A Proposed Design For An Extended DES

by

Lawrence Brown,
Dept. of Computer Science,
University College, UNSW,
Australian Defence Force Academy,
Canberra. ACT. 2600.

6 November 1999

Abstract

The Data Encryption standard (DES) has achieved wide utilization, especially in the financial industry. Whilst DES is a standard, the design criteria used in its development have been classified by the US government. This paper reviews what is known about the design criteria for the S-boxes, P-boxes, and key scheduling in the current DES. It then indicates how this information could be used to design an extended scheme with a double length key. There are two main objectives in doing this. One is because of increasing doubts about the ability of DES to withstand an attack based on exhaustive key-space searches, using specialized hardware. The other is to develop an encryption scheme for which the design rules used are known, and hence open to analysis and criticism.

A Proposed Design For An Extended DES

1. Introduction

With the advent of computers, data encryption schemes underwent a revolutionary change. The use of computers enabled the cracking of hitherto secure schemes, but also enabled the implementation of schemes which until then were too complex for use by a manual cipherclerk. Based on the work of Shannon [Sha49], a new family of cryptosystems, known as **Substitution-Permutation networks** were developed. These led to the development of a system known as Lucifer at the IBM Research Laboratories [Fei73], and subsequently to the Data Encryption Standard (DES) [NBS77]. It has since achieved wide utilization, particularly in the banking and electronic funds transfer areas, where it is now an Australian standard [ASA85].

However, there was a considerable amount of controversy at the time of its introduction, primarily with the choice of a 56-bit key [DiH77], [Hel79]. This was on the grounds that such a key length was small enough to be exhaustively searched on machines which could be constructed in the foreseeable future. With the march of computer technology, this risk appears to be rising. However, on all other accounts DES appears to be an excellent cryptosystem. No short-cuts have been found to aid in the cryptanalysis of DES other than by exhaustive key-space search. With the current significant use of DES (especially in banking), there is interest in designing and building a DES-type cryptosystem with an extended key length, for several reasons. These include:

- Doubts about the ability of DES to withstand an attack based on exhaustive key-space searches on specialized hardware (cf. the decision of the US NBS to withdraw certification of DES for all but banking applications). This may be overcome by increasing the key length.
- The difficulty in obtaining DES systems outside the US, due to export restrictions.
- The desire to develop a scheme for which all the design criteria used are known, and hence open to criticism.

This paper will discuss the design criteria used in the DES, both those reported in the literature, and those noted by the author, and will illustrate how they will be applied in building an extended scheme.

2. Description of the DES

2.1. Fundamental Concepts

There are two basic cryptographic primitives: **substitution** and **transposition** (or permutation). In a substitution function, each character is replaced by some other character (see Fig 1). The particular substitution used is selected by some **key**. If characters are

Fig 1 - Sample S-box

represented by a small number of bits, then such a scheme may be defeated by enumerating all possible cases to find the substitution used. If made sufficiently large and a suitably complex function is used, such a scheme is in practice unbreakable, since it would require the enumeration of all $n!$ cases (where n is the number of possible characters). Unfortunately it is also unworkable, since the number of permutations, and hence keys, would be equally large.

In a transposition function, traditionally, a set of characters were permuted (rearranged) into a new order, again with the particular rearrangement selected by some key. In modern implementations, this becomes a permutation of bits or a wire-crossing (see Fig 2). Since the complexity of such an operation is only linear in the number of bits used, it may be easily broken by testing each bit in turn to find its new location. It also implies that it is easy to build such a function for large numbers of bits. Because of this, the concept of **product ciphers** arose, whereby a series of substitution and transposition functions are combined to form a function which is much

Fig 2 - Sample P-box

more secure than any of its components. Prior to the introduction of rotor machines, only simple product ciphers were possible. The development of rotor machines, and subsequently computers, changed this significantly.

