Journal of Cryptology
© 1990 International Association for Cryptologic Research

Enumerating Boolean Functions of Cryptographic Significance[1]

Chris Mitchell
Department of Computer Science, Royal Holloway and Bedford New College,
University of London, Egham Hill, Egham,
Surrey TW20 0EX, England

Abstract. In this paper we describe applications of functions from $GF(2)^m$ onto $GF(2)^n$ in the design of encryption algorithms. If such a function is to be useful it must satisfy a set of criteria, the actual definition of which depends on the type of encryption technique involved. This in turn means that it is important to ensure that the selected criteria do not restrict the choice of function too severely, i.e., the set of functions must be enumerated. We discuss some of the possible sets of criteria and then give partial results on the corresponding enumeration problems. Many open problems remain, some of them corresponding to well-known hard enumeration questions.

Key words. Cryptography, Boolean function, Enumeration, S-boxes, Look up tables.

1. Introduction

Functions from $GF(2)^m$ onto $GF(2)^n$ ($m \geq n$) are used in a variety of ways in the construction of encryption algorithms. Such functions are used in both stream ciphers and block ciphers as important components of the ciphering operation. In all cases the functions used have to be chosen with great care so that the resulting cipher is hard to break.

Theoretical and practical studies reveal criteria which functions must satisfy for use (sometimes the criteria are the same for use in very different ciphers, such as the need for nonlinearity). Having specified criteria, it is important to know that there exist suitable numbers of functions satisfying them.

As a result the problem arises of enumerating sets of functions satisfying various criteria. We discuss a number of such enumeration problems, many of which equate to classical counting questions with no previous obvious application. We concentrate our attention here on criteria identified as being of particular relevance to the design of stream ciphers, although some of these criteria are also relevant to the design of functions used in block ciphers.

[1] Date received: November 28, 1988. Date revised: November 7, 1989.

In Section 2 below we describe how these functions are used, and then discuss some of the criteria which arise from the particular applications. This leads on to some specific enumeration problems and results, which are discussed in Section 3.

2. Applications and Selection Criteria

2.1. *Introduction*

We start by describing in general terms how functions from $GF(2)^m$ onto $GF(2)^n$ are used in the construction of stream-cipher and, very briefly, block-cipher algorithms. Such functions are, in practice, often represented as a complete listing of all 2^m n-bit outputs, and they are thus often referred to as *Look Up Tables* (*LUTs*). For convenience we now refer to all such functions as (m, n)-*LUTs*, where $m \geq n$.

The chief component of a stream-cipher encryptor is a pseudorandom binary sequence generator; for further details on the design and applications of stream-cipher algorithms see, for example, [5] or [27]. The output from this sequence generator (which is initially "seeded" using a secret key) is combined with the binary data sequence using modulo 2 addition. For such a cipher to be strong (i.e., resist cryptanalysis) the sequence generator must satisfy a number of properties, not least of which are that the output sequence should appear random, and that the output should not be a linear function of the key. To these and other constraints should be added the desirability of straightforward and fast implementation.

Sequences generated using linear feedback shift registers are certainly easy and cheap to implement and they also have many properties required of stream-cipher generators (such as pseudorandomness). However, they fall down on the linearity constraint. One commonly used way of rectifying this problem is to combine two or more linear sequence generators using nonlinear feed-forward logic to produce a pseudorandom nonlinear sequence. In essence this means using the outputs from a number of registers as the inputs to an (m, n)-LUT, and using the output as the enciphering sequence. For added complexity without using LUTs of vast size, this "look up" process can be repeated a number of times.

Block ciphers operate in rather a different way, and involve encrypting groups of data bits simultaneously. The basic idea is to combine a block of plaintext data with a key to produce a block of ciphertext, with the property that a small change to either key or plaintext results in a large, unpredictable change to the resulting ciphertext. For further details on desirable properties for block ciphers see, for example, Section 7.3 of [5] or [14]. There are many ways to construct good block-cipher algorithms, but we are concerned here with just two closely related families of techniques, namely SP-networks and Feistel Ciphers.

Following a suggestion of Shannon [29], SP-networks have been proposed as good candidates for constructing block ciphers (see [14], [18], [19], and [1]). We do not describe the technique here, but suffice it to say that the use of (m, n)-LUTs is fundamental to their operation. The same is true of Feistel Ciphers, one particularly well-known example of which is the DES algorithm (see, for example, Section 7.3 of [5] or Section 3 of [9]).

For both types of application we need to be very careful about the selection of

LUTs to use. To assist in the selection of suitable LUTs we introduce lists of selection criteria which LUTs must satisfy before they have the potential to be useful in the construction of strong ciphers. We now look at possible sets of criteria in more detail, concentrating on the stream-cipher application.

2.2. *Function Selection Criteria*

We have already outlined the main motive for using LUTs in constructing stream-cipher algorithms in Section 2.1 above. We now need to consider some desirable properties for stream-cipher sequences in order to appreciate how to choose these LUTs.

As we have already mentioned, sequences used in stream-cipher applications must be both pseudorandom in appearance and nonlinear functions of the key. In addition, every key bit should affect the output sequence. These.simple requirements immediately give us three conditions on any (m, n)-LUT L used to combine linear sequence generators. Note that throughout this paper all algebra uses GF(2) except where otherwise stated.

