Rijndael for AES

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

1. Introduction

In this document we give a short overview of the reasons why Rijndael should be selected as the AES. We have divided our arguments into four categories:

- **Security:** Rijndael has the same objective security level as the other finalists, and can easily be implemented in a secure way.
- **Efficiency:** Rijndael has a large "performance margin" compared to the other candidates.
- **Design philosophy:** The clear design has many advantages: easy implementable on a wide range of platforms, easy to get confidence in the claimed security level, ...
- **Extensions:** Rijndael is easily extendable to other key and block lengths.

Finally, we discuss the issue of multiple AES algorithms.

2. Security

2.1 Objectively demonstrable security

Until now, for none of the 5 AES finalists, an attack has been published that demonstrates a weakness inherent in the design. Hence, from a cryptanalytical point of view, all 5 ciphers are equivalent.

2.2 Suitability for secure implementation

In software, Rijndael can be implemented using the operations bitwise XOR, table-lookup and 8-bit shifts. Serpent requires no table-lookups but more general shifts and rotations and bitwise boolean operations.

Twofish additionally requires 32-bit addition and both MARS and RC6 even require 32-bit multiplication and shifts over data-dependent off-sets. The presence of these operations makes the latter three algorithms harder to implement in a secure way on smart cards [DaRi99].

2.3 Adding rounds

For all well-designed block cipher, the complexity of published cryptanalytic attacks increases with the number of rounds in the cipher. This has already been taken into account in the Rijndael design: the increasing number of rounds for increasing key lengths assures a growing security marging against cryptanalytic attacks.

In fact, the number of rounds is a parameter that can be increased further, *without a need for any additional specifications*. In applications where the confidence in Rijndael's security doesn't match the importance of the confidentiality/integrity, or in the hypothetical case that an effective attack on Rijndael would be published, a Rijndael version with an increased number of rounds can be used.

3. Relative efficiency

The relative efficiency of the different finalists can be shown by comparing optimal implementations on several platforms. Given the fact that the different design teams have taken different security margins, the question rises how to compare the algorithms on equal footing. One approach is to determine a minimum number of rounds that has to be used in order to resist currently known attacks, and to add some rounds extra [Bi99]. Unfortunately, not all ciphers have been subjected to the same amount of study. Furthermore, there is no consensus on how many rounds one should add to get an adequate security margin. For instance, how should the added security of an extra round of a (generalised) Feistel cipher be compared with a round of an S-P-network ?

On the other hand, the performance of all the algorithms has been evaluated on many different platforms, and all algorithms got their fair share of attention. Therefore we propose to compare the other AES finalists to Rijndael variants with an adapted number of rounds, such that both algorithms execute in the same time. In Table 1 we list for each AES finalist the number of Rijndael rounds (including the implied round key generation) that can be executed in the same time. Nominally, Rijndael has 10 rounds (for 128-bit keys).

We consider the following platforms:

- **Pentium II/Pro:** representative processor for PCs today;
- **Motorola 6805:** representative processor for smart cards today.

Moreover, we give numbers for different amounts of data treated with the same key:

- **many blocks:** indicative if the same key is used for a considerable amount of data (say at least some Kbytes).
- **4 blocks:** indicative if AES is used to secure a small amount of data. In most financial transactions the amount of data that is subject to a MAC is indeed below 64 bytes. This includes electronic purse, debit/credit and ticketing transactions that will be used in timing-critical applications such as public transport and toll-road payment automation.
- **1 block:** relevant if AES is used as the compression function of a hash function, for PIN code encipherment/decipherment or for session/instance key derivation (in smart card, terminal and/or Host security module) typical for payment systems.

Processor	# blocks	source	DES***	MARS	RC6	Serpent	Twofish
Pentium II/Pro	many	[Li00]	-	13	9	38	12*
	4	[Co99]	-	28	15	33	27
	1	[Co99]	-	46	22	36	25
Motorola 6805**	many	[Ke99]	30	30	28	110	23
	4	[Ke99]	32	52	45	107	23
	1	[Ke99]	37	114	91	100	22

Table 1 Number of rounds in Rijndael, given the same number of cycles

* The Twofish design team measures the performance of Twofish with code that has the used key compiled into the executable. We use the code by Aoki and Lipmaa, slightly slower than the self-modifying (!) code by the Twofish team.

****** For Twofish, only the results of the designers are available. For MARS and RC6 we use the implementations for smartcards with massive RAM available. For Rijndael, we average cipher and inverse cipher speed.

******* For DES, the number of blocks is doubled as the block length is only 64 bits

4. Design philosophy

In the Rijndael design, we have tried to keep everything as simple as possible. Complexity has been added only when necessary to thwart attacks. One example is the key schedule, that is very simple and efficient compared to that of other AES finalists.

Other “simplicity” properties include:

- Symmetry in the round transformation and across the rounds,
- Orthogonality of components,
- Absence of arithmetic operations.

These properties lead to a number of advantages that are treated in the following sections.

MARS and Twofish, on the contrary, have both a very complex round function, with many different operations. According to the documentation given by the respective design teams this is partly due to the fact that during the design, whenever complexity could be added ‘at no additional cost', it was added. ‘At no additional cost' should be understood as `no additional cost *on the Pentium Pro*'. On other ‘unknown' platforms [Cl99], these extra operations could be cheaply available, or not.

Serpent introduced asymmetry across the rounds by adopting 8 different S-boxes and asymmetry in the round transformation by having shift (instead of cyclic shift) operations. RC6 has a reasonably symmetrical design. However, it still mixes XOR and arithmetic addication operations and it uses 32-bit multiplication.

Another important advantage of Rijndael is that it was designed right from the start to support 128 bit block lengths. Twofish and RC6 on the other hand, are obviously upgrades from their 64-bit predecessors, respectively Blowfish and RC5, and this shows in the design.

