Report on the AES Candidates

Olivier Baudron[1], Henri Gilbert[2], Louis Granboulan[1], Helena Handschuh[3], Antoine Joux[4], Phong Nguyen[1], Fabrice Noilhan[5], David Pointcheval[1], Thomas Pornin[1], Guillaume Poupard[1], Jacques Stern[1], and Serge Vaudenay[1]

[1] Ecole Normale Supérieure – CNRS
[2] France Telecom
[3] Gemplus – ENST
[4] SCSSI
[5] Université d'Orsay – LRI

Contact e-mail: `Serge.Vaudenay@ens.fr`

Abstract This document reports the activities of the AES working group organized at the Ecole Normale Supérieure. Several candidates are evaluated. In particular we outline some weaknesses in the designs of some candidates. We mainly discuss selection criteria between the candidates, and make case-by-case comments. We finally recommend the selection of Mars, RC6, Serpent, ... and DFC. As the report is being finalized, we also added some new preliminary cryptanalysis on RC6 and Crypton in the Appendix which are not considered in the main body of the report.

Designing the encryption standard of the first twenty years of the twenty first century is a challenging task: we need to predict possible future technologies, and we have to take unknown future attacks in account.

Following the AES process initiated by NIST, we organized an open working group at the Ecole Normale Supérieure. This group met two hours a week to review the AES candidates. The present document reports its results.

Another task of this group was to update the DFC candidate submitted by CNRS [16, 17] and to answer questions which had been omitted in previous reports on DFC.[1] This issue is subject to another report.

1 On the AES Candidates

1.1 On the AES Evaluation Platform

In order to compare the AES candidates, we had to agree on a platform. For this purpose NIST chose an "AES Evaluation Platform" based on a 200MHz Pentium Pro. Although we have to compare all candidates on the same platform,

[1] Due to the significant overlap between the AES working group of ENS and the designers of DFC, the present report obviously favors DFC.

this must be considered with caution. First of all, Pentium Pro's have long been superseded (we believe they are no longer being produced by Intel) and some candidates may be handicapped by this outdated technology. Second, we cannot reasonably forecast what the dominant technology will be during the period of validity of the AES.

We believe that future technologies will be based on 64-bit architectures rather than on 32-bit ones. Microprocessor manufacturers are currently moving towards 64-bit RISC technology so this forecast seems reasonable. We believe that Intel will popularize the 64-bit microprocessor Merced in the next few years, that this will become a *de facto* standard and that 32-bit microprocessors will gradually disappear, the remaining ones being cheap designs for on-board applications. Judging that the twenty-first century will mainly use 64-bit architecture, Alpha microprocessors appear to be the best technology available for comparing the AES candidates today.

Most of the AES candidates have been designed so that moving from 32-bit to 64-bit architectures does not yield much benefit. Only DFC and HPC have been purposefully designed for the latter architecture. Although their performance on Pentium machines is average, the benefit of moving to Alpha is huge. Actually, DFC appears to be the candidate which admits the fastest implementation of all on Alpha.

We compared the AES candidates on Alpha 21164a 500MHz with the best available implementations.[2] For this we started with the optimized code provided on the AES-CD2 and replaced them once better implementation (even in assembly code) were available to us.[3] Table 1 reports the timing (in clock cycles) of one block encryption with a 128-bit key (timing of the key setup is not included). Timing for E2 is from NTT. Timing for DFC is based on Harley's C implementation (which uses one Alpha ASM instruction if it is available). The timing for E2 is from NTT. Timing for HPC is from assembly code (by the author of HPC). All other timings are from Gladman's implementations [18]. All of our experiments were compiled with the DEC C COMPILER. We note that Safer+ is very slow on Alphas older than the 21164a. The 21164a and successors have extra byte-manipulation instructions which lead to a significant speed-up when appropriate compiler flags are used. Our measurements can be compared with Almquist's [3] estimates which are given below.

1.2 Interoperability

The rules for the AES implementations implied the use of the C language as defined by the ANSI norm. However this suffers from several defects. The ANSI-C

[2] Other researchers did the same. For instance, Almquist [3] implemented several candidates in Alpha pseudo-code and gave some (optimistic) estimates of the encryption timing.

