A Note on Comparing the AES Candidates

Eli Biham

Computer Science Department
Technion – Israel Institute of Technology
Haifa 32000, Israel
Email: biham@cs.technion.ac.il
WWW: http://www.cs.technion.ac.il/~biham/

Revised Version

Abstract. The comparison of the AES candidates should take into consideration the security and the efficiency of the ciphers. However, due to different design methodology, the ciphers were developed in different emphasis of the importance of security and efficiency. In this paper we propose measures to compare the AES candidates under the same security assumptions. These measures reduce the effect of the different design methodologies.

1 Introduction

The AES process had attracted 15 submissions of blockciphers. These submissions were designed under different criteria, and were optimized differently. For example, some designs concentrated on the speed of the cipher, with less attention to its security, while others concentrated on security, paying some price on their speed. Therefore, the AES decision may finally have to select between ciphers which were optimized in different ways.

In this paper we describe how we can compare the AES candidate ciphers fairly using comparable measures. The author believe that the AES process should first have a decision on the relative importance of all the various criteria, and on bounds on the speed and strength, and then make the selection based on these decisions. In such a case, the ciphers might be modified in the obvious ways (adding more rounds, or removing some of the rounds) to make the cipher fit the decided criteria.

2 The 15 Submissions

Table 1 lists the 15 AES submissions. These submissions were designed under different criteria and structures. We briefly describe the general structures in Table 2.

These designs vary in

1. High level design ideas
   - Based on existing design?
   - Totally new cipher?
   - Feistel / SP network?
   - Design of the round functions
   - How many rounds?
2. Used Instructions
   - XOR

Cipher	Submitted by	Country
CAST-256	Entrust	Canada
Crypton	Future Systems	Korea‡
Deal	Outerbridge	Canada†
DFC	ENS–CNRS	France
E2	NTT	Japan
Frog*	TecApro	Costa Rica
HPC*	Schroeppel	USA
LOKI97*	Brown, Pieprzyk, Seberry	Australia
Magenta	Deutsche Telekom	Germany
Mars	IBM	USA†
RC6	RSA	USA†
Rijndael*	Daemen, Rijmen	Belgium‡
Safer+*	Cylink	USA†
Serpent*	Anderson, Biham, Knudsen	UK, Israel, Norway
Twofish*	Counterpane	USA†

* Placed in the public domain; † and foreign designers; ‡ foreign influence

Table 1. The 15 AES Submissions.

Cipher	Type	Rounds	Using
CAST-256	Ext. Feistel	48	
Crypton	Square	12	
Deal	Feistel	6, 8, 8	DES
DFC	Feistel	8	Decorrelation modules, mult.
E2	Feistel	12	
Frog	Special	8	BombPermu
HPC	Omni	8	Hasty Pudding
LOKI97	Feistel	16	
Magenta	Feistel	6, 6, 8	
Mars	Ext. Feistel	32	Var. rot., mult., non-crypt. rounds
RC6	Feistel	20	Var. rot., mult.
Rijndael	Square	10, 12, 14	
Safer+	SP network	8, 12, 16	PHT
Serpent	SP network	32	Bitslice
Twofish	Feistel	16	

Var. rot.=Variable rotation. Mult.=Multiplication.

Table 2. General Structure.

- S Boxes: 4x4, 8x8, 8x32, 11x8, 13x8, 8x32
- Addition, Subtraction
- Rotate, Shift
- Multiplication:
  - modulo 2^{32}: Efficient on Pentium II, but very slow on older processors
  - modulo 2^{64}
  - modulo $2^{64} + 13$
- Variable Rotations

3. Techniques
   - Bitslicing
   - PHT
   - Decorrelation
   - Non-cryptographic rounds
   - Using other ciphers (DES)
4. Optimization Target: The designers had to decide which target platform to optimize for

- Pentium? MMX? Pro?
- Pentium II?
- 64-bit processors?
- 16-bit processors?
- 8-bit processors?
- Smartcards?
- Hardware?

5. and various others

Each of the above decisions may crucially affect the design.

3 Speed Comparisons

The figures in the papers of the submitted AES candidates claim speeds based on various measurement assumptions.

