Did Filiol Break AES ?

Nicolas T. Courtois[1], Robert T. Johnson[2], Pascal Junod[5], Thomas Pornin[3], and Michael Scott[4]

[1] SchlumbergerSema, Louveciennes, France, courtois@minrank.org
[2] University of California, Berkeley, USA, rtjohnso@cs.berkeley.edu
[3] Cryptolog International, France, thomas.pornin@cryptolog.com
[4] Shamus Software Ltd, Ireland mscott@indigo.ie
[5] LASEC, École Polytechnique Féd. de Lausanne, Switzerland, pascal.junod@epfl.ch

Abstract. On January 8th 2003, Eric Filiol published on the eprint a paper [11] in which he claims that AES can be broken by a very simple and very fast ciphertext-only attack. If such an attack existed, it would be the biggest discovery in code-breaking since some 10 or more years.
Unfortunately the result is very hard to believe. In this paper we present the results of computer simulations done by several **independent** people, with independently written code. Nobody has confirmed a single anomaly in AES, even for much weaker versions of the bias claimed by the author. We also present a plausible explanation for the possible source of error in the results of [11].

Key Words: block ciphers, AES, boolean functions, linear cryptanalysis, ciphertext-only attacks, stream ciphers.

1 Introduction

The Advanced Encryption Standard (AES) is the new Federal Information Processing Standard (FIPS) and is intended to use by U.S. Government organisations to protect sensitive (unclassified) information. It is also believed to become a (de facto) world standard for commercial applications that use cryptographic techniques. On October 2nd, 2000, NIST has selected Rijndael [9] as the Advanced Encryption Standard. It has been designed with the state of the art in cryptographic attacks on block ciphers, and at the time of submission it was believed very secure. The security of AES is however under constant scrutiny on behalf of the cryptographic community. Since there is no proof of security of AES, legitimate questions about its security are frequently raised.
A certain structural weakness in the design of the S-boxes of AES has been demonstrated in [4]. It cannot be denied, however so far nobody is able to say for sure, whether it leads to an attack on AES that would be faster than exhaustive search. Nobody were able to demonstrate that it will not work either. The results from [4] should be in fact considered as (rather approximative) lower bounds on the complexity of an algebraic attack on AES. Moreover, even if the attacks from [4] work very well, AES still remains by far much more secure than DES. To summarise, nobody really believes that the security of AES will be compromised in a very serious way during the next 5 or 10 years.
It is not the first time that Eric Filiol claims to have found a weakness in AES. In the paper [10] he claimed several results on AES, DES and other ciphers, however all the claims of this paper proved false after verification, see [5]. Later, the author updated his paper and the updated results showed that none of the initially claimed properties holded for AES, DES and hash functions. Still in the updated paper, the authors claims that some other biases in the Boolean functions that constitute DES and AES exist, for different tests, somewhat more complex to verify, see [10]. To the best of our knowledge, these have not yet been confirmed by an independent reviewer.
On January 8th 2003, Eric Filiol published a second paper, in which he aims more specifically AES [11] with much stronger claims, see also [12]. The main claim is that AES can be broken

in about 2^{31} operations by a very simple and very fast ciphertext-only attack. If such an attack existed, it would more or less mean nothing of what we know in cryptography is really very secure. Several people tried to verify this attack since, and nobody has succeeded so far.

2 Claims on AES

Let p_i, c_i and k_i be respectively the plaintext, the ciphertext, and the key bits of AES, The notation is explained in details in the Appendix A of [11] and is based on a standard byte-wise implementation of AES. The bits are always numbered from 0 to 127, 0 being the most significant bit of the first byte, and 127 being the least significant bit of the last byte. For example c_{71} is the least significant bit in the 9-th byte of the ciphertext.
On page 12 of the paper [11], the author claims that an equation of the following form is true with probability of about $p = 0.50003$:

$$1 + c_{71} = k_2 + k_3 + k_4 + \ldots + k_{123} + k_{126} \qquad (\#)$$

when the plaintext is randomly chosen in the subset of plaintexts of the form:

$$A = \left\{ P\&EFEFEFEFEFEF \ldots EFEF | P \in GF(2)^{128} \right\}$$

He also presents another (second) equation, involving the output bit c_{19}.

