Date: Mon, 22 Feb 1999 15:38:25 +1100
From: Emanoil Daneliuc <emanoil@akyman.com.au>
X-Mailer: Mozilla 4.07 [en] (Win98; I)
To: aesfirstround@nist.gov
Subject: official comments on AES candidates

Dear Sir/Madam,

On behalf of Akyman Financial Services Pty. Ltd., I send you our comments on the *AES candidate algorithms* (attached in "my opinion for NIST AES.doc" document).
Please note that it is a *Microsoft Office 97 - Winword* document.

Regards,

Emanoil Daneliuc

(Firmware Engineer)

AKYMAN FINANCIAL SERVICES PTY. LTD.

(Incorporated in Victoria A.C.N. 006 668 962)

19-Feb-99

Dear Sir/Madam,

We, at Akyman, have analysed the "*Advanced Encryption Standard Development Effort*" CD-ROM titled "*CD-2 Algorithm Code*", that was received from you by *Toni Stojanovski*, who used to be a "Cryptography Specialist" at Akyman.

In the following, I will present our conclusions.

Sincerely yours,

Emanoil Daneliuc

(Firmware Engineer)

We classified the algorithms by their speed and memory requirements. (On IBM-PC the memory requirements only matter by their indirect effect on speed by exceeding the 1st level cache of the CPU (usually a Pentium).)

1 Our conclusion (think of it as Akyman's "vote") is:

The best-preferred algorithm – in our vision – is:	**RIJNDAEL**
The second best is:	**CRYPTON**
The third best is:	**TWOFISH**

1.1 Rationale:

1.1.1 Preselection

Out of the initial algorithms list, which comprises 15 algorithms (CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA, MARS, RC6, RIJNDAEL, SAFER+, SERPENT, TWOFISH), we have to discard:

- DEAL – for reasons:
  - is slow (about 12 times slower than good algorithms on Pentium processors, as a reference)
  - has some security weakness (according to "*CAESAR*" Internet site)
- HPC – requires about 2KB RAM, which practically forbids it on current smart cards
- CAST-256 – requires 6KB ROM, which is going to be expensive on smart cards. It is not particularly fast on either 32- or 8 bit CPU's.
- FROG – for 2 reasons:
  - requires 2,300 Bytes on a smart card, which can not be provided conveniently
  - it seems that it has been already broken (according to the "*AES Discussion Forum*" files, to which the *AES* Internet site points)
- LOKI97 – has been broken (according to tekst.ps file, in the "loki97_ps.gz" gzip file, in the AES Discussion Forum)

- MAGENTA – for 2 reasons:
  - has a serious security weaknesses, according to "*CAESAR*" Internet site
  - it is about 40 times slower than fast algorithms on Pentium
- SAFER+ - is too slow on 8-bit CPU's (over 20 times slower than RIJNDAEL). (Besides, there are 2 strictly academic attacks, as mentioned in the "*AES Discussion Forum*" files.)

So, only 8 algorithms remain in the competition:
CRYPTON, DFC, E2, MARS, RC6, RIJNDAEL, SERPENT, TWOFISH

At Akyman, we develop smart-card-based devices; therefore we keep in mind:
- running speed of algorithms on 8-bit CPUs
- running speed of algorithms on 32-bit CPUs
- security level

1.1.2 Running speed on 8-bit CPU's:

Note: for all speeds, we will count just the encryption/decryption time (without the key preparation time).
The best algorithms are:

