

New Observations on Rijndael

Sean Murphy and Matt Robshaw

Information Security Group, Royal Holloway,
University of London, Egham, Surrey, TW20 0EX, U.K.
S.P.Murphy@rhbnc.ac.uk mrobshaw@supanet.com

Preliminary Draft

7 August, 2000

Abstract

We summarise some observations on the AES finalist Rijndael. The cipher splits very cleanly into a layer of S-box transformations, a linear diffusion layer designed to provide mixing across the block, and a subkey layer. We show that the linear diffusion layer has some unusual properties.

1 Introduction

Rijndael [1] is one of the five finalists being considered for the Advanced Encryption Standard. It is an unusual cipher in that the input plaintext is arranged into a rectangular array of bytes and throughout the encryption process this byte-structure is fully respected. By this we mean that all operations take place in a byte-wise fashion.

Some commentators have already noted that Rijndael uses mathematically simple functions [2, 6]. In this note we identify some remarkable structure within Rijndael and we discuss some of the unusual block cipher properties that result. It is important to note that our observations are concerned with the inherent structure of Rijndael and are therefore applicable irrespective of the number of rounds.

The most significant observation—that **any** input text or **any** input difference to the linear diffusion layer is always mapped to itself after at most 16 applications of the linear diffusion layer—appears to cast some doubt on a diffusion layer that aims to “guarantee high diffusion over multiple rounds” [1].

2 Basic round structure

We refer to the documentation of Rijndael [1] for a full description of the cipher, but we list the significant steps here. We describe the typical round. The first and last rounds of Rijndael have a different (but related) form. Throughout we consider the 128-bit block size variant of Rijndael and our observations apply to any length of key.

2.1 Notation

The 128-bit input block to Rijndael is arranged as a 4×4 array of bytes $A[i][j]$. We consider the bytes in the array $A[i][j]$ to be a sequence of bytes starting from position $(0, 0)$ and working through in the order $(1, 0)$, $(2, 0)$, $(3, 0)$, $(0, 1)$, \dots , $(3, 3)$; that is by columns. Further we adopt the convention that the most significant bit in array position (i, j) is represented by the left-most, and most significant, bit of the hexadecimal representation of a byte.

2.2 Description

1. The value of each element in the array is substituted according to a table look-up. This table look up $S[\cdot]$ consists of three transformations. (Of course in an implementation they are combined as a single table look-up.)
 - (a) Input x is mapped to $y = x^{-1}$ over $GF(2^8)$ (with 0 mapped to 0).
 - (b) Intermediate value y is mapped to $z = L \cdot y$, where L is a linear transformation of an 8-dimensional binary space.
 - (c) The output $S[x]$ is $z + c$ over $GF(2^8)$ for a constant c .
2. There is a linear diffusion (mixing) layer.
 - (a) Each row of the array $A[i][j]$ is rotated by a certain number of byte positions. Byte $A[i][j]$ becomes $A[i][j-i \bmod 4]$.
 - (b) Each column of the array $A[i][j]$ is considered as a 4-dimensional $GF(2^8)$ -vector. A (4×4) $GF(2^8)$ -matrix D is used to map this column. Thus a column of bytes x is replaced by the column of bytes y where $y = D \cdot x$. We note that D has constant row and column sum 1.
3. Each byte of the array $A[i][j]$ is exclusive-ored with a byte from a corresponding array of round subkeys.

2.3 Comments

It is worth highlighting some relevant features of this design.

In the S-box operation labeled 1, steps (b) and (c) represent the action of an affine map. This map is used to disguise the algebraic simplicity of the operation $x \rightarrow x^{-1}$. This simplicity is commented on by the designers of Rijndael [1], page 26, and has also been discussed by Schroeppel [6]. The invertible affine transformation is used to provide a “complicated algebraic expression if combined with the inverse mapping” and thus aims to provide protection against interpolation attacks. The constant in the affine mapping “has been chosen in such a way that the S-box has no fixed points and no opposite fixed points” (since both 0 and 1 are fixed points for the inverse mapping).

The non-linear S-box operations are followed by a set of linear operations. This separation is a deliberate design decision with the linear layer being used to “guarantee high diffusion over multiple rounds” [1], page 8. In step 2b, which is part of the linear diffusion layer, the matrix D has a special form that aims to provide good diffusion. The properties of this matrix form an essential step in the estimates for the resistance to linear and differential cryptanalysis of Rijndael. The elements of the matrix were chosen to provide implementation advantages (see [1], page 27).