3 Methodology and Stats

On his web page [12] (seen on January 29th 2003) the author claims: "The only way to thoroughly verify this cryptanalysis is to perform the 100 attacks."
We disagree with this. There are probably many ways to convince oneself of recovering two bits of the key, for example by guessing them and generating the key in a biased way etc. For us, the only sensible way to confirm the attack is to confirm the existence of the bias. If there is no bias, there is no attack. However in Appendix B we also did examine the source code and tested the whole attack.

Testing the Bias

When doing N experiments and counting the number c of events that actually occur with a probability p, and $p = 1/2$ will be the plausible assumption for all our experiences, the expected value for c is $N \cdot p = N/2$ and the standard deviation for c is:

$$\sigma = \sqrt{N \cdot p \cdot (1-p)} = \sqrt{N}/2.$$

In order to see if the bias is significative, for each simulation we assume that the expected average is $c = N/2$. In order to mesure the observed deviation from 1/2, we will compute a signed deviation of c from the average, divided by to the standard deviation, which gives:

$$\text{deviation} = \frac{c - (N/2)}{\sigma} = \frac{c - (N - c)}{\sqrt{N}}.$$

Only results when $|deviation| \geq 5$ may be claimed to be significative. It is known that (for the normal distribution) the probability of not being within 5 standard deviations is about 1 in a 1.7 million.

Non-significative Results

According to the Appendix B of the paper [11], doing $N = 1.5 \cdot 10^9$ AES computations would be enough to detect a bias of order of $p - 1/2 = 0.00003$. It is however easy to see that it is not. For example if the equation (#) is totally unbiased, and true with probability exactly 1/2, we have $\sigma \approx 20000$ and we have approximatively $p = 0.50003 = 1/2 + 2.2\sigma/N$. This a perfectly normal result, within about 2 standard deviations, and does not prove anything.
In Appendix A we explain this in more details, and comment on the analysis done in the Appendix B of the paper [11].

4 Main Simulation Results

Done by	AES encryptions N	observed p	deviation
Courtois	3.7×10^9	0.5000039	0.474
Courtois	70×10^9	0.5000002	0.112
Courtois	386×10^9	0.5000003	0.369
Courtois	609×10^9	0.5000009	1.455

Fig. 1. The Simulations on Filiol's first equation with mask `0xEF`.

We did not confirm the alleged bias in AES.

4.1 Fool-proof Simulations - Fixed Key

In order to make sure that there is no mis-interpretation in the bit numbering somewhere, we also did many simulations for special cases of the Filiol equations, for example when the key is fixed.

Done by	the fixed key	AES encryptions N	observed p	deviation
Courtois	`00010203050607080A0B0C0D0F101112`	43×10^9	0.4999982	-0.724
Scott	`00010203050607080A0B0C0D0F101112`	50×10^9	0.500002	0.894
Pornin	`00010203050607080A0B0C0D0F101112`	152×10^9	0.4999928409	-1.101
Pornin	`15638C7F811F1F5D53123EC357A8E35A`	55×10^9	0.4999989481	-0.494
Pornin	`DAD83319EA7973B85FA8FFE5CDCAA45C`	141×10^9	0.4999984029	-1.199
Pornin	`65A67005017C53A90D77EB0EA10695B7`	59×10^9	0.5000002107	0.102

Fig. 2. The Simulations on Filiol's first equation when the key is fixed with mask `0xEF`.