Shannon [Sha49], in a key paper, described a particular class of product ciphers which used a number of small (and hence implementable) substitution boxes, termed S-boxes, connected by large permutation boxes, termed P-boxes (see Fig 3). He termed these **mixing transformations**, whereby the S-boxes provided **confusion** of the input, and the P-boxes provide **diffusion** of the S-box outputs across the inputs to the next stage. Such schemes are now termed **Substitution-Permutation (S-P) networks**. Two key properties of such S-P networks are the **avalanche** property, identified by Feistel [Fei73]; and the **completeness** property, identified by Kam and Davida [KaD79]. The avalanche property states that changing one input bit should result in approximately half of the output bits changing. The completeness property states that each output bit must be a complex function of every input bit. These properties appear to be important criteria for providing cryptographic strength in the resultant system. More formally, these may be defined (see [WeT86]) as follows. A function f is said to have a good avalanche effect if:

> for each bit i, $0 \le i < m$, one can divide the 2^m plaintext vectors into 2^{m-1} pairs X and X_i such that each pair differs only in bit i; and when forming the 2^{m-1} exclusive-or sums, termed avalanche vectors
>
> $$V_i = f(X) + f(X_i)$$
>
> find that about half of these sums are 1.

A function f is complete if:

> for each bit j, $0 \le j < m$ in the ciphertext output vector, there is at least one pair of plaintext vectors X and X_i, which differ only in bit i, for which $f(X)$ and $f(X_i)$ differ in bit j

For a practical scheme however, it is obviously necessary for the recipient to be able to decipher the message. This implies that one must be able to compute the inverse of the enciphering function. For the simple S-P network shown above, it would be necessary to provide a separate decryption stage incorporating the inverses of the substitutions and permutations used. This is expensive in terms of both hardware and software. It is preferable to be able to use the same hardware for both encryption and decryption. This may be done by an elegant scheme which partitions the input and output blocks into two halves, $L(i-1)$ and $R(i-1)$, and uses only $R(i-1)$ at each round i (see Fig 4). This half $R(i-1)$, along with a key $K(i)$, is used as input to the encryption function g (which incorporates one stage of the S-P network), and is also copied to $L(i)$ for the next round. The output from g is exclusive or'd (XOR'd) with $L(i-1)$ to form $R(i)$ for the next round. Thus when decrypting, $L(i)$ is copied back into $R(i-1)$ where it is used as input to g in order to recover $L(i-1)$. Thus it can be used to recover the other half of the input upon decryption.

Fig 3 - Sample S-P Network, with Avalanche Characteristic

Fig 4 - Invertible DES transform

2.2. Description of DES

The basic DES enciphering chain is shown in Fig 5. In brief, the input block is permuted by an initial permutation IP, is then subjected to a complex key-dependent computation, and is finally permuted by IP^{-1}, the inverse of the initial permutation. On deciphering, the same process is used, except that the complex component is run backwards, using keys in reverse order, via the method shown above. In more detail, the input block X of 64 bits is first permuted by the initial permutation[1] IP, and then divided into two halves $L(0)$ and $R(0)$. The complex key-dependent function consists of sixteen rounds in which the left component $L(i-1)$ is exclusive-or'd (XOR'd) with the output of a non-linear function g whose input is both the right component $R(i-1)$ and part of the key $K(i)$. The halves are then interchanged to form the input for the next round (with the exception of the final round). Finally the output is permuted by IP^{-1} (the inverse of the input permutation IP) to yield the output ciphertext Y.

The cryptographic security of DES rests, in the main, on the complex function g. It is shown in more detail in Fig 6. Its purpose is to ensure that the final DES output vector Y (see Fig 5.) is a complex nonlinear function of the input, such that there is no easy way of deriving the input knowing only the output, short of enumerating all possible keys. In detail, in round i, $1 \leq i \leq 16$, the right component of the message input $R(i-1)$ is first expanded from 32 to 48 bits via an expansion function $E(R(i-1))$. These are then XOR'd with a 48-bit component extracted from the key $K(i)$ to form vector A. This vector is partitioned into eight 6-bit blocks which are used as input to eight substitution boxes (S-boxes). Each S-box produces a 4-bit output, all of which are concatenated to form the vector B. This vector is permuted by P and then XOR'd with the left component $L(i-1)$ of the message. In more detail, each S-box may be regarded as being composed of four substitution functions (row

Fig 5 - DES Enciphering

[1] for details of the actual permutation used in IP (and all the other functions in the DES) see [ASA85], or any of the modern textbooks on cryptography.

