AES Workshop

To Discuss the AES Evaluation Criteria and Submission Requirements

Miles Smid, Ed Roback & Jim Foti
National Institute of Standards and Technology

April 15, 1997

AES Workshop Goals

- Summarize Received Comments
- Discuss:

  1) comments and proposed responses

  2) proposed AES development process

  3) key issues
- Gain participants' insights
- Clarify any misunderstandings
- Address your questions
- Engage interested parties in AES process

AES Announcement of January 2

- Intent to Develop AES
- Proposed Minimum Acceptability Requirements and Evaluation Factors
- Proposed Draft Submission Requirements
- April 15 Workshop Announced
- Call for Comments (by April 2)
- Total of 33 Comments Received

Initial NIST Goals

- Strong Cryptoalgorithm for Government and Commercial Use
- Support of Standard Codebook Modes
- Significantly more Efficient than DES^3
- Variable Key Size so that security could be increased when needed
- Selected in a Fair and Open Manner
  - Publicly Defined
  - Publicly Evaluatable

General Comments on the AES Effort

- “an excellent idea...”
- “support the open and collaborative approach being taken”
- “Are you serious?”
- “public visibility and input are critical factors”
- “essential component of a national strategy for securing the computing and telecommunications infrastructure”

Part A: *Minimum Acceptability Requirements and Evaluation Criteria*

A.1 *AES shall be publicly defined*

- *Comments:*
  - AEA Computations Publicly Defined
  - All Analysis made public
  - Math. logic of table generation made public
- *Proposed Responses:*
  - AEA Computations shall be public
  - All Unclassified Analysis sent to NIST will be made public
  - Math. rationale encouraged

A.2 *AES shall be a symmetric block cipher*

- *Comments:*
  - Consider stream ciphers
  - Select optimum algorithm for each mode and application
  - Block sizes of 128 and/or 256
- *Proposed Responses:*
  - BC compatible w/ existing & well-understood DES modes
  - BC most compatible w/ existing DES applications
  - Large block sizes can result in efficient block ciphers
  - Need to specify block sizes

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

- *Comments:*
  - We agree
  - What does this mean?
  - Just use one big key size
  - Don't preclude DES^3
- *Proposed Responses:*
  - NIST is open as to what key sizes should be required (topic for discussion)
  - NIST intends to recognize DES^3 when it becomes an ANSI standard. AES needs to offer significant advantage over DES^3

A.4 *AES implementable in hardware and software*

- *Comment:*
  - All algorithms can be implemented in both hardware and software
- *Proposed Response:*
  - Agree. The purpose of this requirement was to make it clear that there could be no restrictions to hardware only or software only.

A.5 *AES either a) freely available, or b) available consistent w/ANSI patent policy*

- *Comments:*
  - Algorithm shall be available royalty free worldwide (Majority View)
  - Don't exclude the payment of royalties (Small Minority View)
- *Proposed Responses:*
  - Option 1: royalty-free world-wide
  - Option 2: weigh royalty-free submissions heavily in evaluation

A.6 *Algorithms will be judged according to: a) Security*

- *Comments:*
  - Tables should be generated in mathematical manner
  - No shortcut attacks
- *Proposed Responses:*
  - Strongly encourage public explanation of rationale for table generation
  - Submitter shall state work factor
  - All attacks below work factor will be evaluated for practicality

A.6 *b) Computational efficiency*

- *Comments:*
  - Optimize for 8-bit processors (yes and no)
  - Implement in Java instead of C
  - Specify allowable key setup time
  - Specify minimum speed requirement
  - Specify big or little endian processor
  - NIST should provide specs of its test system

A.6 *b) Computational efficiency, cont'd.*

- *Proposed Responses:*
  - Flexibility credit should be given for efficiency in 8-bit processor
  - Two submissions: Reference and Optimized
  - Flexibility credit should be given for short key setup time
  - Significantly more efficient than DES^3
  - Efficiency tests will be on little endian processor
  - Specs of NIST test system will be publicly specified in call

A.6 *c) Memory requirements*

- *Comments:*
  - Consider code size for software
  - Consider efficiency vs. memory requirements
  - Consider various processors
- *Proposed Responses:*
  - Efficiency and memory requirements will be considered for C implementation on PC
  - Submitters may also provide results for other platforms