[3] These experiments are reported at the following URL which is regularly updated to track the latest developments.
`http://www.dmi.ens.fr/~granboul/recherche/AES.html`

cipher	measure	estimate	cipher	measure	estimate
Cast256	749	600	Magenta	5074	
Crypton	499	408	Mars	507	478
Deal	2752	2528	RC6	559	467
DFC	323	304	Rijndael	490	340
E2	587	471	Safer+	1502	656
Frog	2752		Serpent	998	915
HPC	402	380	Twofish	490	360
Loki97	2356				

Table1. Timing (in cycles) of one block encryption on 21164a for the best software implementation. The measurement has been performed by Granboulan based on real implementations. The estimates are due to Almquist [3].

norm entails more than just some syntactic requirements. It makes the programs portable by disallowing any endianess dependency or restriction to 32-bit processors. Actually, some of the C implementations provided in the AES-CD2 fail to run when tested on big endian machines or 64-bit microprocessors. In particular, some problems were noted for E2, HPC, Mars, RC6, Serpent, Twofish and Loki97. Typically, if the 128-bit message is represented as an array of 16 bytes (of `unsigned char` type), any cast into some `int` array may cause a bus error, because of alignment problems. Although this is not an important flaw (since the implementations were not required to run on these machines) it illustrates the limits imposed by strict adherence to ANSI-C.

Another issue is that ANSI-C does not guarantee the availability of 64-bit words, which causes a performance penalty for those candidates, DFC and HPC, that are dedicated to 64-bit architectures. Indeed, we believe that comparing only pure ANSI-C implementations is not entirely satisfactory. We should compare the best software implementations, which may take advantage of assembly code and architecture-specific features.

One might consider that comparing the JAVA implementations would be more relevant as Java produces a portable byte-code from one architecture to another. But there are two drawbacks: the first one is that JAVA is a new language and thus, even though the virtual machine is well-defined, compilers of JAVA into JVM byte-code, interpreters of byte code and JIT compilers are changing rapidly. Since July and the end of the submission phase, there have been considerable speed-ups of Java Virtual Machines (mainly due to JIT compilers). The second drawback is that AES candidates have not always been optimized in JAVA. Nevertheless, a comparison of AES candidates in JAVA may be fairer than comparison of ANSI C and JAVA processors should be an interesting platform for the future. Results on UltraSparc running Solaris 2.6 are given in table 2.

1.3 Endianess

Some AES candidates are dedicated to big endian or little endian architectures. Actually, comparing them with the AES API may be unreliable because some candidates may have first to convert from one endianess to the other. Some other

cipher	measure	cipher	measure
Cast256	18360	Magenta	56580
Crypton	10750	Mars	8840
Deal	38620	RC6	6110
DFC	11350	Rijndael	7770
E2	10670	Safer+	30190
Frog	23920	Serpent	10050
HPC	9970	Twofish	14990
Loki97	56580		

Table2. Timing (in cycles) of one block encryption on UltraSparc-I using JDK-1.2 with JIT compiler. The measurement has been performed by Noilhan using NIST API.

candidates have portable C programs which convert `char` streams into `int` arrays, incurring a penalty in both cases. This penalty is somehow unfair because it penalizes interoperability which should instead be rewarded, and because most microprocessors already have some endianess conversion routines (`BSWAP` for Intel) which cost little or nothing, but ANSI-C does not make them available. This needs to be considered when comparing the submitted implementations.

1.4 Trustworthiness

In order to avoid the controversial arguments that were raised against DES [1], we believe that look-up tables should be trustworthy, which means that they should come from a "simple" public algorithm. Otherwise the designer could be suspected of having introduced a concealed trapdoor.

1.5 Simplicity of the Algorithm

Another criterion which should not be underestimated is the simplicity of the algorithm. Usually, simplicity is related to compactness of implementation. It also leads to greater levels of trust by the user, and by the cryptanalyst: for a complicated (insecure) algorithm, there is a longer latency delay before an expert comes up with a practical attack. Since the schedule of the AES process is quite tight, it does not seem reasonable to adopt a complicated algorithm as a standard.

1.6 State of the Art of Symmetric Encryption

In the early beginnings of public research on symmetric encryption, security used to be empirical: there was no specific design argument and a cipher was presumed to be secure until someone came up with an attack. After the discovery of differential and linear cryptanalysis and their variants, security became heuristic: designers could argue that no characteristic had a significant probability and it was believed that this was enough to protect against these attacks.

Public research has now entered a new stage where provable security against given classes of attacks can be attained. First of all, the Luby-Rackoff Theorem [28] (and all its refinements due to Patarin [36–38] and Pieprzyk [39]) can be used. This applies to the Deal AES candidate [35, 23]: assuming that DES looks like a random permutation (see below however), six rounds of a Feistel network with DES round functions are secure.

The Nyberg-Knudsen Theorem [33, 34] and all its extensions due to Aoki and Ohta [5] have been used in the Misty cipher [31, 32]. One problem with this approach is that it does not allow much freedom in the design, and the designer is actually limited by some algebraic properties. In some simple cases, this is dangerous, as shown by Jakobsen and Knudsen [21] because it leads to new kinds of attacks namely, interpolation attacks.

