DESIGNING S-BOXES FOR CIPHERS RESISTANT TO DIFFERENTIAL CRYPTANALYSIS (Extended Abstract)

CARLISLE M. ADAMS

Bell-Northern Research, Ltd., P.O. Box 3511 Station C, Ottawa, Ontario, Canada, KI Y 4117

STAFFORD E. TAVARES

Dept. of Electrical Engineering, Queen's University, Kingston, Ontario, Canada, K7L 3N6

Abstract - This paper examines recent work in the area of bent-function-based substitution boxes in order to refine the relationship between s-box construction and immunity to the differential cryptanalysis attack described by Biham and Shamir. It is concluded that $m \times n$ s-boxes, $m < n$, which are *partially* bent-function-based are the most appropriate choice for private-key cryptosystems constructed as substitution-permutation networks (SPNs)[1]. Since s-boxes of this dimension and with this property have received little attention in the open literature, this paper provides a description of their construction and shows how they can be incorporated in a design procedure for a family of SPN cryptosystems with desirable cryptographic properties.

1. INTRODUCTION

In [1] the relationship between substitution box (or s-box) construction and immunity against Biham and Shamir's differential cryptanalysis [6] of DES-like cryptosystems was discussed. It was stated that s-boxes with a flat distribution of so-called "Output XORS" would be immune to this attack[2] and it was proven that bent-function-based s-boxes are guaranteed to possess this flat distribution.

Since the initial submission of [1] (Jan., 1991), two researchers have published results which relate directly to this work. In [11b] Nyberg answers a question left open in the discussion at the end of [1], and in [5] Biham offers a counter-example which shows that for a particular

[1] Note that Nyberg and Knudsen [11c] have recently proposed a DES-like cipher with provable security against differential cryptanalysis. However, that system differs from ours in that it does not use s-boxes and at each round maps m bits to n bits where $m \geq n$.

[2] This was also (independently) stated by Dawson and Tavares [7] and by Nyberg [11b].

class of s-boxes the flat Output XOR distribution is not sufficient to guarantee immunity against differential cryptanalysis. In this paper we discuss these results in light of our previous work in order to refine the relationship between s-box construction and Biham and Shamir's attack. We then describe a design procedure for substitution-permutation networks (SPNS) which incorporates these higher-immunity s-boxes and leads to a family of cryptosystems which are conceptually simple, easily implemented, and very efficient in terms of encryption/decryption speed.

2. S-Box CONSTRUCTION BASED ON BENT FUNCTIONS

In [1] the following theorem was proven:

Theorem 1: An mxn s-box S represented as a $2^m \times n$ binary matrix with columns ϕ_i will have equiprobable output XORs if there exist min(m,n) ϕ_i such that all nonzero linear combinations (modulo 2) of these ϕ_i correspond to bent functions.

We call s-boxes which satisfy the above theorem "bent-function-based s-boxes" (details regarding bent functions can be found in [15, 20, 3, 11a, 13], for example). Although we had been looking specifically at mxn s-boxes (m << n) which were only partially bent-function-based, we stated at the end of [1] that it would also be interesting to determine whether 6x4 s-boxes satisfying the above theorem can be constructed (note that 6x4 is the size of the s-boxes used in the Data Encryption Standard (DES)).

Nyberg has looked at constructing bent-function-based (which she calls "perfect nonlinear") s-boxes and has shown that the above question can be answered in the negative (i.e., there are no bent-function-based 6x4 s-boxes). In fact, Nyberg's result is much more general: for an mxn s-box to be bent-function-based, m must be at least twice as big as n. This means that if mxn s-boxes are required for a cryptosystem where m<2n, then bent-function-based s-boxes cannot be used; instead s-boxes which partially satisfy Theorem 1 above must be used for the cryptosystem (such s-boxes are discussed briefly in [1] and will be discussed further in Section 3).

2.1. Weakness in bent-function-based s-boxes for m>n

Although Nyberg's result states that mxn bent-function-based s-boxes only exist for m≥2n, recent work by Biham [5] can be generalized to show that bent-function-based s-boxes would have a weakness for any m>n. In particular, Biham has observed that if 6x4 bent-function-based s-boxes were to be used in DES, then DES could be broken using approximately 2^{30} chosen plaintext pairs. This is because such s-boxes would, by definition, have equiprobable Output XORs (the value 4 for all entries in the XOR table except those corresponding to a zero Input XOR), meaning that "the Input XORs which modify only private input bits of the s-boxes (which are not replicated to two s-boxes) may cause zero Output XOR with probability 4/64 = 1/16" [5]. Biham therefore derived a 2-round characteristic[3] with this probability of success; such a characteristic, when iterated over the full 16 rounds of DES, leads to its cryptanalysis.

