Designing DES-Like Ciphers with Guaranteed Resistance to Differential and Linear Attacks

Carlisle M. Adams
Bell-Northern Research

Abstract. Because differential and linear cryptanalytic attacks on a DES-like cipher have a computational complexity which is computed as the product of the attack probabilities on individual rounds, these attacks can always be made computationally infeasible by adding rounds. As a practical guideline, the number of rounds which need to be added may be limited by the size of the key: just enough rounds can be added that the complexity of the attack on the overall cipher is higher than the work factor for exhaustive search of the key space. However, adding rounds has an adverse impact on the performance of the cipher, since the number of rounds is inversely proportional to speed (higher means slower).

In this paper we discuss a technique which will make a DES-like cipher not just computationally immune but *intrinsically* immune to both differential and linear attacks (in that the attacks can no longer be mounted), with negligible performance impact in some practical environments.

1. Introduction

Differential [4, 7] and linear [11] cryptanalysis appear to be fairly general-purpose attacks which may be applied to a variety of substitution-permutation network (DES-like) ciphers. The successful (in a theoretical sense) attacks on DES have been widely reported, as have attacks on FEAL, LOKI, Snefru, and generalized versions of DES, among others [5, 6, 8]. Both methods work on the principle of finding high-probability attacks on a single round and then building up "characteristics" (sets of consecutive rounds which interact in useful ways); characteristics which include a sufficient number of rounds can lead to cryptanalysis of the cipher. The probability of a characteristic is equal to the product of the probabilities of the included rounds; this "characteristic probability" determines the work factor of the attack. If the work factor of the attack is less than the work factor for exhaustive search of the key space, the cipher is theoretically broken.

From the above description it can be seen that adding rounds to a DES-like cipher can always be done to increase the work factor of a differential or linear attack, until the work factor surpasses that of exhaustive key search. This makes the cipher resistant to these attacks. The disadvantage of this approach is that the encryption/decryption performance of the cipher is reduced, perhaps drastically, since each added round slows the cipher down by a factor of $1/N$, where N is the number of rounds in the original cipher.

An alternate approach which has been pursued by a number of researchers is to decrease the attack probability of an individual round by improving the properties of the round substitution boxes (s-boxes); see [2, 3, 9, 13-15], for example. This results in a lower characteristic probability for the same number of rounds and therefore has the potential to make the cipher resistant to these attacks without degrading performance. However, there is always the possibility that for a given cipher the round probability cannot be made low enough to avoid the need to add rounds. Furthermore, there is always the possibility that for a given cipher the best characteristic has not yet been found (and when it is found, it will render the cipher breakable).

The approach taken in this paper is neither of the above. Instead, a slight alteration is proposed for the round function which renders it intrinsically immune to differential and linear cryptanalysis. Such an alteration is generally applicable to all DES-like ciphers. However, for certain ciphers with certain implementation characteristics, it may be possible to add this immunity with little or no performance degradation.

The remainder of this paper is organized as follows. Section 2 gives a very brief review of both differential and linear cryptanalysis in preparation for Section 3, which describes the technique to provide intrinsic immunity to these attacks in the round function of a DES-like cipher. Section 4 discusses the application of this technique to the CAST family of ciphers, including some analysis of the resulting performance impact. Finally, some concluding comments are given in Section 5.

2. Review of the Attacks

Differential and linear cryptanalysis (chosen- and known-plaintext attacks, respectively) are similar in flavour in that both rely on s-box properties to formulate an attack on a single s-box. Each then defines a *general approach* to attacking the entire round function and builds on this approach to formulate a variety of *particular* round function attacks. These particular attacks are then extended to create a number of characteristics for the overall cipher. The most successful characteristic (that is, the one with highest probability) is essentially the principle differential or linear attack on the cipher, and the cipher is theoretically broken if the work factor for this principle attack is less than the work factor for exhaustive search of the key space (even if the attack requires an impractical amount of chosen or known plaintext).

