in a DES key was a particularly poor idea. As use of these bits imbeds redundancy inside a key, they may be able to be used to help cryptanalyze a key. The parity bits also pose interoperability problems, as a system may decide to set and test, just set, or not set at all, each with different attributes and advantages and disadvantages. I suggest that any symmetric key definition contain only key bits and that any redundancy function on the key bits to ensure integrity be defined independent of the key definition.

==========

Return-Path: <zimmer@dlcc.com>
From: zimmerman <zimmer@dlcc.com>
To: " 'aes@nist.gov' " <aes@nist.gov>
Subject: NIST Encryption Effort
Date: Tue, 21 Jan 1997 15:59:05 -0800
Encoding: 12 Text

Are you serious?
What makes you think that any user who has privacy
concerns would believe that their communications are
secure. NO government branch has any credibility for
concerns such as these.

davez
David Zimmerman
Hardware Engineer
Diamond Lane Communications Corp
zimmer@dlcc.com
707.792.2946 x144

==========

Memorandum

Date: February 18, 1997

To:

From: M. Blake Greenlee
Subject: Comments on AES

I recommend that the AES have the following:

A. Technical Characteristics:
   1. Input block sizes of 64,128 and 256 bits.
   2. Key lengths (in bits) of:
      a. 80
      b. 96
      c. 128
   3. Minimum internal structure (e.g., "S-box") length = 48 bits
   4. Speed to support at lease T3 encryption
   5. Implementable in hardware or software

These characteristics might be met in a set of algorithm specifications that defined three configurations as indicated in the table, below.

Key Length	Block Size	Use
80	64	General purpose, hardware or software encryption; adequate for all but highest value transactions.
96	128	General purpose, perhaps much faster in hardware than software; bulk encryption.
128	256	High speed encryption devices (probably hardware); bulk encryption of files and data bases; high speed link encryptors.

If the above is attainable with a "flexible" algorithm, many needs can be met, yet tailoring of implementation complexity to risk will be possible. Fully flexible key length and block size may sound "good", but will result in additional costs and design difficulties. In particular, key and block lengths that are not multiples of 8 bits should be forbidden.

B. Patent Licensing Characteristics
   1. The AES must be available with a free license to anyone implementing it as a part of implementing a F.I.P.S. ISO or ANSI standard.

      Unless a submitter of an algorithm agrees to this condition, the submission should not be accepted – even for initial evaluation. Under no circumstances should the power of N.I.S.T. to "bless" a replacement for DES place any vendor or individual in a monopolistic position.

   2. In the unfortunate event that N.I.S.T. decides to accept an AES candidate that is patented, licensees must be able to implement the AES in an embodiment of their own choosing. Under no circumstances must licenses restrict users to "toolkits."

==========

Comments from the Fast Software Encryption Workshop,
Submitted by Ross Anderson on 3/6/97

Advanced Encryption Standard

Draft minimum requirements and evaluation criteria

Abstract: This is the minute of a discussion held at the Fourth Fast Software Encryption Workshop, Haifa, Israel, on Monday January 20, 1997 from 15.30 to 16.30 on the NIST call for comments on the Advanced Encryption Standard proposal. The discussion was held in the presence of over 50 workshop participants from all over the world. These comments were collected during the discussion by Ross

Anderson (the discussion chair), Bart Preneel, and Eli Biham, and then circulated by email to the participants who submitted a few further comments. The final draft was prepared by Ross Anderson.

General Comments

1. It was asked whether there should be a standard at all, or whether a diversity of algorithms might be safer and more adapted to applications. (This argument had been advanced by the NSA in opposition to the adoption of triple DES as a standard.) The counterarguments were:

a) that a standard would be adopted whether we like it or not and we might as well help make it a good one

b) for due diligence reasons, many clients would only use an algorithm with a government seal of approval

c) that a new standard would give an opportunity for many existing systems to be redeveloped and serious vulnerabilities in protocols etc removed

d) that a new standard would concentrate cryptanalytic effort on a single target, which (if unsuccessful) would increase confidence in that target

e) that the AES initiative presented an opportunity to establish a standard supported from the outset by government, industry and the academy.

2. Public trust in the algorithm will be harder to build if the rationale behind design decisions is not made fully public, and if the public does not participate in the evaluation process. So the rationale behind all design decisions should be completely explicit.

3. It would be helpful if any S-boxes, constants etc should be chosen by some convincing method (such as at random from a sufficiently large space). There are two reasons for this. Firstly, if all the design choices are made by a single person or organization, then the algorithm will be less likely to be trusted; trapdoors will be suspected. On the other hand, we do not want a ``committee'' design. A customisable design is probably the best balance between these concerns. Secondly, there are users who will want to customise a standard algorithm (see 11 below).

4. We would favour a process in which the initial submissions are whittled down to a short list of perhaps 3-4 candidates. This would enable the community to concentrate the analysis and evaluation effort on them rather than dispersing it on dozens of targets. (In this workshop alone about ten ciphers were suggested, some of them having several variants.)

5. NIST should clarify the role of non-US citizens. Clearly, a new US standard will (like DES) become widely used in other countries. Will non-US submissions be acceptable?

6. There is concern that the proposed timetable does not leave enough time for serious cryptanalysis.

General Requirements

7. It is not clear that one cipher can satisfy the requirements for all applications, and on all kinds of processors (or special hardware). The question arise whether we should have a family of ciphers, appropriate for different environments.

For example, the majority of fielded DES implementations are on 8-bit processors such as smartcards and microcontrollers, and used in applications such as banking, power metering, pay-TV key management, door locks, road tolls and the like. In such applications, the main `improvement' sought from a DES successor is a reduction in code size.

On the other hand, the importance of intellectual property protection is growing and there is wide use of stream ciphers in, for example, pay-TV systems. Here, speed is a definite requirement and code size is relatively unimportant. So NIST should consider whether there should be two standards: a block cipher suitable for 8-bit processors, and a stream cipher optimised for speed.

8. There was wide condemnation of the draft proposal, that C source code be evaluated on a PC. Ideally, a survey of applications, both fielded and planned, should be undertaken so that the relative importance of different performance metrics (speed, code size, etc) could be evaluated and a realistic benchmark suite be specified. At the very least, NIST should be much more explicit about the performance requirements. We expand on this below.

9. NIST should also provide a ranking for the various evaluation criteria to clarify their relative importance.

Technical Requirements

10. There should be procedures agreed in advance for dealing with any weakness of the algorithm that arises later. This might be predictable, such as an advance in chip technology that makes a longer key necessary; unpredictable but minor, such as the discovery of a new but rare class of weak keys; or catastrophic, such as a new shortcut attack that forces a change to a completely different algorithm.

Several mechanisms are thus likely to be necessary including a review body or process, a `backup algorithm' and perhaps (as suggested by NIST) a means of increasing the keylength. There was no unanimity on this last point however; an alternative would be to adopt an algorithm with a keysize well beyond possible exhaustive search (e.g., 256 bits) and use part of the keyspace as appropriate.

One possible `backup algorithm' is using the same algorithm with different parameters, such as with a different set of S boxes. This could provide a rapid and low-cost means of recovering from all but a total break.

11. There are other reasons to support customization by other means

than the key. In addition to the building public confidence in the absence of trapdoors, as mentioned above, parametrisation will appeal to those users who want a compromise between a proprietary algorithm and a standard one - such as those who at present use DES with nonstandard S-boxes or other modifications to prevent keysearch. The successor to DES should be chosen so that it is not as difficult to choose strong values of the S-boxes or other constants as it is in the case of DES.

12. An increasing number of applications involve cryptographic authentication protocols (Kerberos being an example). Here, the 64-bit blocksize of DES is a disadvantage; the real requirement is to encrypt variable length blocks. Many implementers use DES-CBC but this can be vulnerable to cut and paste attacks. A block cipher of variable width would be ideal for such applications.

13. Some people felt that a 64 bit blocksize was inadequate for security reasons, as once large volumes of data start to be encrypted the volume limits set by the birthday paradox may be approached.

14. Given that the algorithm may be of variable width and may also have a variable key length, thought needs to be given on how such parameters will be securely expressed. The RC5 approach of packaging the key in a `control block' with such parameters might provide inspiration here, as could the IBM approach of `key control vectors' to enforce a functional partition of the keyspace where applications require this. We probably need an algorithm version number as well, and `fields to be defined later'.

15. In the event that the standardized algorithm is simply another 64-bit block cipher, there is a need for a standard mode of operation that allows a variable length block to be encrypted with error extension in both directions. More generally, it is time to look not just at modes of operation but also at other supporting structures such as APIs and lower level interface definitions.

16. The algorithm should approximate to a random permutation as closely as possible, e.g. there should be no equivalent keys, no complementation properties, no related keys and no weak keys.

17. The bit naming convention should be explicitly defined.

Security Requirements

18. The types of attacks that the cipher must withstand must be made explicit (e.g., known plaintext, chosen plaintext, adaptive chosen plaintext/ciphertext, related-key).