Ranking (1 = best)			Algorithm	CPU clock cycles for encrypt or decrypt 128-bit block with 128-bit key	RAM requirements	ROM requirements	Comments
By speed	By RAM requirements	By ROM requirements					
3	2	4	CRYPTON	**12,000**cycles on a hypothetical CPU approx. = 8051 power	52Bytes	<2KB	good
4	4	5	DFC	35,500cycles on 6805 CPU (= approx **29,000**cycles on 8051 CPU)	<60 Bytes	<2KB	If not given more than 100Bytes RAM => multiply speed by 6
8	8	6	E2	6,300cycles (= approx. **4,800** cycles on 8051 CPU)	256 Bytes	<2KB	Rather much RAM required
6	6	7	MARS	5,000cycles on a hypothetical CPU (1 cycle/instruction) = **12,000** cycles on 8051 CPU	160 Bytes	2KB	
7	7	2	RC6	**13,900**cycles on 8051 CPU	210 Bytes	Approx 1KB	A bit high RAM requirements
1	1	3	RIJNDAEL	**3,100**cycles for encryption on 8051 CPU, **6,200** for decryption	52Bytes	<2KB	Best performance, least requirements

Ranking (1 = best)			Algorithm	CPU clock cycles for encrypt or decrypt 128-bit block with 128-bit key	RAM requirements	ROM requirements	Comments
By speed	By RAM requirements	By ROM requirements					
2	3	1	SERPENT	34,000 on 6805 CPU (= approx 28,300cycles on 8051 CPU) if coded in Ada => some **9,000**cycles for 8051 CPU in assembler	<60 Bytes	<1KB	More than 9,000 8051 CPU cycles required when using only the 60Bytes RAM quoted left
5	5	8	TWOFISH	26,500 on 6805 CPU (= approx 22,000cycles on 8051 CPU) or 37,100on 6805 CPU (= approx **31,000**cycles on 8051 CPU)	60Bytes	2.2KB or 1.76KB	

So we can eliminate E2 and RC6 from the competition, as being on the last places for speed on 8-bit CPU's.

1.1.3 Running speed on Pentium CPU

Note: The Minimal Secure Rounds value is the ultimate estimation, so we will use it.

Ranking (1 = best)				Algorithm	CPU clock cycles for encrypt or decrypt 128-bit block with 128-bit key			Comments
Weighted average (coef. = 1:3:2)	By algorithm Specification	By Min. Security Requirements	By *Nistefficiency1.pdf* document		Specification	Minimal Secure Rounds (*aes-performance.pdf*)	*Nistefficiency1.pdf* document	
4	4	6	1	CRYPTON	390	358	630	Best by *Nistefficiency1.pdf* document
7.67	6	8	8	DFC	750	844	4413	Worst by Min. Sec. Req. and by *Nistefficiency1.pdf* document
5	7	5	4	E2	843	342	898	
3.33	5	1	6	MARS	393	200	984	Best by Min. Sec. Req.

Ranking (1 = best)				Algorithm	CPU clock cycles for encrypt or decrypt 128-bit block with 128-bit key			Comments
Weighted average (coef. = 1:3:2)	By algorithm Specification	By Min. Security Requirements	By *Nistefficiency1.pdf* document		Specification	Minimal Secure Rounds (*aes-performance.pdf*)	*Nistefficiency1.pdf* document	
2.67	1	3	3	RC6	254	250	842	Best by algorithm specification
3.17	3	4	2	RIJNDAEL	320	256	800	Worst by algorithm specification
7.17	8	7	7	SERPENT	1730	478	3506	
3	2	2	5	TWOFISH	285	211	937	

So we can eliminate DFC and SERPENT from the competition, by being ranked on the last 2 places on at least 2 comparison criteria.

1.1.4 Security Level

There have been noted some security weaknesses in:

Ranking (1 = best)	Algorithm	Security weaknesses	Comments
4	CRYPTON	1 fixed point has been noticed, but it doesn't look dangerous	2nd choice
7	DFC	Uses multiplication so one must be careful about potential *timing attacks*	3rd choice
3	E2	No security comment	
8	MARS	2^{16} effort to find an equivalent key (because feedback cancels out when size = 160 bits)	A bit suspicious
6	RC6	Timing attacks have to be kept in mind (because of the *data dependent rotations* and *multiplications* being used).	Best by algorithm specification
1	RIJNDAEL	No security comment	OK
2	SERPENT	No security comment	OK
5	TWOFISH	Slight weakness for keys of the type: 0001,8000,0001,8000	

So we can eliminate here: MARS and DFC

1.1.5 Conclusions:

After eliminating the algorithms mentioned above (based on the respective criteria), the following algorithms remain:
CRYPTON, RIJNDAEL, TWOFISH.