Although this is not stated by Biham, it is easily seen that Biham's result is not restricted to 6x4 s-boxes or to DES. Any s-box with m>n has more input vectors than output vectors; this necessarily means that there will be at least one case where two or more inputs are mapped to the same output (i.e., where one or more Input XORs have an Output XOR of zero). The weakness with having such an s-box be bent-function-based is that such cases have a fixed, non-negligible probability of occurrence (such as the 1/16 mentioned above) which may be exploited in a characteristic and used in cryptanalysis (this would be true for any DES-like cryptosystem). If m≤n this avenue of attack is typically not available because each input is mapped to a unique output (so that an Input XOR never has an Output XOR of zero). Note that the 6x4 s-boxes defined for DES are not bent-function-based and so the simple characteristic described by Biham cannot be used against DES (instead, more sophisticated characteristics must be employed - see [6]).

2.2. Remarks on s-box construction

Recent work by Nyberg and by Biham allows the relationship between s-box construction and immunity to differential cryptanalysis to be further clarified. Although a flat Output XOR distribution avoids differential cryptanalysis (by removing high-probability Output XORS, which may be exploited), and although bent-function-based s-boxes have the ideal Output XOR distribution,

---

[3] A *characteristic* is defined as an (r+l)-tuple $\Omega(\Delta X, \Delta Y_1, \Delta Y_2, ..., \Delta Y_r)$, where ΔX is a plaintext difference and the ΔY_i are the ciphertext differences at each of r consecutive rounds of an R-round cipher, R≥r [12].

it turns out that this can be a weakness for s-boxes with more input bits than output bits. Since bent-function-based mxn s-boxes only exist for m≥2n, it seems that without extra precautions this may be too strict a requirement for s-box design[4].

The conclusion to be drawn from this work is that the best approach in s-box design may be to construct mxn s-boxes (m≤n) which are *partially* bent-function-based but which still satisfy other properties which have been deemed to be necessary ill the literature (see, for example, [2, 7, 10, 12, 17, 19]) - these are discussed in more detail in Section 3 below. This should provide resistance to differential cryptanalysis by ensuring that there are no high-probability Output XORs in the s-boxes which may be exploited in an attack (even though the Output XOR distribution is not perfectly flat). Note that such s-boxes require a slightly different approach to SPN cryptosystem design since s-boxes in the open literature have generally used m≥n; this approach is the subject of the remainder of this paper.

3. DESIGN FOR A FAMILY OF SUBSTITUTION-PERMUTATION NETWORK CRYPTOSYSTEMS

The design for a family of SPN cryptosystems requires two distinct phases: the design of suitable s-boxes; and the design of the overall algorithm structure itself. These phases are discussed in the following subsections.

3.1. mxn s-box design

As was stated previously, mxn (m≤n) s-boxes which are partially bent-function-based should provide resistance to differential cryptanalysis. We define an s-box S to be *partially bent- function-based* if the columns of the binary matrix representing S are bent, but at least one of the nonzero linear combinations of these columns (summed modulo 2) is not bent[5]. For this design

---

[4] It is quite conceivable that bent-function-based s-boxes can still be used in substitution-permutation network design, but other measures would have to be used to avoid differential cryptanalysis (such as having no input bits to an s-box which are not replicated to other s-boxes, or using the ideas discussed in [17] with respect to "equivalence classes of s-boxes", for example).

[5] Recall the s-box representation: an mxn s-box is represented as a 2^mxn binary matrix M where each column is a vector which corresponds to a Boolean function of the m input variables and which defines the

let n be an integer multiple of m; in particular, let n=r*m where r>l. Such s-boxes can be constructed as follows. Choose n binary bent vectors ϕ_i of length 2^m such that linear combinations of these vectors sum (modulo 2) to highly nonlinear vectors (Nyberg's work shows that these linear combinations cannot all be bent since m<2n; however, it is important that they be nonlinear or the resulting s-box will not satisfy the Output Bit Independence Criterion[6]). Furthermore, choose half of the ϕ_i to be of weight $(2^{m-1} + 2^{(m/2)-1})$ and the other half to be of weight $(2^{m-1} - 2^{(m/2)-1})$; these are the two weights possible for binary bent vectors of length 2^m. Set the n ϕ_i to be the columns of the matrix M representing the s-box.