Another possibility is to use decorrelation theory. This approach provides more freedom in the design and allows one to prove security against several classes of popular attacks. Of course, this has to be considered with great care, because some simple decorrelated designs can be broken by new kinds of attacks. We mention Wagner's boomerang attacks [52] against Coconut98 [48]. The reasonable conclusion is that the cipher has to be heuristically secure without the decorrelation modules (which was not the case of Coconut98), and that the decorrelation properties provide an additional level of security.

We think that it is important for the forthcoming encryption standard to take research advances into consideration. Although provable security is not a panacea outside its domain of applicability, we believe it provides added value to new designs. Accordingly, we sorted the candidates into three classes: those which have empirical security, heuristic security, and finally some form of provable security. The class of candidates based on empirical security is quite small since it is restricted to HPC [46] and Frog [15]. The class of provable security is equally reduced: Deal and DFC.

1.7 Smart Cards

It is obvious that implementability on smart cards is an important advantage for the AES candidates, since smart cards are now used in many applications. Accordingly we have to consider two requirements.

- On most popular smart cards, the amount of free RAM is limited to one or two hundred bytes.
- Some architectures based on the Motorola 6805 (which is very popular in smart cards) only have a single byte rotation by one position. This makes rotation of 32-bit words, especially data-dependent rotations (as used in Mars [11] and RC6 [42]) complex or slow or both.

The evolution of the technology of smart cards guarantees that some more modern microprocessors (such as 68000 or even ARM) will be available.

1.8 On the Rotations

Mars and RC6 are two candidates which use data-dependent 32-bit rotations. This is inherently ill-suited to both smaller and larger word sizes.

This can be seen clearly in the optimized implementations which are used by `distributed.net`. On a Pentium Pro, RC5 [41] key search takes about twice as long as DES whereas on Alpha the ratio is close to nine! This is in large part attributable to the fact that with 64-bit words, a 32-bit rotation of x by y requires *five* instructions:

```
x<<(y & 31) | x>>(32-(y & 31))
```

(and possibly a mask operation to clear high-order bits). Thus data-dependent 32-bit rotations are clearly ill-suited to the processors of the future.

1.9 A few Candidates

Crypton [27] is a variant of the Square cipher [12]. We noticed the existence of weak keys (which has been independently discovered by Johan Borst). Actually, the key expands onto something of the form

$$xxxxyyyyxxxxyyyyzzzzttttzzzztttt$$

where x, y, z and t are bytes (we thus have 2^{32} possibilities), then the set of all texts with the form

$$ababcdcdababcdcd$$

is stable by the whole encryption. We thus have 2^{32} 256-bit long weak keys. We do not think this is a serious flaw. In the Appendix we added a preliminary attack.

DFC When considered suing these criteria, DFC is quite fast: at the time of writing, the fastest software implementation on Alpha 21164a is 323 cycles per encrypted block and 232 cycles on a 21264 prototype (which leads to an encryption rate of over 300Mbps on current 575 MHz models with 500Mbps possible in the near future), and 482 cycles on Pentium Pro. DFC is quite simple: a Feistel scheme combining a conservative CP permutation and the decorrelation module $(ax + b) \bmod 2^{64} + 13 \bmod 2^{64}$. The look-up table comes from the hexadecimal expansion of e so that there can be no question of a trapdoor. The decorrelation takes advantage of research results and provides an additional provable security feature (as discussed below). It is portable to simple chips as well (see [40]): we have two implementations on Motorola 6805. The first one requires less than a hundred bytes of RAM by computing the round keys "on the fly", and is faster than triple-DES. The second one requires less than two hundred bytes of RAM and is much faster than DES.

During the first phase of the AES, DFC has been subject to several criticisms. Namely, the number of rounds was said to be too low (Biham [7] suggested that we add one more round), and drawbacks were found in the key scheduling algorithm (weak keys, as noticed by Coppersmith). These criticisms do not raise any concerns of significant weakness in our design, but can nonetheless be addressed easily as shown in [6].

Deal [23, 35] One problem with Deal is that it is clearly slower than triple-DES (it requires at least 6 DES computations plus some extra operations in order to encrypt 128-bit blocks while triple-DES requires only 3 DES computations in order to encrypt 64-bit blocks). Furthermore, it is based on the DES design which was adapted to the technology of the 70s. We now have more efficient designs available.

As noticed by Lucks [29], the key schedule of Deal makes the first round key depend on too few bits of the secret key. This leads to a variant of Knudsen's "impossible differential attack" [23]. We can refine it by noticing that we can take advantage of badly distributed keys. For instance, if we use a 192-bit key which consists of 24 lower case alphabetical ASCII characters, we have a diversity of roughly 2^{120}, which may appear sufficient. However the first round key only has 2^{40} possibilities instead of 2^{56}, so that we can improve on Lucks' attack in this setting: we need 2^{70} chosen plain-texts and a complexity of 2^{109} easy tests. This is still quite high, but is definitely a design flaw.

