Organization: ESAT, K.U.Leuven, Belgium
Date: Wed, 14 Apr 1999 23:19:11 +0200 (METDST)
From: Bart Preneel <Bart.Preneel@esat.kuleuven.ac.be>
To: AESFirstRound@nist.gov
cc: Bart Preneel <bart.preneel@esat.kuleuven.ac.be>
Subject: Some observations on the 1st round

Please find attached my comments on the Round 1 AES
Candidate Algorithms. The document format is postscript.

Yours sincerely,

Bart Preneel

Katholieke Universiteit Leuven   tel. +32 16 32 11 48
Dept. Electrical Engineering-ESAT / COSIC   fax. +32 16 32 19 86
K. Mercierlaan 94, B-3001 Heverlee, BELGIUM

bart.preneel@esat.kuleuven.ac.be
http://www.esat.kuleuven.ac.be/~preneel

Some Observations on the 1st Round of the AES Selection Process

Bart Preneel*

Katholieke Universiteit Leuven, Dept. Electrical Engineering-ESAT
Kardinaal Mercierlaan 94, B–3001 Heverlee, Belgium
bart.preneel@esat.kuleuven.ac.be

April 14, 1999

1 Introduction

This note contains a few observations on the 1st round selection process of the Advanced Encryption Standard (AES). The issues which are discussed are:

- Should we have one or more AES algorithm?
- Performance measurements.
- Key lengths and number of rounds.
- Recommendation of five finalists.
- The name AES.

2 One or More Algorithms

I believe that the outcome of the AES algorithm should be a single algorithm, that offers a good performance on a wide variety of platforms. Selecting more algorithms will create interoperability problems, and will increase costs. The price paid for this is that a single standard becomes an attractive target for attacks. But that is why security should be the first consideration in the selection process.

One can expect that independently of the AES process, several finalists will be used in commercial products anyway. It will be harder for AES to compete successfully in such an environment if AES itself consists of many algorithms.

*F.W.O. postdoctoral researcher, sponsored by the Fund for Scientific Research – Flanders (Belgium).

Only if it turns out that none of the finalists can offer the required flexibility, one should consider to recommend two algorithms. If it would have been the intention from the beginning to select more than one algorithm, the second algorithm should not have been a block cipher but an additive stream cipher.

3 Performance Issues

At the 2nd AES conference, the comment was made that the standard should not be crippled by taking into account the performance on low-end devices (8-bit processors with 128 or 256 bytes of RAM). This may be a valid argument if the main goal of the AES is to encrypt information stored on PCs and transferred over the Internet (or its successor).

However, I believe that the goal of the standard should be more ambitious, and that any inexpensive device should be capable of implementing the standard. That will reduce interoperability problems, and will make it more likely that secure encryption is used everywhere. If the AES does not fit in such environments, these devices will either not use encryption or use weaker (and/or proprietary) algorithms instead.

I would also warn against taking performance too seriously. I believe that any algorithm that fits in small devices and that falls within a factor 1.5-2.5 of the best algorithm on most processors should be considered satisfactory. The truth is that we really don't know what our computers will look like 10-15 years from now.

4 Key Lengths and Number of Rounds

There are two aspects to the security level:

- The key length in bits, which determines the (maximum) strength against **exhaustive key search**. The call for algorithms has requested at least three key lengths (128-192-256); the motivation is that this should correspond to three levels of security. Whether this is desirable or not is a separate matter, which will not be addressed here.

- The security against **shortcut attacks**. It is very hard to estimate this security level. The best we can do is study the algorithm thoroughly, and try to estimate the minimum number of rounds to resist a shortcut attack with 2^x known/chosen plaintext/ciphertexts and 2^y off-line encryptions, where x and y should be chosen 'large enough' (note that for simplicity we make abstraction of other parameters such as the storage and the amount of parallelism in the attack). For specific classes of shortcut attacks we may even be able to prove that an attack takes a certain effort, which is very desirable, but not sufficient.

  In order to take into account future developments in cryptanalysis, most AES designers have chosen x to 128, y to the key length in bits, and have added a number of rounds for extra security margin.