Done by	the fixed key	AES encryptions N	observed p	deviation
Pornin	`00010203050607080A0B0C0D0F101112`	152×10^9	0.5000009909	0.772
Pornin	`15638C7F811F1F5D53123EC357A8E35A`	55×10^9	0.4999976404	-1.106
Pornin	`DAD83319EA7973B85FA8FFE5CDCAA45C`	141×10^9	0.4999995254	-0.356
Pornin	`65A67005017C53A90D77EB0EA10695B7`	59×10^9	0.4999966314	-1.638

Fig. 3. The Simulations on Filiol's second equation when the key is fixed with mask `0xEF`.

We also received some reports of another person, that wished to remain anonymous, who tried 4 billion encryptions with mask $0x7F$, and didn't find anything significative either.

4.2 More Fool-proof Simulations

In order to make sure that the author did not get the bits inside one byte in the wrong order, Michel Scott did some more simulations. He used Filiol's own random number generator, and the plaintext mask `0x66`. This mask covers a subset of all the 4 possible cases `0x7F, 0xEF, 0xF7, 0xFE` which might arise due to some misunderstanding concerning nibble/byte ordering. He also used the same fixed (test) key `000102...1112`. Moreover, in case the bit numbering inside a byte were different, he tested all the bits c_{16} through c_{23} and also (in case the bits are numbered the other way around), all the bits c_{104} through c_{111}. Here are the results:

Done by	output bit	AES encryptions N	observed p	deviation
Scott	c_{16}	12×10^9	0.499999	-0.2
Scott	c_{17}	12×10^9	0.499999	-0.2
Scott	c_{18}	12×10^9	0.499997	-0.7
Scott	c_{19}	12×10^9	0.500004	0.9
Scott	c_{20}	12×10^9	0.500005	1.1
Scott	c_{21}	12×10^9	0.500007	1.5
Scott	c_{22}	12×10^9	0.499992	-1.8
Scott	c_{23}	12×10^9	0.500002	0.4
Scott	c_{104}	12×10^9	0.499998	-0.4
Scott	c_{105}	12×10^9	0.499991	-2.0
Scott	c_{106}	12×10^9	0.500003	0.0
Scott	c_{107}	12×10^9	0.500000	0.0
Scott	c_{108}	12×10^9	0.500000	0.0
Scott	c_{109}	12×10^9	0.500006	1.3
Scott	c_{110}	12×10^9	0.499997	-0.7
Scott	c_{111}	12×10^9	0.500006	1.3

Fig. 4. The Simulations on different output bits, when the key is fixed to `0001020305060708OA0B0C0D0F101112` and with mask `0x66`.

Similar test, but for all output bits, and for much more different keys, have been done at Berkeley by Robert T. Johnson. Since Filiol proposed two equations, and also in order to circumvent a possible ambiguity in the bit-numbering for AES ciphertexts and the keys, it has been decided to measure the average absolute deviation of every single bit of the ciphertext. Again, the key has been fixed, to avoid problems in bit numbering here, and the experiment has been repeated for a few hundred keys, averaging the deviations across all the keys. For each of the 336 randomly chosen keys, exactly $N = 2^{32}$, i.e. 4 billions plaintexts has been tested for each key. A SIMD parallel computer consisting of a cluster of 336 CPUs has been used to perform the computation, that took about 3 days. For each of the 336 keys, it is expected that the average number of times when one output bit, say c_0, is equal to 1, is 2^{31}. The absolute value of the deviation is expected to be about $\sqrt{N}/2 \approx 2^{15}$. These values have been averaged over all keys. Here are the resulting absolute values of the deviations obtained, divided by 2^{15} in order to be compatible with the deviations given in other results of this paper. For all the output bits of AES in order, the results are:
[.7709, .7698, .7232, .8044, .8292, .8443, .8381, .7823, .7642, .8039, .7730, .7765, .7803, .8353, .7641, .7954, .8086, .7978, .7825, .8094, .8095, .7659, .7701, .8527, .8338, .8541, .7739, .7693, .8315, .8442, .7921, .7954, .7938, .8043, .7781, .7826, .8261, .8041, .8157, .7902, .7865, .7840, .7946, .8613, .8431, .7729, .7980, .8315, .7917, .7739, .8142, .7376, .8458, .7909, .8035, .8236, .7579, .8101, .8222, .7954, .8049, .7196, .8257, .8270, .7998, .7995, .8127, .8294, .8572, .8095, .7290, .8232, .8246, .7947, .7873, .7860, .8467, .7667, .7859, .7805, .8187, .7820, .7809, .8190, .7447, .8213, .8378, .8170, .7967, .7997, .8394, .8197, .8153, .7758, .8475, .7879, .8207, .7780, .8336, .7986, .8576, .8629, .7706, .7758, .8042, .7872, .7990, .8115, .8101, .7818, .7799, .8210, .8079, .8375, .7756, .8072, .7676, .7993, .7949, .8592, .7951, .8531, .7834, .7990, .7779, .8048, .7947, .8057]
We see that all these are perfectly normal. This simulation rules out the possibility of one of output bits of AES having a bias that would hold for many keys. However it does not exclude that a bias could exist for one, very special key. Such a bias however would be (in principle) useless in cryptanalysis.