Rijndael is very well engineered against conventional linear and differential cryptanalysis. The observations in this note are unlikely to change this. Our concern, however, is the possibility of more opportunistic attacks dedicated to the structure of Rijndael itself.

3 Re-grouping the operations

We can group the round operations in Rijndael in a slightly different way.

3.1 The S-box

Within the S-box we note the following. The same constant c is exclusive-ored to every byte following the action of the linear map L . The two subsequent operations are the row shift and the mapping of columns by the $GF(2^8)$ -matrix D . The row shift merely permutes bytes and the action of the $GF(2^8)$ -matrix D can be accounted for. Thus the constant c could just as easily be included after the application of D . Hence the exclusive-or with bytes of the constant c could be incorporated into the round subkeys as a part of a (slightly) modified key schedule. In this note, we use this equivalent description of Rijndael.

By moving the constant c , the remaining linear component of the affine mapping L can be considered as part of the linear diffusion layer. By grouping the operations like this we feel that we have a more natural description of Rijndael.

3.2 The linear diffusion layer

We have the following equivalent description of Rijndael.

S-box layer. The value of each element in the array is substituted according to a table look-up.

1. Input x is mapped to $y = x^{-1}$ over $GF(2^8)$.

Linear diffusion layer. The following operations take place on the array $A[i][j]$.

1. The value of each element in the array $A[i][j]$ is substituted by the value $L \cdot A[i][j]$ under the action of a linear mapping L over $GF(2)^8$.
2. Each row of the array $A[i][j]$ is rotated by a certain number of byte positions which changes between rows.
3. Each element in a column of the array $A[i][j]$ is combined by means of a (4×4) $GF(2^8)$ -matrix D where each column of bytes \mathbf{x} is replaced by the column of bytes \mathbf{y} where $\mathbf{y} = D \cdot \mathbf{x}$.

Subkey layer. Each byte of the array $A[i][j]$ is exclusive-ored with a byte from a (slightly modified) corresponding array of round keys.

Since the entirety of the modified linear diffusion layer, which consists of two-thirds of the original S-box transformation and the original row and column mixing operations, is a $GF(2)$ -linear map, its action can be represented by a 128×128 binary matrix, M .

Both the *characteristic* polynomial $c(x)$ ($\text{Det}(M + xI)$) and *minimal* polynomial $m(x)$ (the polynomial of smallest degree such that $m(M) = 0$) of M are remarkably simple. It turns out that

$$\begin{aligned}c(x) &= (x + 1)^{128} = x^{128} + 1, \text{ and} \\m(x) &= (x + 1)^{15}.\end{aligned}$$

Two aspects of the form of $m(x)$ are immediately noteworthy:

1. $m(x)$ has an exceptionally simple form.
2. $m(x)$ has an exceptionally small degree.

Furthermore, since $m(M) = 0$ and $(x^{16} + 1) = (x + 1) \times m(x)$, we have that

$$M^{16} = I.$$

That is the minimum number of iterations (the exponent) of the linear diffusion layer that give the identity transformation is 16. This has the following immediate consequence:

Any 128-bit input (or difference) to the linear diffusion layer of Rijndael is mapped to itself after **at most** 16 repeated applications of the linear diffusion transformation.

This is a rather surprising property.

3.3 Comparison with DES

It is hard to come up with a parallel for such a minimum polynomial in another block cipher design. Perhaps the closest, and most illustrative, comparison we can make is to DES [5].

In several ways the structure of Rijndael is similar to that of DES. Both ciphers have round functions with three layers (though in a different order): an S-box layer (including the expansion function E for DES), a linear diffusion layer and a subkey exclusive-or layer. To make our comparison to DES, we will assume that the only diffusive properties in DES derive from the bitwise permutation P at the end of each round, and the Feistel structure itself (though this is, of course, an over-simplification). The characteristic and minimum polynomials for such a linear diffusion layer for DES are

$$\begin{aligned}c(x) &= (1 + x + x^2)^8(1 + x + x^2 + x^3 + x^4)^4 \\ &\quad (1 + x + x^2 + x^4 + x^6 + x^7 + x^8)^4, \text{ and} \\ m(x) &= (1 + x + x^2)^4(1 + x + x^2 + x^3 + x^4)^4 \\ &\quad (1 + x + x^2 + x^4 + x^6 + x^7 + x^8)^4.\end{aligned}$$

This overly-simplified linear diffusion layer for DES appears to be more complicated than the linear diffusion layer of Rijndael. The minimum polynomial has degree 56 (for a 64-bit cipher). For Rijndael the minimum polynomial has degree 15 (for a 128-bit cipher). We also observe that the exponent (the minimum number of iterations of the linear diffusion layer to give the identity) of the linear diffusion layer for DES is 1020 compared with 16 for Rijndael.