19. The security targets must be quantified, e.g. ' 2^10 related key queries, 2^40 chosen plaintexts, 2^50 storage, 2^60 known plaintexts, 2^80 effort'.

20. There must be minimum values set for security parameters,

such as number of rounds, block size and key size, in order to prevent loss of confidence in the standard following a published attack on a legitimate implementation.

Efficiency requirements

21. As noted above, it was widely felt to be unwise to evaluate the candidate algorithms solely on a PC, as the majority of DES implementations are believed to run on 8-bit processors in embedded applications. It appears to be prudent engineering practice to optimise an algorithm for the slowest processor on which it will be widely used - which might mean the 8051 (although 4-bit processors are still used, and GOST appears to have been designed with these in mind). It should also run adequately in Java, as the commercial success of this language cannot be ignored.

PCs will be important, but we do not know whether the typical PC CPU in five years time will be a RISC processor such as Alpha, a VLIW processor such as Philips' TriMedia, or a combination superscalar/SIMD such as Klamath. Similarly, hardware/firmware implementations (FPGA, ASIC, standard cell,...) should be considered.

22. Some applications, such as B-ISDN require fast key setup. The evaluation criteria should therefore define a maximum key scheduling delay; this might defined relative to encryption as a function of key length. A possible alternative would be ability to cache a number of round keys. However, while 1024 keys might be sufficient for current ATM switches, more keys might be needed by future equipment.

23. There should be targets for code size and memory size, especially for implementations on smartcards and other 8-bit processors. For hardware implementations, there should be a target gate count; and for power-critical applications (such as contactless smartcards) there should be a power target of microjoules per block encrypted.

Evaluation and interface requirements

24. The process of evaluation should involve bounties to attract serious and sustained attack. It is suggested that NIST offer a large sum (say \$1m) for a significant shortcut attack. This should ensure that anyone outside the sigint community who discovers such an attack will report it rather than seek to exploit it. The shorter term evaluation procedure should be also clarified: what incentives will there be for outside contributors to invest effort in it?

25. When reducing a large number of candidates to a shortlist, one possible approach to the performance issue would be to define a minimum speed relative to known ciphers such as DES or triple-DES. However, some participants felt that many people are unaware of, or have no access to, fast DES code for comparison.

26. In any case, a thorough examination of the performance aspects of shortlisted candidates should be carried out. As mentioned above,

there would ideally be a study of existing and planned applications leading to the development of a benchmark suite. In the absence of such an exercise, then at the very least the following should be considered for each shortlisted candidate:

a) code and memory size, especially on common smartcards and microcontrollers

b) speed, not just on currently common chips such as 8051 and Pentium but also RISC and VLIW chips

c) gate count for simplest and fully pipelined hardware implementations. Tradeoffs between speed and gate/count should be considered, as well as the minimum number of microjoules per block encrypted

d) whether software implementations are significantly different (or more difficult) according to whether the processor is big endian or little endian

e) key agility, or round key memory requirements if cacheing is preferred for B-ISDN applications

f) whether there is a well understood tradeoff between number of rounds and attack effort

27. NIST should define a standard interface for the algorithms in order to facilitate validation by the wider crypto community.

28. Ease of validation is important. A single test vector is not enough: the algorithm designer should supply a full set of test vectors, plus a validation suite that exercises them via the standard interface mentioned above and performs any other tests required to check all single points of failure and thus ensure that an implementation is correct.

29. Submissions should include not just one or more implementations optimized for speed or memory size on various processors but also an easy-to-read endian-indifferent one, so that correspondence with the description of the cipher can be readily checked.

30. Finally, the evaluation criteria should be more carefully drafted. For example, criteria (b), (c) and (d) overlap, and it is not clear what exactly is meant by `simplicity' and `flexibility'.

==========

Comments on AES Federal Register Notice
3/9/97

Dear Sirs,

I applaud your invitation for comments on a proposed AES. I am submitting comments based on my 20 years experience as an information security practitioner in the financial services industry, probably the

largest private sector implementor of national cryptographic standards. In general I agree with your proposed criteria, with the following input:

1. I would make one modification to A.5: AES must be freely available (i.e., no license fee), as is DES.

The remainder of my comments are additions to A.1 - A.6.

2. AES must be exportable.

3. I agree with Don B. Johnson's (Certicom) comments #1 and 3, dated 1/17/97. I assume he sent them to you, so I will not repeat them here. If you do not have them please feel free to contact me. (I also agree with his comment #2, but think it should be a separate request as it does not deal directly with an AES algorithm.)

4. There must be a well defined (backward compatible) migration path from DES to AES (or some variant thereof). The banking industry has a huge installed base of DES and could not afford to scrap it all.

5. An AES validation process must be in place at the time the AES is announced.

If you have any questions on the above, please feel free to contact me.

Respectfully submitted,
Sandra M. Lambert
Lambert & Associates
Voice & fax: (213) 469-6978

==========

Return-Path: <rivest@theory.lcs.mit.edu>
From: rivest@theory.lcs.mit.edu (Ron Rivest)
Date: Sat, 22 Mar 97 14:09:29 EST
To: AES@nist.gov
Subject: Comments on the proposed FIPS for AES

To: Director, Computer Systems Laboratory
Attn: FIPS for AES comments
Technology Building, Room A231, NIST, Gaitherburg, MD 20899
From: Professor Ronald L. Rivest, MIT Lab for Computer Science
Date: March 22, 1997
Re: AES criteria

Here are some comments on the Advanced Encryption Standard proposal and procedures, as per your request for comments. These comments are listed individually in no particular order.

1. In general, it's nice to see that you are finally getting around to replacing DES. It's about time!

2. While I presume that 3DES will be submitted, you should be sure to include it as a candidate in any case. (This suggestion is somewhat inconsistent with my other comments, such as the one

on block size.)

3. All submissions should be made public, and there should be a public comment period on the submissions. You should not allow proprietary submissions that may not be so published. Any such submissions, and the publication of such submissions by NIST, should of course be exempt from export regulations. You should post all of the submissions on the World Wide Web.

4. All deliberations and considerations on the selection of the AEA should be public. In particular, the role of the NSA in the evaluation procedure should be explicit and public. I would propose that NSA stay out of the picture until the number of viable candidates is down to a small handful, at which time NSA may wish to publically comment on the security of one or more of them. The timetable for the selection of AES should be sufficiently relaxed that sufficient public comment and scrutiny is achieved. In particular, the final candidate should be announced and comments solicited before the decision is made final.

5. Criterion A.3 should be reworded to say: "AES shall be designed to accept keys of variable length, from 0 to at least 256 bits, inclusive, in one-bit increments." (The wording you have is unnecessarily vague.)

6. You don't say enough about the block length. Criterion A.2 should be more specific here. I think that a block length of at least 128 bits should be specified, with explicit encouragement for larger block sizes or even variable block sizes. The small block size of DES has been the source of many difficulties. A block length of 160 bits or greater makes "birthday"-type attacks succeed with probability at most 2**(-80), which is satisfactorily small.

7. You don't give any guidelines regarding key setup time. A short setup time promotes key agility, which is needed in many network contexts. A longer setup time hinders brute-force attacks. What would you like? (Of course, you can have a variable setup time, too.)

8. You don't mention a natural consideration for software-oriented algorithms: whether it should be "little-endian" or "big-endian" in style. I refer, of course, to the issue of the order in which bytes are stored in a word. A little-endian bias favors Intel architectures, whereas a big-endian approach favors other architectures. Such "fine points" can noticeably affect the speed of the algorithms, according to which machine they are run on. (While you do specify a PC, you don't suggest such a bias.) Of course, for some algorithms this issue is irrelevant. I suggest that you specify that the "target PC" is little-endian.

9. An algorithm should be accompanied by some indication as to why it is free of "trap-doors". The derivation of and justification for any tables or arbitrary parameters should be required as part of the submission. If programs were used to derive the tables or parameters, they should be included with the submission.

10. The submitter should include not just a single input/output example, but a proposed "validation suite" of examples to test an implementation to see if it is correct. (This might be of an iterative nature, to minimize the size of the suite given. The encryption algorithm might be repeatedly applied to its own output, for example, and the result after 100 iterations given. For each iteration the key used would be derived from the previous outputs as well.)

11. A submission should clearly indicate if there are any known weak keys or semi-weak keys. The submission should indicate what special care, if any, should be taken in key selection, if this process is other than just randomly picking a bit string of the appropriate length.

12. It is often desirable for organization such as banks to have their own "proprietary" version of a standard algorithm. You may wish to encourage submitters to indicate how such variants could be derived from the standard algorithm in a way that does not affect the security of the resulting algorithm.

13. If there are other limitations on the algorithm design, such as whether it should be implementable on current smart cards, or whether it should be efficiently parallelizable on high-end processors, then these limitations and criteria should be made explicit, or at least listed as explicit biases for the evaluation.