Fig 6 - DES Function g

functions or rows). The (outer) two bits of the 6-bit input selects one of these four functions, which then operates on the (inner) 4 input bits to give a 4-bit output. This output is then permuted by P. Together these form one stage of an S-P network. The repetition of this procedure sixteen times forms the full S-P network.

The other critical component in the DES algorithm is the **key scheduling** stage. It is responsible for forming the sixteen 48-bit sub-keys used in the rounds of the encryption procedure. It is shown in Fig 7. This function is important since if the same key is used on successive rounds, it can weaken the resulting algorithm. This trap is mostly avoided with the specified key scheduling algorithm, except for a small number of keys known as **weak** or **semi-weak** keys (see [GrT78], [MeM82], [MoS87], [MoS86], and [ASA85]). In detail, the 64-bit key is permuted by $PC1$. This permutation performs two functions, first it strips the eight parity bits out, and then distributes the remaining 56 bits over two 28-bit halves $C(0)$ and $D(0)$. Subsequently these form two distinct paths. At each round, each 28-bit register is rotated left either one or two places according to the following schedule:

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Shift	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

After the shift, the resultant 28-bit vectors are permuted by $PC2$ (which in fact consists of two 28-bit permutations, not one 56-bit permutation), and are concatenated together to form the sub-key for that round. Note that the total number of shifts is 28; hence $C(16) = C(0)$, and $D(16) = D(0)$. When decrypting, it is necessary to generate the keys in reverse order. Because the final vector is the same as the initial vector, this can be done by rotating right, reading the table in reverse order (and with no shift before generating the first key). Thus, with minor changes, the same hardware is used for both encryption and decryption.

Fig 7 - DES Key Scheduling

3. Known DES Design Criteria

All the functions used in the DES are specified in tabular form. No description is provided on why they are chosen. In fact it is stated that the design criteria are classified. Nonetheless, some of the criteria have been released, and others may be deduced by examining the tables as given. The search for the design criteria is motivated by two reasons, first to aid in the cryptanalysis of the DES, and second in designing new and extended DES style schemes. The S-boxes and P-boxes will be examined independently.

3.1. S-box Design Criteria

The S-boxes in DES are the prime source of complexity in the algorithm, and are largely responsible for the security of the scheme. Consequently, considerable time has been spent on analysing them, and hence more is known about them than the P-boxes. The actual criteria used are classified secret. It has been stated[2] that there were 12 (possibly 13) criteria used, which resulted in about 1000 suitable S-boxes, of which the implementors chose 8. Some of the criteria used were reported to an NBS workshop on cryptography [BGK77], and are:

- 1 Each row function in an S-box is a permutation of the integers 0 to 15
- 2 No S-box is a linear or affine function of the input.
- 3 Changing 1 input bit to an S-box results in changing at least 2 output bits
- 4 $S(x)$ and $S(x + 001100_2)$ must differ in at least 2 bits (where $S(x)$ is the output of an S-box given input vector x).

The following rules were labelled (see [BGK77]) as resulting from design criteria:

- 5 $S(x) \neq S(x + 11ef00_2)$ for any choice of e or f.
- 6 The S-boxes were chosen to minimize the difference between the number of 1's and 0's in any S-

[2] by Alan G. Konheim at a Seminar series on Cryptography in Amsterdam, 1986 [Kon86]

box output when any single input bit is held constant.

