The Twofish Team's Final Comments on AES Selection

Bruce Schneier* John Kelsey† Doug Whiting‡
David Wagner§ Chris Hall¶ Niels Ferguson‖
Tadayoshi Kohno** Mike Stay††

May 15, 2000

1 Introduction

In 1996, the National Institute of Standards and Technology initiated a program to choose an Advanced Encryption Standard (AES) to replace DES [NIST97a]. In 1997, after soliciting public comment on the process, NIST requested proposed encryption algorithms from the cryptographic community [NIST97b]. Fifteen algorithms were submitted to NIST in 1998. NIST held two AES Candidate Conferences—the first in California in 1998 [NIST98], and the second in Rome in 1999 [NIST99a]—and then chose five finalist candidates [NIST99b]: MARS [BCD+98], RC6 [RRS+98], Rijndael [DR98], Serpent [ABK98], and Twofish [SKW+98a]. NIST held the Third AES Candidate Conference in New York in April 2000 [NIST00], and is about to choose a single algorithm to become AES.

We, the authors of the Twofish algorithm and members of the extended Twofish team, would like to express our continued support for Twofish. Since first proposing the algorithm in 1998 [SKW+98a, SKW+99c], we have continued to perform extensive analysis of the cipher, with respect to both secu-

*Counterpane Internet Security, Inc., 3031 Tisch Way, 100 Plaza East, San Jose, CA 95128, USA; schneier@counterpane.com.

†Counterpane Internet Security, Inc. kelsey@counterpane.com.

‡Hi/fn, Inc., 5973 Avenida Encinas Suite 110, Carlsbad, CA 92008, USA; dwhiting@hifn.com.

§University of California Berkeley, Soda Hall, Berkeley, CA 94720, USA; daw@cs.berkeley.edu.

¶Princeton University, Fine Hall, Washington Road, Princeton, NJ 08544; cjh@princeton.edu.

‖Counterpane Internet Security, Inc. niels@counterpane.com.

**Reliable Software Technologies, 21351 Ridgetop Circle, Suite 400, Dulles, VA 20166, USA; kohno@rstcorp.com.

††AccessData Corp., 2500 N. University Ave, Suite 200, Provo, UT 84606, USA; staym@accessdata.com.

rity [SKW+98b, Fer98, WW98, WKS+99, Fer99, FKS+00b, Kel00] and performance [WS98, SKW+99b]. We feel that Twofish offers the best security/performance tradeoff of all the AES finalists, and urge NIST to choose Twofish as the single AES standard.

These comments are an expanded version of the comments we submitted to the Third AES Candidate Conference [SKW+00].

2 Security

Security is not only the most important, but also the most difficult characteristic to compare. In the absence of any theoretical ways of measuring security, we can only fall back on estimates and guesses. "I can't break this algorithm, and all those other smart people can't either" is the best we can say. Hence, all discussions about security rely on this type of non-rigorous argument.

When looking at the published cryptanalysis on the AES finalists, it is important to keep in mind what the data mean. Historically, cryptanalytic results against any algorithm have improved over time [KFS+00]. Initial results might cryptanalyze a simplified variant of the algorithm, or a version of the algorithm with fewer rounds. Later results improve on those initial results: more rounds or less simplification. Finally, there may be a successful attack against the full algorithm.

This is why the published cryptanalysis against the AES finalists, even though none of the results approach practicability and none of the attacks are of any use against the full version of the algorithms, are so important. By comparing how close the published attacks come to breaking the full algorithms, we can get some inkling about how the algorithms' security compares. This is not a perfect comparison by any means, but it is the best we have to go on.

2.1 Safety Factors

The best measure of security that we have come across is the *safety factor*. Eli Biham first compared the AES candidates in this manner when he calculated the "minimal secure rounds" [Bih99]. Lars Knudsen also used this factor when he discussed the AES candidates in his first-round comments [Knu99].

Let n be the number of rounds of the full cipher, and b be the largest number of rounds that has been broken. The safety factor σ is defined as $\sigma := n/b$. A broken cipher has a safety factor of 1. A safety factor of 2 corresponds to a cipher for which a version with half the rounds has been broken.

In this context we are very liberal in our definition of what it entails to break a cipher. The most straightforward type of breaking is to find a key-recovery attack: an attack that recovers the key faster than a brute-force search. However, we also include any other non-random property that can be detected faster than an exhaustive search of the key space in our definition of "breaking" a cipher. This can include a statistical test that distinguishes the cipher from a random permutation, detectable relationships between encryptions with different keys, or more generally any detectable property that an ideal cipher would not have.