5 Stream Ciphers vs. Block Ciphers

In the very same paper [11], the author claims that block ciphers are inherently insecure. We believe, as most of the people, that most of the block ciphers are very secure and will still be so for a very long time. For example DES resisted very well to 20 years of quite massive effort to break it, see for example [3], and the triple DES encryption is widely believed to resist at least 20 more years. However, in [11], the author advocates stream ciphers, as a replacement for the block ciphers, and claims that these (stream ciphers) are much more secure. On page 2 we read:

"Though we can strongly affirm that a very consequent theory for stream encryption exists, the block encryption theory does not provide more than a few cryptanalytic techniques and results on the constituent primitives at the round level."

Claiming that one type of ciphers are better than the other, certainly does not belong to a scientific paper: very few objective criteria allowing to compare them are known. However since the paper [11] has been largely circulated, and could be taken seriously by many readers, we need to say this is a very worrisome affirmation. It seems even that, at least for the ciphers that are popular today, the opposite could be true. This could be seen in the Nessie project [14]. Among the stream ciphers submitted, no cipher managed to convince the reviewers about its security, except one, that is not a stream cipher, but rather a mode of operation of a block cipher. All the other submissions have proven to be flawed. Moreover, recently several new and very general attacks on stream ciphers have been published, see [7, 6, 8, 1]. Finally, the paper [11] also proclaims that it is "easy" to be certain that there is no trapdoor in a stream cipher. This claim is again not justified, see for example [8, 2].

6 Conclusion

In order to verify the recent claims on the bias in the AES output bits, we have done many computer simulations, that have been programmed independently by several people. Our results show that AES has none of the two biases suggested by Filiol. Even for much weaker special cases, and worse probabilities, we did not find a single anomaly in AES. We also explain that the alleged bias in AES was due to insufficient number of tests done. It is moreover quite hard to believe, that such strong properties as suggested in [11, 12], would exist for a modern cipher that has several rounds. Further affirmations of the paper that most of the block ciphers are not secure and they should be replaced by stream ciphers, have no scientific foundation whatsoever.

In addition, in Appendix B, we examine the source code from [12] for the whole alleged attack on AES. We present a possible explanation for the apparent success rate obtained in [11]. We conclude that all the results of the paper [11] are erroneous.

It is remarkable how many misleading or plainly wrong remarks on AES are published on the Internet on the regular basis. This paper demonstrates quite clearly the importance of peer review by other members of the cryptographic community, and the caution that must be exercised when reading articles published on the Internet prior to independent examination.

Finding new results on AES is probably hard. It is hard to believe that a new amazing result on AES will appear every week. It is not even certain that anyone will ever find an essentially better attack on AES than those that are already known.