A.6 *d) Hardware and software suitability*

- *Comments:*
  - Should make efficient for 8-bit processors
  - For hardware, should provide gate count
- *Proposed Responses:*
  - Although primary applications are for processors with larger word sizes, flexibility to run on 8-bit processors will be valued
  - Some submitters may not be able to provide gate count
  - Some submitters may provide VHDL representations

A.6 *e) Simplicity*

- *Comments:*
  - What does this mean?
- *Proposed Responses:*
  - Simplicity of design
  - Simplicity of mathematical basis for design and security
  - Ease of implementation

A.6 *f) Flexibility*

- *Comments:*
  - What do you mean?
  - NIST should define standard interface
  - Should allow variant proprietary versions
  - Fix block size, key size, and number of rounds to promote interoperability and ease of evaluation

A.6 *f) Flexibility, cont'd.*

- *Proposed Responses:*
  - Flexibility: ability to implement on differing platforms for various applications.
  - NIST will consider defining a "standard" interface for testing purposes.
  - Variant algorithms would make security evaluation more difficult and reduce interoperability. However, one could use portion of key space as variant.
  - NIST open to discussion of appropriate block and key sizes. Fix rounds for given block and key size.

A.6 *g) licensing requirements*

- *Comments:*
  - Algorithm shall be available royalty free worldwide (Majority View)
  - Don't exclude the payment of royalties (Small Minority View)
- *Proposed Responses:*
  - Option 1: royalty-free world-wide
  - Option 2: weigh royalty-free submissions heavily in evaluation

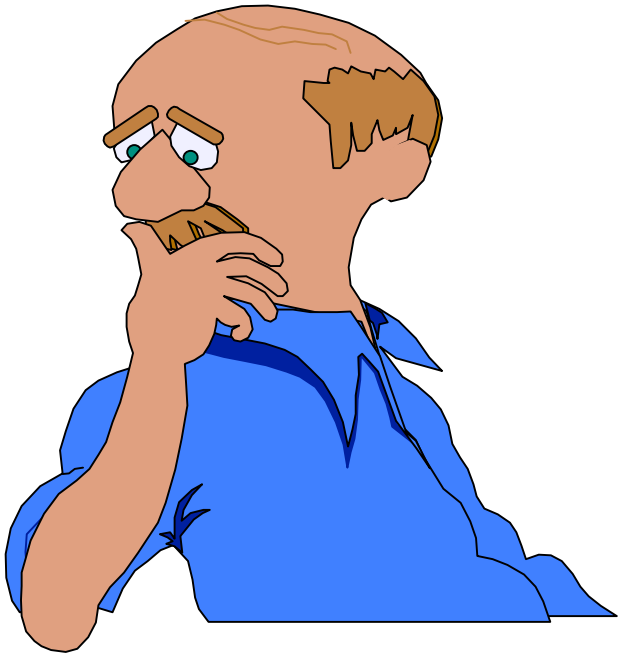
General Comments

- *Comments:*
  - Lifetime of the algorithm should be 20-30 years
  - The A.6 evaluation factors could be grouped into three categories: Security, Efficiency, and Cost
  - The A.6 evaluation factors should be ranked in order of importance
  - Submitted Algorithms should not be export controlled
  - Algorithm development should be independent of export control considerations

General Comments, cont'd

- *Proposed Responses:*
  - Agree (lifetime)
  - Agree (grouping)
  - Agree
    - Security > Efficiency
    - Efficiency = Cost
  - Export policy is beyond NIST control
  - Export laws must be complied with
  - AEA should be at least as strong as DES^3

Questions?



Part B: *Proposed Draft Submission Requirements*

(Contents of the Submission Package)

B.1 *Complete written specification of the algorithm & necessary parameters, tables, equations.*

- *Comments:*
  - Minimum values for security parameters should be specified by NIST.
  - Complete design rationale should be required.
- *Proposed Responses:*
  - Key & Block size values will be specified in the call
  - Submitter encouraged to provide non-proprietary design rationale.

B.2 *Provide software implementation & source code in ANSI C, for a PC - used for comparison of algorithms.*

- *Comments:*
  - Reference AND Optimized implementations.
  - Specify configuration to be used by NIST for eval.
  - Specify medium for submissions.
- *Proposed Responses:*
  - Reference implementation (ANSI C and/or Java?)
  - Optimized implementation (ANSI C) suitable for IBM-compatible PC running Win95, with 16MB RAM, Pentium XXMHz processor.
  - One 3.5” 1.44MB floppy for each impl. (max.)

