Linear Redundancy in S-Boxes

Joanne Fuller and William Millan

Information Security Research Centre,
Queensland University of Technology,
GPO Box 2434, Brisbane, Queensland 4001, Australia
{fuller,millan}@isrc.qut.edu.au

Abstract. This paper reports the discovery of linear redundancy in the S-boxes of many ciphers recently proposed for standardisation (including Rijndael, the new AES). We introduce a new method to efficiently detect affine equivalence of Boolean functions, and hence we study the variety of equivalence classes existing in random and published S-boxes. This leads us to propose a new randomness criterion for these components. We present experimental data supporting the notion that linear redundancy is very rare in S-boxes with more than 6 inputs. Finally we discuss the impact this property may have on implementations, review the potential for new cryptanalytic attacks, and propose a new tweak for block ciphers that removes the redundancy. We also provide details of a highly nonlinear 8*8 non-redundant bijective S-box, which is suitable as a plug in replacement where required.

1 Introduction

The properties of substitution boxes (called S-boxes) form the basis for arguments regarding the security of symmetric encryption algorithms, and their importance is undoubted. Following Shannon's theory of secrecy systems proposing confusion and diffusion in SP-networks [16], and the popularity of the subsequent Data Encryption Standard [12], S-boxes have been the working heart of many efficient and secure encryption algorithms. A look up table can be implemented in a single software instruction, so S-boxes are attractive for fast software encryption. In fact, the vast majority of high quality proposals for symmetric encryption algorithms include the specification of one (or more) S-boxes, together with a list of security criteria these S-boxes were selected to meet. Clearly a lot of attention has been given to S-boxes, yet still many open problems remain, and some important properties, such as the one presented in this paper, have previously gone unnoticed.

Many papers have investigated the linear approximation and differential (autocorrelation) properties of S-boxes. It has been clearly demonstrated that powerful generic statistical attacks such as differential and linear cryptanalysis can be resisted by the selection of nearly optimal Boolean functions as components for the S-boxes. However, it is known that tradeoffs exist with respect to optimising Boolean functions for several security criteria simultaneously. Several methods

T. Johansson (Ed.): FSE 2003, LNCS 2887, pp. 74–86, 2003.

to generate cryptographically useful S-boxes exist, such as random generation, using finite field operations and heuristic algorithms. Of these, the finite field operation of inversion with respect to a polynomial basis achieves best known combination of high nonlinearity, low autocorrelation and high algebraic degree. For these reasons the finite field operations have become popular in symmetric cryptography.

Finite field operations have been used in many ciphers proposed since 1996, including Shark [14], Square [3], Rijndael [4], as well as several of the NESSIE [18] proposals Camelia, Hierocrypt, and SC2000. More recently the Japanesse Government's CRYPTREC [19] standardisation process has had several proposals that use finite fields (including many of the abovementioned ciphers plus Cipherunicorn A (and E)). In CRYPTREC, the only block cipher proposal *not* based on finite field S-boxes is RC6! The South Korean Government is also undertaking an encryption standardisation process[20] and their block cipher submissions that use finite fields include Seed and Zodiac. Also we note the stream ciphers BMGL (a NESSIE submission) and MUGI (a CRYPTREC submission) also use finite field based S-boxes. Typically, the designers of these algorithms have chosen to wrap the finite field inversion inside a bitwise affine transformation, claiming that this would prevent algebraic attacks over $GF(256)$. However, in this paper we report the discovery of a property of algebraic linear redundancy that is inherent in the finite field exponentiation operations, including inversion, and which is *not* removed by any surrounding affine transformation. Apart from finite field operations, S-boxes can possess linear redundancy that stems from other sources, in particular a small number of inputs (as in Serpent and Q) [11] and also low order functions (as in Misty, Kasumi and CAST).

In Section 2 of this paper we introduce the concepts relating to equivalence classes of Boolean functions and present an efficient algorithm to detect affine equivalence. Our initial discovery of redundancy in the AES S-box is presented in Section 3. In Section 4 we consider the special case of finite field power permutation based S-boxes, and give a simple and direct proof, due to Wagner, that *all* the linear combinations of the output bits are given by Boolean functions that are equivalent under affine transform! We first demonstrated this fact by a computer experiment on Rijndael's S-box (that revealed eight transforms between each pair of functions) but after an early draft of this paper [7] was circulated many people then offered algebraic proofs for this case. The first proof we received was from David Wagner [17] (and longer but somewhat similar proofs were found by Eric Garrido [8], Don Coppersmith [2] and very probably many others). It is amazing that a property with this simple a proof went so long unnoticed.