These criteria and rules were analysed by Brickell et al. [BMP87]. Criterion 1 states, in effect, that each row of each S-box is a 1-1 function. If this were not true, then information would be lost, and recovery of the input would be impossible. Criterion 2 ensures that the DES algorithm as a whole is not linear or affine. If it were, it would substantially reduce the security of the scheme, and enable reasonably easy recovery of a key with a set of plaintext/ciphertext pairs. The next two criteria 3 and 4, seem to incorporate the concept of the avalanche effect, in that it ensures that small changes in the input result in large changes in the output. Rule 5, reduces the probability that two different inputs would be mapped onto the same output (nb: function g is neither 1-1 nor *onto*, it must only generate the same output from the same input and key); and that the image space (ie: the range of possible values generated by g) is as large as possible. Criteria 1, 3, and 4 seem to imply rule 6 (which accords with the original statement that it is a consequence of the design criteria).

Another consequence of design criteria was noted by Meyer and Matyas in their book [MeM82], namely that:

- 7 the S-boxes chosen required significantly more logical minterms[3] to implement than a random choice would require.

A random choice of S-box would lead to a distribution with a mean around 32. However Meyer and Matyas stated that as they added more design rules, the average number of minterms rose from around 46 in the early stages, to around 56 in the final design (with the distribution mostly ranging from 52 to 59). The final choice of 8 boxes was taken from the low end of this range (52 and 53) in order to make hardware implementation easier. This rule would indicate that the S-boxes are considerably more complex than a random choice would provide.

3.2. P-box Design Criteria

There are a number of permutation networks in the DES design, IP, IP^{-1}, E, P, $PC1$, and $PC2$. They appear to be performing various functions. No design criteria for them have been released. However by examination of the various networks, certain features can be deduced. In general these all appear to be motivated by the avalanche and completeness criteria, in that there is a wide dispersion of outputs across the inputs to the next stage. We will examine these networks in turn.

3.2.1. IP and IP^{-1}

These two permutations form the initial and final steps respectively in the enciphering chain for DES. It has been argued [DDF83] that these permutations have no cryptographic significance, since exhaustive plaintext searches can be conducted on messages after these transformations have been applied. This argument probably does not hold when DES is used in one of its chaining modes. Other reasons put forward for their inclusion are that hardware implementations would benefit, and that it might help thwart other forms of attack. It was also noted that IP is extremely regular. As is shown in [HGD84], if the input block is broken into 8-bit bytes, then it may be implemented as eight iterations of an 8-bit permutation {2,4,6,8,1,3,5,7} into eight shift registers: a much more compact hardware realization, than a true 64-bit permutation would imply. Its functionality seems to be to distribute bits from each input byte over all eight bytes in the input block. Also it sends all even numbered bits to the left hand vector, and odd numbered bits to the right hand vector.

3.2.2. $PC1$

This transformation performs a very similar function to IP, and additionally strips off the parity bits to reduce the 64-bit key value entered down to the 56-bit key actually used. It is nearly as regular as IP, except for an asymmetry around the two 28-bit halves. As with IP, it may be implemented with eight iterations of a 7-bit permutation {0,1,2,3,4,5,6,7} (the parity bit 8 is discarded). However now the permuted value is read out by shifting two 28-bit halves toward the centre, rather than to one end, as in IP (see [HGD84]).

3.2.3. P and E

It has been noted by Davies [Dav82] and Davio et al [DDF83], that P and E need to be analysed together as the functional composition $P.E$ (in fact $S.P.E$ should be analysed thus, being one round in an S-P network).

The expansion function E is very regular. If the 32-bit input block is broken into eight 4-bit blocks, then the 6-bit output block is formed by concatenating the adjacent bit from the neighboring blocks. If the 6 bits are labelled *abcdef* then bits *ab* and *ef* are repeated in the neighboring blocks (and include the S-box selector bits), whilst the *cd* bits are message bits only.