2.1. Differential Cryptanalysis

In differential cryptanalysis the s-box property which is exploited is its "input XOR" to "output XOR" mapping. It was noted by Biham and Shamir that for a given pattern of change in the input of the DES s-boxes, some output change patterns are much more likely to occur than others, over the space of all possible inputs. (This analysis was repeated for all possible input change patterns, so that an XOR table could be created for each s-box [4].) Thus, for any specific input, a given change to that input will cause a given output change with a certain probability. The attack takes advantage of the large probabilities (some as high as 0.25) which occur in some s-boxes.

The *general* round attack which uses the above mapping is as follows. In DES the input to the round is the right half of the data, R, and the subkey, K, for that round. After an expansion step for R, the data and key are XOR'ed together, so that the input to the set of s-boxes is $X = E(R) \oplus K$. The output of the set of s-boxes is $Y = S(X)$, which is then permuted to form the output of the round function $R' = P(Y)$. Since the expansion step is linear, two data vectors R_1 and R_2 which differ by ΔR (so that $\Delta R = R_1 \oplus R_2$), produce $\Delta X = X_1 \oplus X_2 = E(R_1) \oplus K \oplus E(R_2) \oplus K = E(R_1) \oplus E(R_2) = E(\Delta R)$ during two encryptions with the same key. It is therefore not difficult to choose ΔR such that ΔX has a desired pattern. Furthermore, since the permutation step is also linear, it is not difficult to find a ΔY such that $\Delta R' = P(Y_1) \oplus P(Y_2) = P(\Delta Y)$ has a desired pattern.

The implication of the above comments is that it is possible to choose a particular ΔR such that a particular ΔX occurs (one which presents a desired set of input XORs to a desired set of s-boxes). With a known probability, a particular ΔY will result (a desired set of output XORs from the s-boxes), which will be reflected in a particular $\Delta R'$. Finding ΔR pairs which result in useful $\Delta R'$ pairs constitutes the set of *particular* round function attacks on the cipher. In this context, a $\Delta R'$ pair is "useful" if it can act as a desired ΔR pair in the following round, so that particular round function attacks can be iterated and concatenated into characteristics with high overall probability.

2.2. Linear Cryptanalysis

In linear cryptanalysis the s-box property which is exploited is linearity. It was noted by Matsui that for some of the DES s-boxes, the XOR sum of a subset of the input bits is equal to the XOR sum of a subset of the output bits significantly more or less often than half the time, over the space of all possible inputs. (This analysis was repeated for all possible input and output subsets, so that a distribution table could be created for each s-box [11].) Thus, for any specific input, the input sum will be equal to the output sum with a certain probability. The attack takes advantage of the fact that in some s-boxes the probabilities are significantly different from the ideal of 0.5 (some as low as 0.19).

The *general* round attack which uses the linearity property is as follows. For DES the values R, K, X, Y, and R', and the operations $E(\bullet)$ and $P(\bullet)$ are as discussed above. With a probability given by the distribution tables, a subset of the bits of Y will sum to a value (zero or one) which is equal to the sum of a subset of the bits of X. Thus, $\Sigma(Y) = \Sigma(X)$, where the operation $\Sigma(\bullet)$ is taken to mean the XOR sum of a specific subset of the bits in the argument. This implies that $\Sigma(Y) = \Sigma(E(R) \oplus K) = \Sigma(E(R)) \oplus \Sigma(K)$, and so $\Sigma(K) = \Sigma(E(R)) \oplus \Sigma(Y)$. Now, knowing R immediately yields $\Sigma(E(R))$ and knowing R' immediately yields $\Sigma_p(R') = \Sigma_p(P(Y)) = \Sigma(Y)$, where the operation $\Sigma_p(\bullet)$ is defined to be the XOR sum of the permuted indices of the subset of bits used in $\Sigma(\bullet)$. Therefore, knowledge of R and R' can be used to solve for the equivalent of one key bit, $\Sigma(K)$, with a known probability.

The *particular* round function attacks on the cipher consist of those $\Sigma(R)$ subsets which result in useful $\Sigma(R')$ subsets. In this context, a $\Sigma(R')$ subset is "useful" if it can be XOR'ed with a desired $\Sigma(L)$ subset from the previous round (where L is the left half of the data) to yield a desired $\Sigma(R)$ subset for the following round, so that particular round function attacks can be iterated and concatenated into characteristics with high overall probability.