References

1. Frederik Armknecht: *A Linearization Attack on the Bluetooth Key Stream Generator*, Available on `http://eprint.iacr.org/2002/191/`.
2. Paul Camion, Miodrag J. Mihaljevic, Hideki Imai: *Two Alerts for Design of Certain Stream Ciphers: Trapped LFSR and Weak Resilient Function over GF(q)*. SAC 2002.
3. Don Coppersmith, *The Data Encryption Standard (DES) and its strength against attacks*. IBM Journal of Research and Development, Vol. 38, n. 3, pp. 243-250, May 1994.
4. Nicolas Courtois and Josef Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations;* Asiacrypt 2002, LNCS 2501, Springer. A preprint with a different version of the attack is available at `http://eprint.iacr.org/2002/044/`.
5. Nicolas Courtois: *About Filiol's Observations on DES, AES and Hash Functions (draft)*, Available at `http://eprint.iacr.org/2002/149/`.
6. Nicolas Courtois: *Higher Order Correlation Attacks, XL algorithm and Cryptanalysis of Toyocrypt;* ICISC 2002, LNCS 2587, Springer. An updated version is available at `http://eprint.iacr.org/2002/087/`.
7. Nicolas Courtois and Willi Meier: *Algebraic Attacks on Stream Ciphers with Linear Feedback*, Eurocrypt 2003, Warsaw, Poland, LNCS, Springer.
8. Nicolas Courtois: *Fast Algebraic Attacks on Stream Ciphers with Linear Feedback*, Preprint, January 2003, available from the author.
9. Joan Daemen, Vincent Rijmen: *AES proposal: Rijndael;* The latest revised version of the proposal is available on the Internet, `http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf`
10. Eric Filiol: *A New Statistical Testing for Symmetric Ciphers and Hash Functions*, The preliminary version published on eprint on 23th of july 2002, and revised on October 1st 2002. Will be published at ICICS 2002. `http://eprint.iacr.org/2002/099/`.
11. Eric Filiol: *Plaintext-dependent Repetition Codes Cryptanalysis of Block Ciphers - the AES Case*, Published on eprint on 8th of January 2003. `http://eprint.iacr.org/2003/003/`.
12. Eric Filiol, a web page about the PDRC attack on AES from [11], `http://www-rocq.inria.fr/codes/Eric.Filiol/PDRC.html`.
13. Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language, Second Edition*, Prentice Hall, Inc., 1988. ISBN 0-13-110362-8 (paperback), 0-13-110370-9 (hardback).
14. Nessie Security Report v1.0., available from `www.cryptonessie.org`.
15. J. A. Rice. *Mathematical statistics and data analysis.* Duxbury Press, 1995.

A Comments on the Appendix B of the Paper [11]

In this appendix we show in more details that there is a mistake in the statistical reasoning of the Appendix B of [11].

A.1 Simple Random Sampling

Most of the material in this section is a condensed version of Chapter 7.3 in [15].
In this section, we are interested in the following problem: we would like to estimate experimentally the probability that a boolean equation

$$f_1(P) \oplus f_2(C) = f_3(K) \tag{1}$$

holds where f_1, f_2 and f_3 are arbitrary functions defined as follows:

$$\begin{aligned} f_1 &: \{0,1\}^\ell \to \{0,1\} \\ f_2 &: \{0,1\}^\ell \to \{0,1\} \\ f_3 &: \{0,1\}^k \to \{0,1\} \end{aligned}$$

with ℓ is equal to the block length and k to the key length. The way to obtain such a probabilistic relation is irrelevant for the following discussions.
We can model the fact whether (1) holds or not with help of a Bernouilli random variable X defined as follows:

$$\begin{cases} X = 1 \text{ if } (1) \text{ holds} \\ X = 0 \text{ if } (1) \text{ does not hold} \end{cases} \tag{2}$$