We noticed that we can use the complementation property of DES in order to decrease the exhaustive search complexity by a factor of 4. (Here we have two complementation properties: one for the even rounds, and one for the odd rounds.) We can use these properties with the meet-in-the-middle attack and break a 192-bit key (resp. a 256-bit key) within 2^{166} encryptions (resp. 2^{222}) in the worst case.

Deal uses an interesting security feature. Assuming that DES behaves like a random permutation, Patarin's Theorem [38] asserts that one needs at least 2^{48} known plain-texts to break it with unlimited computation. However this depends on an assumption which is quite debatable. Actually, there are only 2^{56} possibilities for each round instead of $2^{64}! \approx 2^{2^{70}}$, so the assumption consists of approximating $2^{2^{70}}$ by 2^{56}... How can we manage to prove something without any assumption? We are not aware of any formal result which proves that the the security of 6 rounds of a Feistel cipher with DES as a round function requires a complexity which is, say, squared from the complexity of attacks on DES. We can still prove that Deal has at least the same security as DES (without assuming that DES looks random). In particular, unless we find an attack against DES better than Matsui's, we need at least 2^{43} known plain-texts to break Deal. We know that this provable security result is not enough.

Frog and HPC [15, 46] Actually these two candidates do not reference any research results. Their security is empirical in the sense that they will be considered secure until someone finds an attack. We think this paradigm ignores all recent advances in research and should not be considered.

Magenta [20] Due to the weakness of the key schedule which has been noticed by Biham *et al.* [8] and the strange property that the round function is not bijective (and actually collapses on fixed points when iterated as noticed by the authors themselves) we do not think that Magenta is a good candidate for the AES.

Mars [11] uses the MUL instruction of 32-bit microprocessors. It also uses data-dependent 32-bit rotations, which are inherently ill-suited to both smaller and larger word sizes. Overall, the implementation requires a 2KB lookup table, which is far too much for smart cards. The rationale for the substitution boxes is very mysterious (and actually not simple enough to demonstrate that no trapdoor is concealed there). Mars also suffers from being complicated. Over all the AES candidates, Mars seems to have required considerable resources, which does not necessarily mean that it is secure: designers are notorious for overlooking weaknesses in their own algorithms. If there exists some attack against Mars, we believe it will take more than six months (the time between the first AES workshop and the deadline for submitting the present report) to notice it. We believe that Mars deserves further investigations though.

RC6 [42] uses the MUL instruction of 32-bit microprocessors. It also uses data-dependent 32-bit rotations, which are inherently ill-suited to both smaller and larger word sizes.

We also noticed that the internal f function has many fixed points. Namely, any 32-bit word whose 16 least significant bits are set to zero is a fixed point.

In the Appendix we added a preliminary statistical attack.

Safer+ [30] is a byte-oriented candidate. Its performance on NIST's suggested 32-bit platform and on 64-bit platforms is thus poor. Its security is only heuristic. The design includes a so-called Armenian shuffle which has many mysterious properties. For instance, we noticed that the Armenian shuffle does not mix up odd and even positions: it shuffles the odd positions and the even ones separately. We noticed that the round matrix consists of two rows and some permutations of it (at even and odd positions). We also noticed some properties like stable subspaces: for instance, the set of all byte-vectors $(a, b, a, b, \ldots)$ is stable by the round linear transformation. Although we were not able to forge an attack based on these observations, we wonder if they may imply some weakness.

Serpent [4] is an overly conservative design. It is based on tricky optimized implementations on a conservative architecture (32-bit based). It shows no advantage when moving towards 64-bit (the C implementation provided by NIST actually takes more cycles on an Alpha than on a Pentium Pro.) Its security is only heuristic. The authors provide an heuristic argument for a version of Serpent reduced to 8 rounds and they choose an unexpectedly large 32 rounds in order to have confidence in its security. (This may however be the most trustworthy way to get real security.) Another concern is the unclear origin of the substitution boxes. Again, this may open the way to the hidden-trapdoor controversy.

Twofish [45] is designed for 32-bit architectures, with little consideration for future 64-bit ones. Its key schedule is very complicated and has a high complexity. One important feature of Twofish is key-dependent S-boxes. This was used in the Blowfish [43, 44] cipher. Since the so called key-dependent S-boxes of Twofish consist of two fixed S-boxes which are transformed by XORing with two key bytes, we believe they should no longer be called "key-dependent S-boxes", but a regular design with fixed S-boxes. Twofish uses many other tricks taken from other block cipher algorithms and actually consists of a collection of patches which fix various problems. The result lacks simplicity, and it is quite hard to make any serious analysis of this algorithm, which does not necessarily mean it is secure. When compared to Mars, we do not think that this design comes from deep investigations.