14. You do not say whether submissions from outside the U.S. are welcome. I presume therefore that they are welcome. It would help to be more explicit on this point. You also could be more explicit regarding international patent issues. (Would you accept an algorithm that was patented only in Japan, but not in the U.S., when there was no guarantee of reasonable licensing availability in Japan? This would affect our multinational corporations.)

15. It would help to inspire confidence in the algorithm chosen if you were to adopt an explicit program of continual review, both by the NSA and by non-governmental cryptanalysts (who might be paid by their employers or by government grants). This program would continue indefinitely.

16. It should be explicit that use of the new algorithm is independent of any other issues of cryptographic policy. If, for example, the new algorithm were patented and licenses were to be made available only within the context of a key-recovery or key-escrow mechanism, then the whole standards effort is likely to fail as it falls into this policy tar-pit. It should be clear that the new algorithm can be used with no more constraints than any published, unpatented, algorithm could be, with respect to such requirements.

---- end ----
==========

Return-Path: <vallhonrat@worldnet.att.net>
From: "Eleanor & Juan Vallhonrat" <vallhonrat@worldnet.att.net>

To: "AES NIST" <AES@nist.gov>
Cc: "Lang Wedgeworth" <hpwlaw@prysm.net>, "Joe Morgan" <morganj@arn.net>
Subject: AES Comments from Gemini Systems
Date: Tue, 25 Mar 1997 11:08:05 -0600
X-Msmail-Priority: Normal

25 March, 1997

Director, Computer Systems Laboratory
Attn.: FIPS for AES Comments

Submitted by: Joseph M. Morgan and Juan B. Vallhonrat
Gemini Systems

Ladies and Gentlemen:

In your announcement of the AES workshop you include the following text from the 1993 reaffirmation of DES:

``At the next review (1998), the algorithm specified in this standard will be over twenty years old. NIST will consider alternatives which offer a higher level of security. One of these alternatives may be proposed as a replacement standard at the 1998 review."

The way this wise statement is worded it implies to us that the alternative could be a truly new standard altogether, rather than just a new algorithm.

In reading the proposed draft Minimum Acceptability Requirements it appears to us that concepts which are 20 years old are being perpetuated. We respectfully suggest that perhaps what the encryption standard needs is a complete paradigm change. The requirements enumerated, for example, a symmetric block cipher, are based on DES with the exception that the key length may be increased as needed. Why would a new algorithm be restricted to be symmetric? Why would it be restricted to operate on fixed length blocks of data? Relaxation of these two factors would, in itself, offer a higher level of security.

Instead of endorsing a single new fixed algorithm, our suggested paradigm change envisions an AES which would provide the umbrella under which algorithms could be built by invoking a sequence of AES protocol compliant encryption components. The AES should define the framework for a component based encryption system (CBES). Such a

system would consist of a set of encryption primitives from which the components would be built. These primitives would include logical and mathematical operations, bit movement and management operations, data and key management operations, etc.

The components, which are created from the AES sanctioned primitives, would be categorized into functional groups such as:

Key Manipulations and Management
Data Delivery
Permutations
Data Separation and Concatenation
Substitutions
etc.

There would be an AES defined protocol for each group of components. The components would be developed by numerous sources, yet they each would need to comply with the AES protocol for the category for which they were designed. Encryption algorithms would be designed by combining several components via a component base interface (CBI).

With the ever increasing and varying needs for encryption it is hard to imagine how one single algorithm could satisfy all requirements. With a CBES like standard, the component protocols become the standard, not a specific algorithm. This would provide a fertile environment where algorithms would be developed to fulfill application specific requirements. There would not be one single criteria with which to judge speed or memory requirements, as those criteria would be application dependent and proportional to the degree of security required in the given application.

A CBES like standard would encourage creativeness and allow for the emergence of numerous third-party component vendors. For example, some vendors might perfect key management, while others may perfect substitution. The encryption developer would pick and choose from numerous components to fulfill the security requirements of the specific application by instructing the CBI as to which components to use and in which order to use them during the encryption process.

In order to allow for a smooth transition to a new standard the existing

algorithms can be
implemented under this proposed CBES like standard exactly as the algorithms are
presently defined. If a further evolutionary transition is desired one could also easily
strengthen such algorithms. For example, the components used to implement DES can be
modified to increase the key length, and/or add more rounds, and/or vary key shifting,
and/or change the S-boxes, etc. The possibilities are endless.

If CBES is to be endorsed, the Minimum Acceptability Requirements and Evaluation
Criteria together with the Submission Requirements would have to be drafted from
scratch. For example, when DES was implemented 20 years ago, memory requirements
for an algorithm was a terribly important consideration. Today memory requirements take
on an entirely different light. The same goes for hardware implementation when one
considers that a good part of present day hardware is really firmware.

While in the last 20 years there have been revolutionary changes in hardware, software,
and communications, the encryption standard has remained unchanged. Some may praise
its staying power, while others may consider it an anchor to progress. Let us now
consider a truly flexible and dynamic standard which can adapt and evolve with the advent
of new technology.

We are looking forward to some lively discussions during the upcoming workshop.

END

==========

Return-Path: <ritter@io.com>
X-Sender: ritter@mail.io.com (Unverified)
Date: Tue, 25 Mar 1997 16:56:45 -0600
To: AES@nist.gov
From: Terry Ritter <ritter@io.com>
Subject: COMMENTS ON CHOOSING THE ADVANCED ENCRYPTION STANDARD

COMMENTS ON CHOOSING THE ADVANCED ENCRYPTION STANDARD

In the National Institute of Standards and Technology:
A Response to [Docket No. 960924272-6272-01] by:

Terry Ritter, Registered Professional Engineer
ritter@io.com http://www.io.com/~ritter/

Ritter Software Engineering
2609 Choctaw Trail
Austin, Texas 78745
(512) 892-0494

Here are my recommendations for a new ciphering standard intended to assist The United States well into the next century:

1. To be considered for the AES, candidate ciphers should have a keyspace of 120 bits or more.

2. To be considered for the AES, candidate ciphers should have a block size of 128 bits.

3. Ideally, the AES would also support much larger blocks, 64 bytes wide and beyond. Large blocks are important to support features generally not possible in a smaller block, and thus not currently available to DES users. These include: ciphering without needing plaintext randomization (allowing ECB mode), zero latency dynamic keying, strong validation values, and homophonic control.

4. Ideally, this standardization process would produce multiple acceptable ciphers for each of multiple categories of ciphering application. We should resist the idea that there can be only *one* "standard" cipher. Cipher users are the appropriate authority for any selection among acceptable ciphers.

5. It is difficult to know what draft requirement A.3 (requiring candidate ciphers to be able to "increase key length") is about:

   If A.3 is about supporting a key of arbitrary length, presumably through some associated hashing process, that is a good idea.

   But if A.3 is about having a parameter to adjust the internal keyspace, this is probably not a good idea. The typical way to provide such a parameter would essentially *reduce* the native strength of the cipher *without* any compensating advantages in resource use or throughput. This would be a mistake.

6. Draft requirement A.6(b) (computational efficiency) should be more detailed, such as:

   > Ciphering rate, in bytes per second, for repeatedly ciphering a single block in memory, using any named operating system, and any named CPU chip X at clock speed Y. The ciphering rate must be listed for a version in high level C, but additional values may be presented for other versions, possibly using assembly-language.
   >
   > Setup or initialization time, in seconds, for each ciphering rate measurement.

   It might be useful to additionally require values for some widely-available CPU, with results normalized to a standard

clock speed, such as Pentium CPU (w/o MMX) normalized to 100 MHz.

A.6(b) is also the appropriate place for comments regarding the advantages of custom hardware realizations, although no such hardware need actually exist, so that actual performance measurements will not be available.

7. Draft requirement A.6(f) (flexibility) seems like an attempt to get a single cipher for all applications. But, in ciphering, one size does *not* fit all.

   Certainly a cipher *technology* can be scalable across a wide range of resource costs and throughput targets. But a particular *cipher* probably will have some inflexible decisions which will optimize it for a particular environment.

   We should instead define *groups of applications* with similar requirements, and then define the appropriate ciphers for each group.

   Smart cards might be a group which would be particularly sensitive to computation and memory requirements. Larger systems, where memory is not constrained, might be another group (even entry-level PC's now often have 16MB of RAM).

8. Draft requirement B.2 (source code) should also be more detailed:

   > Source code for each system measured in A.6(b), should be delivered both as printed text and as ASCII text files on a 3.5" 1.44 MB floppy in IBM PC format. Simple drivers for A.6(b) and B.4 should be included.

9. I am aware of sentiment to restrict the AES to unpatented algorithms or to require a free grant of rights by the patent holder. But considering that patent rights are granted by the very same U.S. Government now making a selection, any attempt to ignore patented designs would be very disturbing.