In Section 5 we present some experimental results on the equivalence class variety possessed by random S-boxes, and hence propose a new randomness criterion: that all output functions should have distinct equivalence classes. In Section 6 we offer some discussion about the consequences for encryption algorithms. We consider improved implementation tradeoffs that result from the redundancy, and discuss some possible avenues towards new cryptanalysis. Finally we suggest altering (or Tweaking [9]) S-boxes affected by these results, as a barrier against any future cryptanalysis that may result from this kind of non-randomness. We show that this tweak removes the class redundancy without greatly reducing the nonlinearity and differential security properties of the S-box.

In Appendix A we give examples of the matrix transforms that map between the output bits of the Rijndael S-box, and their inverses appear in Appendix B. Appendix C contains the look-up-table for the best non-redundant 8*8 S-box we have so-far generated by tweaking the AES S-box. It has nonlinearity 106 and algebraic order 7, which is the best combination known for a non-redundant S-box. We propose this s-box as a suitable plug-in replacement for ciphers such as AES.

2 Equivalence Classes of Boolean Functions

Boolean functions are represented by their truth tables. When there exists an affine transformation that maps between two Boolean functions, then those functions are said to be affine equivalent and are grouped together in the same equivalence class. Two n-input Boolean functions f and g are considered equivalent if there exists a non-singular binary matrix D, two n-element binary vectors a, b and a binary constant c such that

$$g(x) = f(\mathcal{D}x^T \oplus a^T) \oplus b \cdot x^T \oplus c,$$

where $b \cdot x^T = b_1x_1 \oplus b_2x_2 \oplus \cdots \oplus b_nx_n$ denotes a linear function of x selected by b. The study of Boolean functions can be greatly enhanced by considering equivalence classes. Many properties of cryptographic interest are unchanged by affine transform, such as algebraic degree and nonlinearity. More generally, the absolute values of the Walsh transform and the autocorrelation function are both re-arranged by affine transform.

It seems little has been written on equivalence classes since the 1972 Berlekamp and Welch paper [1] on $n = 5$ described all 48 classes in terms of their Algebraic Normal Form (ANF). It seems to be well known that the number of equivalence classes increases exponentially with n, for example see [5]. Concretely, the 1991 Maiorana [10] paper states that there exist 150,357 classes for $n = 6$, including 2082 different WHT distributions, but there is no analysis of structure for cryptology. More recently, equivalence classes have been used to provide restricted inputs to random and heuristic searches seeking better Boolean functions [13]. However, it has remained an open problem to easily distinguish between equivalent functions and indeed determine such mappings for functions of any n.

In seeking other methods to approach the class distinguishing problem, we investigated the local structure by considering the set of functions at Hamming distance one from a given Boolean function.

Definition 1. The *1-local neighbourhood* of a Boolean function f consists of all 2^n Boolean functions f_i, $i \in \mathbf{Z}_2^n$, constructed such that $\texttt{dist}(f,\, f_i)$=1. Furthermore the *connected* functions are given by

$$f_i(x) = \begin{cases} f(x) & x \neq i \\ f(x) \oplus 1 & x = i \end{cases}$$

We now prove that if f and g are equivalent, then there exists a function g_j at distance 1 from g, that is equivalent to a corresponding function f_i at distance 1 from f under the same affine transform that relates f and g. This result also provides a useful property for consideration when trying to determine whether two functions are equivalent.

Theorem 1. If f_i is a connecting function of $f(x)$, defined as above, then there exists a connecting function g_j of $g(x) = f(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c$ such that $g_j(x) = f_i(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c$ and $j = (\mathcal{D}^{-1}(i^T \oplus a^T))^T$.

Proof. Let $g(x) = f(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c$ and

$$f_i(x) = \begin{cases} f(x) & x \neq i \\ f(x) \oplus 1 & x = i \end{cases}$$

Therefore,

$$f_i(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c = \begin{cases} f(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c & (\mathcal{D}x^T \oplus a^T)^T \neq i \\ f(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c \oplus 1 & (\mathcal{D}x^T \oplus a^T)^T = i \end{cases}$$

$$f_i(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c = \begin{cases} g(x) & x \neq ((\mathcal{D}^{-1}(i^T \oplus a^T))^T = j \\ g(x) \oplus 1 & x = ((\mathcal{D}^{-1}(i^T \oplus a^T))^T = j \end{cases}$$

And hence, $f_i(\mathcal{D}x^T \oplus a^T) \oplus b^T \cdot x^T \oplus c$ is equivalent with g_j, a connecting function of g such that $j = (\mathcal{D}^{-1}(i^T \oplus a^T))^T$.

