Round 2 Discussion Issues for the AES Development Effort

The following is a list of some of the AES related issues that need to be addressed during Round 2 of the AES development effort. NIST encourages the public to provide their comments to AESround2@nist.gov.

1. How Many AES Algorithms?

There are differences of opinion in the AES community with regard to the number of AES algorithms that should be chosen. Some people favor the selection of a single AES algorithm; others favor the selection of multiple algorithms.

Some of the arguments submitted during Round 1 by the public in favor of multiple AES algorithms are:

a. Resiliency: if an AES algorithm is broken, there will be at least one more algorithm available and implemented in products. Some of the commenters have expressed the concern that the extensive use of a single algorithm would place a lot of critical data at risk if that algorithm were to be shown to be insecure.

b. Intellectual property concerns could surface at a later time, calling into question the royalty-free availability of a particular algorithm.

c. A set of AES algorithms could cover a wider range of desirable traits than a single algorithm; in particular, it may be possible to offer both high security and high efficiency to an extent not possible with a single algorithm.

Arguments have also been submitted during Round 1 by the public in favor of a single AES algorithm:

a. Multiple AES algorithms would cause interoperability problems and raise costs when multiple algorithms are implemented in products.

b. Multiple algorithms would provide multiple targets for a cryptanalyst, increasing the chance that at least one AES algorithm will be broken.

c. If any AES algorithm is broken, it would substantially decrease public confidence in the remaining algorithms.

Specific questions to consider:

- NIST has stated its goal that the AES should specify an algorithm(s) that will provide strong security for protecting sensitive data for 20-30+ years. How would

the selection of one versus multiple algorithms affect the likelihood of achieving this goal?

- If only one algorithm is selected, how will sensitive data be protected (with AES-comparable security) in the event that the AES algorithm is broken?

- What type of attack on an AES algorithm would be sufficient to "break" the algorithm? A practical attack or a purely theoretical one? That is, when is an algorithm considered to be broken? Another way to think of this is to consider what sort of attack would cause users to lose confidence in the AES algorithm(s).

- If multiple algorithms are selected, what effect would this have on interoperability? Note that there are currently multiple algorithms available which may provide confidentiality and other security services.

- If multiple algorithms are selected, how many should there be?

- If multiple algorithms are selected, what sort of guidance or standards would be useful?

2. What about the speed versus security margin tradeoff?

Consider the security margin of an algorithm to be defined as the number of rounds that is performed for that algorithm after the security threshold is reached. For example, if an algorithm is designed so that an encryption is performed using 16 rounds, and it is currently considered insecure until at least 12 rounds have been performed, 12 is the security threshold, and 16 – 12 = 4 is the security margin. Note that this assumes that essentially the same functions are performed for each round. What are the views of users as to how NIST should weigh the security versus efficiency trade-offs in making the AES selection?

3. How important are low-end smart cards and related environments when selecting the AES algorithm(s)?

Some issues have arisen about AES with respect to smart cards:

a. Implementability: Implementability applies to any memory-restricted environment. Smart cards are a primary example of such environments. In such environments, implementability is largely a function of RAM and ROM requirements alone. It is entirely possible that encryption technology will be embedded in many different kinds of restricted-memory environments, not all of which share the characteristics of today's smart cards.

b. Defense against attacks: Smart cards have been claimed to be vulnerable to timing, power analysis, and similar attacks. Operational procedures and careful implementation, combined with well designed algorithms, may make it possible to protect against attacks on low-end smart cards. However, it may also be possible to design high-end smart cards to resist these attacks with fewer operational restrictions.

c. Viability of Low-End and High-End Cards: It has been argued that low-end cards are inherently insecure, and hence that the question of whether candidates can be implemented on such cards is irrelevant. There are various other aspects of this issue that need to be explored more fully. For example, the characteristics of an algorithm (e.g., the use of certain operations such as addition or multiplication) and its implementation (e.g., the use of software balancing) may also be relevant.

d. Cost: The extra expense of purchasing a high-end smart card may be prohibitive in many cases.

Are there any other issues?

4. What is the relative importance of hardware vs. software performance in the selection of the AES algorithm(s)?

Timings were performed on the software for the algorithms during Round 1, and additional timings will be performed on the revised code on the same and different platforms during Round 2. The candidate algorithms will also be fully described in the hardware design language VHDL in order to obtain a variety of implementation and performance metrics relevant to Field Programmable Gate Arrays (FPGAs) (and possibly Application-Specific Integrated Circuits (ASICs)). What should be the relative importance of these metrics during the AES selection process?

5. What modes of operation should be available for the AES algorithm(s)?

Four modes of operation were defined for DES: electronic codebook (ECB), cipher block chaining (CBC), cipher feedback (CFB – in several sizes), and output feedback (OFB). Each mode had its advantages and disadvantages (see FIPS 81, DES Modes of Operation). Are these modes appropriate for AES? Are there other modes that would be appropriate?