3.4 Simplified form of the linear diffusion layer

Given the matrix M we can find alternative representations by using a “change of basis” matrix. In particular, by analysing the minimum polynomial for M , we can identify the block diagonal form of M which we denote by R . It is a strikingly simple matrix. For some matrix P we have $R = P^{-1} \cdot M \cdot P$. Here we present the block diagonal matrix R . (The matrices M , P , and P^{-1} are provided in the Appendices.) To emphasize the remarkable simplicity of R we use $.$ to represent 0. The line breaks in the presentation of this matrix represent a division into invariant subspaces which we discuss in Section 3.5.

Any subspace, including V_1, \dots, V_{15} , can be defined as the kernel of some linear transformation. For a subspace defined by a collection of elements of the new basis (columns of P or rows in Appendix B), this linear transformation is given by removing the corresponding rows of P^{-1} (rows in Appendix B). For a subspace V_i , this means removing the appropriate collection of rows between line breaks. Such a linear transformation with kernel a subspace of dimension n can be regarded as a set of $(128 - n)$ parity checks (rows in Appendix B).

3.5.1 Quotient spaces and cosets

The subspace $U_i = V_1 + \dots + V_i$ is the sum of the first i subspaces and is invariant under the linear diffusion layer. The quotient space $W_i = V/U_i$ ($i = 1, \dots, 15$) is the vector space of cosets of U_i in V . The coset of U_i to which a vector belongs is determined by the values of the parity check equations for U_i . The effect of the linear diffusion layer on the cosets of U_i is given by an appropriate lower right submatrix of R , and maps any given coset to another coset of U_i . One promising approach for a cryptanalyst, therefore, might be to consider how such quotient spaces are mapped under the full cipher [3, 4].

4 Some consequences

The block matrix R that describes the linear diffusion layer of Rijndael is very simple and allows considerable algebraic structure to be unearthed. Some immediate consequences of the structured linear diffusion layer of Rijndael are given below. These consequences are applicable in both a linear fashion (when the input is a single text) and also in a differential fashion (when the input is a pair of texts and we are interested in the behavior of the difference across the linear diffusion layer). All these effects could be viewed as symptoms of questionable diffusion in Rijndael.

It could well be that many of these properties offer little immediate advantage to the cryptanalyst. However, they are indicative of a very rich structure in Rijndael, a structure for which more subtle properties may become apparent over time.

- All inputs (differences) to the linear diffusion layer are fixed over 16 iterations of the linear diffusion transformation.
- There are inputs (differences) to the linear diffusion layer that are fixed over a very small number of iterations of the linear diffusion transformation.
 - 2^{16} inputs (differences) are fixed over one diffusion layer. Basis vectors for the fixed 16-dimensional subspace are given here.

5c090b8f	df49d0c2	4d5c4d8f	5adf0001
a711966b	ce87aaa8	0dbb3cc1	642d0002
9df03855	0f0baaae	375a92ff	a5a10004
3ff0a9d2	a174a1c0	3492a2b0	00bc0008
8cb32178	c4e73c79	b0da1d11	f88e0010
88eeb685	fddd96e3	1e2d2046	6b1e0020
e73c7ebb	0f049683	71ffe878	99c70040
ef5a233c	c7d19d21	d8511437	5a700080
aeb5ac48	61f6a0c8	a5d7a72a	c03e0100
79fae7b6	6884350b	e45b7a17	5f8f0200
953f90bd	9c760f62	34f73175	97140400
be44b463	3e4e3469	822d880a	02270800
adfaa271	54d35600	286c72b2	47861000
b6c3d0a5	46662000	b6c3d0a5	46662000
6953a7c8	e7a77c69	553a9ba1	dbce4000
248283da	eff98b62	854a2212	e49b8000

- Some inputs (differences) to the linear diffusion layer might never involve more than 12 active S-boxes. For one example (among several) the input (difference)

55336600	33550066	55336600	33550066
----------	----------	----------	----------

is fixed by the linear diffusion layer and only uses 12 S-Boxes no matter how many rounds of Rijndael are used.

- 2^{30} inputs (differences) are fixed over two iterations of the linear diffusion transformation.
 - 2^{58} inputs (differences) are fixed over four iterations of the linear diffusion transformation.
 - 2^{96} inputs (differences) are fixed over eight iterations of the linear diffusion transformation.
 - All 2^{128} inputs (differences) are fixed over sixteen iterations of the linear diffusion transformation.
- There are 2^{16} parity equations whose value is fixed across the linear diffusion layer. These can be derived from the 16 (linearly independent) parity checks below. Such parity equations would be applicable to both inputs and differences.