Corollary 1. Let f and g be affine equivalent. Then the 1-local neighbourhood of f and the 1-local neighbourhood of g are related by a permutation.

Proof. This follows from the non-singularity of $\mathcal{D}$.

3 Redundancy in Rijndael S-Box Functions

It was already known from [4] that the functions in the Rijndael S-box exhibit identical excellent properties of algebraic degree and nonlinearity. However it was not well known that the Boolean functions formed by all 255 linear combinations share the same frequency distributions for absolute walsh transform and absolute autocorrelation values. Given that these properties are known to be conserved under affine transform, it suggests the existence of an affine relationship, but does not prove it.

It turns out that the general problem of determining equivalence between functions of six inputs and greater has been difficult, and to date the only known solution was exhaustive search. However, the results of the previous section provide the theoretical basis for a new technique that we have implemented. This approach reduces the search space significantly.

Theorem 1 indicates that the connecting functions of $f(x)$ and those of $g(x) = f(\mathcal{D}x^T \oplus a^T) \oplus b \cdot x^T \oplus c$ share the same equivalence mapping as f and g. Hence, rather than only two equivalent functions, we in fact have $2^n + 1$ pairs of equivalent functions under the same affine transform. After implementing the following algorithm, we believe that this provides sufficient data to uniquely determine $\mathcal{D}(n \times n$ invertible matrix), $(a, b) \in \mathbf{Z}_2^n$ and $c \in \mathbf{Z}_2$ in an efficient manner.

Test for Affine Equivalence

Input: $f(x)$, $g(x)$ and n
Output: Return $\mathcal{D}(n \times n$ invertible matrix), $(a, b) \in \mathbf{Z}_2^n$ and
$c \in \mathbf{Z}_2$ when test is positive, else return not equivalent.

1. Finding a
   (a) From Theorem 1, we know that connecting function i of f will be equivalent to connecting function $j = \mathcal{D}^{-1}(i \oplus a)$ of g and therefore $i = \mathcal{D}j \oplus a$. When $j = 0$ we know that $i = a$.
   (b) Thus, determine which connecting functions $g_{(j)}$ could be equivalent to connecting function $f_{(0)}$ using algebraic degree and the absolute frequency distribution of the WHT and autocorrelation function of both f and g, the a must be from the set of valid $j's$
   (c) Let the set of possible values for a be denoted $\{a_0, a_1, ...\}$.
2. Finding $\mathcal{D}$
   (a) From 1, we know that connecting function i of f will be equivalent to connecting function $j = \mathcal{D}^{-1}(i \oplus a)$ of g, and therefore $i = \mathcal{D}j \oplus a$ for $a \in \{a_0, a_1, ...\}$.
   (b) When $j = \mathbf{e}_k$ such that $\mathbf{e}_k$ be the unit vector with 1 is position k and 0 elsewhere, we see that i^T will be the k^{th} column of $\mathcal{D} \oplus a$.
   (c) Thus, determine which $f_{(i)}$ could be equivalent to $g_{(j)}$ when $j = \mathbf{e}_k$ $(\forall k \leq n)$, using algebraic degree and the absolute frequency distribution of the WHT and autocorrelation function of both f and g, to find the possible columns of $\mathcal{D} \oplus a$.
   (d) Let the set of possible values for $\mathcal{D}_k \oplus a$ be denoted $\{x(k)_0, x(k)_1, ...\}$.
3. Finding b and c
   (a) From 2, the only two remaining unknown variables of the transform relating f and g are $b \in \mathbf{Z}_2^n$ and $c \in \mathbf{Z}_2$.
   (b) Test each combination of these potential values for a, $\mathcal{D}_k \oplus a$, b and c to establish if a valid affine equivalence mappings exists.
   (c) If a valid mapping is found, return $\mathcal{D}(n \times n$ invertible matrix), $(a, b) \in \mathbf{Z}_2^n$ and $c \in \mathbf{Z}_2$. Otherwise, return no mapping found.

This algorithm provides a general procedure that can be applied to determine an affine equivalence relationship between two functions, or to prove that one does not exist. We should stress that the complexity varies greatly according to the actual pair of functions. We simply note that it is very efficient for testing most pairs of functions, including the functions in the Rijndael S-box.

This algorithm was applied to both the individual outputs and the linear combinations of the Rijndael S-box. In this case, for all pairs of functions considered, it was found that the search space was reduced to 2^{15} possible affine mappings. It was therefore a feasible task to automate this procedure and find the matricies. Exactly *eight* distinct linear transforms were identified relating any pair of functions from the S-box. One example mapping from each set relating the individual output functions is listed in Appendix A. The corresponding inverse matrices are in Appendix B.