1. Some measure the speed of the cipher with NIST API.
2. Some measure the speed of the native procedures. This is usually 10–20% faster than using the NIST API.
3. Some measure the speed using various optimizations, which are incompatible with the NIST API (such as setting the subkeys in a static array, or even statically planting the subkeys into the encryption code in assembler). This might give additional 20% in speed for almost every cipher.

It is clear that the comparisons should be done under some common assumptions. For the figures in this paper we measured the speed of the mathematically optimized C implementations of the designers (with the necessary minor modifications[1]) using a common test program. Table 3 shows the speed of the optimized implementations on Linux/GCC-2.7.2.2/Pentium 133MHz MMX. We did not write our own code for all or some of the candidates for the comparison. Only 128-bit keys are considered.

4 Security/Speed Tradeoffs

The AES candidates design have a huge difference in the design methodology for the relative importance of speed and security.

---

[1] Although the sources CD contains all the code, there are many problems to solve:

Deal assumed that the caller makes memory allocation for it.

DFC receives the input length in bytes, while all others receive in bits. Thus, it seemed eight times slower.

HPC comes without include files, which should be created manually (typed from a printed paper which comes with the CD). It also malloc's memory each makeKey, but fails to free it. So measuring the speed of makeKey is problematic due to memory constraints.

Magenta's implementation fails when the plaintext and ciphertext blocks reside in same memory.

Mars returns wrong return values (0 rather than TRUE).

Rijndael added a non-standard parameter to the API: variable block size. So it cannot be used with the standard calling form.

Some submissions verify that in ECB mode the IV is set to NULL. Some other initialize the IV even in ECB mode. Thus, a single main program cannot work for all submission supplied on the CD.

Many have special optimization macros and qualifiers to set.

Cipher	Encrypt (cycles)	Decrypt (cycles)	Key Setup* encrypt	decrypt	Init	% CPU	
Twofish	1254	1162	18846	18634	20	95.4%	92pf+0w
Rijndael	1276	1276	17742	18886	28	99.5%	98pf+0w
Crypton	1282	1286	758*	824*	24	99.7%	66pf+0w
RC6	1436	1406	5186	5148	30	94.0%	92pf+0w
Mars	1600	1580	4708	5548	18	96.7%	92pf+0w
Serpent	1800	2102	13154	12648	14	94.7%	98pf+0w
E2	1808	1854	7980	7780	24	96.0%	76pf+0w
		— DES with NIST API —					
CAST-256	2088	2080	11412	11478	34	99.9%	67pf+0w
Frog	2182	2668	3857000	3817100	22	95.6%	64pf+0w
HPC	2602	2962	234346	248444	20	64.1%	142pf+5557w
Safer+	4424	4620	4708	4668	38	95.7%	88pf+0w
DFC	5874	5586	23914	25616	534	98.6%	65pf+0w
LOKI97	6376	6118	22756	22490	148	96.7%	108pf+0w
Deal	8950	8910	108396	107996	36	97.3%	68pf+0w
Magenta	23186	23230	1490	1622	24	99.2%	89pf+0w

Table 3. Comparison of the speed of the candidate ciphers on Pentium 133MHz MMX.

1. Small margins: adding a few rounds:
   - RC6: Rivest assumes that there is an attack on 16-round RC6 with complexity 2^{119}. In such a case we expect that there is an attack on 17 rounds which is very slightly faster than exhaustive search. Rivest proposes 20 rounds.
   - DFC: An attack on 6 rounds. 8 rounds are proposed.
   - Deal: Knudsen describes an attack on 6 rounds which is slightly faster than exhaustive search (2^{120}). He proposes 6 rounds.
2. Large margins: Doubling the number of rounds in the expense of speed:
   - Serpent: 16 rounds are secure. 32 are proposed.
   - Twofish: The best known attack is on 6 rounds. 9–10 rounds using strong assumptions of related keys against the 256-bit variant. 16 are proposed.

We need to consider the speed of the variants with the same security level. For this, we propose measuring the speed of the variants with the minimal number of rounds for which the cipher is still (believed to be) secure, while still considering the structure of the cipher (i.e., a full multiple of a pass). In order to ensure that the cipher is secure, and no attack or a slight improvement of an attack can succeed, we actually define the minimal number of rounds by adding two passes to the variant with the largest number of rounds whose complexity of analysis (of any kind) is *smaller* than the complexity of exhaustive search.[2] These figures are then taken from the designers' papers, or from other external (including our own) sources on the ciphers.