3. Round Modification

The goal behind modifying the round function is to eliminate the possibility of both differential and linear cryptanalytic attacks against the cipher. This is done by inserting a nonlinear, key-dependent operation before the s-box lookup to effectively mask the inputs to the set of s-boxes. If these inputs are well "hidden", then s-box properties (such as the input XOR to output XOR mapping, or linearity) cannot be exploited in a general round function attack.

More specifically, the following round modification is proposed:

$$f(R, K) = f(R, K_1, K_2) = S(g(R \oplus K_1, K_2))$$

where $g(\bullet, \bullet)$ is an operation with properties as defined below. For DES, the expansion operation can be placed either around R or around $(R \oplus K_1)$ – that is, $f(R, K) = S(g(E(R)$

$\oplus K_1, K_2$)) or $f(R, K) = S(g(E(R \oplus K_1), K_2))$ – depending on whether K_1 is 32 or 48 bits in length. As well, the permutation operation can be placed around $S(\bullet)$ as is done in the current round definition.

Several properties are required of the function $g(\bullet, \bullet)$. These will be discussed in Sections 3.1 and 3.2 below, but they are enumerated here for reference.

- Firstly, $g(\bullet, \bullet)$ must not be distributive with respect to the XOR operation.
- Secondly, it must represent a nonlinear mapping from its input to its output, so that any linear change in either input leads to a nonlinear change in the output vector.
- Thirdly, it must be relatively simple to implement in software (in terms of code size and complexity).
- Fourthly, it must execute efficiently (no more slowly than the remainder of the round function, for example).
- Finally, it must effectively "hide" its R (or $E(R)$) input if K_1 and K_2 are unknown (in the sense that there must be no way to cancel the effect of the keys in the round function using an operation on a single R value or a pair of R values).

A function which appears to encompass all the properties listed above is modular multiplication, for an appropriate choice of modulus. If R, K_1, and K_2 are 32 bits in length (as can be done in a CAST cipher [1, 3], for example), two candidate moduli are $(2^{32} - 1)$ and $(2^{32} + 1)$. Meijer [12] has given a simple algorithm to carry out multiplication modulo $(2^{32} - 1)$ in a high-level language using only 32-bit registers, and has shown that multiplication with this modulus is a "complete" operation (in that every input bit has the potential to modify every output bit [10]), so that this modulus appears to satisfy nonlinearity, simplicity, and data hiding. However, this modulus does not satisfy the fifth property ideally, since zero always maps to zero, and $(2^{32} - 1)$ always maps to either $(2^{32} - 1)$ or zero (depending on the implementation), regardless of the key in use. (Note, however, that in a practical implementation it is a simple matter to ensure that the computed subkey K_2 is never equal to 0 or to (2^{32} - 1), and masking R with K_1 ensures that it is not easy for the cryptanalyst to choose R such that ($R \oplus K_1$) is equal to 0 or to (2^{32} - 1).)

The modulus (2^{32} + 1) may be a better choice with respect to property five than (2^{32} - 1) if either of two simple manipulations are performed. Firstly, each input can be incremented by one, so that the computation is actually done with (R+1) and (K+1). Thus the arguments belong to the set [1, 2^{32}] rather than [0, 2^{32} - 1], avoiding both the zero and the (2^{32} + 1) "fixed point" inputs. Alternatively, the inputs can be left as is (so that the computation is done with R and K), with only the zero input mapped to the value 2^{32} (and the 2^{32} output mapped back to zero). Implementation of multiplication using this modulus is thus only slightly more difficult using a high-level language with 32-bit registers than for the modulus (2^{32} - 1), and on platforms where the assembly language instructions give access to the full 64-bit result of a 32-bit multiply operation, the modular reduction can be accomplished quite simply and efficiently. Furthermore, as for (2^{32} - 1), multiplication with this modulus represents a nonlinear mapping from input to output.

It is also important to mention that in order to ensure that the modular multiplication does not perform badly with respect to property five, it is necessary that the subkey K_2 be relatively prime to the modulus. Thus, when the subkeys are being generated, the K_2 used in each round must not have 3, 5, 17, 257, or 65537 as factors if the modulus n = (2^{32} - 1), and must not have 641 or 6700417 as factors if n = (2^{32} + 1).