C1. Balance. Over the complete set of possible inputs, each possible n-bit output should occur 2^{m-n} times, i.e., if $\mathbf{y}$ is any n-bit vector, then

$$|L^{-1}(\mathbf{y})| = 2^{m-n}.$$

C2. Nonlinearity/Affinity. L must be a nonlinear and nonaffine function for all n outputs, i.e., for every $i (1 \leq i \leq n)$ there must not exist a vector $\mathbf{h}$ in $\mathrm{GF}(2)^m$ and a fixed scalar a such that

$$L(\mathbf{x})|_i = \mathbf{x} \cdot \mathbf{h} + a \text{ for every } \mathbf{x} \text{ in } \mathrm{GF}(2)^m,$$

where $\mathbf{y}|_i$ denotes the bit in the ith position in vector $\mathbf{y}$.

C3. Nondegeneracy. Each of the n outputs of L must depend on all the m inputs; i.e., if each of the n output variables is expressed as an equation in the m input variables, then each equation must involve all of the m input variables.

Note that C1 is essential if the output is to appear pseudorandom (of course C1 does not in itself guarantee pseudorandomness). Condition C2 is present to ensure the nonlinearity and nonaffinity of the output sequence, and C3 ensures that every key bit affects each bit of the output sequence.

The above criteria are widely accepted, and Beale [4] has given a recurrence for the number Q_m of $(m, 1)$-LUTs satisfying C1–C3. Beale goes on to suggest that, since Q_m grows very quickly with m, some particular schemes he suggests are secure. However, although some such schemes may be secure, C1–C3 are by no means sufficient to guarantee this. We now consider some further criteria of importance.

C4. Uncorrelatedness. Given any vectors

$$\mathbf{x} = (x_1, x_2, \ldots, x_m), \qquad \mathbf{y} = (y_1, y_2, \ldots, y_n)$$

for which

$$L(\mathbf{x}) = \mathbf{y},$$

then

$$\Pr(x_i = y_j) = 0.5 \qquad \text{for any} \quad i, j\,(1 \le i \le m \text{ and } 1 \le j \le n).$$

The importance of C4 is clear from the recent work of Siegenthaler [30]–[33], Rueppel [26], and Retter [25]. Basically, if C4 is not satisfied, then it may be possible to cryptanalyse a sequence generator by attacking one component at a time. For a full discussion of the correlation property the interested reader is advised to consult the literature, since the definition of C4 above is a rather simplified version of the criterion developed by Siegenthaler.

C5. Symmetry. If P is any m by m permutation matrix, then

$$L(\mathbf{x}) = L(\mathbf{x}P)$$

for any $\mathbf{x}$ in $\mathrm{GF}(2)^m$.

This property was introduced by Brüer [7], who suggests that it is important because it means that no input is of any greater or lesser significance than any other input. Condition C5 means that (given w is the Hamming weight function) if $w(\mathbf{x}) = w(\mathbf{y})$, then $L(\mathbf{x}) = L(\mathbf{y})$. Condition C5 is probably overrestrictive as we discuss in Section 3.2 below.

Many other similar *ad hoc* constraints can be devised. Of particular relevance are strengthened versions of C2, which require the function to be nonaffine in all nontrivial subsets of the m input variables. For example, if the $(m-1, n)$-LUT L' obtained from L by setting one input variable to 0 (or 1) does not satisfy C2, then the function is probably not "nonaffine enough" to form a cryptographically strong sequence generator. One very important point to note is that just because a function satisfies a list of criteria (C1–C3 say), it does not guarantee that it will produce a strong cipher. Other types of attack using other properties of the LUT cannot easily be ruled out.

On the other hand, if too large a set of criteria is imposed, it may well happen that no such LUTs exist! It is at this point that the question of enumeration becomes of great importance. While some criteria (such as C1–C3) are of fundamental importance, others (such as C5) are perhaps less vital. The result of enumeration and classification work should help the algorithm designer decide which set of criteria to use.

Before proceeding, we briefly mention three criteria of particular relevance to block ciphers. Before considering selection criteria in detail we need to specify exactly what we mean by S-box functions. As usually defined, an S-box is a collection of 2^r invertible (n, n)-LUTs. If we let $m = n + r$, then an alternative definition of an S-box is that it is an (m, n)-LUT L satisfying C6 below.

C6. For any

$$(a_1, a_2, \ldots, a_r) \text{ in } \mathrm{GF}(2)^r,$$

the (n, n)-LUT L^* defined by

$$L^*(x_1, x_2, \ldots, x_n) = L(a_1, a_2, \ldots, a_r, x_1, x_2, \ldots, x_n)$$

is one-to-one.

If we think of an S-box as consisting of 2^r permutations on the set of all n-bit vectors, then a further possible criterion seems natural. When used in an SP-network, key bits are used to select which permutation is used, and so it might be desirable to make the permutations as different from one another as possible. One way in which we might do this is by requiring an S-box to satisfy C7 below.

C7. For any

$$(b_1, b_2, \ldots, b_n) \text{ in } \mathrm{GF}(2)^n,$$

the function L^{**} mapping $\mathrm{GF}(2)^r$ into $\mathrm{GF}(2)^n$ defined by

$$L^{**}(x_1, x_2, \ldots, x_r) = L(x_1, x_2, \ldots, x_r, b_1, b_2, \ldots, b_n)$$

is one-to-one.