4.1 Symmetry

There is only a single S-box, since until now, no advantage has been demonstrated for the use of different S-boxes (as in Serpent, Twofish and MARS). This S-box is applied in parallel to all state bytes. Similarly, the linear transformation and the round key addition treat all state bytes in the same way and have rotational symmetry. The round function is the same for the complete cipher execution (unlike Serpent and MARS) as the differences in the round keys are considered to introduce sufficient asymmetry. This gives Rijndael the following advantages:

- **Parallelism:** among the finalists, Rijndael is by far the best suited to be implemented on processors with a parallel architecture[Cl99], that is expected to be the architecture of the future (Merced, McKinley, …). Moreover, a dedicated hardware implementation in which the Rijndael round is fully hardwired can give very high speed thanks to its short critical path[DaRi98].

- **Compactness:** the single S-box and the simplicity of the linear transformation allow to code Rijndael in a small number of bytes, relevant on smart cards. Moreover, a minimal dedicated hardware implementation of Rijndael can be built by hardwiring a single S-box and a single 4-byte to 1-byte linear transform[DaRi98].
- **Absence of arithmetic operations:** the description of Rijndael does not make any (hidden) assumptions on the coding of integers as a sequence of bits. One of the advantages of this is that Rijndael is immune for so-called big endian/little endian confusion and conversion problems.

4.2 Orthogonality of the components

In Rijndael, the round function is composed of a number of components each with their own contribution: S-boxes for non-linearity, round key XORing for key dependence and asymmetry, byte transposition for inter-word diffusion and an MDS transform for intra-word inter-byte diffusion. This design feature allows to get more easily a view on the security of the algorithm.

We have provable lower bounds for linear and differential probabilities based on the interaction of these components. These proofs make use of only a few macroscopic properties of the components and leave a lot of freedom on how these properties are actually attained. The advantage of this modular approach is that components may be replaced without affecting these lower bounds as long as the macroscopic properties hold. For example, in the hypothetical case that an attack would be launched that makes use of some specific property of the current S-box, it could be replaced by another one without affecting the lower bounds.

For the other AES finalists, the interaction between the different components is intricate and much harder to analyse and the act of replacing a single component turns a lot of the analysis performed obsolete.

4.3 Confidence

As a consequence of its clarity of design and good performance results, Rijndael attracted by far the most attention from cryptanalysts outside the design team. Although the other finalists seem to have been analyzed quite thoroughly by their own designers, history has shown that `friendly' cryptanalysis is not as effective. A number of attacks on reduced versions has been published. We can conclude that Rijndael has a sufficient security margin, and do so with a high level of confidence.

5. Extensions

Rijndael is the only AES finalist that supports other block lengths than 128 bits, namely 192 bits and 256 bits. Moreover, extensions are defined for all combination of block lengths and key lengths between 128 and 256 bits in steps of 32 bits [DaRi98].

The added value of the longer block lenghts is that the cipher can be used as the compression function of a collision-resistant iterated hash function. Note that a length of 128 bits was considered to be insufficient for SHA-1.

6. Multiple algorithms

The technical reasons for having multiple algorithms for the AES would be the fact that a single algorithm cannot be efficiently and securely implemented on all target platforms, or to have a backup in case the primary algorithm has been broken.

If Rijndael is chosen as the AES, there is no need for an alternative algorithm for the first reason as Rijndael is very efficient on all target platforms. Of course, if MARS or RC6 would be chosen, smart card application developers will see their performance and RAM availability go down and will tend to stick to good old Triple-DES if no alternative AES is available.

In practice, “having a backup in case the primary algorithm is broken” is a very expensive and cumbersome undertaking. It implies coding, testing and integrating both the primary and the backup algorithms in all products and applications where this backup is really taken seriously. If Rijndael is chosen as AES, the “backup” could be a Rijndael version with the number of rounds doubled. In this respect it is worth while to consider the actual risk. For the current standard DES, the most practical attack to date is exhaustive key search, an attack that was already known before its publication. The more sophisticated attacks, such as linear and differential cryptanalysis are very interesting and have learnt us a lot on how to design ciphers, but are no threat in the real world. The design teams of the AES finalist algorithms know their literature and have all used the experience obtained from analysing DES, FEAL, IDEA, … to build their ciphers. Hence, although new attacks may always be found, we think it is unlikely that they will be a security threat in real-world applications, whatever choice is made among the finalists.

7. References

[Bi99] E. Biham, "A note on comparing the AES candidates", AES 2.

[BAK98] E. Biham et al., “Serpent, a proposal for the Advanced Encryption Standard”, AES 1.

[Cl99] C. Clapp, “Instruction-level parallelism in AES candidates”, AES 2.

[Co99] B. Schneier et al., “Performance comparison of the AES submissions”, AES 2.

[DaRi98] J. Daemen and V. Rijmen, “Rijndael”, AES 1. Updated version from `http://www.esat.kuleuven.ac.be/~rijmen/rijndael`

[DaRi99] J. Daemen and V. Rijmen, “Resistance against implementation attacks: a comparative study of the AES proposals”, AES 2.

[Ke99] G. Keating, “Performance analysis of AES candidates on the 6805 CPU core”, AES 2. Updated version from `http://www.ozemail.com.au/~geoffk/aes-6805/`.

[IBM98] C. Burwick et al., “Mars - a candidate cipher for AES”, AES 1.

[RC98] R. Rivest et al., “The RC6™ block cipher”, AES 1.

[Co99] B. Schneier et al., “Twofish, a block encryption algorithm”, AES 1.

[Li00] H. Lipmaa, AES cipher performance cross-table, available at `http://home.cyber.ee/helger`