   If you want this selection process to continue well into the new millennium, all you need to do is to treat some participant class unfairly, and the whole issue will end up in court for a very long time.

10. I am also aware of comments that AES should be a stream cipher, because (it is said) a stream cipher can be 10 times as fast as a block cipher. Having personally developed and patented new fundamental technology for both stream and block ciphers, and having personally implemented a wide variety of stream and block ciphers, I question such conclusions. In my experience, software realizations of stream and block ciphers with comparable strength tend to have surprisingly comparable throughputs.

    In hardware realizations, it seems likely that block cipher architectures which support massively parallel operations

must be much faster than "equivalent" stream ciphers.

11. I am also aware of sentiment that the AES should run on all platforms from "smart cards" to Alpha workstations. I think this would necessarily compromise the strength of the AES by inherently eliminating many of the advantages that two decades of semiconductor progress have bought us, including large amounts of storage and modern CPU design.

    If it is necessary to have a cipher for "smart cards," that should be a different standard.

    If there are to be limits on RAM storage, they should reflect the situation of a modern desktop computer with 16 megabytes of RAM. By the time a new standard is effective, 64 megabytes could be very common. AES should make use of common computation capabilities to deliver serious strength and performance.

12. I am also aware of attempts to limit candidate designs based on ratios of enciphering vs. deciphering speeds or setup vs. ciphering, etc. Such comparisons are misguided, in that they could eliminate a cipher which is superior in absolute terms.

13. My last comment is that while cipher "strength" is clearly our ultimate goal, it is *only* a goal, because it cannot be measured. Rather than relying upon such analysis as may have occurred in the "open academic literature," I would hope that the National Security Agency would be enlisted to perform a scientific analysis of each acceptable candidate. While there may be some motive to minimize the content of such comments, they must be sufficiently factual and forthcoming to allow external comparison across a wide range of categories and benefits, instead of being simple "yes / no" conclusions.

Terry Ritter ritter@io.com http://www.io.com/~ritter/

==========
Return-Path: <dkatz@aba.com>
Date: Wed, 26 Mar 1997 14:39:00 -0500
From: Deborah Katz <dkatz@aba.com>
To: AES@nist.gov
Subject: ASC X9 Comments on AES

(Embedded in email message AND attached as a Word 6.0 file)

March 26, 1997

TO: Director, Computer Systems Laboratory
Attn: FIPS for AES Comments
Technology Building, Room A231,
National Institute of Standards and Technology
Gaithersburg, MD 20899

From: Cindy Fuller, X9 Secretariat

Accredited Standards Committee X9 - Financial Services, an ANSI-accredited committee, has achieved consensus on the following list of comments for the first phase of the Advanced Encryption Standard (AES) development.

1) Consideration should be given to the transition time for the financial industry to migrate to any proposed new encryption standard. The transition time is expected to take from two to five years once the new AES has been selected and approved. Along with the next federal register announcement regarding the algorithm selection process, a proposed timetable should be published to include the following:

a) Algorithm testing and selection schedule
b) A one year public scrutiny period used to find flaws and fix them
c) Validation (NVLAP, etc.) methodology availability

2) Many in the financial industry are beginning to use ANSI X9.52, Triple DES. Even though this may be a transition tool being used until widespread AES availability, NIST should recognize this reality and continue to support DES in this context and other appropriate contexts for as long as is prudent from a security perspective.

3) The financial industry of the United States must protect the flow of funds and related information both inside and outside of this country*s borders. X9 realizes that making an algorithm exportable under the current regulations are outside the scope of the selection of any symmetric encryption algorithm. However, once AES has been selected, every consideration (including changing current regulations) must be given to ensure that our financial industry can continue to compete on an international basis while protecting its data with an algorithm of sufficient strength.

4) If AES is to become a cost effective solution, it must be free of any unreasonable patent issues.

In addition, AES must be freely implementable in any way that a vendor chooses. No license constraint (associated with any patent) should dictate that the AES can only be implemented in a specific manner or with a specific vendor*s *tool kit* or software package. This will allow the marketplace to provide more competitive and secure alternatives.

AES must be royalty-free.

5) The AES algorithm should have an associated (NVLAP or other) initiative that provides a mechanism that will validate AES implementations.

6) Ideally only one AES should be selected to satisfy all requirements. However, there are many uses that AES will be subject to, such as:

a) MAC calculation (as in ANSI X9.9 and X9.19)
b) Block data encryption (as in ANSI X3.106 CBC mode)
c) Stream data encryption (as in ANSI X3.106 OFB mode)
d) Symmetric key encryption (as in ANSI X9.17)
e) Random number generation (as in ANSI X9.17 or ANSI X9.30)
f) Nonce generation
g) Key-derivation techniques, such as DUKPT (as in ANSI X9.24, Section 4.8 and Appendix E)

7) AES should not use parity bits for keys. The use of these bits imbeds redundancy inside a key, which may be used to cryptanalyze a key. The parity bits also pose interoperability problems based on the different ways that communicating systems use these parity bits. There is anecdotal evidence in the financial industry concerning implementation delays and costs associated with the use of parity bits for keys. No corresponding benefit for the use of parity bits for keys has been documented. Therefore, symmetric key definitions should contain only key bits. Any redundancy function on the key bits to ensure integrity should be defined independent of the key definition.

8) There are only three criteria to be considered for the evaluation of AES. These criteria listed in order of relative importance are:

a) Security - which defines the capaibility of the AES to withstand cryptanalysis or exhaustive key search.

b) Total cost - including CPU cycle cost (computational efficiency) and memory allocation costs.

While cost factors are considered in the aggregate, any single drawback (e.g., computational efficiency) may rule out an algorithm if this attribute makes AES infeasible. However, NIST should be careful not to discard too quickly a poor implementation of a good AES candidate. History has shown that large differences in computational efficiency for the same algorithm depend on the implementation techniques used. Once AES is exposed to commercial development and optimization, computational efficiency advances will be made. For measurement purposes - many very large financial institutions that use the fastest commercial mainframe processors use DES engines with a single-engine throughput of about 112 million bytes per second.

c) System Feasibility - the ability to design, develop, implement, and operate the AES on a variety of industry platforms (for example, ATMs, POS devices, user workstations, servers, and mainframes); using a variety of technology support devices (for example, dongles, smart cards, PCMCIA cards, various PC boards, and a variety of integrate encryption engines).

==========

Return-Path: <Bill_Poletti@MASTERCARD.COM>
To: aes <aes@nist.gov>
Cc: Melinda Yee/NYC/MASTERCARD <Melinda_Yee/NYC/MASTERCARD@MASTERCARD.COM>,
Dennis Allen/STL/MASTERCARD <Dennis_Allen/STL/MASTERCARD@MASTERCARD.COM>,
dkatz <dkatz@aba.com>
From: Bill Poletti/STL/MASTERCARD
<Bill_Poletti@MASTERCARD.COM>
Date: 26 Mar 97 10:47:54 EDT
Subject: MasterCard comments regarding AES

Director, Computer Systems Laboratory
Attn: FIPS for AES Comments
Technology Building
Room A231
National Institute of Standards and Technology
Gaithersburg, MD 20899.

Dear Sir:

MasterCard appreciates the opportunity to respond to the AES FIPS proposal. The following represents the position of MasterCard in addition to the comments previously submitted by the ANSI X9 Secretariat.

"The use of keys should not be restricted to a single or fixed length mechanism. Key lengths should be variable with the extension of key size effectively not limited by the algorithm. Recommended minimum key lengths should be made as part of the standard, but additional key lengths should be easily implemented without the need to re-write the standard."

If you have any questions, please let me know eiher by email or a call.

Bill Poletti
Manager, Information Security, Cryptography
MasterCard International, Inc.

==========

27 March 1997

Director, Computer Systems Laboratory
Attn: FIPS for AES Comments
Technology Building, Room A231
National Institute of Standards and Technology
Gaithersburg, MD 20899

RE: Advanced Encryption Standard

Dear Mr. Director:

This letter is in response to your request for comments on the proposal to develop a FIPS for an Advanced Encryption Standard (AES). As author of Applied Cryptography and cryptography consultant to dozens of different hardware and software companies, I feel I have a good inkling of what the commercial community needs from an AES.

In general, I think it is an excellent idea for NIST to oversee the development and adoption of a standard enryption algorithm to replace DES. While DES is an excellent encryption algorithm, its key length is clearly too small for today's security needs. The commercial security community is unlikely to converge on a single replacement algorithm on their own, and a new NIST standard will go far to increase public confidence in cryptography.

I agree that the AES should be a publicly defined algorithm, but I hesitiate to require it to be a block cipher. I find that most of my clients are satisfied with triple-DES if they need a block cipher. The biggest hole in my array of good algorithms is a fast stream cipher with a low gate count. Steam ciphers can probably run about ten times faster than comparible block ciphers; it makes more sense for triple-DES to continue to be used for applications where a block cipher is required, and that the AES address the bulk-encryption problem with a suitable stream cipher.