Check that M has 2^m distinct rows (this is not guaranteed) and that the Hamming weight of each row and the Hamming distance between pairs of rows is close to n/2 (i.e.., that the set of weights and the set of distances each have a mean of n/2 and some suitably small - but nonzero - variance)[7]. If these conditions are not satisfied, continue choosing suitable bent vectors (i.e., candidate ϕ_i) and checking the resulting matrix until the conditions are satisfied. Note that

- requiring each row to have approximately half zeros and half ones ensures that the s-box will provide good "confusion" [16, 8],
- requiring the sum (modulo 2) of any pair of rows to have approximately half zeros and half ones ensures that the s-box will provide good avalanche [8, 10], and
- using bent vectors as the matrix columns ensures that each output bit will respond "ideally" (in the sense of highest-order Strict Avalanche Criterion [2, 4][8]) to changes in the input vector

---

response of a single output bit to any given input. Row i of M, $1 \le i \le 2^m$, is therefore the n-bit output vector which results from the i^{th} input vector.

[6] The (output) Bit Independence Criterion (BIC) states that s-box output bits j and k should change independently when any single input bit i is inverted, for all i, j, k [18, 19].

[7] Note that this is impossible if $m \ge n$ but is quite feasible if $2^m \le C(n,n/2)$, where C(a,b) is "a choose b".

[8] This has independently been called the Propagation Criterion of degree n in [13].

(further discussion of the above points will be given in the full paper). In other words, these s-boxes appear to be perfectly suited for the approach to SPN cryptosystem design given below.

3.2. SPN cryptosystem design

A design for substitution-permutation network cryptosystems which incorporates mxn s-boxes where m<< n is as follows. Let n = r*m (where r is an integer greater than 1) and let 2n be the block size of the cryptosystem. The general framework of the algorithm is identical to the Data Encryption Standard: the plaintext is initially broken into halves of length n; at each round, one half is modified, is added modulo 2 to the other half, and the two halves are interchanged; after R rounds, the two halves are concatenated to form the ciphertext. However, the modification of a message half at each round is implemented completely differently from DES: here r+s mxn s-boxes from separate compatibility classes[9] are used, where each s-box is constructed according to the procedure given in Section 3. 1.

The (keyed) message half modification at each round is quite straightforward and is accomplished as follows. The n-bit message half is broken into r m-bit pieces and the subkey for that round is broken into s m-bit pieces. Each of the r+s pieces is input to a separate mxn s-box and the n-bit outputs are summed modulo 2 to form the n-bit modified message half. (Optionally, other key bits may be used to mask the r m-bit pieces of the message before they are input to the s-boxes, as is done in DES, but this may not be required.) Note that although each s-box necessarily causes data expansion (since m<n), using the set of s-boxes in this way results in *no expansion of the message half* and so the SPN has input and output blocksizes which are equal (an important consideration in some applications). Furthermore, the use of bent functions which are nonlinearly related guarantees *"ideal" behaviour from any s-box: any* change in the m input bits (from single-bit, to multi-bit, to total complementation) causes each of the n output bits to change with probability 1/2, virtually independently of all the other output bits[10] . Therefore, on average,

---

[9] A *compatibility relation* on a set is a relation which has the reflexive and symmetric properties. We define two s-boxes to be in the same compatibility class if they have one or more Boolean functions in common, or if a Boolean function in one is a linear modification of a Boolean function in the other (its bitwise complement, for example). The importance of compatibility classes in s-box construction is given in [2] and will be discussed in the full version of this paper.

[10] Note that the relationship between columns of the matrix M representing an s-box determines the performance of that s-box on the Output Bit Independence Criterion (BIC) [18, 19]. In particular, if col-

approximately half of the output bits will change and because the s-box outputs are summed modulo 2, this means that approximately half of the bits in the modified message half will be inverted. This remains true regardless of how many s-boxes have their inputs modified (since randomly changing approximately half the bits in any nonzero set of binary vectors randomly changes approximately half the bits in their modulo 2 sum) and so it is clear that this SPN should exhibit statistically ideal highest-order SAC and BIC. Note finally that this design eliminates the need for permutation layers. In other SPN designs, permutations are essential because s-boxes are small relative to the size of the message half; permutations serve to spread the avalanche from a single s-box over the block or message half so that these changes are input to several s-boxes in the next round. In this design, each s-box affects the entire message half directly and so changes are guaranteed to affect many s-boxes in the next round without a permutation layer.