4 Finite Fields

The polynomial basis representation of finite fields with characteristic 2 are discussed in the AES submission of Rijndael [4]. That document describes how to generate the S-box but does not examine its cryptographic properties in any great depth. We now present the first known proof of the linear redundancy in finite field inversion, due to David Wagner after he saw our initial posting to the IACR e-print archive [7].

Theorem 3. [17] The component output functions of finite field inversion are related by linear transform.

Proof. Let $Tr : GF(2^8)->GF(2)$ denote the trace function, and let $S : GF(2^8)->GF(2^8)$ be the AES S-box, i.e., inversion: $S(x) = x^{-1}$.

The basic fact required is that the linear function $f_i : GF(2^8)->GF(2)$ extracting the i-th bit of its input can be expressed in the form $f_i(x) = Tr(c_i x)$ for some constant c_i in $GF(2^8)$ that depends only on i.

Now we can see that

$$\begin{aligned} f_j(S(x)) &= Tr(c_j x^{-1}) \\ &= Tr(c_i d_{i,j}^{-1} x^{-1}) \\ &= f_i((d_{i,j} x)^{-1}) \\ &= f_i(S(d_{i,j} x)) \end{aligned}$$

where the constant d_ij in $GF(2^8)$ is given by $d_{i,j} = c_i c_j^{-1}$.

Of course, multiplying (in $GF(2^8)$) by any constant in $GF(2^8)$ is a $GF(2)$-linear map, hence for each i, j there is an $8 * 8$ matrix $M_{i,j}$ over $GF(2)$ so that $d_{i,j} x = M_{i,j} x$. We find that $f_j(S(x)) = f_i(S(M_{i,j} x))$, which is the result claimed.

We briefly note that Wagner's proof works for any linear combination of the outputs and moreover it can be adjusted to apply to any single term exponentiation. Inversion is the specific power mapping given by $x^{-1} = x^{2^{n-2}}$.

5 Proposing a New Criterion for S-Boxes

In this section we discuss the variety of different equivalence classes and consider it as an indicator of non-randomness for S-boxes. These results show how unusual it is for an S-box on 6 or more inputs to have any linear redundancy at all. For the case $n \leq 4$ there are so few equivalence classes that we expect there to be

Table 1. Class Redundancy of Random Bijective S-boxes $n = 5$, 1000 Trials

# classes	frequency
5	4
6	20
7	86
8	220
9	246
10	228
11	146
12	39
13	8
14	3

some redundancy. Consider that there are only 8 clases for $n = 4$, so every $4 * 4$ S-box must have some linear redundancy as there are 15 linear combinations.

When $n = 5$ there are 48 equivalent classes. For a 5*5 S-box to have all different classes, 31 are required, so it is difficult to avoid some redundancy. The distribution of number of classes for 5*5 bijective S-boxes is shown in the table 1. The average number of different classes is 9.

For n=6 and more there are so many available classes that it is difficult to find linear redundant S-boxes at random. Our experiments show that 3.3% of random 6*6 bijections have 62 different classes, and one class used twice. The rest had no redundancy with 63 classes. Our experiments at $n = 7, 8$ found all S-boxes had no redundancy (in trials of 1000 random bijections). From these results we see that linear redundancy is very rare for 6 or more inputs, hence we propose a new randomness criterion for S-boxes.

Proposition 1. Let B[.] be an S-box with 6 or more variables. Then B[.] fails the equivalence class variety test iff the S-box has any linear redundancy.

In the previous section we showed that finite field power mappings inherently possess saturated linear redundancy and so they clearly fail our new test for non-randomness.

6 Discussion

In this section we discuss some of the potential impact of these results on implementation, security and design.

6.1 Impact on Implementation

The most obvious consequence from these observations is the impact on the minimum size of hardware implementations. Clearly any implementation that used combinatorial logic to implement one S-box (with the time/space tradeoff) could achieve a much smaller size now by implementing only one Boolean function instead of eight. The hardware cost for some additional XORs is much less than the hardware savings, however the speed of the reduced size implementation will be reduced by a factor of 8. We note that very small hardware implementations may be economically suitable for future ubiquitous computing.