If a block algorithm is required, I suggest requiring a 128-bit block. The current generation of 64-bit block ciphers are becomming more and more vulnerable to attacks based on the block size.

I agree that the AES should have a variable key length and implementable in both software and hardware. However, I strongly feel that any government-endorsed algorithm should be available free for all uses, like DES is. Patented algorithms should not be considered, unless the patent-holder is willing to grant free world wide rights as IBM did with DES.

With regards to your factors for judging, the issues are far more subtle than your list indicates. "Computational efficiency" and "hardware and software suitability" are very complex metrics. You have to differentiate between the time required to set up a key with the time required to encrypt an amount of plaintext after key setup. Some algorithms have very efficient key-setup routines; others are very slow. This efficiency and suitability also depends strongly on the type of processor. An AES will be used on platforms ranging from 8-bit microprocessors on smart cards to 64 bit Alpha workstations, as well as platforms that haven't been developed yet. On hardware, speed is often a function of gate count. Algorithms can often run very quickly if they are implemented in a large number of gates, and slowly if they are implemented in a small number of gates.

I feel that efficiency should be judged on slow 8-bit platforms. The desktop machines are getting faster every year; almost any algorithm is efficient on those platforms. I recently wrote a paper on fast implementations of algorithms on Pentium machines; the number of clock cycles required for encrypting a single block of plaintext (20 clocks per byte encrypted for Blowfish; 24 cocks per byte encrypted for CAST) were remarkably close. A factor of 4 or 5 is not very much when processor speeds double every 18 months. The low end, however, will always be with us, and it is constrained both in processor power and available RAM. And as cryptography becomes more of a consumer item, it will find its way more into the low end.

Hardware efficiency should be judged on the basis of flexibility: the algorithm should be implementable in both small-gate-count and large-gate-count variants, with appropriate variances in speed.

In any case, there should be clearly-defined minimum acceptability requirements for efficiency. I suggest the following:

> Encryption no slower than DES on any platform (e.g. at most 360 clock cycles per block on a Pentium).
>
> Key setup no more than 5 times the speed to encrypt one block.
>
> Encryption and decryption speeds within 10% of each other.
>
> Implementable in hardware with a total table size of less than 256 bytes.
>
> Hardware enryption throughput of one block per clock cycle (given enough gates), with a maximum encryption/decryption latency of 16

clock cycles.

Minimum RAM requirements (RAM only, not code or tables) of no more than 64 bytes on an 8-bit smart-card processor.

Software implementation should favor little-endian machines.

With regards to your draft submission requirements, I suggest that you provide standard function calls in ANSI C that the software implementation should conform to. This will greatly simplify comparison testing, by providing a standard interface for comparison. These calls should test bulk encryption as well as key-setup.

You should also require test vectors (possibly a standard suite) that can be used to verify any implementation of the algorithm, as well as a copyright-free reference implementation. And in addition to a cryptanalysis of the algorithm, you should require an explanation of the design rationale.

Finally, I think we need to think more about the process of evaluation. Assuming you are looking to choose an algorithm in 1998, any set of candidates will only get a year or so of analysis before a choice is made. Unless you are sure that an existing block cipher with more than a couple-years' analysis (i.e. triple-DES, IDEA, Blowfish, RC5, Khufu, CAST, and SAFER) meets your requirements, this is far too short a time to develop and analyze a new algorithm. Perhaps it might be smarter to adopt triple-DES as a short-term fix, and spend the next few years developing a completely new algorithm for long-term use.

The benefits of this approach is that we can then develop an algorithm with all sorts of useful features not generally present in the list of algorithm suggested above:

Both block modes and a stream modes, with the steam modes at least five times faster than the block modes.

A standard hash-function mode. (While I understand that SHA-1 is NIST's standard for hashing, many hardware modules cannot afford the silicon necessay to implement SHA-1. If they are already using AES, they will want to use it for hashing as well.)

A standard MAC (Message Authentication Code) mode.

A mechanism for improving the algorithm, in the field, in the event that an unforseen weakness is discovered after approval.

Variablility in the algorithm to provide a family key for different applications. (Sometimes companies want their algorithm to be proprietary in some way; it makes sense to give them a harmless way to do this.)

In any case, you should consider finding a panel of cryptanalytic experts to quickly weed out bad candidates, spending money on public cryptanalysis of the top contenders, and offering bounties for successful cryptanalyses of top contenders. I suspect you will get many algorithms that are not

worthy of serious consideration, and eliminating them quickly will allow the serious contenders to receive more analysis.

I applaud your efforts to develop a new encryption standard, and I look forward to attending your AES Criteria Workshop on 15 April 97 to futher discuss these issues.

Sincerely,

/s/ Bruce Schneier

==========

(From RSA Laboratories)

March 28, 1997

The Director
Computer Systems Laboratory
Attn.: FIPS for AES Comments
Technology Building
Room A231
National Institute of Standards and Technology
Gaithersburg, MD 20899

Dear Director,

It is with great interest that RSA Laboratories has noted the intended development of a forthcoming Advanced Encryption Standard.

The existing draft Minimum Acceptability Requirements and Evaluation Criteria cover many important issues and offer good working guidelines to some of the issues that must be addressed during the review of a submitted algorithm. We note, however, that the draft listed in the Federal Register on January 2, 1997 leaves various aspects of the process unspecified.

We expect that many of the outstanding detailed technical considerations will be discussed at the meeting of April 15, 1997. At this point, therefore, we would like to highlight some issues that might usefully be noted prior to that meeting. All are general in nature and most are related to issues of procedure.

1. The exact aim of the Advanced Encryption Standard should be clarified.

    •Is it intended that proposals for the Advanced Encryption Standard be judged solely on their merit as encryption algorithms? While block ciphers are primarily used for encryption, they are often used in other ways. As an example, block ciphers have been used as the basis for hash functions and as a building block in the process of providing message authentication codes. It would be valuable to specify at the outset exactly what the uses and applications of the forthcoming Advanced Encryption Algorithm are

intended to be. By doing this, criteria more relevant to an application not under direct consideration can be separated from those used to assess an algorithm for it's intended roles.

2. The anticipated timetable and procedures adopted during the development of an Advanced Encryption Standard should be clearly defined.

• Is the aim to have an Advanced Encryption Standard in place by 1998, so that it can be immediately deployed as a replacement to DES? If so, the whole effort will be under a great deal of time pressure.

• Is there adequate time to allow for a thorough analysis of the different submissions?

• What happens if all the submissions are rejected as being inadequate? Will there be a second call for submissions? Will an alternative algorithm be proposed for immediate adoption so as to avoid a period during which no encryption algorithm would be approved for government use?

3. It can be anticipated that several algorithms will satisfy certain Minimum Evaluation Requirements and that further, more detailed, evaluation will be required.

•Which groups or persons will be responsible for this advanced assessment?
•Will the results of this assessment be made public, including any cryptanalytic techniques that might have been used in assessing the security of a proposed cipher?
•Will the open cryptographic community be involved in the process of assessing all aspects of the suitability of some cipher?
•It is very likely that some aspect of a candidate algorithm might appear to be in conflict with another. A common example would be that of security and performance. How is it intended that any potential conflict between a "fast but sufficiently-secure" cipher and a "secure but sufficiently-fast" cipher be resolved?
•It is very important that a clear prioritization between the possible attributes of a submission be established ahead of time. It might in fact be desirable to consider quantifying and setting certain minimum goals that a candidate algorithm must surpass. Perhaps a specific target speed of encryption for different environments should be set, along with some minimum goals on the amount of data required for a successful cryptanalytic attack along with a minimum required work effort to recover a key.
•How will the final decision on which submission is the most suitable be made?

4. There is some ambiguity in some of the terms used in the current draft Minimum Acceptability Requirements and Evaluation Criteria.

•Terms and criteria such as "simplicity", "hardware and software suitability" and "statement of computational efficiency in hardware and software", for example, are vague and open to a great deal of conflicting interpretation. We would suggest that the final Minimum Acceptability Requirements and Evaluation Criteria and the final Submission Requirements be more specific in such issues so as to avoid misinterpretation.

5. Some issues with regard to the assessment of an algorithm might be considered for inclusion in the Minimum Acceptability Requirements and Evaluation Criteria.

• In the evaluation of any algorithm, it is important to recognize whether the algorithm has already been the subject of widespread scrutiny and analysis. In this way, the algorithm might already have achieved a certain level of trust, thereby making the process of acceptance by the cryptographic community that much easier. As a consequence, a list of related and published cryptanalytic work would make an essential addition to any

algorithm submission. It is also important to realize that there is a great deal of difference between whether an algorithm has been available in the public domain for a number of years and whether it has, in fact, been the subject of a serious cryptanalytic effort.