Keying is, of course, a crucial aspect of cryptosystem design. A key schedule is required that provides some guarantee of key/ciphertext Strict Avalanche Criterion and Bit Independence Criterion; that is, each ciphertext bit should depend in a complicated, nonlinear way on every key bit and any change in the key should cause each ciphertext bit to change with probability one half, virtually independently of the other ciphertext bits. On the other hand, we would like the key schedule to be as simple as possible, in keeping with our objective of design simplicity for the entire network. Our approach to key scheduling is as follows. Let the keysize be equal to the blocksize of 2n bits. At each round, as described above, take s*m key bits, use these as the inputs to s mxn s-boxes, and add the s n-bit outputs to the modulo 2 sum of the r n-bit outputs from the message half. Because of the properties of the s-boxes, any changes to the key bits at any round should randomly change approximately half the bits in the modified message half at that round. Note that the s s-boxes used for keying should be from separate compatibility classes and

---

umns ϕ_j and ϕ_k sum modulo 2 to a linear vector, then s-box output bits j and k will either always change together or never change together when any input bit i is inverted (i.e., they will have a correlation coefficient of ±1). At the other extreme, if ϕ_j and ϕ_k sum to a bent vector, then j and k will change independently for any input change. Because it is impossible for all column sums to be bent (since m<2n), we are suggesting the use of s-boxes which are partially bent-function-based, where one or more of the column sums is nonlinear but not bent. Using the highest (non-bent) nonlinearity possible ensures that bits j and k will act "virtually" independently (i.e., will have a correlation coefficient which is nonzero, but as small as possible), for all input changes.

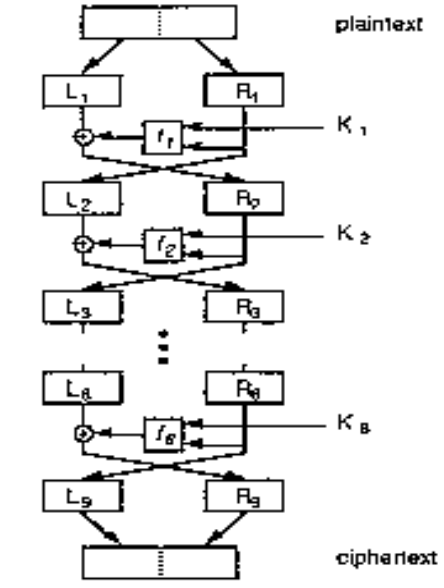
furthermore should be in separate classes from the other r s-boxes used in message half modification (so that there is no guaranteed cancellation of s-box outputs). Note as well that the s*m key bits selected in round i should be different from the s*m key bits selected in round i+l (this is due to the work of Grossman and Tuckerman [9], who showed that DES-like cryptosystems without a rotating key can be broken). Note finally that if any key bit is used in round R (the last round) for the first time then the network fails the key/ciphenext completeness test, since complementing that bit can only affect a single message half. All key bits must therefore be used by round R-1; in fact, we recommend that they be used by round R/2 and reused, possibly in a different order, in the lower half of the network (this ensures good key avalanche for both encryption and decryption).

3.3. An example SPN cryptosystem

We have constructed an example network based on the design procedure of this section to illustrate how it may be used. This network has a blocksize and keysize of 64 bits, uses six 8x32 s-boxes S_1 ... S_6, uses 16 bits of key in each round, and uses 8 rounds. At round i, the message half modification block f_i takes as input the four bytes of R_i and two bytes of the key:

$$f_i(R_i, K_i) = \sum S_j(R_{i,j}) \oplus S_5(K_{i,1}) \oplus S_6(K_{i,2})$$

where $R_{i,j}$ is the j^{th} byte of R_i and the summation is from j =1 ... 4.