2 On Decorrelating the AES

The main added value of DFC is the kind of provable security provided by the decorrelation paradigm. We should however be careful about what this really means. First because it provides formal proofs of security in some classes of attacks. In particular, it does not provide any proof of security against other attacks. As an example we can mention Wagner's Boomerang attack [52] which breaks Coconut98. Second because this provides formal proofs of security in some formal security model. History shows that some crypto-systems can be secure in some security model, but that some real-life attacks may bypass this model. The best example is Bleichenbacher's impressive attack against RSA PKCS#1 [9] which does not break the plain RSA scheme at all.

As an interesting feature, we outline that the decorrelation theory focuses on average complexity over the distribution of the secret key. Some restricted sets of keys can still be "weak" (as shown by Coppersmith for instance), but these sets will necessarily be of limited size because of the average case. In particular, our results do not rely on some unproven assumption on the stochastic equivalence of the keys (as for instance the theory of Markov ciphers [24–26]).

Interestingly, the construction suggested by decorrelation theory is very flexible in the sense that — for instance — we can freely choose the CP Confusion Permutation and still keep the decorrelation property. This should be compared to the Nyberg-Knudsen provable security approach. This does not mean that CP is useless. It means that we can construct a cipher in some conservative way with CP and add the decorrelation module to get the provable security added value. For this we choose some conservative CP function with which the cipher is heuristically secure even when all ARK_i values are known by the attacker. (These are the "a coefficient" parts in the RK_i round keys.)

Of course the decorrelation modules could be added to other AES candidates in order to provide provable security as well. This suggests an avenue for future research.

3 Conclusion

Based on the discussion above, we recommend that the following candidates not be selected.

– HPC and Frog, due to the lack of any reference to research advances.
– Magenta, due to security flaws.
– Deal, due to insufficient security and outdated technology.

Although we did not find any serious flaw we are reluctant to recommend the following algorithms.

– Crypton and Rijndael [13], for security.
– Safer+, because of its 8-bit oriented (outdated) design.
– Twofish, because of its obscure design.

Since we did not investigate Cast256 [2], E2 [22] and Loki97 [10], we will not comment on them.

Finally, we recommend the following algorithms.

– Mars, RC6, Serpent,

... and of course DFC.

Acknowledgements

We wish to thank Don Coppersmith for identifying the weak keys of DFC. We also thank Brian Gladman and Robert Harley for their impressive implementations of DFC as well as Eli Biham for valuable discussions, Kazumaro Aoki and Doug Whiting for their comments and the NIST for organizing this exciting work.

References

1. *FIPS 46*, Data Encryption Standard. U.S. Department of Commerce — National Bureau of Standards, National Technical Information Service, Springfield, Virginia. *Federal Information Processing Standard Publication 46*, 1977.
2. C. Adams. The Cast-256 Encryption Algorithm. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
3. K. Almquist. AES Candidate Performance on the Alpha 21164 Processor (version 1). Published in the `sci.crypt` Usenet Newsgroup. 23rd of December, 1998.
4. R. Anderson, E. Biham, L. Knudsen. Serpent: a Flexible Block Cipher with Maximum Assurance. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
5. K. Aoki, K. Ohta. Strict Evaluation of the Maximum Average of Differential Probability and the Maximum Average of Linear Probability. *IEICE Transactions on Fundamentals*, vol. E80-A, pp. 1–8, 1997.
6. O. Baudron, H. Gilbert, L. Granboulan, H. Handschuh, R. Harley, A. Joux, P. Nguyen, F. Noilhan, D. Pointcheval, T. Pornin, G. Poupard, J. Stern, S. Vaudenay. DFC Update. Submitted.