In simple terms C7 has the effect of requiring that no two permutations agree in any position (and hence C7 can only be satisfied if $r \le n$). Alternatively, if we think of the 2^r permutations in the S-box as forming rows in a 2^r by 2^n matrix, C6 and C7 are precisely the same as requiring that the matrix form a *Latin Rectangle.* As a result we call S-boxes satisfying C6 and C7 *Latin Rectangle S-boxes.*

Enumerating S-boxes satisfying C6 or C7 in isolation is very straightforward; by contrast, enumerating S-boxes satisfying C6 and C7 in combination is equivalent to enumerating Latin Rectangles. Computing this number is a "classical" hard problem. Asymptotes (for r small with respect to n) are known for this number, due to Erdös and Kaplansky [13] and extended by Yamamoto [35], [36]. The van der Waerden–Egorycev theorem and the Minc–Brégman upper bound can also be used to give bounds on the size of the number. For further details see Section 3.1 of Minc's update [24] to his earlier book [23].

Having looked at two possible requirements for S-boxes we next note that C1–C4 above are also very significant for S-boxes. Condition C1 is in fact guaranteed by C6 (N.B. they are equivalent if $m = n$, i.e., if $r = 0$). A further criterion of relevance to S-box design is the following, originally proposed by Webster and Tavares [34].

C8. Strict Avalanche Criterion. Define the probability p_{ij} as follows. Let $\mathbf{c}_i$ be the m-vector with a one in the ith position and zeros elsewhere. Then, if $\mathbf{x}$ is any m-vector, p_{ij} is defined to be the probability that

$$(L(\mathbf{x}) + L(\mathbf{x} + \mathbf{c}_i))|_j = 1.$$

Then L satisfies the *Strict Avalanche Criterion* (*SAC*) iff

$$p_{ij} = 0.5 \qquad \text{for every} \quad i, j\ (1 \le i \le m,\ 1 \le j \le n).$$

Before proceeding we briefly mention other existing work on the enumeration of

S-boxes satisfying certain criteria. For the combination of the two criteria C6 and C2, certain enumerative results have been achieved by Gordon and Retkin [16] and Ayoub [2], [3], albeit always for the $m = n$ case. The emphasis of this work has been to demonstrate that for sufficiently large m almost all S-boxes satisfy a certain minimal set of criteria. Their purpose is to show that for sufficiently large m it is safe to choose S-boxes at random. Their results are given as corollaries of more general enumeration results in Section 3 below. Additionally, Lloyd [20]–[22] has recently enumerated those $(m, 1)$-LUTs which satisfy certain cases of Forre's generalized version of C8 [15].

Finally we note that the criteria used to select the DES S-boxes remain classified. As a result a large effort has gone into trying to deduce the criteria used, and additionally to find weaknesses in the selected S-boxes. Some interesting work of this type can be found in a number of recent papers [6], [8], [10]–[12], [28]. This work is also of significance in selecting new S-boxes for future block ciphers.

3. Enumeration Problems

We now consider the enumeration of (m, n)-LUTs satisfying various subsets of the criteria given in Section 2.2 above. To some extent the results are for those subsets of criteria for which enumerations have proved tractable, rather than necessarily those (probably larger) sets of criteria of direct cryptographic significance.

As we have already stated, we concentrate our attention here on those criteria of particular relevance to stream ciphers, i.e., C1–C5. It is clearly of importance to know how many (m, n)-LUTs exist satisfying combinations of C1–C5, and in particular those satisfying all of C1–C3 together with one or both of the other conditions. We consider each of C1–C5 in turn.

3.1. *Condition C1–Balance*

We immediately have:

Theorem 3.1.1. *The number of (m, n)-LUTs satisfying C1 (i.e., the number of balanced (m, n)-LUTs) is given by*

$$b_{m,n} = M!/[(M/N)!]^N,$$

where $M = 2^m$, $N = 2^n$, and $C(n, k)$ denotes the binomial coefficient $n!/k!(n - k)!$, as it does throughout.

Proof. Clearly,

$$\begin{aligned} b_{m,n} &= \prod_{i=0}^{N-1} C(M - i \cdot M/N, M/N) \\ &= \prod_{i=0}^{N-1} (M - iM/N)!/[(M/N)! \cdot (m-(i + 1)M/N)!]. \end{aligned}$$

The result follows. □

3.2. *Condition C2—Nonlinearity/Affinity*

We now enumerate those LUTs satisfying C1 and C2.

Lemma 3.2.1. *The number of (m, n)-LUTs satisfying C1 and for which some chosen set of k outputs are all affine functions of the inputs is given by*

$$a_{m,n,k} = K \cdot \prod_{i=0}^{k-1} (M - I) \cdot [(M/K)!]^K / [(M/N)!]^N,$$

where $I = 2^i$, $K = 2^k$, $M = 2^m$, *and* $N = 2^n$.

Proof. Suppose L is any (m, n)-LUT with the desired properties. Then consider first L^*, the (m, k)-LUT obtained from L by restricting attention to the k outputs which must be an affine function of the inputs. L^* must be of rank k in order for L to be balanced. Hence there are

$$K \cdot \prod_{i=0}^{k-1} (M - I)$$

possibilities for L^*.