Example SPN Cryptosystem

The key schedule is as follows:

Key at round i	Byte of input key K
K1	k1,k2
K2	k3,k4
K3	k5,k6
K4	k7,k8
K5	k4,k3
K6	k2,k1
K7	k8,k7
K8	k6,k5

This key scheduling algorithm has some features worth noting. - Firstly, each key byte is used twice, once in the first four rounds and once in the last four rounds. Secondly, each key byte is used once as input to s-box S_5 and once as input to s-box S_6, so that no key byte gives the same result in two different rounds. Thirdly, no key bytes are used in two consecutive rounds. Fourthly, the key bytes in the last four rounds are not the reverse order of the key bytes ill the first four rounds (which may have facilitated some sort of "meet-in-the-middle" attack). Finally, it is clear from the above diagram that the left half of the initial plaintext is modified by f_1, f_3, f_5, and f_7 and the right half is modified by f_2, f_4, f_6, and f_8. The key schedule is such that every key byte is used as input to f_i, i odd, and every key byte is used as input to f_i, i even. If encryption is performed with the 64-bit (8-byte) key $K_{enc} = k_1k_2k_3k_4k_5k_6k_7k_8$, then decryption is identical to encryption with the key $K_{dec} = k_6k_5k_8k_7k_2k_1k_4k_3$. For such a system, then, we suggest that keys be chosen such that $k_1=k_6$, $k_2=k_5$, $k_3=k_8$, and $k_4=k_7$ do not all simultaneously hold (otherwise double encryption will result in the original plaintext again). Furthermore, so that the subkey is not the same in two consecutive rounds (i.e., $K_i \neq K_{i+1}$, for i=1,2,...,7), it is important that $K_i \neq K_{i+2}$ for i=1,2,...,6. Although there are 2^{64} valid keys, we may think of the above conditions as leading to possibly weak keys; this results in slightly less than 2^{64} "recommended" keys, but a larger keyspace can easily be provided by using, for example, 12 key bytes (3 in each round).

As with any SP network, the security of this system rests primarily in its s-boxes. We have outlined the required properties for each s-box, but there are requirements on the set of s-boxes as a whole as well. Because of the way that s-box outputs are combined it is important that very few (ideally none) of the s-box outputs be repeated within the set. For our example network, this means that ideally there should be 6x256 unique n-bit outputs. Furthermore, the property that outputs should be pairwise a Hamming distance of approximately n/2 bits apart should hold not only within each s-box but between s-boxes as well so that at each round i, L_i is well randomized by the message half modification block f_i.

Note that the example cryptosystem given here can be modified in several simple ways for applications where this is desired. For example, a different key length can be used with a different key scheduling algorithm (note that any extra key bytes may go into extra s-boxes, or may be used to mask message bytes in the message half modification block f_i). As well, the blocksize can be changed to any (even) value. Finally, the number of rounds can be changed in the network to vary encryption/decryption speed. This design procedure truly represents a family of cryptosystems, like RSA [14] and several other systems - but unlike DES - because the parameters are so flexible (extending -DES from 64 bits (with a 56-bit key) to 128 bits (with a 112-bit key) requires some effort but is not prohibitive; however, extending it to 80 or 96 bits (with any key length) would necessitate a re-design of major portions of the algorithm). This suggests that the design given above, if it is secure, would be appropriate for a wide variety of applications.

3.4. Remarks on SPN cryptosystem design

The design procedure suggested in this section has some interesting features, such as a theoretical framework (because of the use of bent vectors), guaranteed higher-order SAC and BIC (for both plaintext/ciphertext and key/ciphertext pairs), feasibility (since bent vectors of the necessary size can be easily generated; see [11a, 13, 3], for example), speed (since bit permutations are avoided), and simplicity (since all phases of the design are defensible and readily understandable). Furthermore, it can easily be implemented in both software and hardware. This design therefore looks promising and, as always, investigation/analysis by the cryptographic community is encouraged.

We have done some preliminary analysis on the example system given in Section 3.3 and found not only that a (non-optimized) implementation in a high-level language is three to four times

faster than a (somewhat optimized) Assembly-language version of DES, but also that its performance is comparable to DES on a variety of both simple and relatively sophisticated statistical tests (see [2] for further details).

4. CONCLUSIONS

This paper has examined the relationship between substitution boxes in private-key cryptosystems and resistance to the Differential Cryptanalysis attack described by Biham and Shamir. Recent work by Nyberg and by Biham can be interpreted in light of our own previous work to show that the requirement that s-boxes be bent-function-based is too strong (in that, without extra precautions, systems incorporating such s-boxes are vulnerable to differential cryptanalysis). This leads to the conclusion that s-boxes which have fewer input bits than output bits and which are partially bent-function-based may be preferable for these systems. Since such s-boxes have received little attention in the open literature, this paper then focused on a design procedure for these s-boxes and for a family of Substitution-Permutation Network cryptosystems which incorporates them. The SPN design procedure given is conceptually simple, is easily implemented, is very efficient, and appears to produce ciphers with good cryptographic properties (including resistance to differential cryptanalysis because of the component s-boxes). This does not, of course, guarantee that the procedure is secure or suggest that its immediate use is recommended. Rather it lends some credibility to the design procedure and may serve to encourage more extensive examination by both expert and amateur cryptanalysts.

