Date: Wed, 14 Apr 1999 17:11:24 +0200
From: "Daemen Joan" <daemen.j@protonworld.com>
X-Mailer: Mozilla 4.04 [en] (Win95; I)
To: aesfirstround@nist.gov
CC: Vincent Rijmen <vincent@ii.uib.no>
Subject: Public comments from the Rijndael Team

Dear,

please find in attachment a public comment of the Rijndael team. We would appreciate it if you could confirm receipt of our comments? Thanks in advance.

Kind regards,

Joan Daemen & Vincent Rijmen

AES Public Comment from the Rijndael Team

Joan Daemen, Proton World, **daemen.j@protonworld.com**
Vincent Rijmen, KULeuven, **vincent.rijmen@esat.kuleuven.ac.be**

1. Introduction

In this public comment to NIST we address a number of diverging issues:

- provable security vs. provable properties;
- suitability of AES candidates for hashing;
- Rijndael software performance issues:
  - Java timing,
  - improved key setup code.
- Rijndael hardware implementation issues:
  - the inverse cipher,
  - the S-box.

2. Provable security

An important topic of discussion at the 2nd AES conference and the FSE workshop directly following it, was so-called provable security of block ciphers. We would like to stress here the difference between provable security and provable properties of block ciphers.

Designing a block cipher that is provably secure in an absolute sense seems for now an unattainable goal. Reasonings that have been presented as proofs of security have been shown to be based on (often implicit) assumptions that make these “proofs of security” irrelevant in the real world.

Still, we consider that having provable properties for a block cipher is an important feature of the design. Therefore we would like to draw your attention to Section 7 and 8 of the Rijndael documentation [2]. In these sections we prove minimum bounds on the number of active S-boxes in linear and differential trails over 4 rounds. In fact, comparing these results with the claimed results on the “provably secure” DFC [11] with respect to resistance against linear and differential attacks, we see that the results on Rijndael are stronger! (Furthermore, they are not susceptible to the comments in [4] and [12].) In order to avoid confusion however, we do not refer to these results as ”provable security”.

3. Hash suitability: the importance of variable block length

A block cipher can be used as the compression function of an iterated hash function by adopting the Davies-Meyer, Matyas-Meyer-Oseas or Miyaguchi-Preneel mode [7,Sect. 9.4.1]. In these modes the length of the hash result (and also the chaining variable) is the block length.

A length of 128 bits was considered to be insufficient for SHA-1. In this light it is desirable that the AES supports a block length of 160 bits or higher. This is the only the case for the candidates Rijndael and the Hasty Pudding Cipher [10].

4. Rijndael software performance

4.1 Java timings

We noticed a significant difference between the JAVA timing results in NIST's overview document [8] and Alan Folmsbee's paper [3]. NIST's timing classifies Rijndael at the top, with 7 Mbit/s, while Alan Folmsbee places it a much lower ranking, with 513 kbits/s for the MCT test performance, and 184 kbit/s for the KAT test performance.

We think that this phenomenon can be explained by a suboptimal implementation of the tests: we used the vanilla test code provided by the Cryptix team, but provided optimised JAVA code for the block cipher written by Paulo Barreto.

4.2 Key schedule improvement

A second performance issue that we want to mention, is that we significantly improved the performance of the C key setup routine of the cipher. We will provide new optimised ANSI C code before May 15.

5. Rijndael hardware implementation issues

This section is an answer to the following two comments that have been made regarding Rijndael hardware implementations:

1) The S-box of Rijndael eats a lot of chip area[9],
2) The inverse operation of the cipher needs the inverse of the S-box. This effectively doubles the hardware requirements if both forward and backward operation need to be implemented [9].

While it is true that we did not address the problem of hardware implementation until now, we feel that this needs to be put in perspective.

5.1 Inverse cipher

Firstly, we want to point out that all AES candidates that have not adopted the Feistel structure need to implement the inverse of the S-boxes for the inverse cipher operation: Crypton[5], SAFER+[6] and Serpent[1] are in the same situation.

Crypton and SAFER+ have a structure similar to Rijndael. Serpent on the other hand, with its bit-sliced structure and different S-boxes for different rounds, is expected to adopt another strategy for hardware implementations.