7. E. Biham. Invited talk given at the Asiacrypt'98 Conference. Slides available on `http://www.cs.technion.ac.il/~biham/`
8. E. Biham, A. Biryukov, N. Ferguson, L. R. Knudsen, B. Schneier, A. Shamir. Cryptanalysis of Magenta. Draft distributed during the *First Advanced Encryption Standard Candidate Conference*, August 1998.
9. D. Bleichenbacher. Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS#1. In *Advances in Cryptology CRYPTO'98*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 1462, pp. 1–12, Springer-Verlag, 1998.
10. L. Brown, J. Pieprzyk. Introducing the new Loki97 Block Cipher. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
11. C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Julta, S. M. Matyas Jr., L. O'Connor, M. Peyravian, D. Safford, N. Zunic. Mars - a Candidate Cipher for AES. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
12. J. Daemen, L. Knudsen, V. Rijmen. The Block Cipher Square. In *Fast Software Encryption*, Haifa, Israel, Lectures Notes in Computer Science 1267, pp. 149–171, Springer-Verlag, 1997.
13. J. Daemen, V. Rijmen. Rijndael. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
14. H. Feistel. Cryptography and Computer Privacy. *Scientific American*, vol. 228, pp. 15–23, 1973.
15. D. G. Georgoudis. Frog. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
16. H. Gilbert, M. Girault, P. Hoogvorst, F. Noilhan, T. Pornin, G. Poupard, J. Stern, S. Vaudenay. Decorrelated Fast Cipher: an AES Candidate. (Extended Abstract.) In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
17. H. Gilbert, M. Girault, P. Hoogvorst, F. Noilhan, T. Pornin, G. Poupard, J. Stern, S. Vaudenay. Decorrelated Fast Cipher: an AES Candidate. Submitted to the Advanced Encryption Standard process. In *CD-ROM "AES CD-1: Documentation"*, National Institute of Standards and Technology (NIST), August 1998.
18. B. Gladman. `http://www.seven77.demon.co.uk/crypto_technology.htm`
19. H. Handschuh, H. Heys. A Timing Attack on RC5. To appear in SAC'98, LNCS.
20. K. Huber, M. J. Jacobson Jr. The Magenta Block Cipher Algorithm. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
21. T. Jakobsen, L. R. Knudsen. The Interpolation Attack on Block Ciphers. In *Fast Software Encryption*, Haifa, Israel, Lectures Notes in Computer Science 1267, pp. 28–40, Springer-Verlag, 1997.
22. M. Kanda, S. Moriai, K. Aoki, H. Ueda, M. Ohkubo, Y. Takashima, K. Ohta, T. Matsumoto. E2 - a Candidate Cipher for AES. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.

23. L. R. Knudsen. DEAL - A 128-Bit Block Cipher. Submitted to the Advanced Encryption Standard process. In *CD-ROM "AES CD-1: Documentation"*, National Institute of Standards and Technology (NIST), August 1998.
24. X. Lai. *On the Design and Security of Block Ciphers*, ETH Series in Information Processing, vol. 1, Hartung-Gorre Verlag Konstanz, 1992.
25. X. Lai, J. L. Massey. A Proposal for a New Block Encryption Standard. In *Advances in Cryptology EUROCRYPT'90*, Aarhus, Denemark, Lectures Notes in Computer Science 473, pp. 389–404, Springer-Verlag, 1991.
26. X. Lai, J. L. Massey, S. Murphy. Markov Ciphers and Differential Cryptanalysis. In *Advances in Cryptology EUROCRYPT'91*, Brighton, United Kingdom, Lectures Notes in Computer Science 547, pp. 17–38, Springer-Verlag, 1991.
27. CRYPTON: a 128-bit Block Cipher - Specification and Analysis. Submitted to the Advanced Encryption Standard process. In *CD-ROM "AES CD-1: Documentation"*, National Institute of Standards and Technology (NIST), August 1998. Extended abstract in *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
28. M. Luby, C. Rackoff. How to Construct Pseudorandom Permutations from Pseudorandom Functions. *SIAM Journal on Computing*, vol. 17, pp. 373–386, 1988.
29. S. Lucks. On the Security of the 128-Bit Block Cipher DEAL. Preprint. 1998.
30. J. L. Massey, G. H. Khachatrian, M. K. Kuregian. Safer+. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
31. M. Matsui. New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis. In *Fast Software Encryption*, Cambridge, United Kingdom, Lectures Notes in Computer Science 1039, pp. 205–218, Springer-Verlag, 1996.
32. M. Matsui. New Block Encryption Algorithm MISTY. In *Fast Software Encryption*, Haifa, Israel, Lectures Notes in Computer Science 1267, pp. 54–68, Springer-Verlag, 1997.
33. K. Nyberg, L. R. Knudsen. Provable Security against a Differential Cryptanalysis. In *Advances in Cryptology CRYPTO'94*, Santa Barbara, California, U.S.A., Lectures Notes in Computer Science 839, pp. 566–574, Springer-Verlag, 1994.
34. K. Nyberg, L. R. Knudsen. Provable Security against a Differential Cryptanalysis. *Journal of Cryptology*, vol. 8, pp. 27–37, 1995.
35. R. Outerbridge, L. R. Knudsen. AES Candidate DEAL. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
36. J. Patarin. *Etude des Générateurs de Permutations Basés sur le Schéma du D.E.S.*, Thèse de Doctorat de l'Université de Paris 6, 1991.
37. J. Patarin. How to Construct Pseudorandom and Super Pseudorandom Permutations from One Single Pseudorandom Function. In *Advances in Cryptology EUROCRYPT'92*, Balatonfüred, Hungary, Lectures Notes in Computer Science 658, pp. 256–266, Springer-Verlag, 1993.
38. J. Patarin. About Feistel Schemes with Six (or More) Rounds. In *Fast Software Encryption*, Paris, France, Lectures Notes in Computer Science 1372, pp. 103–121, Springer-Verlag, 1998.
39. J. Pieprzyk. How to Construct Pseudorandom Permutations from a Single Pseudorandom Functions. In *Advances in Cryptology EUROCRYPT'90*, Aarhus, Denemark, Lectures Notes in Computer Science 473, pp. 140–150, Springer-Verlag, 1991.