We will assume without loss of generality that

$$\Pr[X = 1] = 1 - \Pr[X = 0] = \frac{1}{2} + \epsilon \tag{3}$$

where $\epsilon > 0$ for a fixed key and plaintexts drawn at random. The goal is to derive experimentally an accurate estimation $\hat{\epsilon}$ of ϵ. Note that

$$\mu \triangleq \mathrm{E}[X] = \frac{1}{2} + \epsilon \tag{4}$$

and

$$\sigma^2 \triangleq \mathrm{Var}[X] = \left(\frac{1}{2} + \epsilon\right) \cdot \left(\frac{1}{2} - \epsilon\right) = \frac{1}{4} - \epsilon^2 \tag{5}$$

Thus, we would like to estimate accurately $\mathrm{E}[X]$; this completely determines the underlying probability distribution.
The most elementary form of sampling is called *simple random sampling*. For a population of size N, there is $\binom{N}{n}$ possible samples of size n; simple random sampling is the situation where each sample of size n is taken without replacement, and that all of these samples are uniformly distributed. We will assume that a simple random sampling procedure has been used in [11], which is the most probable case.
In the following, we will denote the sample values by

$$X_1, \ldots, X_n \tag{6}$$

It is important to realize that each X_i is a *random variable*. The so-called *sample mean*

$$\overline{X} \triangleq \frac{1}{n} \sum_{i=1}^{n} X_i \tag{7}$$

is an estimate of the population mean. Since each X_i is a random variable, so is the sample mean; its probability distribution is called its *sampling distribution*. This distribution will determine how accurately $\overline{X}$ will estimate $\mathrm{E}[X]$. As a measure of the center of the sampling distribution, we will use $\mathrm{E}[\overline{X}]$ and as measure of dispersion the standard deviation $\sqrt{\mathrm{Var}[\overline{X}]}$. The well-known key results that one can obtain are that the sampling distribution is centered at μ and that its spread is inversely proportional to the square root of the sample size n:

Theorem 1. *With simple random sampling,* $\mathrm{E}[\overline{X}] = \mu$.

$\overline{X}$ is said to be an *unbiased estimator* of μ.

Theorem 2. *With simple random sampling,*

$$\mathrm{Var}[\overline{X}] = \frac{\sigma^2}{n}\left(\frac{N-n}{N-1}\right) = \frac{\sigma^2}{n}\left(1 - \frac{n-1}{N-1}\right) \tag{8}$$

The value

$$1 - \frac{n-1}{N-1} \tag{9}$$

is called the *finite population correction*. Frequently, it is very small in which case the standard deviation of $\overline{X}$ is

$$\sigma_{\overline{X}} \approx \frac{\sigma}{\sqrt{n}} \tag{10}$$

Some forms of central limit theorems adapted to the simple random sampling have been proved; indeed, in sampling finite population without replacement, the X_i are not independent, and it makes no sense to have n tend to infinity while N remains fixed. However, one can show that if n is large, but still small relatively to N, then $\overline{X}$ is approximately normally distributed.

Using this fact, one can derive a *confidence interval* for the population mean μ. It is a random interval calculated from the sample that contains μ with some specified probability:

$$\Pr\left[\overline{X} - z(\alpha/2)\sigma_{\overline{X}} \leq \mu \leq \overline{X} + z(\alpha/2)\sigma_{\overline{X}}\right] \approx 1 - \alpha \tag{11}$$

where $z(\alpha)$ is defined such that

$$1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z(\alpha)} e^{\frac{-t^2}{2}}\, dt = \alpha \tag{12}$$

In order to be able to compute this confidence interval, one should have an estimation of $\sigma_{\overline{X}}$. For this, the following theorem is generally used.