We note that simplified hardware for finite field inverse is already known from [15]. Here we point out that method implemented $GF(2^n)$ inversion using nonlinear Boolean logic surrounding a single instance of inversion in $GF(2^{n/2})$. Our redundancy result holds for that finite field operation also, so our improvement can be applied to that method to achieve further space reductions. We futher note that the combined construction can be used to implement inversion over $GF(2^{16})$ using only a single Boolean function of 8 inputs, thus making designs using these larger S-boxes easier to implement.

6.2 Impact on Security

The impact on security is more difficult to assess. It takes time for the cryptographic community to consider the many attacks that may be possible. To begin the discussion, we suggest these avenues for cryptanalysts to investigate:

- A distinguishing attack may be possible on reduced round ciphers using linear redundant S-boxes. There is more research needed to discover just how the surrounding structures influence the equivalence property over multiple rounds. Whenever redundancy persists over several rounds, then the cipher does not display random behaviour and could be easily distinguished from random.
- The linear redundancy could be exploited to reduce plaintext requirements of some existing attack techniques, open the door for new kinds of related key attacks, or improve the efficiency of other cryptanalyses such as using (perhaps multiple) non-linear approximations, higher order derivatives, interpolation, the square/integral attack, and algebraic attacks.
- The single formula to represent Rijndael, presented at SAC2001 [6], is simplified by this result, since the division operation in the continued fraction is really inversion in the finite field. We invite cryptanalysts to investigate how this redundancy affects the complexity of solving the equation from [6].
- Some ciphers, including Rijndael, use an inversion based S-box in the key schedule. We wonder what security consequences this might have! What is the effect of linearly redundant round keys on the effectiveness of linear and differential cryptanalysis?

It seems clear that there are many potential ways to investigate the application of linear redundant S-boxes to cryptanalysis. We challenge the cryptographic community to find these new attacks or prove they do not exist. Until such proof appears, there must remain some doubt about the security of ciphers using redundant S-boxes, given the extreme non-randomness of these structures.

6.3 Proposal for Tweaking Redundant S-Boxes

In case new attacks are found that exploit linear redundancy in S-boxes, we make the following suggestion for tweaking [9] the S-box of Rijndael, or any

Table 2. Properties of Tweaked AES-Sbox 10000 Trials

nonlinearity	order	DDT	frequency
96	7	8	1
98	6	8	4
98	7	6	1
98	7	8	1
100	6	6	21
100	6	8	48
100	6	10	3
100	7	6	9
100	7	8	13
100	7	10	1
102	6	6	382
102	6	8	723
102	6	10	20
102	6	12	1
102	7	6	146
102	7	8	307
102	7	10	8
104	6	6	2037
104	6	8	2776
104	6	10	56
104	6	12	1
104	7	6	895
104	7	8	1246
104	7	10	19
106	6	6	506
106	6	8	414
106	6	10	7
106	7	6	181
106	7	8	172
106	7	10	1

other cipher with a redundant 8*8 S-box. Divide a 128-bit public tweak value into 8 pairs of bytes. Then for each pair, swap entries in the S-box indexed by the bytes. This process can be done quickly in software and our experimental results show that only a few saps are reqired to remove the linear equivalence property, and that at worst the tweaked S-box is as good as random with regard to linear and differential probabilities. The designers of Rijndael noted that, given effectiveness of the wide-trail strategy, a random S-box should be enough to provide security against differential and linear cryptanalysis. The tweaked S-boxes we propose are typically higher nonlinearity than random 8*8 S-boxes.

Using this tweak, an $8 * 8$ S-box is altered in (up to) 16 places, which is $\frac{1}{16}$ of the S-box. Each round of Rijndael, for example, uses 16 S-boxes, so there is a good probability that these changes have some effect in each round of the encryption. The equivalence class property is removed by this tweak and if the key-schedule is allowed to use the original S-box, then the per-block encryption speed is not affected by this tweak. Alternatively, (if tweaking is not allowed) an extra 128-bit sub-key can be generated from the keyschedule and used to tweak the S-box before encryption.

We have performed some experiments to discover the distribution of properties in S-boxes generated by this tweak. An iteration of the experiment is swapping the outputs for a pair of randomly chosen inputs, and analysing the linearity and class variety of the resulting S-box. In 1000 trials we found that an average of 7.62 iterations was sufficient to eliminate any linear redundancy. The average S-box nonlinearity after this process was found to be 103.90. Table 2 shows the frequency distribution of the final S-box nonlinearity, starting with the Rijndael S-box in all cases.

In Appendix C we present the best S-box we have found so far by this method. It is the best from those found to have nonlin=106, order=7 and maximum XOR difference distribution table (DDT) value 6. We believe this S-box is suitable as a drop-in replacement for any (currently redundant) 8*8 S-box.