This definition is fairly arbitrary, but it is the best we have found given the situation. Excluding certain types of attacks as "unfair," or certain detectable properties as "unimportant," would be even more arbitrary. Including unsubstantiated claims of the form "I think I can break x rounds," or "this property might lead to an attack" make the measurement completely arbitrary. It is also reasonable not to consider any other type of simplifications, such as modifying the rounds themselves. It is trivial to attack Rijndael with a linear S-box, or Twofish without the PHT. Taking any attacks of that type into consideration would also lead to completely arbitrary measurements. The biggest inherent problem in our definition of safety factor is that it favors ciphers on which little cryptanalysis has been done. Unfortunately, this is unavoidable if we want to try to maintain at least some kind of objectivity.

There are two problems in applying this definition to the AES finalists. The first one is that most attacks are against the 256-bit-key versions of the algorithm. Thus we might have a reasonable amount of data on how many rounds we can break of each cipher in 2^{256} steps, but there is very little information on how many rounds we can break in either 2^{128} or 2^{192} steps. We therefore use the largest number of rounds broken by any attack on any one of the key sizes. This gives a fairly accurate result for 256-bit key sizes, and introduces a bias for smaller key sizes. At first glance this seems unfair to Rijndael, since we compare the number of rounds attackable under 256-bit keys to the number of rounds in the 128-bit-key cipher. However, this is exactly what we do for all the other algorithms. If we had more information on attacks on 128-bit-key versions we could compute safety factors for each of the key sizes, but we simply don't have that information. Rijndael's reduced number of rounds for smaller keys gives it a speed advantage, but it also reduces the safety factor for those key sizes.

The second problem is MARS. Because of its heterogeneous structure, there are several ways of defining reduced-round versions. It is unfair to only count the 16 core rounds, but it is equally unfair to give all 32 rounds the same weight. We suggest the following: give the core rounds a weight of 1, and the mixing rounds a weight of α (where α is a parameter that we still have to choose). An attack on c core rounds and m mixing rounds would thus give a safety factor of $(16 + 16\alpha)/(c + m\alpha)$, and all attacks are measured using this metric. Finally, we choose α in such a way as to maximize the resulting safety factor. Thus, the weight α is chosen to favor the algorithm as much as possible. This is not ideal, but it is the only reasonable way we have found of getting a number that is somewhat comparable to what we get for the other ciphers.

The best attacks on MARS that we know of are on 11 core rounds [KKS00a] and on the full mixing layers plus 8 core rounds [Fer00]. This gives us a safety factor of 1.90, with $\alpha = 0.30$.

The other ciphers are easier. For RC6, statistical methods can distinguish 17 rounds from a randomly chosen permutation for 2^{-80} of all keys in less than 2^{128} steps [KM00]. Given 2^{80} keys to attack, we can find nonrandom behavior in about 2^{208} steps. This gives a safety factor of $20/17 \approx 1.18$. Rijndael has an

Algorithm	Safety factor
MARS	1.90
RC6	1.18
Rijndael	1.11/1.33/1.56
Serpent	3.56
Twofish	2.67
34-round RC6	2.00
18-round Rijndael	2.00
24-round Rijndael	2.67

Table 1: Safety factors for AES finalists and some increased-round variants.

attack on 9 rounds [FKS+00a], and for the three key sizes has a safety factor of 1.11/1.33/1.56 (for the three key sizes, respectively). Serpent has a 9-round attack [KKS00b], for a safety factor of 3.56. Twofish has a 6-round attack [Fer99], for a safety factor of 2.67.

The results are tabulated in Table 1 and are not very surprising. RC6 and Rijndael have the smallest safety factors. MARS does better, and Twofish better still. As expected, Serpent has the highest safety factor.

Keep in mind that a safety factor of 1 corresponds to a broken cipher. Thus, even moderate advances in cryptanalysis could endanger RC6 and Rijndael. In his first-round comments Lars Knudsen recommended that AES should have a safety factor of at least 2 [Knu99]. We strongly support that notion. The worst thing that could happen to AES is a successful attack a decade from now, even an "academic attack." Not only would this create havoc in many systems, it could also endanger confidential data that was encrypted before the break. Given the very sketchy information we have to go on, we simply cannot afford to gamble on a relatively small safety factor.