40. G. Poupard, S. Vaudenay. Decorrelated Fast Cipher: an AES Candidate well suited for Low Cost Smart Cards Applications. To appear in Cardis 98, LNCS.
41. R. L. Rivest. The RC5 Encryption Algorithm. In *Fast Software Encryption*, Leuven, Belgium, Lectures Notes in Computer Science 1008, pp. 86–96, Springer-Verlag, 1995.
42. R. L. Rivest, M. J. B. Robshaw, R. Sidney, Y. L. Yin. The RC6 Block Cipher. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
43. B. Schneier. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In *Fast Software Encryption*, Cambridge, United Kingdom, Lectures Notes in Computer Science 809, pp. 191–204, Springer-Verlag, 1994.
44. B. Schneier. The Blowfish Encryption Algorithm. In *Dr Dobb's Journal*, pp. 38–40, April 1994.
45. B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson. Twofish - a Block Encryption Algorithm. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
46. R. Schroeppel, H. Orman. Introduction to the Hasty Pudding Cipher. In *Proceedings from the First Advanced Encryption Standard Candidate Conference*, National Institute of Standards and Technology (NIST), August 1998.
47. S. Vaudenay. An experiment on DES — Statistical Cryptanalysis. In *3rd ACM Conference on Computer and Communications Security*, New Delhi, India, pp. 139–147, ACM Press, 1996.
48. S. Vaudenay. Provable Security for Block Ciphers by Decorrelation. In *STACS 98*, Paris, France, Lectures Notes in Computer Science 1373, pp. 249–275, Springer-Verlag, 1998.
49. S. Vaudenay. Feistel Ciphers with L_2-Decorrelation. To appear in SAC'98, LNCS.
50. S. Vaudenay. The Decorrelation Technique Home-Page. URL:`http://www.dmi.ens.fr/~vaudenay/decorrelation.html`
51. S. Vaudenay *Vers une Théorie du Chiffrement Symétrique*, Dissertation for the diploma of "habilitation to supervise researches" from the University of Paris 7, Technical Report LIENS-98-15 of the Laboratoire d'Informatique de l'Ecole Normale Supérieure, 1998.
52. D. Wagner. The Boomerang Attack. Personal communication.

A Appendix: Preliminary Results

When the present report was being terminated, we discovered some interesting attacks on RC6 and Crypton. Since we did not check everything, we warn the reader about correctness issues. We think that these properties might be of some interest anyway.

A.1 Statistical Attack on RC6

RC6 consists of iterating r times the following transformation on four w-bit registers A, B, C, D.

$$(A, B, C, D) \leftarrow (B, ((C \oplus u) <<< t) + S_{2i+1}, D, ((A \oplus t) <<< u)) + S_{2i}$$

where $t = f(B)$, $u = f(D)$, and f is such that $f(x) \bmod w$ (which is the useful part for the rotations) is a pseudo-random function of x. Our basic observation is that if $t \bmod w = u \bmod w = 0$, then the mod w part of A and C is just added with some key-bits. Thus, if this event occurs in one out of two rounds (which holds with probability $w^{-2\lfloor \frac{r}{2} \rfloor}$), then the mod w part of two output registers will consist of the mod w part of two input registers added with some key-bit constants.

Let us call Δ_1 and Δ_2 the mod w difference of the corresponding two registers. We can mount a distinguisher between RC6 and a random permutation by just making an experimental distribution table of all (Δ_1, Δ_2) values. Following [47], the Euclidean distance between the uniform distribution and the expected distribution is $D = w^{1-r}$ for a space of w^2 values. This leads to a χ^2 attack within a complexity of $O(1/(wD^2))$ which is $O(w^{4\lfloor \frac{r}{2} \rfloor - 3})$. Since the number of samples is limited to 2^{4w}, we can break up to $r = \lceil \frac{2w}{\log_2 w} + \frac{3}{2} \rceil$ rounds. For the nominal choice of RC6 $w = 32$, this leads to $r = 15$ for which the complexity is 2^{125}.