7 Conclusion

The property of linear redundnacy in S-boxes has been introduced, following the discovery of that property in the S-box of Rijndael. Moreover it is now proven that finite field representations over polynomial basis have exponentiation operations (including inversion) that display linear redundancy in all their component Boolean functions and all of their linear combinations. This kind of S-box is used my many recently proposed ciphers. The immeadiate impact of these discoveries on the design, implementation and cryptanalysis of ciphers using redundant mappings has been discussed, but clearly much more needs to be done on this topic. A non-randomness property is defined and we propose a way to remove the redundancy from affected S-boxes using some key-material or a tweak value. A highly nonlinear and non-redundant 8*8 S-box has been provided as a possible replacement.

Acknowledgements

The authors would like to thank Ed Dawson, Lauren May and Matt Henricksen for encouragement, assistance and helpful discussions. We also are grateful to the anonymous reviewers whose comments helped improve the presentation of this paper. We also give thanks to Dave Wagner for permission to present his original proof of redundancy for finite field inversion.

References

1. E.R. Berlekamp and L.R. Welch. Weight Distributions of the Cosets of the $(32, 6)$ Reed-Muller Code. *IEEE Transactions on Information Theory*, 18(1):203–207, January 1972.
2. D. Coppersmith. Personal communication, September 2002.

3. J. Daemen, L. Knudsen, and V. Rijmen. The Block Cipher SQUARE. In *Fast Software Encryption, 1997 Haifa Workshop, LNCS*, pages 149–165, 1997.
4. J. Daemen and V. Rijmen. AES proposal: Rijndael.
5. J.D. Denev and V.D. Tonchev. On the Number of Equivalence Classes of Boolean Functions under a Transformation Group. *IEEE Transactions on Information Theory*, 26(5):625–626, September 1980.
6. N. Ferguson, R. Schroeppel, and D. Whiting. A Simple Algebraic Representation of Rijndael. In *Proceedings of Selected Ares in Cryptology, SAC'01*, Lecture Notes in Computer Science, page 103, 2001.
7. J. Fuller and W. Millan. Linear redundancy in the aes s-box, aug 2002. manuscript 2002/111 on IACR E-print Archive.
8. E. Garrido. Personal communication, August 2002.
9. M. Liskov, R. Rivest, and D. Wagner. Tweakable Block Ciphers. In *Advances in Cryptology - Crypto '02, Proceedings*, Lecture Notes in Computer Science, page 31, 2002.
10. J.A. Maiorana. A Classificationn of the Cosets of the Reed-Muller code $r(1, 6)$. *Mathematics of Computation*, 57(195):403–414, July 1991.
11. S. Mister. Analysis of the building blocks of Serpent, 2000.
12. National Bureau of Standards (U.S.). Data Encryption Standard (DES). *Federal Information Processing Standards*, 1977.
13. E. Pasalic, T. Johansson, S. Maitra, and P. Sarkar. New constructions of resilient and correlation immune boolean functions achieving upper bounds on nonlinearity, 2001.
14. V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers, and E. De Win. The Cipher SHARK. In *Fast Software Encryption, 1996 Cambridge Workshop, LNCS*, pages 99–111, 1996.
15. V. Rijmen. Efficient Implementation of the Rijndael S-box. Presented at an AES conference.
16. C.E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949.
17. D. Wagner. Personal communication, August 2002.
18. The New European Schemes for Signatures, Integrity and Encryption (NESSIE) process maintains a web-site via http://www.cryptonessie.org.
19. The CRYPTREC process has a web-site at http://www.ipa.go.jp/security/enc/CRYPTREC.
20. The South Korean standards process has a web-site with downloads at http://www.kisa.or.kr/seed/algorithm.htm.

Appendix A – Rijndael Equivalence Relationships

The AES sbox functions are $b_i(x) = 1\&\{\texttt{AES}[x] >> (i-1)\}$, for $1 \leq i \leq 8$.