Finally, it appears that either modulus can be used to satisfy property one, since modular multiplication is not distributive with respect to the XOR operation.

3.1. Intrinsic Immunity to Differential Cryptanalysis

Property five listed above prevents a differential attack as described by Biham and Shamir, and property two prevents a simple modification to their description. Recall the equation given in Section 2.1:

$$\Delta X = X_1 \oplus X_2 = E(R_1) \oplus K \oplus E(R_2) \oplus K = E(R_1) \oplus E(R_2) = E(\Delta R)$$

during two encryptions with the same key. This is the critical component of the differential attack because it shows that the XOR sum of two data inputs (R_1 and R_2) completely determines the input XOR for the round s-boxes. This is why this attack would ideally be mounted using chosen plaintext (so that the cryptanalyst can select the input XORs which will construct the highest-probability characteristic). Property five prevents such an attack by the requirement that no operation on a pair of R values can cancel the effect of the key. Modular multiplication appears to achieve property five in the modified equation

$$\begin{aligned}\Delta X &= X_1 \oplus X_2 \\ &= g(R_1 \oplus K_1, K_2) \ \oplus \ g(R_2 \oplus K_1, K_2) \\ &= (((R_1 \oplus K_1) * K_2) \bmod n) \ \oplus \ (((R_2 \oplus K_1) * K_2) \bmod n)\end{aligned}$$

since knowledge of R_1 and R_2 does not seem to reveal ΔX if K_1 and K_2 are not known. Thus, the input XOR to output XOR mapping of the round s-boxes cannot be exploited through knowledge/choice of R_1 and R_2.

Modular multiplication also appears to satisfy property two because it is not obvious that any simple modification to the differential attack will cause knowledge of R_1 and R_2 to reveal information about ΔX if K_1 and K_2 are not known. This is not true of arbitrary operations which may be proposed for $g(\bullet, \bullet)$. For example, if $g(\bullet, \bullet)$ is real addition (modulo n), then re-defining ΔX to be subtraction (modulo n) yields

$$\begin{aligned}\Delta X &= (X_1 - X_2) \bmod n \\ &= (g(R_1 \oplus K_1, K_2) \ - \ g(R_2 \oplus K_1, K_2)) \bmod n \\ &= (\ (((R_1 \oplus K_1) + K_2) \bmod n) \ - \ (((R_2 \oplus K_1) + K_2) \bmod n)\ ) \bmod n\end{aligned}$$

$$= ((R_1 \oplus K_1) - (R_2 \oplus K_1)) \bmod n$$

In such a situation the difference between R_1 and R_2 (XOR or real subtraction) reveals a significant amount of information about ΔX which may be used in subsequent rounds to construct a characteristic.

3.2. Intrinsic Immunity to Linear Cryptanalysis

Property one given in Section 3 prevents a linear attack as described by Matsui. Recall the equation given in Section 2.2:

$$\Sigma(Y) = \Sigma(X) = \Sigma(E(R) \oplus K) = \Sigma(E(R)) \oplus \Sigma(K)$$

$$\text{Therefore,}\ \ \Sigma(K) = \Sigma(E(R)) \oplus \Sigma(Y)$$

This is the critical component of the linear attack because the distributive nature of the XOR operation with respect to the subset sum operation $\Sigma(\bullet)$ – which is another XOR operation – allows the equivalent of one key bit to be computed using only knowledge of $\Sigma(E(R))$ and $\Sigma(Y)$. This is why this attack would typically be mounted using known plaintext (so that the cryptanalyst can use knowledge of $\Sigma(plaintext)$ and $\Sigma(ciphertext)$ to work through intermediate rounds to solve for various key bits). Property one prevents such an attack by the requirement that $g(\bullet, \bullet)$ not be distributive with respect to the XOR operation. Modular multiplication appears to achieve this requirement, as seen in the modified equation

$$\Sigma(Y) = \Sigma(X) = \Sigma(((R \oplus K_1) * K_2) \bmod n)$$

since it appears that this equation cannot be rearranged in any way to solve for subset sums of K_1 and K_2 given only subset sums of R and Y. (Note that either $E(R)$ or $E(R \oplus K_1)$ may be substituted in the above equation, if required.)