For example, Serpent has 32 rounds. The authors claim that 16 rounds are already secure, and thus the longest variant which is not as secure as exhaustive search might have 15 or fewer rounds. Serpent is a SP network, and thus each pass contains one round, and two passes contain two rounds. Therefore, the minimal number of rounds is at most 17.

Mars has 32 rounds of two types, divided into 8 4-round passes. We know that 12-round Mars is insecure. Thus, the minimal number of rounds is at least 20. We did not make the full analysis of Mars, and therefore left 20 as the minimal number of rounds. We believe that it should be even higher as we expect that Mars with a few rounds more than 12 should still be insecure.

[2] Other methods for choosing the minimal number of rounds are also possible, e.g., adding 10% of the rounds to the longest insecure variant.

The designers of Rijndael propose that 6 rounds are still insufficient. Therefore, the minimal number of rounds is 8.

Twofish has 16 rounds. The authors mention attacks on 6 rounds of the 128-bit variant. Twofish is a Feistel cipher, and thus each pass contains two rounds, and two passes contain four rounds. Therefore, the minimal number of rounds is 10.

The designers of Crypton propose that 9 rounds are not secure. Therefore, the minimal number of rounds is 11.

E2 is a Feistel cipher with 12 rounds. The designers of E2 propose that 8 rounds are not enough. Therefore, the minimal number of rounds should be $8 + 4 = 12$. However, as the initial and final transformations are more complex than in the other ciphers, we feel that they should be counted as rounds. Therefore, the minimal number of rounds we use is 10 (plus the initial and final transformations).

RC6 is a Feistel cipher with 20 rounds. The designers claim that 16 rounds are still attackable with complexity 2^{119}. We expect that if this is the case, then 17 rounds are still attackable with complexity very slightly faster than exhaustive search. Therefore, the minimal number of rounds is $17 + 4 = 21$.

CAST-256 has 48 rounds. It is not clear from the paper what is the most successful attack against CAST-256. We are aware of 20-round impossible differential of CAST-256, and believe that 32 (and even more) rounds of CAST-256 is still breakable faster than exhaustive search. Each pass contains 4 rounds. Therefore, the minimal number of rounds is at least $32 + 8 = 40$.

Safer+ is a SP network with 8 rounds. A 5-round variant is still breakable. Thus, the minimal number of rounds is at least 7.

DFC is a Feistel cipher with 8 rounds. The designers describe an attack on a 5-round variant. Thus, the minimal number of rounds is at least 9.

Deal is a Feistel cipher with 6 rounds. The designer describes an attack on the full 6-round Deal with complexity 2^{120} which is faster than exhaustive search. Thus, the minimal number of rounds is $6 + 4 = 10$.

LOKI97 is a Feistel cipher with 16 rounds. It was shown that its complexity of analysis is no more than 2^{56}. We expect that it can be analyzed up to 34 rounds. Thus, the minimal number of rounds is at least 38.

Magenta is a Feistel cipher with 6 rounds. We believe that variants up to 7 rounds are still breakable. Thus, the minimal number of rounds is at least 11.

Frog and HPC have 8 rounds. Although we know that other cryptographers analyzed Frog, we decided not to guess the minimal number of rounds of this cipher. We also did not analyze HPC, and cannot predict the minimal number of rounds. We expect that this lack in predicting the minimal number of rounds of these two ciphers will not affect the choice of the final AES cipher.

Table 4 summarizes the minimal number of rounds and the speeds of the AES candidates.

5 Hardware Implementations

Table 5 describes the figures of the hardware size as described by the designers.