Now consider the composition $P.E$. The input of P is divided into the eight 4-bit outputs from the S-boxes, and the output from E is divided into the eight 6-bit

[3] a minterm is a logical AND (boolean product) of input bits (and their negations), which form a necessary (but not sufficient) condition for a particular output bit to be asserted. An output bit may have its value completely described by the logical OR (boolean sum) of all its minterms. The number of minterms in such an expression (after simplification) is a measure of its complexity. In the worst case, for n input bits, 2^n minterms may be needed to describe each output bit.

inputs to the S-boxes at the next stage. Examination of this network by the author revealed the following:

- 1 each bit in a given 6-bit output comes from a different 4-bit input
- 2 no bit in a given 6-bit output comes from the corresponding 4-bit input
- 3 in each 4-bit input block, two bits go to *ab* or *ef* bits (ie bits shared by two output blocks), the other two go to *cd* bits

These rules all appear consistent with the implementation of the avalanche and completeness effects. They ensure that every output bit becomes a function of each input bit in as few rounds as possible. Meyer [Mey78] (also in [MeM82]) has shown that with the networks specified in DES, and assuming S-box output bits are functions of all their input bits, then after 5 rounds, every output bit is a function of all input bits. This would appear to be an important consequence of the design criteria.

It is worth noting that both Davies [Dav82] and Davio et al [DDF83] have commented that the actual order of output bits from each S-box is irrelevant. This is because a functionly equivalent $S.P.E$ combination can be formed by rearranging the order of bits in each S-box (by rearranging columns), and then altering permutation P to compensate. As Hoornaert et al [HGD84] show, this is desirable in order to design an efficient hardware implementation which minimizes the number of wire crossings (and hence area) in P. It certainly illustrates that any analysis of these components needs to be performed on them as a whole.

3.2.4. $PC2$

The permutation $PC2$ is responsible for selecting 48 of the 56 key bits available on each round. The output of this permutation can be analysed in terms of the eight 6-bit blocks which become S-box inputs after being XOR'd with the expanded message block. As with the other components in the key scheduling stage, it can be regarded as two distinct 28 to 24-bit selections and permutations. There is no interaction across the two halves. The author has noted that in each 6-bit output block, if the bits are sorted into ascending order of their input bits, then:

- 1 no two bits may be spaced closer than 3 bits apart
- 2 the span from lowest to highest input bit number must be at least 22 (alternatively, the average spacing must be at least 3 2/3)

The first rule implies that the same bit of the key will not be used as input to the same S-box on successive rounds (since in the key schedule, bits are shifted only 1 or 2 places). The second results from application of the avalanche and completeness effects to ensure that the output depends on all key bits in as few rounds as possible. Meyer [Mey78] (also in [MeM82]) has shown that with the current DES each output bit will be a function of all key bits after 5 rounds.

3.3. Key Scheduling

Aside from the S-boxes and P-boxes, the other component critical to the security of DES is the **key scheduling** algorithm. Several authors [Ber82], [Dav82], [QDD86] have noted that DES in fact uses a 768-bit key (formed from a 48-bit key to each of 16 stages), which is generated from the 56-bit key supplied. If either the key scheduling is poor [GrT78], or a bad choice of key is made [Dav82], then the same subkey might be repeated for more than one round, leading to insecurities in the scheme. No design rules are known for the scheme currently in use. However the author notes that:

- 1 each bit is used as input to each S-box
- 2 no bit is used as input to the same S-box on successive rounds
- 3 the total number of bits rotated is 56 (which implies that $K(0) = K(16)$, enabling the decryption operation to use right shifts in reverse order).

Suggestions for alternate forms of key scheduling have been made in [Ber82] and [QDD86].

4. An Extended DES

Having analysed the existing DES, and determined as many of the design rules as possible, a proposed design of an extended DES style scheme may be outlined. In doing this it is proposed to extend the work of Morris-Yates [Mor86], who used a subset of the criteria detailed previously. It involves widening all the data paths to twice the current width, and extending or redesigning all the S-boxes, P-boxes, and key scheduling. The initial motivation for doing this was to overcome the doubts about the ability of DES to withstand an attack based on exhaustive key-space searches by doubling the key length. This raises the effort needed from 2^{55} trials to 2^{111} trials[4], an impossibility in the foreseeable future. It will also result in a scheme for which all the design criteria used are known, and hence open to criticism; and which is not bound by foreign government restrictions.