4. Application of this Technique to the CAST Cipher

The CAST encryption algorithm [1, 3], implemented with a blocksize and keysize of 64 bits, four 8×32 s-boxes $S_1 ... S_4$ for the encryption step, and 32-bit subkeys in each round, can be shown to have a work factor for differential and linear attacks which is greater than exhaustive search of the key space if 12 or more rounds are used. Since it requires no expansion or permutation steps, the round function f is computed simply as:

$$f(R, K) = S_1(B^{(1)}) \oplus \ldots \oplus S_4(B^{(4)})$$

where $B = R \oplus K$ and $B^{(j)}$ is the j^{th} byte of B. Application of the technique described in Section 3 to CAST yields the modified computation of the round function, where f remains identical but B is now computed as

$$B = ((R \oplus K_1) * K_2) \bmod n.$$

Examination of the assembly language instructions required for the modular multiplication step alone (using either $(2^{32} - 1)$ or $(2^{32} + 1)$ as the modulus) shows that multiplication takes approximately 75% of the time taken for the remainder of the round on a 486- or a Pentium-class PC, so that every four multiplications added is equivalent in performance impact to the addition of 3 original CAST rounds.

Security analysis so far indicates that it may not be required to include the modular multiplication step in every round; rather, using it only in the first two and the last two rounds appears to be sufficient to render the cipher intrinsically immune to both differential and linear attacks. Furthermore, since these attacks appear to be the primary keysize-limiting attacks on the cipher, the added strength of these four modified rounds may allow the total number of rounds to be reduced to six or eight with no security degradation. These conjectures need to be examined further, but if they are true then a 6-round modified CAST (with an execution time equivalent to a 9-round implementation of the original CAST) or an 8-round modified CAST (with execution time equivalent to 11 original rounds) can be used in place of a 12-round unmodified CAST. Thus, with no negative performance impact (and possibly with a significant performance increase), the modified CAST can provide intrinsic immunity to differential and linear attacks. Note that the modified CAST includes every operation present in the unmodified CAST, and there is no indication that the insertion of modular multiplication can interact badly with these operations; thus, security should remain equivalent to the original cipher in all other respects.

5. Conclusions

This paper has proposed a general modification to the round function of DES-like ciphers to combat certain cryptanalytic attacks. It was shown that ciphers incorporating this modification are made to be intrinsically immune to the round function attacks of both differential and linear cryptanalysis (in that the attacks, as described by Biham and Shamir and by Matsui, can no longer be mounted). A specific round modification (modular multiplication) was discussed and it was shown that for certain environments and certain ciphers, the proposed modification may have negligible impact on encryption/decryption speed (the CAST algorithm on a 486-class processor or better was given as a concrete example).

The primary purpose of this proposal is to stimulate further research into various ways of modifying DES-like round functions to enhance security against general classes of cryptanalytic attack, with a view toward minimizing the resulting performance degradation. The cryptologic community is thereby encouraged to analyze this proposal and to suggest others in order to solidify the principles of DES-like cipher design in this area.

References

[1] 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.

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

[3] C. M. Adams and S. E. Tavares, *Designing S-Boxes for Ciphers Resistant to Differential Cryptanalysis*, Proceedings of the 3rd Symposium on the State and Progress of Research in Cryptography, Rome, Italy, Feb., 1993, pp.181-190.

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

[5] E. Biham and A. Shamir, *Differential Cryptanalysis of FEAL and N-Hash*, in Advances in Cryptology: Proc. of Eurocrypt '91, Springer-Verlag, 1992, pp.1-16.

[6] E. Biham and A. Shamir, *Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI, and Lucifer*, Advances in Cryptology: Proc. of CRYPTO '91, Springer-Verlag, 1992, pp.156-171.

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

[8] E. Biham and A. Shamir, *Differential Cryptanalysis of the full 16-round DES*, in Advances in Cryptology: Proc. of CRYPTO '92, Springer-Verlag, 1993, pp.487-496.