$$b_2(x) = b_1(\mathcal{D}_{12}x)$$

$$b_3(x) = b_1(\mathcal{D}_{13}x) \oplus 1$$

$$b_4(x) = b_1(\mathcal{D}_{14}x) \oplus 1$$

$$b_5(x) = b_1(\mathcal{D}_{15}x) \oplus 1$$

$$b_6(x) = b_1(\mathcal{D}_{16}x)$$
$$b_7(x) = b_1(\mathcal{D}_{17}x)$$
$$b_8(x) = b_1(\mathcal{D}_{18}x) \oplus 1$$

$$\mathcal{D}_{12} = \begin{bmatrix} 1&0&1&1&0&0&1&0\\ 0&1&0&1&0&1&1&0\\ 0&0&0&1&0&0&1&0\\ 1&0&0&1&1&0&0&0\\ 0&1&0&1&1&1&0&0\\ 1&0&0&0&1&1&1&0\\ 1&1&0&1&0&0&0&0\\ 1&0&0&0&1&1&0&1 \end{bmatrix} \mathcal{D}_{13} = \begin{bmatrix} 1&1&1&1&0&1&1&0\\ 0&1&1&0&0&1&0&0\\ 0&0&0&1&0&0&0&0\\ 1&0&1&1&0&0&1&1\\ 1&1&0&1&0&0&0&0\\ 0&0&0&1&0&0&1&0\\ 1&1&1&0&1&1&1&0\\ 0&1&1&0&0&0&1&0 \end{bmatrix}$$

$$\mathcal{D}_{14} = \begin{bmatrix} 0&1&0&1&0&0&1&0\\ 0&0&0&1&0&1&0&0\\ 0&1&0&0&0&1&0&1\\ 1&1&0&1&0&0&0&1\\ 1&1&1&0&0&1&1&0\\ 1&0&1&0&1&0&1&1\\ 0&1&1&0&1&0&1&0\\ 0&1&0&0&1&0&0&0 \end{bmatrix} \mathcal{D}_{15} = \begin{bmatrix} 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&0&0&0\\ 1&0&0&0&0&0&1&0\\ 1&0&0&0&0&1&0&1\\ 0&0&0&0&0&0&1&0\\ 1&0&0&0&1&1&0&0\\ 1&1&1&1&0&1&1&0\\ 0&0&1&0&1&1&1&0 \end{bmatrix}$$

$$\mathcal{D}_{16} = \begin{bmatrix} 1&1&0&1&1&0&0&0\\ 1&0&0&0&0&0&1&0\\ 0&1&1&0&1&1&0&0\\ 1&1&0&0&0&0&0&1\\ 0&1&1&0&1&1&1&0\\ 1&0&1&1&1&0&1&0\\ 1&0&1&1&0&1&0&1\\ 0&0&0&0&0&1&0&1 \end{bmatrix} \mathcal{D}_{17} = \begin{bmatrix} 0&1&0&0&0&0&0&0\\ 1&1&0&0&1&1&0&1\\ 1&0&0&1&1&0&1&1\\ 0&0&0&0&0&0&1&1\\ 0&0&1&0&1&0&0&1\\ 0&1&0&1&0&0&0&1\\ 0&1&0&0&1&0&1&0\\ 1&0&0&1&0&0&1&1 \end{bmatrix}$$

$$\mathcal{D}_{18} = \begin{bmatrix} 1&0&1&1&0&1&1&0\\ 1&1&1&0&0&0&0&0\\ 0&1&0&1&1&0&1&1\\ 0&1&1&1&0&0&0&0\\ 1&0&0&1&1&0&1&1\\ 1&1&1&0&1&1&1&0\\ 1&0&1&0&1&1&0&1\\ 1&1&0&0&0&0&0&1 \end{bmatrix}$$

Appendix B – Inverse Equivalence Relationships

$$\mathcal{D}_{21} = \begin{bmatrix} 0&1&1&1&1&0&0&0\\ 0&1&0&1&0&1&0&0\\ 1&1&0&1&1&0&0&0\\ 0&0&1&0&1&1&1&0\\ 0&1&0&0&0&1&1&0\\ 0&0&1&1&0&1&0&0\\ 0&0&0&0&1&1&1&0\\ 0&0&0&0&1&0&1&1 \end{bmatrix} \mathcal{D}_{31} = \begin{bmatrix} 1&1&0&0&0&1&0&0\\ 1&1&1&0&1&1&0&0\\ 1&1&0&0&1&0&0&1\\ 0&0&1&0&0&0&0&0\\ 1&0&1&0&0&0&1&0\\ 0&1&1&0&0&1&0&1\\ 0&0&1&0&0&1&0&0\\ 0&0&0&1&1&0&0&1 \end{bmatrix}$$