We now consider how many ways there are of extending L^* to a balanced (m, n)-LUT. If we examine the k outputs determined by L^* over all 2^m possible inputs, each possible pattern of k output bits occurs 2^{m-k} times. If we consider one collection of 2^{m-k} inputs all having the same k outputs, then, in order for L to be balanced, the other $n - k$ outputs must take each of their 2^{n-k} possibilities 2^{m-n} times each. As in Theorem 3.1.1, the number of ways this can happen is simply

$$\prod_{j=0}^{N/K-1} C(M/K - j \cdot M/N, M/N) = (M/K)!/[M/N)!]^{N/K}.$$

This applies equally to all 2^k possible values for the k outputs determined by L^* and the result follows. □

Using Lemma 3.2.1 we can now obtain:

Lemma 3.2.2. *The number of balanced (m, n)-LUTs which are affine in precisely k of their outputs is*

$$\sum_{i=0}^{n-k} (-1)^i \cdot C(n, k + i) \cdot C(k + i, k) \cdot a_{m,n,k+i}.$$

Proof. This result follows immediately from Lemma 3.2 on application of the inclusion–exclusion principle (see, for example, Section 2.1 of [17]). □

As an immediate corollary of Lemma 3.2.2 we now have:

Theorem 3.2.3. *The number of (m, n)-LUTs satisfying C1 and C2 (i.e., the number of balanced (m, n)-LUTs nonaffine in all their outputs) is given by*

$$d_{m,n} = \sum_{i=0}^{n} (-1)^i \cdot C(n, i) \cdot a_{m,n,i}.$$

Note that the above results generalize the work of Gordon and Retkin [16], who studied the special case $m = n$. In fact, they explicitly studied (m, m)-LUTs satisfying C6 and C2, which in this case turns out to be the same as enumerating (m, m)-LUTs satisfying C1 and C2.

3.3. *Condition C3—Nondegeneracy*

We next consider the number of (m, n)-LUTs satisfying C3. Before commencing note the following trivial yet useful result:

Lemma 3.3.1. *Suppose x_m is the number of $(m, 1)$-LUTs satisfying some combination of C2, C3, C4, and C5. Then the number of (m, n)-LUTs satisfying the same set of conditions is simply $(x_m)^n$.*

Proof. The lemma follows immediately from the definitions of C2–C5. □

Using this result we now have:

Theorem 3.3.2. *Let $e_{m,n}$ denote the number of (m, n)-LUTs satisfying C3 (i.e., the number of nondegenerate (m, n)-LUTs). Then we have the following results enabling the simple computation of $e_{m,n}$:*

(i) *$e_{m,1}$ satisfies the recurrence*

$$e_{m,1} = 2^M - \sum_{i=0}^{m-1} c(m, i) \cdot e_{i,1},$$

where $M = 2^m$.

(ii) $e_{0,1} = 2$.

(iii) $e_{m,n} = (e_{m,1})^n$.

Proof. (i) There are 2^M possible $(m, 1)$-LUTs. Each such function will be a nondegenerate function of some subset of the set of m input variables, and hence we have

$$\sum_{i=0}^{m} C(m, i) \cdot e_{i,1} = 2^M.$$

The desired recurrence immediately follows. (ii) is trivial and (iii) follows immediately from Lemma 3.3.1. □

When we consider C3 in combination with C1 and C2, the problem becomes rather more complex. However, for the case $n = 1$ the problem is tractable, and we have the following result (previously obtained by Beale and Monaghan [4]):

Theorem 3.3.3. *The number Q_m of $(m, 1)$-LUTs satisfying C1–C3 obeys the following recurrence:*

$$Q_m = d_{m,1} - \sum_{i=1}^{m-1} C(m, i) \cdot Q_i,$$

where $d_{m,1}$ is as in Theorem 3.2.3 *above. In addition we have the initial condition*

$$Q_1 = 0.$$

Proof. The result follows by observing that an $(m, 1)$-LUT which does not satisfy C3 is simply a nondegenerate $(k, 1)$-LUT for some subset of k of the input variables. The recurrence then follows immediately. Finally, note that

$$d_{1,1} = 0,$$

and hence

$$Q_1 = 0.$$

3.4. *Condition C4—Uncorrelatedness*

We next consider C4. Let $u_{m,n}$ denote the number of (m, n)-LUTs satisfying C4 (i.e., the number of uncorrelated (m, n)-LUTs). As for C3, because of Lemma 3.3.1, we need only consider $u_{m,1}$. However, even for this case the enumeration problem is rather difficult. What we can say is as follows:

Lemma 3.4.1. *$u_{m,1}$ is the number of ways the elements of* GF $(2)^m$ *can be partitioned into two sets A, B (possibly empty) such that, if*

$$\mathbf{x} = (x_1, x_2, \ldots, x_m)$$

is in A, and

$$p_i = \Pr(x_i = 1),$$

then

$$p_i = 0.5 \qquad \textit{for every } i\ (1 \le i \le m).$$

Proof. For any $(m, 1)$-LUT L, let A and B denote the sets of m-vectors which are mapped by L onto 0 and 1, respectively. Then it is clear that L satisfies C4 if and only if A and B have the properties specified above. The lemma follows. □

In the absence of a precise enumeration, a simple method of guaranteeing uncorrelatedness is of potential interest. We have the following:

Theorem 3.4.2. *If an $(m, 1)$-LUT (with $m \ge 2$) satisfies*

$$L(\mathbf{x}) = L(\mathbf{x} + \mathbf{i}) \tag{$*$}$$

for all m-vectors $\mathbf{x}$ *(where* $\mathbf{i}$ *is the m-vector of all ones), then L satisfies C4, i.e., L is uncorrelated. Hence*

$$u_{m,1} \ge 2^{M'} \qquad \textit{where} \quad M' = 2^{m-1}.$$

Moreover, the number $bu_{m,n}$ of (m, n)-LUTs satisfying C1 and C4, i.e., the number of balanced uncorrelated (m, n)-LUTs, satisfies

$$bu_{m,n} \ge b_{m-1,n},$$

where $b_{m-1,n}$ is the number of $(m - 1, n)$-LUTs satisfying C1 (see Theorem 3.1.1).