Theorem 3. *An unbiased estimate of* $\mathrm{Var}(\overline{X})$ *is*

$$s^2_{\overline{X}} \triangleq \frac{s^2}{n}\left(1 - \frac{n}{N}\right) \tag{13}$$

where

$$s^2 \triangleq \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X})^2 \tag{14}$$

To summarize, here is a procedure which allows to compute a confidence interval for the bias of (1):

1. Let n be the chosen number of samples and α the accepted error probability.
2. Evaluate n times (1) for a fixed key and random plaintexts and define $X_i = 1$ if the relation holds for sample i and $X_i = 0$ otherwise.
3. Compute $\overline{X}$ and $s^2_{\overline{X}}$ out of the X_i.
4. The confidence interval for μ is then given by (11).

A.2 Link to Filiol Results

In [11], Filiol claims to have discovered a relation on AES which holds with probability 0.500029 and another one which holds with probability 0.500028; these values have been derived experimentally. Unfortunately, Filiol does not give any confidence interval for them, and he does not give the experimental variance $s^2_{\overline{X}}$ of his experiments.
However, in the Annex B of [11], he derives the necessary number n of samples in order to get an error probability equal to $\alpha = 0.0001$: he estimates that $n = 1'520'000'000$ samples are sufficient. Let us assume that both equation are *not* biased, and that $\mathrm{E}[X] = \frac{1}{2}$ (thus $\mathrm{Var}[X] = \frac{1}{4}$), and let us compute a confidence interval for $\overline{X}$ with the prescribed n and α: we have

$$z(\alpha/2) \approx 3.89 \tag{15}$$

If we neglect the finite population correction (the plaintext subspace used by Filiol has still an enormous cardinality), we get

$$\mathrm{Var}[\overline{X}] = \frac{\mathrm{Var}[X]}{n} = \frac{1}{4n} \approx 1.645 \cdot 10^{-10} \tag{16}$$

Thus, for the prescribed error probability, we get the following confidence interval:

$$0.500029 - 3.89 \cdot \sqrt{1.645 \cdot 10^{-10}} \leq \mu \leq 0.500029 + 3.89 \cdot \sqrt{1.645 \cdot 10^{-10}} \tag{17}$$

which is equivalent to

$$0.499979 \leq \mu \leq 0.500079 \tag{18}$$

We note that Filiol results *don't contradict the hypothesis* $\epsilon = 0$.
Now, let us assume that $\mathrm{E}[X] = 0.500029$, *i.e.* that the given experimental value is correct. We would like to estimate the number of needed samples in order to get an accurate result up to 6 digits. We can compute

$$\sqrt{\mathrm{Var}[X]} \approx 0.4999999992 \tag{19}$$

So, in order to be able to give a confidence interval valid with a probability equal to 0.9999, one should sample the equation

$$n = \mathrm{Var}[X] \cdot \left(\frac{3.89}{0.0000005}\right)^2 \approx 2^{43.8} \tag{20}$$

times.

A.3 Conclusion

In this appendix we showed that there is obviously a mistake somewhere in the statistical reasoning of the Appendix B of [11].

B The Full Filiol's Attack on AES - Source Code Analysis

Recently, the author published also the source code for this attack, see [12]. We analysed this code.

B.1 The Biased Guessing Procedure - Or "the $N2$ Bug"

If we look carefully at the code, we can see that the algorithm A.1 is not implemented properly, as shown in the following excerpt (out of `cry_aes.c`):

```
#define N 49999
...
N2 = (N+1)/2;     /* N2 = 25000 */
N2 *= N;          /* N2 = 1249975000 */
...
for(i0 = 0L;i0 < N;i0++){
  for(i1 = 0L;i1 < N;i1++)
   {
...                       /* Loop executed 2499900001 times */
   }
}

/* ML Decoding */
if(rec[i1] > N2) SOL[i1] = 1;
    else SOL[i1] = 0;
```

At this point, one can see a flaw: the ML decoding rule should be

```
  if(rec[i1] >= 1249950001) SOL[i1] = 1;
    else SOL[i1] = 0;
```

Instead of this, it is implemented as

```
  if(rec[i1] >= 1249975000) SOL[i1] = 1;
    else SOL[i1] = 0;
```

The problem is the definition of $N2$. It is $N(N+1)/2$ instead of $(N*N)/2$. We see that this rule will guess the key parity in a biased manner (and this decision rule is therefore not a maximum-likelihood one).