We can still increase the number of vulnerable rounds by making an attack which has a low probability of success: assuming that the same attack limited to 2^{4w} queries will have a probability of success of $2^{4w}.w^{-4\lfloor \frac{r}{2} \rfloor - 3}$, we can mount an attack on the nominal RC6 ($r = 20$) with a probability of success of 2^{-57} and a complexity of 2^{128}. This would mean that this attack holds for 192-bit or 256-bit keys with complexity better than exhaustive search.

This attack can be used with the standard tricks, for instance Matsui's attack which consists of guessing the key in one round.

We also considered using other attack models, namely the timing attacks which were investigated in [19].

A.2 Differential Attack on Crypton

Notation : $A_i = (a_{3,i}, a_{2,i}, a_{1,i}, a_{0,i})^t$ denotes one of the four 32-bit column vectors of the A input to the π_o or π_e linear permutation. We introduce the following additional notation to further split each of the $a_{j,i}$ bytes into four 2-bits words: $a_{j,i} = (a_{3,j,i}, a_{2,j,i}, a_{1,j,i}, a_{0,j,i})$. With the above notation, the 128-bit word A is split in 64 $a_{k,j,i}$ 2-bit words. In the sequel, The i and j indexes will be implicitly taken modulo 4. The following invariance properties lead to iterative differential-like and linear-like attacks with only 4 active S-boxes per round.

Differential invariance property. Let $i \in [0,3]$ be any column number, $j \in [0,3]$ be any line number, k be any index in $[0,3]$, d_1, d_2, d_3, d_4 be four 2-bits words. Denote by $\Delta[d_1, d_2, d_3, d_4, k, j, i]$ the following input difference on four or the 2-bit words of A: $\Delta a_{k,j,i} = d_1$; $\Delta a_{k,j,i+2} = d_2$; $\Delta a_{k,j+2,i} = d_3$; $\Delta a_{k,j+2,i+2} = d_4$ (the 60 other 2-bit difference values of δA are taken equal to zero). It can be shown that if $d_1 = d_3$ and $d_2 = d_4$, then the resulting difference on the transformed A' value obtained at the output of the π_o or the π_e linear mapping can be written $\Delta[d_1, d_2, d_1, d_2, k, j', i]$ (where j' is equal or not to j, depending on

i, j, k and the parity of the considered round), and that the resulting difference on the transformed A'' value obtained at the output of the τ byte transposition (on the A' input) can be written $\Delta[d_1, d_1, d_2, d_2, k, j'', i']$, for some j'' and i' values. This induces iterative differential behaviors which involve only four active S-boxes per round. As a matter of fact, assume an input difference of the form $\Delta[d_1, d_2, d_1, d_2, k, j, i] \oplus \Delta[d'_1, d'_2, d'_1, d'_2, k+2, j, i]$ at the input of the π_o or π_e linear transformation; we expect (but did not check yet in detail) that the p probability of obtaining, after one entire ρ Crypton round, a resulting output difference of the form $\Delta[d_1, d_2, d_1, d_2, k, j, i] \oplus \Delta[d'_1, d'_2, d'_1, d'_2, k+2, j, i]$ is at least $p = 2^{-24}$. This provides a differential of probability p, which can be used to mount attacks on Crypton with a limited number of rounds. This way we can expect to mount a differential attack against 6 rounds of Crypton (instead of 12) faster than an exhaustive search.

Linear invariance property. Denote by $\Phi[A, k, j, i]$ the $a_{k,j,i} \oplus a_{k,j+2,i} \oplus a_{k,j,i+2} \oplus a_{k,j+2,i+2}$ sum. Denote by A' the transformed value obtained from A at the output of the π_o or π_e linear mapping, and by A'' the transformed value obtained from A' at the output of the τ transposition. It can be shown that for any (k, j, i) triplet there exist (k', j', i') and (k'', j'', i'') triplets such that $\Phi[A', k, j, i] = \Phi[A, k', j', i']$; $\Phi[A'', k, j, i] = \Phi[A, k'', j'', i'']$. Moreover, for the same i, j, k, i'', j'', k'' values, we have $\Phi[A'', k+2, j, i] = \Phi[A, k''+2, j'', i'']$. Therefore, if B denotes the transformed value obtained after one entire round, the $(4 * \Phi[B, k''+2, j'', i''] \oplus \Phi[B, k'', j'', i''])$ output nibble is correlated to the $(4 * \Phi[A, k+2, j, i] \oplus \Phi[A, k, j, i])$ input nibble. Such correlation properties can be easily concatenated to mount attacks on Crypton. (For instance, the above property implies that there is some correlation between linear combinations of the above input nibble and linear combinations of the above output nibble: this provides chains of linear characteristics involving only 4 active S-boxes per round). For the time being, we have no precise assessment of the performance of the resulting attacks.