In our opinion, Twofish and Serpent have good safety factors. MARS is close, but RC6 and Rijndael clearly need more rounds. The table shows that 34-round RC6 and 18-round Rijndael would have a safety factor of 2. To raise the safety factor of Rijndael to the same level as that of Twofish would require 24 rounds of Rijndael. Of course, an increase in the number of rounds results in a corresponding reduction in performance. This will have to be taken into account in any comparison with increased-round versions.

2.2 Twofish and Security

Twofish was not designed in an ad hoc manner: our primary goal was security. Each component in the cipher exists for a specific purpose; in each case, that purpose is clearly articulated in our submission document. We weighed every proposed change to the cipher, considering its implications for security, performance, and simplicity. Twofish has the most thoroughly explicated design of any AES candidate.

The Twofish submission document [SKW+98a] contains the most extensively documented cryptanalysis of any AES candidate. We have strongly held to the philosophy that anyone cryptanalyzing Twofish should have the most possible information, so as to have a better chance of finding successful attacks. To that end, not only did we disclose all of our design rationale [SKW+98a, Sections 6-7], but we disclosed all of our internal analysis [SKW+98a, Section 8]. Anyone trying to break Twofish has access to everything we were thinking during our design process. By reading through the Twofish design document, a cryptanalyst can better understand the attacks we thought of—and, by extension, the attack avenues we might have overlooked—the assumptions we made, and the prejudices we held. We made every effort we could to facilitate analysis. The AES process was too short to do anything less.

To date, the Twofish round function has proven to be the strongest round function of any of the finalists. The best known attack can break 6 rounds of Twofish compared to at least 9 rounds for any of the other finalists. Moreover, Twofish is the AES candidate with the fewest published attacks: not because people have not tried to cryptanalyze Twofish, but because they have not found anything worth publishing. We believe that these two facts speak to its strength.

Especially interesting are the cryptanalysis papers, written by other people, that have been published about Twofish. In 1999, Fauzan Mirza and Sean Murphy published a note describing a "key separation property" in Twofish [MM99]. In 2000, Murphy published a another note, further describing the same phenomenon [Mur00]. (The Twofish team also published notes on this property [WKS+99, Kel00], which essentially exists in all ciphers.) In two years, neither Murphy nor the Twofish team has been able to turn this property into an attack against even a reduced-round version of Twofish.

At the 7th Fast Software Encryption workshop in New York, Lars Knudsen announced a truncated-differential attack against reduced-round versions of Twofish [Knu00a]. At the Third AES Candidate Conference, two days later, he retracted his result as being wrong [Knu00b]. As it turned out, the very same structure of Twofish that he was using prevented the attack he presented.

The Twofish team had a similar experience with a related-key attack published in the original Twofish document [SKW+98a, Section 8.7]. While attempting to improve our attack, we realized that our assumptions were extremely generous, and after further evaluation we concluded that the attack was more constrained than we thought. This was the "Twofish Retreat" paper we published [FKS+00b].

During the Third AES Candidate Conference, we joked that Twofish was the only algorithm that not only prevented attacks, but actively repelled them. There is a modicum of truth in this statement. We designed Twofish to frustrate attacks: both known attacks and new, unknown attacks. What we have seen over the past two years is that cryptanalytic attacks that seem to work fail when actually applied to the cipher. We feel that this is a direct result of our design rule to break up as much structure as possible: the key-dependent S-boxes, the one-bit rotations, the deliberate non-algebraic representation. We will return to this point in the section on complexity.

It's relatively easy to design ciphers that are secure against known attacks; it's much harder to design ciphers that are secure against unknown attacks. Amplified boomerangs successfully attacked 9 rounds of Serpent [KKS00b] and 11 rounds of the MARS core [KKS00a]. Statistical attacks on data-dependent rotations have been used to attack 15–17 rounds of RC6 [GHJ+00, KM00], and Rijndael's highly structured round function allowed relatively straightforward extensions of attacks that were already known [FKS+00a, Luc00, GM00]. Since the AES candidates were published in 1998, several new cryptanalytic attacks were published: impossible differential cryptanalysis [BBS99a, BBS99b], mod n cryptanalysis [KSW99], boomerang attacks [Wag99, KKS00a], and slide attacks [BW99, BW00]. Twofish's resistance to these hitherto unknown attacks speaks of its security.