B.2 The Random Number Generator

This code uses the `rand()` function as basis and unique initial source of randomness (that function is used to create the seeds which are input into the cryptographic PRNG used), but without calling `srand()` before. The C standard mandates that on a given machine, the stream of values returned by `rand()` is deterministic from the seed provided by `srand()` beforehand, and that if `srand()` is not called, the `rand()` function behaves as if `srand(1)` was called at the beginning of the program. That behaviour is already mentioned in [13] and as such is likely to be implemented in all modern and not-so-modern C compilers and standard libraries.

The bottom line is that, if Filiol's program is ran twice on the same machine, it will give twice the exact same results. It is quite easy, under these conditions, to get "reproducible" results. This is easily checked by adding a couple of `printf()` calls in the code to print out the two seeds used (for instance just before the comment "Beginning of the cryptanalysis itself").

B.3 Other Potential Problems

It is also possible to see that the code is not 64-bit compatible: it will behave wrongly on, for instance, an Alpha or Ultrasparc machine, where "unsigned long" is not exactly 32-bit long.

B.4 Running the Complete Program

Michel Scott run the complete Filiol's program, unchanged except for the deletion of the reference Rijndael implementation, replaced by a much faster implementation (which is known to be functionally equivalent and runs about 20 times faster). Then the program was run on Windows 2000, with a Pentium 4 clocked at 2.33GHz. It took about 26 hours to complete. When the key parity bit has been guessed correctly, `0` is displayed, otherwise `1` is displayed.
Here is the output obtained for the first equation (with c_{71}):

```
[ 1 1 0 0 1 1 1 1 0 0 1 1 1 1 0 0 1 0 1 0 1 0 1 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1
1 0 1 1 0 1 0 1 0 0 1 0 1 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 1 0 1 0 0 1 1 1 1 1 1 0
0 0 1 0 0 0 0 0 1 0 0 1 1 0 0 1 1 0 1 1 0 ]
```

We see that there are 51 `1`'s and 49 `0`'s.
And here is the output obtained for the second equation (with c_{19}):

```
[ 0 1 1 0 1 0 1 0 1 0 0 1 1 1 1 0 0 1 1 1 1 0 1 1 1 0 1 0 0 1 1 1 0 1 1 0 1 1 1
0 1 1 1 0 0 1 1 0 0 0 0 1 1 0 1 0 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 0 0 0 0 1 1
1 1 1 0 0 1 1 1 0 0 1 1 1 0 0 1 1 0 1 0 1 ]
```

We see that there are 59 `1`'s and 41 `0`'s.
In both cases the guess is more frequently wrong than right. We see that even with "the $N2$ Bug", there is no significative result. The result of Filiol was not reproduced, even using the same source code. This could be due to using a different compiler, with a different initial value used by `rand()`.

B.5 Conclusion

The (apparent) result of Filiol could be a combination of "the $N2$ Bug", with "bad luck" in the key generated by the PRNG. Note that after seeding from `rand()`, it is not used much, and the generator may show an initial bias when started with an "unlucky" seed value.
This remark has been confirmed by Pascal Junod, with the "almost deterministic seeding" implemented, and for the equation 2 of [11], the PRNG produces 100 keys having a parity distribution equal to 41/59 (in favour of zeroes...) for the key parity function induced by the equation 2 of [11].
As the bugged decision rule has a success probability equal to about 84% when the key has a parity equal to 0, (and 16% otherwise), one can explain the global success probability obtained in [11, 12] relatively well.