167b3466	22befef7	06748ec3	97bee15d
07efe15c	a8d7be94	2d60cbd3	d70dc14e
13f8c41c	f88a2f6e	13f8c41c	f88a2f6e
16c0652a	9dc24482	e69a9570	6d98b4d8
00d98d67	c1244c9a	558cd832	947119cf
12efba6a	8569dd1c	c83a60bf	0ae9529c
079f6ef6	b489dde0	52ca3ba3	e1dc88b5
04ebc728	1d57de94	51be927d	48028bc1
12311231	b73eb73e	12311231	b73eb73e
04280428	f472f472	04280428	f472f472
0616acbc	f64c5ce6	0616acbc	f64c5ce6
129bb831	e2c1486b	129bb831	e2c1486b
33333333	33333333	33333333	33333333
0fa50fa5	0fa50fa5	0fa50fa5	0fa50fa5
aaaaaaaa	aaaaaaaa	aaaaaaaa	aaaaaaaa
5af05af0	5af05af0	5af05af0	5af05af0

- There are 14 parity equations that are invariant under the linear diffusion layer and which never involve four of the sixteen S-boxes.

00999900	cc5555cc	00999900	cc5555cc
00aaaa00	55ffff55	00aaaa00	55ffff55
00333300	99aaaa99	00333300	99aaaa99
33000033	aa9999aa	33000033	aa9999aa
aa0000aa	ff5555ff	aa0000aa	ff5555ff
99000099	55cccc55	99000099	55cccc55
aa9999aa	33000033	aa9999aa	33000033
55cccc55	99000099	55cccc55	99000099
ff5555ff	aa0000aa	ff5555ff	aa0000aa
99aaaa99	00333300	99aaaa99	00333300
55ffff55	00aaaa00	55ffff55	00aaaa00
cc5555cc	00999900	cc5555cc	00999900
1f001f00	ef5aef5a	1f001f00	ef5aef5a
ef5aef5a	1f001f00	ef5aef5a	1f001f00

- The linear diffusion layer of Rijndael allows input (difference) vectors to be split into cosets.

As a simple example we might consider the following two parity equations.

aaaaaaaa	aaaaaaaa	aaaaaaaa	aaaaaaaa
5af05af0	5af05af0	5af05af0	5af05af0

Evaluating both parity checks will give a two-bit quantity. This specifies in which of four cosets an input (difference) lies. These parity equations

are fixed by the linear diffusion layer and we have thus identified four subsets of size 2^{126} that are fixed by the linear diffusion layer. An input (difference) in one coset can never be mapped by the diffusion layer of Rijndael into one of the other three cosets.

There are many such examples of large sets being mapped to themselves by the linear diffusion transformation.

5 Conclusions

We have noted the striking property that any input (or input difference) to the linear diffusion layer of Rijndael will be mapped to itself after at most 16 iterations of the linear diffusion transformation. This is quite surprising since we might expect the repeated action of row rotation and the matrix multiplication to be very effective at mixing. It implies the presence of considerable inner structure within the diffusion layer, some of which will appear within the single iteration used to map between S-boxes. We have also shown that the affine map, used in some sense to disguise the inverse operation in the S-box, can be moved into a (very slightly) modified linear diffusion layer.

The consequences described in this note are ones that immediately come to mind and demonstrate the structure in the linear diffusion layer. Even if these particular properties offer little advantage to conventional differential and linear cryptanalysis, it remains an open question whether the cryptanalyst can find a more novel way to combine the rich structure in the diffusion layer of Rijndael with the highly structured inverse map.

References

- [1] J. Daemen and V. Rijmen. AES Proposal: Rijndael. Version 2. 1999.
- [2] L. Knudsen and H. Raddum. Recommendation to NIST for the AES. Second round comments to NIST. Available via csrc.nist.gov/aes/.
- [3] S. Murphy. "An Analysis of SAFER", *Journal of Cryptology*, vol. 11, pp 235-251, 1998.
- [4] S. Murphy, F. Piper, M. Walker and P. Wild. Maximum Likelihood Estimation for Block Cipher Keys. Technical Report, Royal Holloway, University of London, 1994. Available at <http://www.cs.rhbnc.ac.uk/~sean>.
- [5] National Institute of Standards and Technology. Data Encryption Standard. FIPS 46-2. 1993.
- [6] R. Schroeppel. Second round comments to NIST. Available via csrc.nist.gov/aes/.