2.3 Power Analysis and Block Ciphers

At the Second AES Candidate Conference, some people looked at the AES submissions with respect to differential power analysis (DPA). We do not feel that this is a relevant criterion by which to judge block ciphers. DPA [KJJ99] is a instance of "side-channel attacks"—cryptanalysis that makes use of information other than the algorithm's inputs and outputs. Examples of side channels include timing [Koc96], power [KJJ99], radiation, etc. [KSWH98].

Paul Kocher's consulting company, Cryptography Research, has done considerable work on side-channel attacks, especially DPA [CR00]. They have a portfolio of patent applications that they license to companies wanting to build DPA-resistant hardware. None of these patent applications affect the mathematics of the algorithm; they all affect the implementation. They include ways to shield power leakage, ways to balance circuits, and ways to randomize power consumption.

Our own research on side-channel attacks bears the same conclusion: the proper place to defend against side-channel attacks is at the hardware level; it is not possible to design in resistance to side-channel attacks, power attacks in particular, at the algorithm level. Many DES implementations are easy to break with DPA, despite its use of only XORs, table-lookups, and bit-permutations as basic operations. It's hard to see why any AES submission would be better in this regard.

3 Flexibility

Twofish is unique in its implementation flexibility. The modular design of the algorithm allows it to be implemented in a variety of ways, depending on which aspect of the algorithm is most important for performance. Twofish can be optimized for bulk encryption, key agility, low gate count, throughput, low latency, or any combination of factors [SKW+98a, Sections 5.1, 5.4]. Twofish can be optimized to fit in the smallest smart cards and to run more efficiently on larger smart cards [SKW+98a, Section 5.2]. This flexibility, although extensively documented, does not seem to be appreciated by the researchers currently evaluating performance. In many of the papers comparing AES candidate performance, the ranking of Twofish could have been improved by properly optimizing the implementation for the particular problem at hand.

We designed this flexibility into Twofish from the beginning, because we feel that it is smarter to build a general algorithm to have good performance everywhere rather than excellent performance in a limited number of implementations and mediocre performance elsewhere. No other AES finalist allows this range of implementation options.

4 Performance

If there's anything we've learned from the various performance comparisons presented at the Third AES Candidate Conference, it is that there are a myriad ways to measure performance. We urge NIST to ignore rankings and minor variances in performance, and instead look at the larger differences. Despite the disparate ways of implementing the algorithms and measuring performance, several trends stand out.

In hardware, Rijndael and Serpent are fastest, Twofish is adequate, and RC6 and MARS are both slow and large. In software, Rijndael and Twofish are fastest, MARS and RC6 are adequate (they're fast on the few CPUs that support fast multiplies and data-dependent rotations, and slower on all others), and Serpent is very slow. RC6 and MARS have key schedules that make them very poor choices for high-performance hardware that has to handle a huge number of different keys (IPsec hardware is a good example) and cheap smart cards with limited RAM.

Twofish was designed to have good performance on a variety of hardware and software platforms, instead of being optimized for a single platform. Unlike some of the other AES finalists, Twofish runs at the same speed for encryption and decryption. In our design we took a variety of platforms and implementations into account, and the results show in all the different performance comparisons performed.

4.1 Measuring Twofish Performance

Several of the algorithms can take advantage of optimization "tricks" to improve performance on different platforms. Serpent makes use of a bit-slice implementation; Rijndael can take advantage of the MMX instruction set on the Pentium II to improve performance [AL00]. Twofish can be implemented by compiling the round keys into the code, speeding up performance on most CPUs [SKW+98a, Section 5.1]. (This is analogous to a Java JIT (just-in-time) compiler generating appropriate code on-the-fly for a Java program, and similar techniques are used in the Windows operating system.) It is our belief that in performance-intensive implementations, programmers will use all tricks possible to speed up AES, and our performance comparisons made use of all tricks for all algorithms [SW00]. Some comparisons have slower numbers for Twofish because they did not implement this trick. (At best, this is a minor issue. Even without the pre-compiling trick, Twofish is still only 20%–30% slower than the fastest algorithms, depending on what tricks you use for the other algorithms.)

Other comparisons implemented Twofish with its fastest block encryption speed and slowest key schedule speed, and then used those numbers to compare key setup. In applications where AES is used to encrypt bulk data, key-setup speed rapidly becomes irrelevant. In applications where a single key is only used to encrypt a few blocks, Twofish can be implemented with a faster key schedule. These performance tradeoffs are unique to Twofish, and did not always get reflected in the comparisons. Presumably all possible tricks will be implemented in production; applications where performance is important will do everything possible to improve performance.