4.1. S-box Redesign

Since sixteen S-boxes will be required in the extended scheme, an additional eight boxes will be required. One possibility is simply to reuse the existing

[4] there is a saving of half resulting from the well known symmetry in DES, namely that $DES(\bar{X}) = \overline{DES(X)}$

eight boxes. This option was rejected both for copyright reasons, and because of doubts about the cryptological security of the current S-box 4 [BMP87], [Sha86]. Hence it was decided that sixteen new S-boxes will be generated. Ideally all possible 6*4 bit functions would be generated and evaluated against the known design rules, and a subset then selected from those which satisfy them. Because of the time this would take, it is envisioned that a random selection of boxes based on the non-linearity requirement will be generated and tested against the remaining criteria. This will be done both to generate suitable boxes for the proposed design, and to extend work (see [BMP87]) on the analysis of the design rules, identifying both redundant rules and possibly new rules.

4.2. P-box Redesign

The use of double width data paths implies a complete redesign of all the permutations. IP, IP^{-1}, and $PC1$ may be easily extended because of their regular design, if they are included at all. Morris-Yates [Mor86] has argued that because of doubts about their cryptological usefulness, they need not be provided at all (with the exception of the parity stripping component of $PC1$), saving a significant portion of silicon real-estate. If they are provided then the efficient implementation suggested by Hoornaert et al [HGD84] should be used.

The expansion function E may also be easily extended because of its regular design. This leaves the important permutations P and $PC2$. Because these are central components of the S-P network, and key scheduling respectively, they are critical to the security of the new scheme. As with the S-boxes, a sample will be randomly generated and tested against the known rules, in an effort to isolate any further design criteria, and to identify $S.P$ combinations which are functionally identical. It is here that this work differs most from Morris-Yates, who chose purely random permutations.

4.3. Key Scheduling Extension

In the extended algorithm, the same style of shift and select key scheduling will be used. A new shift schedule will be designed to satisfy the known design criteria, which will involve some interaction with the design of $PC2$. Since sixteen rounds are planned in the extended design (provided analyses of the dependence of output bits on input and key prove satisfactory), this implies rotations greater than 2 bits in order to have a total rotation of 112 bits in sixteen rounds. This impacts on the minimum spacing requirement between bits selected in $PC2$.

5. Conclusion

In this paper, all the known design criteria for the S-boxes, P-boxes, and key scheduling in the Data Encryption Standard have been identified. These all appear to be related to the avalanche and completeness effects. The paper concludes with a discussion on how these criteria will be used to design an extended DES, as well as aiding in identifying further design criteria.

References

[ASA85] ASA, "*Electronics Funds Transfer - Requirements for Interfaces, Part 5, Data Encryption Algorithm*," AS2805.5-1985, Standards Association of Australia, Sydney, Australia, 1985.

[Ber82] T. A. Berson, "Long Key Variants of DES," in *Advances in Cryptology - |Proc.| of Crypto 82*, D. Chaum, R. L. Rivest and A. T. Sherman (editors), pp. 311-313, Plenum Press, New York, |August.| 23-25, 1982.

[BGK77] D. K. Branstead, J. Gait and S. Katzke, "*Report of the Workshop on Cryptography in Support of Computer Security*," NBSIR 77-1291, National Bureau of Standards, |September.| 1977.

[BMP87] E. F. Brickell, J. H. Moore and M. R. Purtill, *Advances in Cryptology - |Proc.| of CRYPTO'86*, Lecture Notes in Computer Science, no. 263, pp. 3-8, |Springer Verlag|, Berlin, 1987.

[Dav82] D. W. Davies, "Some Regular Properties of the Data Encryption Standard," in *Advances in Cryptology - |Proc.| of Crypto 82*, D. Chaum, R. L. Rivest and A. T. Sherman (editors), pp. 89-96, Plenum Press, New York, |August.| 23-25, 1982.

[DDF83] M. Davio, Y. Desmedt, M. Fosseprez, R. Govaerts, J. Hulsbosch, P. Neutjens, P. Piret, J. Quisquater, J. Vanderwalle and P. Wouters, "Analytical Characteristics of the DES," in *Advances in Cryptology - |Proc.| of Crypto 83*, D. Chaum, R. L. Rivest and A. T. Sherman (editors), pp. 171-202, Plenum Press, New York, |August.| 22-24, 1983.