Overall Ranking	Algorithm	security	8-bit performance	32-bit performance	Hardware performance (according to *aes-performance.pdf*)
2	CRYPTON	1 fixed point has been noticed, but it doesn't look dangerous	Above average	Just under average	Very good
1	RIJNDAEL	No security comment	Best by speed and RAM	Pretty Good	Pretty Good
3	TWOFISH	Slight weakness for keys of the type: 0001,8000,0001,8000	Under average	Good	Good

1.1.6 Important warning

Fighting hardware attacks:

Because all of the algorithms fail to be secure against the *Differential Power Attack* (which consists of monitoring the current consumption of the microprocessor while encryption/decryption is being performed), all smart card hardware manufacturers need to be specifically warned to take hardware countermeasures against a *Differential Power Attack*. I can think here of 2 methods:
1). use a circuit that would equalise the current consumption irrespective of current operation being performed
2). use a "Differential Power Attack confusing circuit", which will simply add some random (or following some well-thought algorithm in relation to data being processed) current consumption.

Both of these methods will increase the power consumption of the smart card. The second will be more economical. The Differential Power Attack confusing algorithm will have to be designed containing clearly defined elements (especially timing), as well as pseudo random elements (to hide the real data).

Appendix:

Algorithm performance parameters as we were able to extract them directly from the Algorithm Specifications provided on "NIST, AES Round 1, CD-2" CD-ROM. Note that later in the document, the data from AES Internet sites have also been used along with it.

Algorithm	Clocks per 128-Bytes block (with 128-Bytes key) encryption		required RAM amount [Bytes]	Security level	Comments
	Pentium	sample 8 bit CPU			

Algorithm	Clocks per 128-Bytes block (with 128-Bytes key) encryption		required RAM amount [Bytes]	Security level	Comments
	Pentium	sample 8 bit CPU			
CAST-256	890	26,000 (6811µP)	4,000		Not suitable for smart cards
CRYPTON	390	12,000 (hypothetical µP 3clocks/instruction ≅ power of 8051µP)	48		Very low requirements: good
DEAL	7,200	233,000 (hypothetical µP, 32 times slower than a Pentium at same clock)	128 (faster with 1,000)		Rather slow
DFC	750	35,000 (6805µP)	200		H/w implementation difficult because of multiplication
E2	843	6,300 (Hitachi H8/300µP)	256		
FROG	800	17,900 (Z80µP)	5,000		Not suitable for smart cards. New approach => hard to assess
HPC	3,500	35,000 (Z80µP)	2,000		Not suitable for smart cards (10KB code memory)
LOKI97	4,800	8,000 (emulated PDP-11µP)	2,000	Gives mathematical proof	Slow, not suitable for smart cards
MAGENTA	23,600	55,000 (Z80µP)	256 (not sure)		Very slow
MARS	393	5,000 (hypothetical 1 cycle/instruction µP)	2,200	Math. proof	Good speed, too much RAM for smart cards
RC6	254	13,900 (8051µP)	176		Very fast on Pentium. Still the best for smart cards
RIJNDAEL	320	3,100 for 8051µP, 8,300 for 6808µP	1,000		Too much RAM for a smart card
SAFER+	2,000	80,000 (8051µP)	**maybe** 256		Quite slow for 8 bit µP's
SERPENT	1,730	34,000 (6805µP)	<1,000 (maybe 256)		Might be OK for smart cards if really 256 Bytes RAM suffice
TWOFISH	285	26,500 (6805µP)	60 +		Very fast if given

Algorithm	Clocks per 128-Bytes block (with 128-Bytes key) encryption		required RAM amount [Bytes]	Security level	Comments
	Pentium	sample 8 bit CPU			
			2.2KB ROM		enough key setup time