4.2 Security/Performance

It is easy to design a secure algorithm by ignoring performance. It is also easy to make an algorithm faster by reducing its security. More interesting than individual performance measures is the ratio of safety factor to performance of the AES finalists. Looking at the five algorithms in this manner—normalizing to the largest number of rounds cryptanalyzed is a reasonable metric—Twofish far surpasses the other four finalists [SW00].

5 Simplicity

"Simplicity" is the NIST criterion that's hardest to describe. Lines of pseudocode, number of mathematical equations, density of lines in a block diagram: these are all potential measures of simplicity. Our worry about simplicity as a measure is that the simplest algorithms are often the easiest to break. The two simplest AES finalists—RC6 and Rijndael—are the ones whose current attacks break the most rounds: 9 of Rijndael and 17 of RC6. Simple algorithms feel like they're waiting for the right hammer: find it and you'll be able to break many rounds of the algorithm.

Simplicity is desirable so that cryptanalysts can understand the cipher well enough to evaluate its ultimate strength. However, it is important to note that simplicity is only one of many goals of a cipher designer, and that cipher design requires tradeoffs like other engineering task. We believe the Twofish structure is easily simple enough for a cryptanalyst to understand it and work with it. Indeed, we claim that this is true for all five of the AES finalists.

Twofish is as simple as it can be, but no simpler. It has a modular design, designed to facilitate understanding. We designed the cipher around a 32-bit invertible keyed function: the g function [SKW+98a, Section 7.1]. The MDS matrix multiplication [SKW+98a, Section 7.3] and the key-dependent S-boxes [SKW+98a, Section 7.2] guarantee certain nice properties in the round function. It is easy to understand the structure of Twofish from a single diagram

[SKW+98a, Page 11], one that contains *both* the round function and the key schedule.

We chose to put the math in the g function and the key schedule [SKW+98a, Section 7.11], and the muddle in the rest of the structure. That is, the g function is easy to understand mathematically, to analyze. The rest of the structure mixes adds, XORs, and rotates to make it hard to use that mathematical structure in an attack [SKW+98a, Section 7.7]. The eight-bit rotation for one of the g function inputs prevents identical values in the input to F from giving identical inputs to the two g functions. The one-bit rotations before and after the F outputs are XORed in to break up the byte-oriented structure of the cipher, so that differential or linear characteristics that line up nicely for the input to one S-box will not line up for the next [SKW+98a, Section 7.9]. These complexities are not arbitrary; they are all there for specific reasons, and those reasons are explained in our submission document [SKW+98a, Section 7].

By contrast, Rijndael's very simple byte-oriented structure is exploited in various attacks on reduced-round versions, including [FKS+00a, Luc00, GM00]. The simplicity of the cipher is what makes these attacks possible.

To be secure, encryption algorithms need both some math and some muddle. You need a mathematical structure so you can prove things about the cipher, such as the best differential and linear paths. But you also need to break up that mathematical structure so that an attacker can't make use of it.

6 Discussion: Choosing AES

The AES process has worked even better than expected. Today we have five good algorithms, and any of the designs would make an adequate AES standard. We recommend increasing the number of rounds for RC6 from 20 to 34, and the number of rounds for Rijndael from 10/12/14 to at least 18, to get an acceptable safety factor. We feel that without these increased numbers of rounds, RC6 and Rijndael would pose unacceptable security risks.

Two of the finalists, MARS and RC6, simply do not work well in certain applications. Any one of the other three algorithms—Rijndael (with the extra rounds), Serpent, or Twofish would make an *excellent* standard.

Choosing among these three algorithms is harder. Serpent is by far the slowest in software, about three times slower than the other algorithms. With 18 rounds, Rijndael is 80% slower than 10-round Rijndael (the variant used for most performance comparisons), and all performance figures need to be readjusted in that light.

Serpent has positioned itself as the conservative choice, while Rijndael (with its original 10 rounds) is clearly the fastest choice. We believe that the results of the straw poll at the Third AES Candidate Conference reflected this dichotomy: those that thought security was paramount chose Serpent, while those more concerned with performance chose Rijndael. Twofish represents a middle road, combining the best of both: very conservative security and very good performance across all platforms.

We believe Twofish is the best single choice for AES: "one standard; Twofish."