Proof. Suppose L is an $(m, 1)$-LUT satisfying the property $(*)$ for all $\mathbf{x}$. Then if A is the set of m-vectors which L maps onto zero, then $\mathbf{x}$ is in A if and only if $\mathbf{x} + \mathbf{i}$ is in A. Hence the elements of A can be divided into pairs of vectors and their complements (where we define the complement of $\mathbf{x}$ to be $\mathbf{x} + \mathbf{i}$). Therefore, in any of the m bit positions, exactly half the m-vectors in A have a one in that position. Therefore L satisfies C4.

The number of complementary pairs of m-vectors is simply

$$M' = 2^{m-1}.$$

A necessary and sufficient condition for an $(m, 1)$-LUT to satisfy property $(*)$ is that the set A consists of some collection of complementary pairs. The number of choices for such an A is simply $2^{M'}$ and the bound for $u_{m,1}$ follows.

A necessary and sufficient condition for an (m, n)-LUT to satisfy C1 is that, for any n-vector $\mathbf{y}$, the set $L^{-1}(\mathbf{y})$ must have cardinality precisely 2^{m-n}. In addition, as above, a sufficient condition for an (m, n)-LUT to satisfy C4 is that, for each n-vector $\mathbf{y}$, the set $L^{-1}(\mathbf{y})$ contains only complementary pairs of vectors. Hence a sufficient condition for an (m, n)-LUT to satisfy both C1 and C4 is that, for each n-vector $\mathbf{y}$, the set $L^{-1}(\mathbf{y})$ contains precisely 2^{m-n-1} complementary pairs of vectors. The desired bound follows. □

The condition $(*)$ in Theorem 3.4.2 is rather restrictive. This is illustrated by the fact that if a $(2N, N)$-LUT satisfies C6 and C7, then it must also satisfy C4. We conclude this section by briefly considering the effect of requiring both C3 and C4. Suppose output y_j does not depend on input x_i; then it is clear that x_i and y_j will be uncorrelated in the sense of C4. This indicates that C3 and C4 are related so that any pair (x_i, y_j) cannot be both independent and correlated. This suggests that enumerating (m, n)-LUTs satisfying C3 and C4 may be a nontrivial task.

3.5. *Condition C5—Symmetry*

We next consider (m, n)-LUTs satisfying C5. This is a strong condition, and there is a very limited set of LUTs which satisfy it. We first note the following trivial result, previously quoted informally following the definition of C5:

Lemma 3.5.1. *If L satisfies C5, i.e., if L is a symmetric (m, n)-LUT, and if $w(\cdot)$ is the Hamming weight function, then*

$$w(\mathbf{x}) = w(\mathbf{x}')$$

implies that

$$L(\mathbf{x}) = L(\mathbf{x}').$$

Having observed this simple result, we can now state:

Theorem 3.5.2. *The number $s_{m,n}$ of symmetric (m, n)-LUTs (i.e., the number of (m, n)-LUTs satisfying C5) is given by*

$$s_{m,n} = (2^{m+1})^n.$$

Proof. Since there are $C(m, i)$ vectors of weight i, by Lemma 3.5.1 the number of symmetric (m, n)-LUTs is simply the number of ways the set of binomial coefficients

$$\{C(m, 0), C(m, 1), \ldots, C(m, m)\}$$

can be partitioned into 2^n sets. The result follows. □

3.6. *Conditions C1–C5*

When we consider C5 in combination with other conditions, the enumeration problem becomes much more difficult. Before attempting to enumerate those (m, n)-LUTs satisfying some combination of C1–C4 in conjunction with C5 we observe the following. It is well known (and elementary to establish) that any (m, n)-LUT can be uniquely expressed as a set of n multinomial equations in m variables:

$$x_1, x_2, \ldots, x_m,$$

where each term is a product of between 0 and m of these variables. In such a multinomial equation, let the *weight* of a term be the number of variables appearing in the term (e.g., the term $x_1 x_5 x_7$ has weight 3 whereas the term 1 has weight 0). Moreover, if s is a term (i.e., a product of some subset of the x_i's) and $\mathbf{x}$ is an m-vector, then s is said to be *agreeable* to $\mathbf{x}$ if all the variables in s have their corresponding positions in $\mathbf{x}$ set to 1. Using this notation we then have:

Lemma 3.6.1. *Suppose L is an $(m, 1)$-LUT with equivalent multinomial equation*

$$y_1 = f(x_1, x_2, \ldots, x_m).$$

Then, if $\mathbf{x}$ is an m-vector of weight k,

$$L(\mathbf{x}) = \sum_{i=0}^{k} N(i),$$

where $N(i)$ represents the number of terms of weight i in f which are agreeable to $\mathbf{x}$.