This is a conservative approach, which is excellent for applications that need long term security (30-50 years or even more). However, for other applications this might be overkill. Another concern is that the candidates have proposed different security margins against shortcut attacks.

It seems that calling 128-bit 'low grade' is a little strange (at least for the first 10-20 years). However, the amount of work to encrypt one block is typically independent of the key length, so it is justified to use a longer key. The best solution would probably have been to use a 256-bit key (shorter keys could have been padded with zeroes, and the key length could have been included in the key schedule algorithm) and to vary the number of rounds to trade performance and security against shortcut attacks.

Given that NIST has required three key sizes, I would make the following suggestions (the first one being more important):

1. Request that for the finalists the number of rounds increases with the key size (some designers have chosen this approach already).

2. Define also low and medium grade AES. A suggestion is included below. I would prefer that this is done by NIST rather than by others later on.

Low and medium grade AES

A first observation is that 128 bits is sufficient for most present day applications (one exception being medical data) and will be sufficient for the next 20 years (even if data has to be kept secret for 20-30 years). Hereby we assumed that "Moore's law" will remain valid for 50 years[1].

On the other hand, there are low and medium grade applications that could also greatly benefit from AES, but that may not be able to afford the performance overhead. A solution for this could be to make the number of rounds variable. I am not in favor of making this a parameter; I prefer a limited set of values. Below an example is included (here the number of chosen plaintext/ciphertexts is equal to 2^x and the number of off-line encryptions is equal to 2^y):

- low grade AES-128: 128-bit key, number of rounds equal to minimum to avoid an attack with $x = 64$ and $y = 128$ plus 25%.

- medium grade AES-128: 128-bit key, number of rounds equal to minimum to avoid an attack with $x = 96$ and $y = 128$ plus 50%.

- nominal AES-128: 128-bit key, number of rounds equal to minimum required to avoid an attack with $x = 128$ and $y = 128$ plus 100%.

- nominal AES-192: 192-bit key, number of rounds equal to minimum required to avoid an attack with $x = 128$ and $y = 192$ plus 100%.

[1] Note that a similar law seems to apply to the cost of the manufacturing installations, which raises the question who will pay for them.

- nominal AES-256: 256-bit key, number of rounds equal to minimum required to avoid an attack with $x = 128$ and $y = 256$ plus 100%.

Notes:

- The cipher would have to be designed in such a way that it is easy to vary the number of rounds. For some candidates this might be easier than for others.
- One would have to make sure that keys for these variants are differentiated within the key schedule; one solution would be to enter the number of rounds in the key schedule.
- I believe that increasing the number of options from 3 to 5 would not create major interoperability problems.
- The 'low grade' variant would have a strength against brute force attacks that is still much better than 2-key triple-DES.

5 Recommendation of Five Finalists

The algorithms which I would recommend for inclusion in the second round are:

1. Rijndael
2. Serpent
3. RC6 (but I prefer the modified key schedule presented at the 2nd AES conference)
4. MARS
5. Twofish

The sixth on my list would be E2. The main reason for recommending them is because I believe that these five candidates potentially offer the best trade-off between security and performance. I have to admit that for the security part, there is little evidence to base these conclusions on.

I want to express my concern about the fact that the evaluation of the first round has concentrated on the performance evaluation and on the security evaluation of 'weaker' candidates. This is inherent to the evaluation process: as researchers were not paid to perform specific evaluations, they tend to focus their attention on algorithms for which the probability to find a potential weakness is higher. This might be desirable from the viewpoint of NIST, because it helps to select the finalists. On the other hand, this process has taken about one year; some of this time should perhaps have been spent on analyzing the stronger candidates.

In view of this, one should give preference to candidates that are easier to analyze, i.e., that do not mix a wide variety of different operations. I believe that the first three candidates on my list have an advantage in this respect.

I am strongly convinced that leaving less than 10 months between the publication of the tweaked candidates and the deadline for 2nd round comments is *not sufficient*. This period should be at least 18 to 24 months.

6 Name

While the name AES will be hard to eradicate, I do not believe that it is a good choice. 'Advanced Encryption Standard' is an acceptable name for the next five to ten years, but will sound a little strange fifteen years from now. Unfortunately, I cannot offer any catchy alternative for the time being.