•An algorithm submission should be accompanied with a full explanation of the design rationale used for the algorithm. In addition, the method used to generate any cryptographic S-boxes, and the method used in choosing any constants or other terms that appear in the algorithm should be described.

The issues highlighted in the points above are merely representative of some of the more detailed questions we have on the draft Minimum Acceptability Requirements and Evaluation Criteria as they have been published.

We anticipate that the meeting of April 15, 1997 will be very useful in addressing many of the issues we have raised. We also expect it to be very productive in setting out more of the details required in assessing the relative merits of any proposal for the forthcoming Advanced Encryption Standard.

Yours sincerely,

Matthew J.B. Robshaw

==========

Return-Path: <Dan_Nessett@3mail.3com.com>
X-Sender: dnessett@tdc.3com.com
Date: Mon, 31 Mar 1997 17:09:48 -0800
To: AES@nist.gov
From: Dan Nessett <Dan_Nessett@3mail.3com.com>
Subject: Evaluation criteria for the advanced encryption standard

To : The Director, Computer Systems Laboratory
National Institute of Standards and Technology

Dear Sir,

Please accept the following evaluation criteria for the Advanced Encryption Standard currently under study by NIST. They are the result of input from various senior engineers and managers at 3Com Corporation :

1. The most important criteria that will affect acceptibility of the standard by commercial vendors is the availability of an exportable version (perhaps tied to key length). Large corporations, such as 3Com, not only ship a large percentage of their products overseas, they also have engineering divisions outside of the United States that work on products requiring cryptographic services. In addition, the engineering organizations, manufacturing facilities and sales channels of such corporations are not designed to differentiate between domestic and international versions of a product.

It must be possible to design, manufacture, sell, deploy and maintain a product that uses cryptography in a single version that can be shipped both domestically and internationally. Achieving these objectives must not

increase the cost of the product substantially (a maximum of 1-2% for minimally acceptible implementations). In addition, the algorithm should allow implementors to trade-off cost versus strength of protection.

To achieve these objective, the algorithm should be designed to support a wide range of cryptographic strengths and allow a single implementation to be tailored in the field for the strength appropriate for particular markets. Such tailoring cannot introduce significant engineering, manufacturing, marketing, sales, deployment or maintenance costs.

2. The standard should be tailorable to a wide variety of applications and implementation platforms. The algorithm should be suitable for high asset value applications, such as business strategic planning and financial transactions as well as low asset value applications, such as protecting casual communications between individuals.

In addition, the standard should be appropriate for implementation in high performance computing equipment as well as mobile or hand-held platforms, such as Personal Digital Assistants. The algorithm should lend itself to efficient (both in terms of gate count and total delay from input to output) implementations in hardware. The algorithm should be structured so that its components could be used for other purposes, such as computing a message authentication code, one-way hash, or digital signature.

Finding a single algorithm with the requisite flexibility, efficiency and strength may be difficult, but is a primary requirement of a useful Federal Encryption Standard. One way to meet this goal would be to choose an algorithm that scales in terms of strength, implementation complexity and (degradation of) performance as the key size increases. The algorithm should have properties (e.g., not form a group under composition), so it can be applied in multiple super-encryption modes, such as triple-DES.

3. The lifetime of the algorithm should be sufficient to justify its deployment. At a minimum, the algorithm should be useful in some form for at least 20 (preferably 30) years after acceptance.

4. The algorithm will be used to provide such services as message authenticity, integrity and confidentiality. Consequently, it will not only be used alone, but also in tandem with other cryptographic algorithms, such as message authentication codes and digital signatures. The algorithm should be designed to compliment these other cryptographic services to the best extent possible.

5. Certain applications, such as email and computer communications, carry content that is variable in sensitivity. The current approach is to protect the content as if all of it had a sensitivity equal to its most sensitive parts. This usually implies higher computational costs, and thus, lower performance than may be necessary.

The standard should allow the interleaving of ciphertext produced by different gradations in the strength of the cryptographic algorithm. Important characteristics in this regard are fast rekeying, and efficient accommodation of different key lengths by a single implementation. If the computational work to produce internal state from an input key is large as compared to the computational work to encrypt a plaintext block, the

algorithm should allow implementations to quickly protect and dump that state to external storage and allow them to quickly and efficiently restore that state at a later point in time. The speed and efficiency of these operations should be measured relative to the speed and work required to encrypt a single block of plaintext. Furthermore, the amount of memory required to represent the internal state associated with a key should be no more than 100 times the key length.

Respectfully yours,

Dan Nessett

==========

Return-Path: <100142.1670@CompuServe.COM>
Date: 01 Apr 97 17:29:01 EST
From: Steve Mathews <100142.1670@CompuServe.COM>
To: AES review <AES@nist.gov>
Subject: Comments

We wish to submit the following comments and observations on the AES proposal.

AES acceptability requirements and evaluation criteria

Comments upon current criteria.

A.2 AES shall be a symmetric cipher which may operate in block mode or in stream mode.

A.3 AES shall be designed such that the key length is alterable such that an increase in the key length equates to an increase in the overall security (i.e. the effort required to cryptanalize),

A.6.b) computational efficiency having regard to the security (i.e. the effort required to cryptanalyze), in both hardware and software,

It is not clear if the criteria in A.5 b) would also meet the ISO/IEC criteria, and thus avoid export and/or problems in other jurisdictions.

The evaluation criteria put security as the first criterion and computational efficiency second. Risk analysis restates Juvenal's comment "Omnia Romae cum pretio," literally 'anything may be had for a price'. There must be a trade of these two, (the others are corollaries of the main price), so they should be considered first among equals rather than first and second.

We are intending to provide a representative to the Gaithersburg meeting.

Kind regards.

Steve Mathews
PCSL, Dallas.

==========

Director, Computer Systems Laboratory
Attn: FIPS for AES Comments
Technology Building, Room A231
National Institute of Standards and Technology
Gaithersburg, MD 20899

Trusted Information Systems, Inc. appreciates the opportunity to provide comments on the draft minimum acceptability requirements for an Advanced Encryption Standard (AES) that NIST has published as a first step in development of a new Federal Information Processing Standard, pursuant to its responsibilities under the Computer Security Act of 1987, the Information Technology Management Reform Act of 1996, Executive Order 13011, and OMB Circular A-130. Public visibility and input are critical factors for the success of this undertaking. By including a mechanism for public comment and other inputs at the beginning of the FIPS development process, NIST's first steps are in the right direction in this regard.

Overall, the draft requirements comprise a reasonable starting point for identification of suitable candidate algorithms. Our comments focus on four areas that merit further attention:

· importance of public scrutiny of the candidate algorithm(s)
· availability of algorithm(s) for worldwide use
· need for a multiyear transition period
· concerns over requirement for variable key length

Importance of Public Scrutiny

The solicitation for comments implies an open process will be used for development of the new FIPS, including public scrutiny of candidate algorithms. Adequate public scrutiny of candidate algorithms, as well as public review and critiquing of testing and evaluation results, will be crucial to public acceptance of and confidence in the NEA.

In effect, the "DES model" should be used for selection and approval of the new standard. Although iterative public inspection and will lengthen the time required to promulgate the new FIPS, the extra time and care at the front end will result in a standard with a longer useful lifetime and wider utilization. The durability of the Data Encryption Standard and its international acceptance (despite initial public skepticism) is due to the thorough public scrutiny it underwent.

The lessons of history with respect to the promulgation of cryptographic FIPS are: (1) mistrust and suspicions fostered by lack of public visibility and participation cannot be overcome by the technical quality of an algorithm and (2) for the purposes of building trust and user acceptance, "vetting" of a proposed standard by limited-membership bodies cannot replace open public inspection

Availability for Worldwide Use

Even though the NES will be promulgated for government use, the reality of the Global Information Infrastructure is that it will come to be used by various communities within the United States and beyond its borders. Therefore, the candidate algorithm(s) must be available for worldwide use.

Multiyear Transition Period

NIST has correctly anticipated that "a multi-year transition period will be necessary to move toward any new encryption standard." Allowing sufficient time for a graceful phase-in of the NES will allow users to recoup the investment that have made in DES and enable producers to gear up production of products implementing the new FIPS algorithm(s). However, because DES has proved to be so popular and durable and has been adopted in numerous other standards, the time remaining until the end of the present certification period is unlikely to be "sufficient time" for a graceful transition to a brand-new FIPS.

Requirement for Variable Key Length

Criterion A.3 states, "AES shall be designed so that the key length may be increased as needed." The requirement for a variable key length is cause for concern, in that it may preclude alternatives like Triple DES from consideration as a candidate for the new FIPS. It is not clear that the objective of a durable new FIPS necessarily requires an adjustable key length.