Proof. If we consider $f(\mathbf{x})$ term by term, then the terms that contribute a 1 to the result are precisely those agreeable to $\mathbf{x}$. The lemma follows. □

We may then state the following lemma, a version of which was informally stated by Brüer [7].

Lemma 3.6.2. *Suppose L is an $(m, 1)$-LUT with equivalent multinomial equation*

$$y_1 = f(x_1, x_2, \ldots, x_m).$$

Then L satisfies C5 if and only if, for every i, f either contains all terms of weight i or no terms of weight i.

Proof. First suppose that L satisfies C5. We prove the desired result by induction on i.

If $i = 0$, then the result is trivially true since there is only one term of weight 0.

Suppose the result is true for every $i < k$. Suppose also that W is the subset of $\{0, 1, \ldots, k - 1\}$ defined so that w is in W if and only if f contains all terms of weight w. Let $\mathbf{x}$ be any m-vector of weight k. By Lemma 3.6.1 we have

$$\begin{aligned} L(\mathbf{x}) &= \sum_{i=0}^{k} N(i) \\ &= \sum_{i \in W} C(k, i) + d, \end{aligned}$$

where $d = 1$ or 0 depending on whether or not the unique term of weight k agreeable to $\mathbf{x}$ is present in f.

But, by C5, $L(\mathbf{x})$ is a constant for all $\mathbf{x}$ of weight k. The induction follows.

Hence, if L satisfies C5, then, for every i, f either contains all or none of the possible terms of weight i. The converse is straightforward since the number of f with the property that, for every i, f contains either all or none of the possible terms of weight i is exactly the same as the number of $(m, 1)$-LUTs satisfying C5. The result follows. □

Using this lemma we can now simply establish:

Theorem 3.6.3. *The number of (m, n)-LUTs satisfying C2 and C5 (i.e., the number of symmetric (m, n)-LUTs for which none of the outputs are affine functions of the inputs) is precisely $(2^{m+1} - 4)^n$.*

Proof. We consider the number of affine symmetric $(m, 1)$-LUTs. It is clear that an LUT is affine if and only if its corresponding multinomial equation only contains terms of weight 0 or 1. By Lemma 3.6.2 there exist precisely four symmetric $(m, 1)$-LUTs with this property. Hence, by Theorem 3.5.2 there exist precisely $2^{m+1} - 4$ nonaffine symmetric $(m, 1)$-LUTs. The result follows on application of Lemma 3.3.1. □

We next observe that, by Lemma 3.6.2, the only $(m, 1)$-LUTs which satisfy C5 and do not satisfy C3 (the nondegeneracy condition) are the trivial functions

$$L(\mathbf{x}) = 0 \qquad \text{for all } \mathbf{x}$$

and

$$L(\mathbf{x}) = 1 \qquad \text{for all } \mathbf{x}$$

which, in addition, are both affine. It is therefore trivial to show:

Theorem 3.6.4. *The number of (m, n)-LUTs satisfying C3 and C5 (i.e., the number of symmetric (m, n)-LUTs nondegenerate in all their outputs) is $(2^{m+1} - 2)^n$. Moreover, if an (m, n)-LUT satisfies C2 and C5, then it also satisfies C3, and hence the number of (m, n)-LUTs satisfying C2, C3, and C5 is $(2^{m+1} - 4)^n$.*

We now consider the effect of requiring C1 in addition to C5. Since there are $C(m, i)$ vectors of weight i, the number of balanced, symmetric (m, n)-LUTs (i.e., the number satisfying C1 and C5) is simply the number of ways the set of binomial coefficients

$$\{C(m, 0), C(m, 1), \ldots, C(m, m)\}$$

can be partitioned into 2^n sets so that the sum of the coefficients in each set equals 2^{m-n}. For the case $n = 1$, two obvious families of examples exist (in fact, these are almost the only examples known to the author for any value of n). These examples can be used to establish the following lower bound:

Theorem 3.6.5. *The number $bs_{m,1}$ of balanced, symmetric $(m, 1)$-LUTs satisfies*

$$bs_{m,1} \geq 2^{(m+1)/2} \qquad \text{if } m \text{ is odd,}$$

$$bs_{m,1} \geq 2 \qquad \text{if } m \text{ is even.}$$

Proof. We establish these bounds by showing how to construct the required numbers of examples of balanced, symmetric $(m, 1)$-LUTs. We write c_i for $C(m, i)$ throughout, and consider partitions of the values c_i into two sets A and B such that $|A| = |B| = 2^{m-1}$.

First suppose that m is odd. Consider the $(m + 1)/2$ pairs

$$\{c_0, c_m\}, \{c_1, c_{m-1}\}, \ldots, \{c_{(m-1)/2}, c_{(m+1)/2}\}.$$

Now suppose that A and B are such that they both contain exactly one element from each of these pairs. It is straightforward to see that a balanced symmetric $(m, 1)$-LUT results. There are $2^{(m+1)/2}$ such partitions, and the desired bound follows.

Now suppose m is even. In this case let

$$A = \{c_0, c_2, \ldots, c_m\} \quad \text{and} \quad B = \{c_1, c_3, \ldots, c_{m-1}\}$$

or vice versa. It is again straightforward to see that both partitions result in balanced symmetric $(m, 1)$-LUTs. The desired bound again follows immediately. □

Note that there do exist examples of balanced, symmetric $(m, 1)$-LUTs not included in the families of Theorem 3.6.5. Two such examples (in fact, the only examples known to the author) are for an $(8, 1)$-LUT and a $(13, 1)$-LUT. In these cases we can achieve the desired balance and symmetry by letting the sets A and B be defined as follows.