B.3 *Statement of estimated efficiency in hardware & software.*

- *Comment:*
  - Statement should include sufficient justification or specific performance figures, if available.
- *Proposed Responses:*
  - Submitter includes efficiency estimates for various platforms, w/ specific details about each platform.
    - bytes/sec for encrypt, decrypt, key setup
    - gate count for hardware, memory requirements
  - Graph with plot of speed vs. memory
  - Used by general public to evaluate efficiency.

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

- *Comments:*
  - Monte Carlo example w/ key, input & output.
  - Submitter proposes a validation suite of examples.
- *Proposed Responses:*
  - Monte Carlo example required - specified by NIST
  - Suite of known answer tests to exercise the algorithm.
  - Allows evaluators to verify correctness of their own implementations of the algorithms.

B.5 *Statement of licensing requirements & patents which might be infringed by algorithm implementations.*

- *Comments:*
  - Submitter should address any domestic AND international patent issues.
  - NIST should assess crypto patents in cooperation with the Patent Office.
- *Proposed Responses:*
  - Call for comments on submissions will request information on ANY known patents & licensing issues pertaining to the submissions.
  - Legal research may be appropriate.

B.6 *Analysis of algorithm with respect to known attacks.*

- *Comments:*
  - Should be NO known equivalent or weak keys, or complementation properties.
  - Submitter shows why no “trap-doors”.
  - Submitter notes published cryptanalyses
- *Proposed Responses:*
  - List known weak or equivalent keys, comp. prop.
  - Can include any math. rationale for “trap-doors”.
  - Reference list of any publications that describe cryptanalysis of the algorithm.

B.7 *Advantages and limitations of the submitted algorithm.*

- *Comment:*
  - What are some examples?
- *Proposed Response:*
  - Addresses efficiency & flexibility criteria.
  - Description of features and advantages offered, with mathematical justification. For Example:
    - mathematically designed S-boxes,
    - variable key setup time
    - fast in 8-bit processors and PCs, etc.

Additional "B" Items (Proposed)

- NIST will not accept any info marked "proprietary" or equivalent (except possibly for optimized implementation).
- Submitter's Statements:
  - Submitting algorithm as a candidate with the understanding that it might not be selected for inclusion in the proposed FIPS.
  - Submitter agrees to waive copyright on submitted materials (but could maintain intellectual property interests for optimized implementation).
  - Statement of expected strength of the algorithm, with supporting rationale.

Making Submissions Public

- NIST receives submission package.
- NIST makes submission packages public.
  - Distribution will comply with U.S. export regulations.
- Public testing and evaluation begins.
- NIST may release test results from using the optimized implementations.

Proposed:
AES Development Process

DRAFT AES Selection Process

- Draft Criteria/Submission Requirements (1/2/97)
- Public Comment Process (Closed 4/2/97)
- Workshop on Criteria / Submission Requirements (4/15/97)
- NIST prepares public call for submissions (~3 mo.)

- Publication of Call for Submissions (4-6 months)
- (during open call) NIST reviews submissions for completeness (allows resubmissions/mods)
- Call for submissions closed
- NIST conducts initial review of submissions (incomplete / improper submissions rejected) (~2 mo.)
- All submissions (including incomplete/improper for the record), made public for review & analysis

- Comments accepted on all competing submissions
- (after 6 months) Interim Workshop
- NIST reviews comments and results of workshop (~3 months)
- Narrowed Candidates published
- Comments accepted on remaining candidates
- (6-9 months from narrowing) Final Workshop

- NIST reviews comments and results of workshop & drafts FIPS
- Draft FIPS published for comment (3 months)
- NIST revises draft as appropriate
- Secretarial approval

Some Key Issues for Discussion

- Block and Key Sizes
- Key Setup Time
- Hardware Efficiency/Complexity Measures
- Tweaking versus Major Changes
- Should the Optimized implementation (software) be proprietary?

Key and Block Size

Key Size:

80 128 192 256

Block Size:

64 80 128 256 512

Key Setup Time

- The shorter the better.
- Variable setup time may be best.

Hardware Efficiency/Complexity Measures

- Gate count?
- Representation in VHDL?
- Etc.?

Tweaking versus Major Changes

- Tweaking allowed
- Major changes not allowed
- What does tweaking consist of?
- Rights of submitter to control tweaks

Should Optimized Software Implementations be Proprietary/Copyrighted?

- Pros:
  - Encourages clever implementations
  - Best implementations often do not come from inventor
- Cons:
  - No withholding of information
  - Everyone could verify optimized implementation