References

[ABK98] R. Anderson, E. Biham, and L. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard," NIST AES Proposal, Jun 1998.

[AL00] K. Aoki and H. Lipmaa, "Fast Implementations of AES Candidates," *Third AES Candidate Conference Proceedings*, Apr 2000.

[BBS99a] E. Biham, A. Biryukov, and A. Shamir, "Miss in the Middle Attacks on IDEA and Khufu," *Proceedings of the Sixth Fast Software Encryption Workshop*, Springer-Verlag, 1999, pp. 124–138.

[BBS99b] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials, *Advances in Cryptology—Eurocrypt '99 Proceedings*, Springer-Verlag, 1999, pp. 12–23.

[Bih99] E. Biham, "A Note Comparing the AES Candidates," revised version, comment submitted to NIST, 1999.

[BW99] A. Biryukov and D. Wagner, "Slide Attacks," *Proceedings of the Sixth Fast Software Encryption Workshop*, Springer-Verlag, 1999, pp. 245–259.

[BW00] A. Biryukov and D. Wagner, "Advanced Slide Attacks," *Advances in Cryptology—Eurocrypt '00 Proceedings*, Springer-Verlag, 2000, to appear.

[BCD+98] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S.M. Matyas, L. O'Connor, M. Peyravian, D. Safford, and N. Zunic, "MARS—A Candidate Cipher for AES," NIST AES Proposal, Jun 1998.

[CR00] Cryptography Research, Inc., "Differential Power Analysis," webpage at `http://www.cryptography.com/dpa/index.html`.

[DR98] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," NIST AES Proposal, Jun 1998.

[Fer98] N. Ferguson, "Upper Bounds on Differential Characteristics in Twofish," Twofish Technical Report #1, 17 Aug 1998. `http://www.counterpane.com/twofish-differential.html`

[Fer99] N. Ferguson, "Impossible Differentials in Twofish," Twofish Technical Report #5, 5 Oct 1999. `http://www.counterpane.com/twofish-impossible.html`

[Fer00] N. Ferguson. "Semi-Equivalent Keys in MARS," Presented at the Third AES Candidate Conference rump session, Apr 2000.

[FKS+00a] N. Ferguson, J. Kelsey, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved Cryptanalysis of Rijndael, *Proceedings of the Seventh Fast Software Encryption Workshop*, Springer-Verlag, 2000, to appear.

[FKS+00b] N. Ferguson, J. Kelsey, B. Schneier, and D. Wagner, "A Twofish Retreat: Related-Key Attacks Against Reduced-Round Twofish," Twofish Technical Report #6, 14 Feb 2000. `http://www.counterpane.com/twofish-related.html`

[GHJ+00] H. Gilbert, H. Handschuh, A. Joux, and Serge Vaudenay, "A Statistical Attack on RC6," *Proceedings of the Seventh Fast Software Encryption Workshop*, Springer-Verlag, 2000, to appear.

[GM00] H. Gilbert and M. Minier, "A Collision Attack on 7 Rounds of Rijndael," *Third AES Candidate Conference Proceedings*, Apr 2000.

[Kel00] J. Kelsey, "Key Separation in Twofish," Twofish Technical Report #7, 7 Apr 2000. `http://www.counterpane.com/twofish-tr7.html`

[KFS+00] J. Kelsey, N. Ferguson, B. Schneier, and M. Stay, "Cryptanalytic Progress: Lessons for AES," Presented at the Third AES Candidate Conference rump session, Apr 2000.

[KJJ99] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology — CRYPTO '99 Proceedings*, Springer-Verlag, 1999, pp. 388-397.

[KKS00a] J. Kelsey, T. Kohno, and B. Schneier, "Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent," *Proceedings of the Seventh Fast Software Encryption Workshop*, Springer-Verlag, 2000, to appear.

[KKS00b] T. Kohno, J. Kelsey, and B. Schneier, "Preliminary Cryptanalysis of Reduced-Round Serpent," *Third AES Candidate Conference Proceedings*, Apr 2000.

[KM00] L. Knudsen and W. Meier, "Correlations in RC6 with a Reduced Number of Rounds," *Proceedings of the Seventh Fast Software Encryption Workshop*, Springer-Verlag, 2000, to appear.

[Knu99] L. Knudsen, "Some Thoughts on the AES Process," comment submitted to NIST, Apr 1999.