For an $(8, 1)$-LUT let

$$A = \{c_0 = 1, c_3 = 56, c_4 = 70, c_8 = 1\}$$

and

$$B = \{c_1 = 8, c_2 = 28, c_5 = 56, c_6 = 28, c_7 = 8\}$$

or any of the eight obvious variants of the above.

For a $(13, 1)$-LUT let

$$A = \{c_0 = 1, c_1 = 13, c_2 = 78, c_3 = 286, c_6 = 1716, c_7 = 1716, c_{10} = 286\}$$

and

$$B = \{c_4 = 715, c_5 = 1287, c_8 = 1287, c_9 = 715, c_{11} = 78, c_{12} = 13, c_{13} = 1\}$$

or any of the 16 obvious variants of the above.

It is interesting to speculate whether further sporadic examples of balanced, symmetric (m, n)-LUTs may exist, in particular whether or not examples exist for $n > 1$. Brüer tabulates the number of balanced symmetric $(m, 1)$-LUTs for all odd $m \leq 17$ and obtains

$$bs_{m,1} = 2^{(m+1)/2}, \qquad m \text{ odd}, \quad m \leq 17, \quad m \neq 13,$$

and

$$bs_{13,1} = 144$$

which confirms that the above "sporadic" examples of balanced, symmetric $(m, 1)$-LUTs are the only such examples for odd m less than or equal to 17.

We now consider which of the examples in the proof of Theorem 3.6.5 satisfy C2–C4. We first consider C3, the nondegeneracy condition. We already observed that the only $(m, 1)$-LUTs which satisfy C5 and do not satisfy C3 are the trivial functions $L = 0$ and $L = 1$. Neither of these are balanced and hence we have:

Corollary 3.6.6. *The number $bns_{m,1}$ of balanced, nondegenerate, symmetric $(m, 1)$-LUTs, i.e., the number of $(m, 1)$-LUTs satisfying $C1$, $C3$, and $C5$, satisfies*

$$bns_{m,1} \geq 2^{(m+1)/2} \qquad \text{if } m \text{ is odd},$$

$$bns_{m,1} \geq 2 \qquad \text{if } m \text{ is even}.$$

We also observed above that the only $(m, 1)$-LUTs which satisfy C5 and do not satisfy C2 are the trivial functions $L = 0$ and $L = 1$, and the two functions L_1, L_2 having multinomial equations:

$$L_1 = x_1 + x_2 + \cdots + x_m$$

and

$$L_2 = x_1 + x_2 + \cdots + x_m + 1.$$

Unfortunately both L_1 and L_2 are balanced. If we let A_i denote the set of m- vectors which L_i maps onto 0 ($i = 1, 2$), then A_1 contains all the m-vectors of even weight and A_2 contains all the m-vectors of odd weight. Therefore, for m even, L_1 and L_2 correspond to both the examples of Theorem 3.6.5, and for m odd, L_1 and L_2 correspond to two of the $2^{(m+1)/2}$ examples. We therefore have:

Corollary 3.6.7. *The number $bans_{m,1}$ of balanced, nonlinear, nonaffine, nondegenerate, symmetric $(m, 1)$-LUTs, i.e., the number of $(m, 1)$-LUTs satisfying $C1$–$C3$ and $C5$, satisfies*

$$bans_{m,1} \geq 2^{(m+1)/2} - 2 \qquad \text{if } m \text{ is odd}.$$

We conclude by considering C4 in conjunction with C1 and C5. Of the examples given in the proof of Theorem 3.6.5, the only ones which are obviously uncorrelated are the two which do not satisfy C2, i.e., L_1 and L_2 (in the above notation). Therefore there are no obvious candidates for $(m, 1)$-LUTs which satisfy all of C1–C5. Indeed, there may well not be any such functions; this is a matter for future research.

In any case, it should be clear from this discussion that C1–C5, when taken together, are too restrictive. While the need for C1–C3 (or something like them) is difficult to dispute, the strict versions of C4 and C5 require some relaxation. Indeed, it is not clear how useful constraint C5 is for stream-cipher applications.

Acknowledgments

The author would like to acknowledge the hospitality and support of the École Nationale Supérieure des Télécommunications, Paris, during September and October 1988, where some of the work described in this paper was undertaken. The author would also like to thank an anonymous referee for suggesting simpler formulations of Theorem 3.1.1 and Lemma 3.2.1, and also giving other suggestions for improvement. Thanks are also due to Hewlett-Packard Ltd., who gave permission for the publication of this paper.

References

[1] D. Andelman and J. Reeds, On the cryptanalysis of rotor machines and substitution–permutation networks, *IEEE Transactions on Information Theory*, **28** (1982), 578–584.

[2] F. Ayoub, Trapdoors, random structures, and encryption functions design, Paper presented at the IEE Colloquium on Techniques and Implications of Digital Privacy and Authentication Systems, London, 1981.

[3] F. Ayoub, Probabilistic completeness of substitution–permutation encryption networks, *Proceedings of the IEE (Part E)*, **129** (1982), 195–199.

[4] M. Beale and M. F. Monaghan, Encryption using random Boolean functions, in: *Proceedings of the IMA Conference on Cryptography and Coding, Cirencester, December 1986*, Oxford University Press, Oxford, 1989, pp. 219–230.