==========

Return-Path: <tmcdermo@missint.missilab.com>
Date: Wed, 02 Apr 1997 14:37:00 -0500 (E)
From: "McDermott, Thomas" <tmcdermo@missint.missilab.com>
Subject: NSA comments on criteria for AES
To: NIST <AES@NIST.GOV>
Cc: Brian Snow <bsnow@radium.ncsc.mil>
Encoding: 66 TEXT

Director, Information Technology Laboratory, NIST

In accordance with our technical advisory role under the Computer Security Act of 1987, we are pleased to offer the following in response to your call for comments on "Proposed Draft Minimum Acceptability Requirements and Evaluation Criteria" for an Advanced Encryption Standard, as published in the Federal Register of January 2, 1997.

The National Security Agency's Information Systems Security Organization strongly supports your proposal to develop a FIPS for an advanced encryption algorithm using a public process and welcomes the opportunity to comment.

While we believe any algorithm can be implemented in hardware or software, certainly computational efficiency is an important consideration; we suggest that minimum specified requirements in this area should be detailed. For example, we recommend that hardware implementations of the selected algorithm must be able to encrypt data at a minimum of 1 Gb/s, pipelined if necessary, in existing technology.

Another additional important factor is key agility; that is the ability

to rapidly change cryptovariables so as to simultaneously support multiple processes in applications such as ATM. Here, we believe, a goal should be that two blocks could be enciphered with different keys in virtually the same time as two blocks could be enciphered with the same key.

Finally, given the requirement for a symmetric block cipher algorithm, we recommend the consideration of a 128 bit block size supporting multiple modes including CBC, ECB, and counter driven modes.

We feel strongly than any algorithm selected should be patent free and/or available to all users free of charge. Patented algorithms should not be considered unless the patent holder is willing to grant free usage as was the case with the adoption of DES.

Finally, in regard to algorithm flexibility, we caution that the more design parameter value combinations allowed, the more difficult it is to evaluate the security of the algorithm and to enable interoperability across a broad range of users and supporting protocols. Ideally, a fixed width for the codebook, a fixed number of steps, and a fixed key length would make for the easiest and quickest evaluation and promote greater interoperability.

If some of these parameters must vary, we point out that the full set of permissible value combinations must be specified, understanding, of course, that each point in the design parameter space yields a distinct algorithm for evaluation.

My point of contact for technical discussions is Brian Snow, INFOSEC Technical Director. He can be reached at (301) 688-8199, (301) 688-3090 facsimile, or bsnow@dockmaster.ncsc.mil.

THOMAS J. McDERMOTT

Deputy Director
for
Information Systems Security
National Security Agency

==========

Return-Path: <romeror@FRB.GOV>
From: romeror@FRB.GOV
Date: Wed, 2 Apr 1997 16:35:12 -0500
To: AES@nist.gov
Cc: romeror@FRB.GOV
Content-Description: cc:Mail note part

April 2, 1997

Dr. Chukri A. Wakid
Director, Computer Systems Laboratory,
Attention: FIPS For AES Comments
Technology Building, Room A231
National Institute of
Standards and Technology
Gaithersburg, MD 20899

Dear Sir,

This letter is in response to the request for comment issued by the National Institute of Standards and Technology (NIST) on the Advance Encryption Standard (AES) draft acceptability requirements and evaluation criteria. Over the past 20 years, the financial services industry has been well served by the current Data Encryption Standard (DES). Technological developments, however, necessitate establishing a more secure standard. The Federal Reserve endorses NIST's current efforts to establish an Advance Encryption Standard and support the open and collaborative approach in which this is being accomplished. We hope you find the comments listed below beneficial in finalizing the AES acceptability requirements and evaluation criteria.

1. Requirement A.2 does not qualify the block size. We recommend that A.2 should qualify a selectable block size or a block size of at least 128 bits.

2. Requirement A.3 will have the effect of disqualifying Triple DES as an AES alternative. We recommend that A.3 be revised to include a minimum key length as an alternative to the ability to increase the key length.

Requirement A.3 should also qualify whether the block size must also be increased to correspond with an increase in the key length.

Requirement A.3 should also specify that parity bits should not be used for keys. The use of these bits imbeds redundancy inside a key, which may be used to cryptanalyze a key. The parity bits also pose interoperability problems based on the different ways that communicating systems use these parity bits. There is anecdotal evidence in the financial industry concerning implementation delays and costs associated with the use of parity bits for keys. No corresponding benefit for the use of parity bits for keys has been documented. Therefore, symmetric key definitions should contain only key bits.

3. Requirement A.5 should also reference the International Standards Organization's patent policy.

4. The seven criteria listed in A.6 (a through g) should be combined into three evaluation criteria. Moreover, the evaluation criteria should be listed in order of importance. Provided below are the evaluation criteria we would recommend listed in order of importance:

a) Security - the strength of AES to withstand cryptanalysis or exhaustive key search. (Includes criteria A.6.a)

b) System Feasibility - the ability to design, develop, implement, and operate the AES on a variety of industry platforms such as ATMs, Point of Sale Devices, User Workstations and Servers, and Mainframes, based on a variety of devices such as smart cards, PCMCIA cards, PC boards, and integrated encryption engines. (Includes criteria A.6.d, e, and f)

c) Cost - total cost of the AES based on licensing fees, computational efficiency, and memory requirements. (Includes criteria A.6.b, c, and g)

5. The AES algorithm should have an associated National Voluntary Laboratory Accreditation Program (NVLAP) or other initiative that provides a mechanism for validating AES implementations.

6. Ideally only one algorithm should be selected as the AES. However, there are many uses that AES will be subject to, such as, message authentication, block data encryption, stream data encryption, symmetric key encryption, random number generation, nonce generation, and key-derivation techniques. A single algorithm, however, may not serve all these requirements effectively. Consequently, NIST may want to consider selecting more than one algorithm based on performance for different applications.

7. Given the significant install base of DES, NIST should also consider features that would allow for a seamless and cost effective transition to a new standard.

8. NIST should consider establishing a timetable for algorithm selection and testing. Moreover, a one year public scrutiny period should be instituted following the selection of an algorithm so flaws can be identified and corrected before the standard is finalized.

9. Consideration should be given to the transition time for the financial services industry to migrate to AES. It is anticipated that the time required to migrate to a new encryption algorithm may take five to seven years after the standard is finalized. During this transition period, DES will continue to play an important role in protecting information. Moreover, many in the financial services industry my utilize a variation on DES, such as Triple DES (ANSI X9.52), during this transitional period as a means of reinforcing existing cryptography infrastructures. The Federal Reserve currently uses DES and is analyzing the use of Triple DES. Therefore, the Federal Reserve encourages NIST to continue support for DES during this transitional period. Attached is a press release from the Federal Reserve related to its evaluation of Triple DES.

If you have any questions about these comments, please contact Mr. Raymond Romero at (202) 452-6474 or via E-mail at romeror@frb.gov.

Sincerely,

/S/

Clyde H. Farnsworth, Jr.

Attachment

For Release: Contact:
April 2, 1997 Joe Elstner, St. Louis - (314) 444-8902
Sandra Conlan, San Francisco - (415) 974-3231
Gwen Byer, Richmond - (804) 697-8105

Federal Reserve is Evaluating Triple DES

ST. LOUIS--The Federal Reserve is evaluating an advanced application of the Data Encryption Standard (DES), known as Triple DES, to protect data that are transmitted electronically between the Federal Reserve Banks and between the Federal Reserve and financial institutions. Federal Reserve officials said that if the new standard proves effective, an announcement about actual implementation can be expected in early 1998.

The Federal Reserve is an active participant in the X9 committee of the American National Standards Institute (ANSI), which is completing a standards document for Triple DES. "Our active role in developing improved data security techniques, of which Triple DES is one component, helps provide assurance that transactions with the Federal Reserve will continue to be safe and secure from cryptographic crime," said Bruce J. Summers, director of automation resources for the Federal Reserve. "This year we will be testing Triple DES and working on an implementation plan, coordinating with vendors of encryption products and our customers."

The Federal Reserve currently uses DES to secure electronic information and will spend the next several months completing its analysis of Triple DES. "Triple DES significantly increases data security because it invokes DES three times," Summers said. "With each iteration, it is possible to use a different encryption key value, which results in a longer overall key value that is far more resistant to attack." Certain Triple DES operating modes are also compatible with the Fed's current DES implementations, which will ensure a smoother transition for Federal Reserve customers.

The Fed is also following a National Institute of Standards and Technology (NIST) project to develop an advanced encryption standard to eventually replace DES. Summers believes that, while the Fed should closely monitor such activities and study other options being developed, it must be at the forefront of data security

implementations and be prepared to use Triple DES to provide continued security until a new standard is ready. "Our evaluation of Triple DES is a continuation of the Fed's efforts to ensure that the highest levels of security are applied to Federal Reserve operations and payment services," said Summers.