$$\mathcal{D}_{41} = \begin{bmatrix} 0&1&1&1&0&0&0&0\\ 0&0&1&0&1&1&0&1\\ 1&0&1&1&1&0&0&0\\ 0&0&0&1&0&1&1&0\\ 0&0&1&0&1&1&0&0\\ 0&1&0&1&0&1&1&0\\ 1&0&1&1&1&0&1&1\\ 0&1&0&1&1&0&1&1 \end{bmatrix} \mathcal{D}_{51} = \begin{bmatrix} 0&0&1&0&1&0&0&0\\ 0&1&1&0&1&0&1&1\\ 0&0&1&0&0&1&0&1\\ 1&0&0&0&0&0&0&1\\ 1&1&0&0&0&0&0&1\\ 1&1&1&0&1&1&0&1\\ 0&0&0&0&1&0&0&0\\ 1&1&0&1&0&1&0&1 \end{bmatrix}$$

$$\mathcal{D}_{61} = \begin{bmatrix} 0&1&1&0&1&0&0&0 \\ 1&1&0&1&1&0&1&0 \\ 0&1&1&1&0&1&1&0 \\ 0&0&0&1&1&1&0&1 \\ 0&0&1&0&1&1&1&1 \\ 1&0&1&0&0&0&1&1 \\ 0&0&1&0&1&0&0&0 \\ 1&0&1&0&0&0&1&0 \end{bmatrix} \mathcal{D}_{71} = \begin{bmatrix} 0&0&1&0&0&1&1&0 \\ 1&0&0&0&0&0&0&0 \\ 1&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&1 \\ 0&0&1&0&0&0&0&1 \\ 0&1&1&1&0&1&0&0 \\ 1&0&1&0&0&0&1&1 \\ 1&0&1&1&0&0&1&1 \end{bmatrix}$$

$$\mathcal{D}_{81} = \begin{bmatrix} 1&0&0&0&1&0&1&0 \\ 1&0&1&0&0&0&1&0 \\ 0&1&1&0&1&0&0&0 \\ 1&1&0&1&1&0&1&0 \\ 1&1&1&1&1&1&0&0 \\ 0&0&1&1&0&1&0&1 \\ 1&0&0&0&1&1&0&1 \\ 0&0&1&0&1&0&0&1 \end{bmatrix}$$

Appendix C – A Replacement S-Box

The following bijective s-box has nonlinearity 106 and algebraic order 7. It contains no fixed points and no linear redundancy. The sbox has a DDT maximum of 6. The distribution of properties over all 255 XOR combinations of the sbox output functions is shown in Tables 3 and 4.

SBox[256]={

```
63, 7C, 77, DD, F2, 6B, 6F, C5,30, 01, 67, 2B, FE, D7, AB, 76,
CA, 82, C9, 7D, FA, 59, 47, F0,AD, D4, A2, AF, 9C, A4, 72, C0,
B7, FD, 93, 26, 36, 3F, F7, CC,34, A5, E5, F1, 71, D8, 31, 17,
04, C7, 23, C3, 18, 96, 05, 9A,07, 12, 80, E2, EB, 27, B2, 75,
09, 83, 2C, 1A, 1B, 6E, 10, A0,52, 3B, D6, B3, 29, 74, 2F, 84,
53, D1, 00, ED, 20, FC, B1, 5B,6A, CB, BE, 39, 4A, 4C, 58, CF,
D0, EF, AA, FB, 43, 4D, 56, 85,45, F9, 02, 7F, 50, 3C, 9F, A8,
51, A3, 40, 8F, 92, 9D, 38, F5,BC, B6, DA, 21, 15, FF, F3, D2,
CD, 0C, 13, EC, 5F, 97, 44, 5A,C4, A7, 7E, 3D, 64, 5D, 19, 73,
60, 81, 4F, DC, 22, 2A, 90, 88,46, EE, B8, 14, DE, 5E, 0B, DB,
E0, 32, 3A, 0A, 49, 06, 24, 5C,C2, D3, AC, 62, 91, 95, E4, 79,
E7, C8, 16, 6D, 8D, D5, 4E, A9,6C, 33, F4, EA, 65, 7A, AE, 08,
BA, 78, 25, 2E, 1C, A6, B4, C6,E8, 7B, E3, 1F, 4B, BD, 8B, 8A,
70, 3E, B5, 66, 48, 03, F6, 0E,61, 35, 57, B9, 86, C1, 1D, 9E,
E1, F8, 98, 11, 69, D9, 8E, 94,9B, 1E, 87, E9, CE, 55, 28, DF,
8C, A1, 89, 0D, BF, E6, 42, 68,41, 99, 2D, 0F, B0, 54, BB, 37}
```

Table 3. Frequency Distribution of Sbox Nonlinearity

nonlinearity	frequency
106	8
108	76
110	147
112	24

Table 4. Frequency Distribution of Sbox Maximum Autocorrelation

maximum autocorrelation	frequency
32	1
40	93
48	134
56	27