REFERENCES

[1] C. M. Adams, *On immunity against Biham and Shamir's "differential cryptanalysis"*, Information Processing Letters, vol.41, Feb. 14, 1992, pp.77-80.

[2] C. M. Adams, A *Formal and Practical Design Procedure for Substitution Permutation Network Cryptosystems*, Ph.D. Thesis, Department of Electrical Engineering, Queen's University, 1990.

[3] C. M. Adams and S. E. Tavares, *Generating and Counting Binary Bent Sequences*, IEEE Transactions on Information Theory, vol. IT-36, 1990, pp. 1170-1173.

[4] C. M. Adams and S. E. Tavares, *The Use of Bent Sequences to Achieve Higher-Order Strict Avalanche Criterion in S-Box Design*, Technical Report TR 90-013, Dept. of Electrical Engineering, Queen's University, Kingston, Ontario, Jan., 1990.

[5] E. Biham, *Differential Cryptanalysis of Iterated Cryptosystems*, Ph.D. Thesis, Weizmann Institute of Science, Rehovot, Israel, 1992.

[6] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-Like Cryptosystems*, Journal of Cryptology, vol.4, 1991, pp. 3-72.

[7] M. Dawson and S. E. Tavares, *An Expanded Set of S-Box Design Criteria Based on Information Theory and its Relation to Differential-Like Attacks*, in Advances in Cryptology: Proc. of Eurocrypt'91, Springer-Verlag, 1991, pp. 352-367.

[8] H. Feistel, *Cryptography and Computer Privacy*, Scientific American, vol.228, 1973, pp. 15-23.

[9] E. Grossman and B. Tuckerman, *Analysis of a Feistel-Like Cipher Weakened by Having No Rotating Key*, Technical Report RC 6375, IBM, 1977.

[10] J. B. Kam and G. 1. Davida, *Structured Design of Substitution-Permutation Encryption Networks*, IEEE Transactions on Computers, vol. C-28, 1979, pp.747-753.

[11a] K. Nyberg, *Constructions of bent functions and difference sets*, in Advances in Cryptology: Proc. of Eurocrypt '90, Springer-Verlag, 1991, pp.151-160.

[11b] K. Nyberg, *Perfect nonlinear S-boxes*, in Advances in Cryptology: Proc. of Eurocrypt '91, Springer-Verlag, 1991, pp.378-386.

[11c] K. Nyberg and L.R.Knudsen, *Provable Security Against Differential Cryptanalysis, in* Advances in Cryptology: Proc. of CRYPTO '92 (to appear).

[12] L. O'Connor, *An Analysis of Product Ciphers Based on the Properties of Boolean Functions*, Ph.D. Thesis, Dept. of Computer Science, University of Waterloo, 1992.

[13] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, *Propagation characteristics of boolean functions*, in Advances in Cryptology: Proc. of Eurocrypt '90, Springer-Verlag, Berlin, 1991, pp. 161-173.

[14] R. L. Rivest, A. Shamir, and L. Adleman, A *Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, vol. 21, 1978, pp. 120-126.

[15] O. S. Rothaus, *On "Bent" Functions,* Journal of Combinatorial Theory, vol. 20(A), 1976, pp.300-305.

[16] C. E. Shannon, *Communication Theory of Secrecy Systems,* Bell Systems Technical Journal, vol. 28, 1949, pp.656-715.

[17] M. Sivabalan, S. E. Tavares, and L. E. Peppard, *On the Design of SP Networks from an Information Theoretic Point of View,* in Advances in Cryptology: Proc. of CRYPTO '92 (to appear).

[18] A. F. Webster, *Plaintext/Ciphertext Bit Dependencies in Cryptographic Systems,* M.Sc. Thesis, Department of Electrical Engineering, Queen's University, 1985.

[19] A. F. Webster and S. E. Tavares, *On the Design of S-Boxes,* in Advances in Cryptology: Proc. of CRYPTO'85, Springer-Verlag, New York, 1986, pp.523-534.

[20] R. Yarlagadda and J. E. Hershey, Analysis and Synthesis of Bent Sequences, IEE Proceedings (Part E), vol. 136, 1989, pp. 112-123.