The designers of Crypton and SAFER+ have "solved" this problem by using both the S-box and its inverse in both forward and inverse operation of the cipher. The inverse operation does not need any extra S-boxes anymore, but a minimal implementation of the forward operation requires double S-box material.

A minimal hardware implementation is useful for instance in a cheap cryptographic coprocessor, that is used to speed up Smart Card applications.

In most Smart Card applications, it is not unreasonable to assume that the inverse cipher does not need to be implemented in the Smart Card:

- MAC verification and generation, the most important use of block ciphers in Smart Cards, does not require the inverse cipher;
- by using the CFB or OFB mode, decryption does not require the inverse cipher;
- if decryption in ECB or CBC mode is inevitable for some (unlikely) reason, encryption can be done using the inverse cipher so that the Smart Card can use the forward cipher to decrypt. While this mechanism is limited to applications in which the encrypting device is no Smart Card, this imposes no actual restrictions to any real applications we know of.

In the case that the cipher and its inverse are required, Crypton, SAFER+ and Rijndael all need to hardwire 2 S-boxes in hardware. However, if the inverse cipher is not required, for Rijndael one S-box is sufficient, while this is not the case for SAFER+ and Crypton.

Secondly, note that the S-box of Rijndael is constructed from two mappings [2]:

$$S(x) = f(g(x)),$$

where $g(x)$ is the mapping:

$$x \Rightarrow x^{-1} \text{ in GF}(2^8)$$

and $f(x)$ is a simple transformation defined as an affine mapping at the bit-level

The mapping $g(x)$ is self-inverse and hence $S^{-1}(x) = g^{-1}(f^{-1}(x)) = g(f^{-1}(x))$. Therefore when we want both S and S^{-1}, we need to implement only g, f and f^{-1}. Since both f and f^{-1} are very simple bit-level functions, the extra hardware can be reduced significantly compared to having two full S-boxes.

5.2 Rijndael S-box

The gate complexity of an implementation of the mapping g: $x \Rightarrow x^{-1}$ depends on the choice of basis in GF(2^8). The current Rijndael S-box was constructed using a basis that is not optimal in this respect and therefore it seems that the most efficient implementation of the mapping $g(x)$ is by a 256-byte table.

However, by choosing a suitable basis in GF(2^8), it is possible to implement $g(x)$ with a circuit with significantly less gates. Changing the basis would mean changing the S-box, so tweaking the cipher. We feel that this change is not major because the particular choice of basis has no further role in the design philosophy and does not influence any of the known analyses of Rijndael.

Furthermore, the design philosophy of Rijndael has always been to make the cipher very modular and robust to changes in the selection of the specific components.

We plan to submit the new S-box by May 15th.

6. References

1 E. Biham, R. Anderson, L. Knudsen, “Serpent, a proposal for the Advanced Encryption Standard”, presented at 1st AES conference

2 J. Daemen, V. Rijmen, “Rijndael”, presented at 1st AES conference

3 A. Folmsbee, ``AES JAVA technology comparisons,'' presented at the 2nd AES conference.

4 L.R. Knudsen, V. Rijmen, ``On the decorrelated fast cipher (DFC) and its theory,'' presented at FSE99, Rome.

5 C.H. Lim, “Crypton, a new 128-bit block cipher” presented at 1st AES conference

6 J. Massey et al., “Nomination of SAFER+ as candidate algorithm for the Advanced Encryption Standard (AES)”, presented at 1st AES conference

7 A. Menezes, P. Van Oorschot, S. Vanstone, Handbook of applied cryptography, CRC Press, 1997.

8 ``NIST efficiency testing results,'' presented at the 2nd AES conference.

9 B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall and N. Ferguson, “Performance Comparison of the AES Submissions” presented at 2nd AES conference

10 R. Schroeppel, “Hasty Pudding Cipher Specification”, presented at 1st AES conference

11 S. Vaudenay et al., “Decorrelated Fast Cipher: an AES candidate”, presented at 1st AES conference

12 D. Wagner, ``The boomerang attack,'' presented at FSE99, Rome.