[DiH77] W. Diffie and M. E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer*, vol. 10, no. 6, pp. 74-84, |June| 1977.

[Fei73] H. Feistel, "Cryptography and Computer Privacy," *Scientific American*, vol. 228, no. 5, pp. 15-23, |May| 1973.

[GrT78] E. K. Grossman and B. Tuckerman, "Analysis of a Weakened Feistel-Like Cipher," in *|Proc.| 1978 IEEE |Conf.| On Communications*, pp. 46.3.1-5, IEEE, 1978.

[Hel79] M. E. Hellman, "DES will be totally insecure within ten years," *SPECTRUM*, vol. 16, no. 7, pp. 31-41, |July| 1979. With rebuttals from George I. Davida, Walter Tuchman, and Dennis Branstad.

[HGD84] F. Hoornaert, J. Goubert and Y. Desmedt, "Efficient Hardware Implementation of the DES," in *Advances in Cryptology: |Proc.| of Crypto 84*, Lecture Notes in Computer Science, no. 196, G. R. Blakley and D. Chaum (editors), pp. 147-173, |Springer Verlag|, Berlin, |August.| 1984.

[KaD79] J. B. Kam and G. I. Davida, "Structured Design of Substitution-Permutation Encryption Networks," *|IEEE Trans. on Computers|*, vol. C-28, no. 10, pp. 747-753,

|October.| 1979.

[Kon86] A. G. Konheim, *Cryptography*, Seminars of Excellence, ORSYS Institute, Amsterdam, |June| 9-11, 1986. notes from a seminar series.

[Mey78] C. H. Meyer, "Ciphertext/plaintext and ciphertext/key dependence vs number of rounds for the data encryption standard," in *AFIPS |Conf.| |Proc.| 47*, pp. 1119-1126, AFIPS Press, Montvale NJ, USA, |June| 1978.

[MeM82] C. H. Meyer and S. M. Matyas, *Cryptography: A New Dimension in Data Security*, |John Wiley & Sons, New York, 1982.

[MoS86] J. H. Moore and G. J. Simmons, "Cycle Structure of the Weak and Semi-Weak DES Keys," *Eurocrypt 86 - Abstracts of Papers* , p. 2.1, Linkoping, Sweden, 20-22 |May| 1986.

[MoS87] J. H. Moore and G. J. Simmons, *Advances in Cryptology - |Proc.| of CRYPTO'86*, Lecture Notes in Computer Science, no. 263, pp. 9-32, |Springer Verlag|, Berlin, 1987.

[Mor86] T. M. Morris-Yates, "*An nMOS VLSI Implementation of a Data Encryption Algorithm*," BEng Honours Thesis, School of Electrical Engineering, Uni. Sydney, |November.| 1986.

[NBS77] NBS, "*Data Encryption Standard (DES)*," FIPS PUB 46, US National Bureau of Standards, Washington, DC, |January.| 1977.

[QDD86] J. Quisquater, Y. Desmedt and M. Davio, "The Importance of Good Key Scheduling Schemes (How to make a Secure DES scheme with <= 48 bit keys)," in *Advances in Cryptology - Crypto 85*, Lecture Notes in Computer Science, no. 218, H. C. Williams (editor), pp. 537-542, |Springer Verlag|, Berlin, 1986.

[Sha86] A. Shamir, "On the Security of DES," in *Advances in Cryptology - Crypto 85*, Lecture Notes in Computer Science, no. 218, H. C. Williams (editor), pp. 280-281, |Springer Verlag|, Berlin, 1986.

[Sha49] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28, no. 10, pp. 656-715, |October.| 1949.

[WeT86] A. F. Webster and S. E. Tavares, "On the design of S-Boxes," in *Advances in Cryptology - Crypto 85*, Lecture Notes in Computer Science, no. 218, H. C. Williams (editor), pp. 523-534, |Springer Verlag|, Berlin, 1986.