[Knu00a] L. Knudsen, "Trawling Twofish," Reports in Informatics, University of Bergen, Apr 2000.

[Knu00b] L. Knudsen, "Trawling Twofish—Revisited," Presented at the Third AES Candidate Conference rump session, Apr 2000.

[Koc96] P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *Advances in Cryptology — CRYPTO '96 Proceedings*, Springer-Verlag, 1996, pp. 104–113.

[KSW99] J. Kelsey, B. Schneier, and D. Wagner, "Mod n Cryptanalysis, with Applications Against RC5P and M6," *Proceedings of the Sixth Fast Software Encryption Workshop*, Springer-Verlag, 1999, pp. 139–155.

[KSWH98] J. Kelsey, B. Schneier, D. Wagner, and C. Hall, "Side Channel Cryptanalysis of Product Ciphers," *ESORICS '98 Proceedings*, Springer-Verlag, 1998, pp 97–110.

[Luc00] S. Lucks, "Attacking Seven Rounds of Rijndael under 192-bit and 256-bit Keys," *Third AES Candidate Conference Proceedings*, Apr 2000.

[MM99] F. Mirza and S. Murphy, "An Observation on the Key Schedule of Twofish," *Second AES Candidate Conference Proceedings*, Mar 1999.

[Mur00] S. Murphy, "The Key Separation of Twofish," comments on AES round 2 submitted to NIST, Mar 2000.

[NIST97a] National Institute of Standards and Technology, "Announcing Development of a Federal Information Standard for Advanced Encryption Standard," *Federal Register*, v. 62, n. 1, 2 Jan 1997, pp. 93–94.

[NIST97b] National Institute of Standards and Technology, "Announcing Request for Candidate Algorithm Nominations for the Advanced Encryption Standard (AES)," *Federal Register*, v. 62, n. 117, 12 Sep 1997, pp. 48051–48058.

[NIST98] National Institute of Standards and Technology, *First AES Candidate Conference Proceedings*, Sep 1998.

[NIST99a] National Institute of Standards and Technology, *Second AES Candidate Conference Proceedings*, Mar 1999.

[NIST99b] National Institute of Standards and Technology, "Status Report on the First Round of the Development of the Advanced Encryption Standard," Aug 1999.

[NIST00] National Institute of Standards and Technology, *Third AES Candidate Conference Proceedings*, Apr 2000.

[RRS+98] R. Rivest, M. Robshaw, R. Sidney, and Y.L. Yin, "The RC6 Block Cipher," NIST AES Proposal, Jun 1998.

[SKW+98a] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-bit Block Cipher," *AES Round 1 Technical Evaluation CD-1: Documentation*, National Institute of Standards and Technology, Aug 1998.

[SKW+98b] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "On the Twofish Key Schedule," *Proceedings of the 1998 SAC Conference*, Springer-Verlag, 1998, pp. 27–42.

[SKW+99a] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Performance Comparison of the AES Submissions," *Second AES Candidate Conference Proceedings*, 1999.

[SKW+99b] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish on Smart Cards," *Proceedings of CARDIS 98*, Springer-Verlag, to appear.

[SKW+99c] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, *The Twofish Encryption Algorithm, A 128-Bit Block Cipher*, John Wiley & Sons, 1999.

[SKW+00] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, and N. Ferguson, "Comments on Twofish as an AES Candidate," *Third AES Candidate Conference Proceedings*, Apr 2000.

[SW00] B. Schneier and D. Whiting, "A Performance Comparison of the Five AES Finalists," *Third AES Candidate Conference Proceedings*, Apr 2000.

[Wag99] D. Wagner, "The Boomerang Attack," *Proceedings of the Sixth Fast Software Encryption Workshop*, Springer-Verlag, 1999, pp. 156–170.

[WKS+99] D. Whiting, J. Kelsey, B. Schneier, D. Wagner, N. Ferguson, and C. Hall, "Further Observations on the Key Schedule of Twofish," Twofish Technical Report #4, 16 Mar 1999. `http://www.counterpane.com/twofish-ks2.html`

[WS98] D. Whiting and B. Schneier, "Improved Twofish Implementations," Twofish Technical Report #3, 2 Dec 1998. `http://www.counterpane.com/twofish-speed.html`

[WW98] D. Whiting and D. Wagner, "Empirical Verification of Twofish Key Uniqueness Properties," Twofish Technical Report #2, 22 Sep 1998. `http://www.counterpane.com/twofish-keys.html`