==========

Return-Path: <Cadams@entrust.com>
From: Carlisle Adams <Cadams@entrust.com>
To: "'AES@nist.gov'" <AES@nist.gov>
Subject: Comments on AES Criteria...
Date: Wed, 2 Apr 1997 18:35:22 -0500
Encoding: 160 TEXT

Entrust Technologies

2 Constellation Crescent

Nepean, Ontario, Canada

K2G 5J9

> 2 April 1997
>
>
>Director, Computer Systems Laboratory,
>Attn: FIPS for AES Comments,
>Technology Building, Room A231,
>National Institute of Standards and Technology,
>Gaithersburg, MD 20899
>
>Dear Director:

Please find below comments with respect to the "Proposed Draft Minimum Acceptability Requirements and Evaluation Criteria" which was published on January 2nd of this year. It is our understanding that these comments will be made part of the public record.

Sincerely,

Carlisle M. Adams, Ph.D.

Senior Cryptographer,

Entrust Technologies

cadams@entrust.com

>PROPOSED DRAFT MINIMUM ACCEPTABILITY REQUIREMENTS AND EVALUATION CRITERIA
>
>The draft minimum acceptability requirements and evaluation criteria are:
>
>A.1 AES shall be publicly defined.
>
>A.2 AES shall be a symmetric block cipher.
>
>A.3 AES shall be designed so that the key length may be increased as needed.

A.3 AES shall be designed so that the key length may be increased as needed (up to some appropriate maximum).

>A.4 AES shall be implementable in both hardware and software.

A.4 AES shall be economically implementable in both hardware and software.

>A.5 AES shall either be a) freely available or b) available under terms >consistent with the American National Standards Institute (ANSI) patent policy.

A.5 AES shall be freely available.

A.6 AES shall be amenable to short messages and to environments in which keys are changed frequently (i.e., any set-up time required for the algorithm, prior to encryption/decryption, shall not be prohibitive).

>A.6 Algorithms which meet the above requirements will be judged based on

A.7 ...

>the following factors:
>
>a) security (i.e., the effort required to cryptanalyze),
>b) computational efficiency,

b) computational efficiency (particularly in software and firmware),

>c) memory requirements,
>d) hardware and software suitability,
>e) simplicity,
>f) flexibility, and
>g) licensing requirements.
>
>Comments are being sought on these draft minimum acceptability criteria and
>evaluation criteria, suggestions for other criteria, and relative importance
>of each individual criterion in the evaluation process. Criteria will be
>finalized by NIST following the criteria workshop.
>
>PROPOSED DRAFT SUBMISSION REQUIREMENTS
>
>In order to provide for an orderly, fair, and timely evaluation of candidate
>algorithm proposals, submission requirements will specify the procedures and
>supporting documentation necessary to submit a candidate algorithm.
>
>B.1 A complete written specification of the algorithm including all
>necessary mathematical equations, tables, and parameters needed to implement
>the algorithm.
>
>B.2 Software implementation and source code, in ANSI C code, which will
>compile on a personal computer. This code will be used to compare software
>performance and memory requirements with respect to other algorithms.

B.2 Software implementation and source code, in ANSI C code, which will compile on an IBM-compatible personal computer. This code will be used to compare software performance and memory requirements with respect to other algorithms.

>B.3 Statement of estimated computational efficiency in hardware and software.

B.3 Statement, with sufficient justification, of estimated computational efficiency in hardware and software (or specific performance figures, if these are available).

>B.4 Encryption example mapping a specified plaintext value into ciphertext.

B.4 Encryption example mapping a specified plaintext value into ciphertext (i.e., test vectors showing expected ciphertext for given plaintext/key pairs so that implementations can be verified for correctness).

>B.5 Statement of licensing requirements and patents which may be infringed

>by implementations of this algorithm.
>
>B.6 An analysis of the algorithm with respect to known attacks.

B.6 A detailed analysis of the algorithm, including published papers evaluating the strength of the algorithm with respect to known attacks.

>B.7 Statement of advantages and limitations of the submitted algorithm.

==========

Return-Path: <alpoplove@its.cse.dnd.ca>
From: "A. L. Poplove" <alpoplove@its.cse.dnd.ca>
To: "'AES@nist.gov'" <AES@nist.gov>
Cc: "A. L. Poplove" <alpoplove@its.cse.dnd.ca>
Subject: FW: comments on AES proposal
Date: Tue, 8 Apr 1997 16:44:56 -0400

Dear Dir/ITL

In accordance with your request, I would like to re-submit the following comments with regard to the AES proposal. Ted Elliott, a colleague of mine at CSE, will submit his comments separately.

Thanks, Alan Poplove Cryptomath UnitHead , Communications Security Establishment, Ottawa, Canada

Regarding A.3
There is some ambiguity about what is meant by "AES shall be designed so the key length may be increased as needed"; i.e. does NIST mean that the users can simply choose their key spaces with an algorithm which allows this variation; or does it mean that the AEA does have a fixed key space, and that future versions/upgrades of the AEA may have larger key spaces? I favour that latter.

Regarding A.6 (a) and B.6; "Algorithms to be judged on Security..." and a submission requirement of "an analysis of the algorithm with respect to known attacks"
Algorithms should be shown, in a mathematically-explicit manner, by the submitting party to be resistant to all potential cryptanalytic attacks. However, as a condition of submission, it should be recognized that the security evaluation may include classified analysis, the results of which may cause an otherwise attractive contender to be dismissed. It should be accepted that the results of any classified analysis will not be released.

Regarding A.6 (g) and B.5: We recommend that AES should be available with a free license to anyone implementing it.

Regarding A.6 (f) "Flexibility" should be defined. Does this refer to supporting multiple uses?

==========

Comments Submitted to NIST Regarding the AES Draft FIPS

I wish to submit the following comments on the AES draft FIPS for NIST. These are my personal recommendations and shall not be taken to represent either the official CSE view nor the official view of the Government of Canada, at this time.

1. U.S. Patent 5,559,993 issued 24 Sept '96 to our Minister of National Defence provides, in hardware, the ability to lock any software, for example, the subject i.e. AES algorithm(s), i.e. AEA, complete, if desired with any related AEA keymat, and/or AEA key generation software, behind read-ONLY hardware technology, the subject of my/our invention. Corresponding Canadian and European patent protection is also assigned to our Minister. An MFM prototype, an IDE, SCSI, and MFM prototype have been built. A further commercial design is being studied for cost and feasibility by one of our allies for a law enforcement application. Our Minister may wish to consider the utility of allowing this technology to be released under suitable terms, for any specified U.S. and/or international use, within such an AES FIPS framework. This is related to Section A.5 of the draft.

2. My paper also refers to this subject, which was published in the Proceedings of the NIST/NCSC 17th National Computer Security Conference, October 11-14, 1994, Vol 1, pages 274-282.

3. Entrust (TM) digital signature, with the corresponding PKI infrastructure of the Government of Canada, may provide an additional mechanism for both wrapping the subject i.e. AES algorithm(s), i.e. AEA complete, if desired with any related AEA keymat and/or AEA key generation software, to assure its integrity, and verify its integrity at any time after creation, forming part of the security of any proposed AEA implementation under the AES.

4. Section A.4 as written does not appear to cover any proposal in software which is implemented "behind" such hardware integrity control. This section implies a proposal must be implement/able/ed in BOTH software and hardware, as presently worded, yet I don't think that was intended. Does AES as drafted include keymat and does it include session generating software code?

5. Section A.6 subsections e), f), g) are assured in my view by our hardware device of item 1. above.

6. Evaluation methodology in this current draft is not in my view sufficiently described to take account of this high integrity

protection approach for the cryptographic components of AES and AEA.

7. I will give my personal support to Mr.
Poplove during his attendance at your workshop, if there are any questions.

Thank you.

T.E. (Ted) Elliott
Tel. 613-991-7506
FAX 613-991-7411

==========

Return-Path: <sori@iss.isl.melco.co.jp>
To: Jim Foti <foti@st1.ncsl.nist.gov>
Cc: etakeda@iss.isl.melco.co.jp, atsuhiro@iss.isl.melco.co.jp,
matsui@iss.isl.melco.co.jp, kondo@syskai.hoku.melco.co.jp,
WBoyles@MSM.mea.com, sori@iss.isl.melco.co.jp
Subject: Re: Preliminary agenda for AES workshop
Date: Thu, 10 Apr 1997 23:30:46 +0900
From: "T.SORIMACHI" <sori@iss.isl.melco.co.jp>

Dear Mr.Jim Foti:

My name is Tohru Sorimachi.

This E_Mail is comments of
"PROPOSED DRAFT MINIMUM ACCEPTABILITY REQUIREMENTS
AND EVALUATION CRITERIA"

1. In order to implement variable key length in H/W,
I think it is required to specify maximum key length to be extended
and key length notch to be increased.

e.g. key length: from 128bits to 256 bits max.
key notch : 32bits i.e. 128bits,160bits,....,224bits,256bits
as a sequence.

We agree with all other criteria.

2. Since AES will be the world wide standard at least as a defacto,
please allow to submitt the proposals as a criteria and also of
algorithms from the other countries besides U.S.

Regards, Tohru Sorimachi.