Cipher	Original (cycles)	Rounds	Minimal Rounds	Time (cycles)
Twofish	1254	16	$6+4=10$	784
Serpent	1800	32	$15+2=17$	956
Mars	1600	32	$12+8=20$	1000
Rijndael	1276	10	$6+2=8$	1021
Crypton	1282	12	$9+2=11$	1175
E2	1808	12	$8+2+\mathrm{IT}+\mathrm{FT}=10$	1507
RC6	1436	20	$17+4=21$	1508
CAST-256	2088	48	$32+8=40$	1740
— DES with NIST API —				
Safer+	4424	8	$5+2=7$	3871
DFC	5874	8	$5+4=9$	6608
Deal	8950	6	$6+4=10$	14917
LOKI97	6376	16	$\geq 34+4=38$	≥ 15143
Magenta	23186	6	$\geq 7+4=11$	≥ 42508
Frog	2182	8	?	
HPC	2602	8	?	

Table 4. Proposed Minimal Rounds for the Ciphers and Their Speed.

Cipher		gates/cycles	gates/cycles
CAST-256	not given		
Crypton		19000 / > 6	50000 / ?
Deal	not given		
DFC	not given		
E2		127000 nand / 16	
Frog	not given		
HPC	not given		
LOKI97	not given		
Magenta	not given		
Mars		70000 cells / 50	
RC6	100 nano-sec		
Rijndael	not given		
Safer+		62000 cells / 134	
Serpent		4500 / 32	70000 / 1 (fully pipelined)
Twofish		14000 / 64	23000 / 16

Table 5. Hardware Figures.

Ciphers always have tradeoffs between hardware size and speed:

- Duplicate bottlenecks
- Increase table sizes
- Unroll
- Interleave blocks

We believe that we should therefore take gates$\times$cycles as the comparison parameter. Table 6 compares this measure. We urge hardware experts to improve this comparison parameter and compare the ciphers in the most fair way, taking to consideration the number of gates and cycles, but also the delays and other factors that are inherent in any hardware technology.

Cipher	gates	cycles/block	gates×cycles	Minimal	variant
Serpent	70000	1	70000	17/32	37187
Crypton	50000	?	100000?	11/12	91667
Twofish	23000	16	368000	12/16	276000
DES	28000?	16?	448000?		
E2	127000 nand	16	2032000	10/12	1693333
Mars	70000 cells	50	3500000	20/32	2187500
Safer+	62000 cells	134	8308000	7/8	7269500

Table 6. Hardware Figures Compared Under the Proposed Measure.

6 Novel vs. Conservative Design and Confidence

Public confidence in the design and strength of ciphers is gained through evidence that the cipher is invulnerable to all known attacks. Such evidence may be proposed by the designers, or be published in the few years after first publication of the cipher. In this environment, even the negative fact that nothing is published on a well known cipher adds confidence to the design of the cipher.

The design of the cipher can take this into consideration, and propose a cipher that can be easily analyzed whether it is vulnerable to widely known techniques. The designers can decide to use novel ideas in the design which might increase the speed or improve other properties of the cipher, and ensure that the widely used techniques (such as differential and linear cryptanalysis) are not applicable. In such designs, however, it is difficult to bound the strength, as the used techniques are new, and no known cryptanalytic techniques are known that can compare the strength to other known ciphers.

On the other hand, some designers prefer conservative designs, where they use known structures, that can be easily analyzed by well known cryptanalytic techniques. Their security is thus easier to study, and they can be shown immune against the standard cryptanalytic tools.

The author believes that the AES cipher should select the later kind of design. Novel designs are important for future advances of blockcipher design and analysis. Although in many cases they may be quite fast, even under the measures proposed in this paper, we should let them longer times till they can get sufficient confidence. Otherwise, somebody might find a new attack against such ciphers. Such a new attack is less expected (although not impossible) against conservative designs.

7 Summary

In this paper we proposed possible measures for the comparison of the AES candidates. We believe that the real question in the AES process is how to compare speed and efficiency to security, i.e., which of them to prefer, and how to choose their relative importance. The author welcomes comments of any kind, and urge other experts to publish their own measures, and their improved variants of these measures.

8 Acknowledgments

We are grateful to Ross Anderson and Serge Vaudenay for their valuable comments on the proposed ideas. We would also like to thank Bruce Schneier who helped a lot in developing the proposed idea, and who already made the first comparison based on this idea during the first AES workshop. Finally, I am grateful to the organizers of ASIACRYPT'98 who gave me the opportunity to present some of this work.