[5] H. J. Beker and F. C. Piper, *Cipher Systems*, Van Nostrand, Wokingham, 1982.

[6] E. F. Brickell, J. H. Moore, and M. R. Purtill, Structure in the S-boxes of the DES, in: *Advances in Cryptology: Proceedings of Crypto 86*, Springer-Verlag, Berlin, 1987, pp. 3–8.

[7] J. -O. Brüer, On pseudorandom sequences as crypto generators, in: *Proceedings of the 1984 International Zurich Seminar on Digital Communications*, IEEE, pp. 157–161. New York, 1984.

[8] H. Cloetens, Y. Desmedt, L. Bierens, J. Vandewalle, and R. Govaerts, Additional properties in the S-boxes of the DES, Paper presented at Eurocrypt 86.

[9] D. W. Davies and W. L. Price, *Security for Computer Networks*, Wiley, Chichester, 1984.

[10] M. Davio, Y. Desmedt, and J. -J. Quisquater, Propagation characteristics of the DES, in: *Advances in Cryptology: Proceedings of Crypto 84*, Springer-Verlag, New York, 1985, pp. 62–73.

[11] Y. Desmedt, Analysis of the Security and New Algorithms for Modern Industrial Cryptography, Doctoral dissertation, Katholieke Universiteit, Leuven, October 1984.

[12] Y. Desmedt, J. -J. Quisquater, and M. Davio, Dependence of output on input in DES: Small avalanche characteristics, in: *Advances in Cryptology: Proceedings of Crypto 84*, Springer-Verlag, New York, 1985, pp. 359–376.

[13] P. Erdös and I. Kaplansky, The asymptotic number of Latin rectangles, *American Journal of Mathematics*, **68** (1946), 230–236.

[14] H. Feistel, Cryptography and computer privacy, *Scientific American*, **228** (1973), 15–23.

[15] R. Forré, The Strict Avalanche Criterion: Spectral properties of Boolean functions and an extended definition, Paper given at Crypto 88.

[16] J. A. Gordon and H. Retkin, Are big *S*-boxes best?, in: *Cryptography: Proceedings of Burg Feuerstein* 1982, Springer-Verlag, New York, 1983, pp. 257–262.

[17] M. Hall, Jr., *Combinatorial Theory* (2nd edition), Wiley, New York, 1986.

[18] J. B. Kam and G. I. Davida, A structured design of substitution–permutation encryption network, in: *Foundations of Secure Computation*, Academic Press, New York, 1978, pp. 95–113.

[19] J. B. Kam and G. I. Davida, Structured design of substitution–permutation encryption networks, *IEEE Transactions on Computers*, **28** (1979), 747–753.

[20] S. A. Lloyd, Counting functions satisfying a higher order strict avalanche criterion, *Proceedings of Eurocrypt 89, Houthalen, Belgium*, to appear.

[21] S. A. Lloyd, Balance, uncorrelatedness and the strict avalanche criterion, submitted.

[22] S. A. Lloyd, Characterising and counting functions satisfying the strict avalanche criterion of order $(n - 3)$, *Proceedings of the Second IMA Conference on Cryptography and Coding, Cirencester, 1989*, to appear.

[23] H. Minc, *Permanents*, Cambridge University Press, Cambridge, 1984.

[24] H. Minc, Theory of permanents, 1978–1981, *Linear and Multilinear Algebra*, **12** (1983), 227–263.

[25] C. T. Retter, A key-search attack on MacLaren-Marsaglia systems, *Cryptologia*, **9** (1985), 114–130.

[26] R. A. Rueppel, Correlation immunity and the summation generator, in: *Advances in Cryptology: Proceedings of Crypto* 85, Springer-Verlag, New York, 1986, pp. 260–272.

[27] R. A. Rueppel, *Analysis and Design of Stream Ciphers*, Springer-Verlag, New York, 1986.

[28] A. Shamir, On the security of DES, in: *Advances in Cryptology: Proceedings of Crypto 85*, Springer-Verlag, New York, 1986, pp. 280–281.

[29] C. E. Shannon, Communication theory of secrecy systems, *Bell System Technical Journal*, **28** (1949), 656–715.

[30] T. Siegenthaler, Correlation-immunity of nonlinear combining functions for cryptographic applications, *IEEE Transactions on Information Theory*, **30** (1984), 776–780.

[31] T. Siegenthaler, Decrypting a class of stream ciphers using ciphertext only, *IEEE Transactions on Computers*, **34** (1985), 81–85.

[32] T. Siegenthaler, Design of combiners to prevent divide and conquer attacks, in: *Advances in Cryptology: Proceedings of Crypto 85*, Springer-Verlag, New York, 1986, pp. 273–279.

[33] T. Siegenthaler, Correlation-immune polynomials over finite fields, Paper given at Eurocrypt 86.

[34] A. F. Webster and S. E. Tavares, On the design of *S*-boxes, in : *Advances in Cryptology: Proceedings of Crypto 85*, Springer-Verlag, New York, 1986, pp. 523–534.

[35] K. Yamamoto, On the asymptotic number of Latin rectangles, *Japan Journal of Mathematics*, **21** (1951), 113–119.

[36] K. Yamamoto, On the number of Latin rectangles, *Science Report, Tokyo Women's Christian College Journal of Mathematics*, 7–**10** (1969), 86–97.