[9] 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, 1992, pp.352-367.

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

[11] M. Matsui, *Linear Cryptanalysis Method for DES Cipher*, in Advances in Cryptology: Proc. of Eurocrypt '93, Springer-Verlag, 1994, pp.386-397, 1994.

[12] H. Meijer, Multiplication-Permutation Encryption Networks, Technical Report #85-171, Queen's University, Department of Computing and Information Science, 1985.

[13] K. Nyberg, *Perfect nonlinear S-boxes*, in Advances in Cryptology: Proc. of Eurocrypt '91, Springer-Verlag, 1992, pp.378-386.

[14] J. Seberry, X-M. Zhang, and Y. Zheng, Systematic Generation of Cryptographically Robust S-boxes, in 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, U.S.A., Nov. 3-5, 1993, pp.171-182.

[15] 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, Springer-Verlag, 1993, pp.260-279.

Securing DES S-boxes against Three Robust Cryptanalysis

Kwangjo Kim Sangjin Lee Sangjoon Park Daiki Lee

Section 0710, P.O.Box 106, Yusong
Electronics and Telecommunications Research Institute
Taejon, 305-600, KOREA

Abstract

In this paper, we propose an expanded set of design criteria for the generation of DES-like S-boxes which enable DES being immunized against three known robust cryptanalysis, *i.e.*, differential, Improved Davies' and linear cryptanalysis and we also suggest a set of new 8 DES-like S-boxes generated by our proposed design criteria in order to replace with the current 8 DES S-boxes. The computer simulation leads us to conclude that the breaking complexity of the strengthened DES (we call s^5DES) by three powerful cryptanalysis is no more efficient than the key-exhaustive search.

1 Introduction

Until now, three powerful cryptanalysis have been published to break DES (Data Encryption Standard) [1] more efficiently than the 56-bit key exhaustive search. One is the DC (Differential Cryptanalysis) proposed by Biham and Shamir [2],[4] in 1990. The DC is a kind of chosen plaintext attack in a sense that an attacker has to choose 2^{47} plaintexts and their corresponding ciphertexts to find an unknown 56-bit DES key. The other attack, known as LC (Linear Cryptanalysis), is more feasible than DC and was proposed by Matsui [6] in 1993. The overall complexity to break DES by LC has been reduced [14] to be 2^{43} known plaintexts and ciphertexts pairs comparing to the initial complexity of 2^{47}. Moreover, the Improved version of Davies' attack proposed by Biham and Biryukov [13] was found to break DES with the complexity of 2^{50}. The common point of three cryptanalysis is to use the cryptanalytic properties of DES S-boxes which play an important role in making DES work as nonlinear cryptographic functions.

Two researches in [3] and [5] have been reported to strengthen DES resistant against DC by only replacing the current DES S-boxes with new S-boxes based on the different design criteria rather than the well-known 6 design criteria of DES S-box. Two DES-like cryptosystems are named as s^2DES and s^3DES, respectively. By the efficient search techniques [10],[11] evaluating on the overall strength of DES against DC and LC, the relative security of the full-round DES and s^iDES was found to be $s^2\text{DES} < DES < s^3\text{DES}$ against DC and $s^3\text{DES} < DES < s^2\text{DES}$ against LC.

In this paper, we propose an expanded set of design criteria for the generation of DES-like S-boxes which make DES be immunized against three robust attacks. We also suggest a set of DES-like S-boxes satisfying our proposed design criteria to replace the current DES S-boxes with new S-boxes. Finally, we evaluate the breaking complexity of new DES (we call s^5DES) by three powerful attacks.

2 Design Criteria of DES S-boxes

The followings are the well-known 6 design criteria of DES S-boxes :

(S-1) No S-box is a linear or affine function.

(S-2) Changing one bit in the input of an S-box results in changing at least two output bits.

(S-3) The S-boxes were chosen to minimize the difference between the number of 1's and 0's when any single bit is held constant.

(S-4) S(**x**) and S(**x** $\oplus$ (001100)) differ at least two bits.

(S-5) S(**x**) $\neq$ S(**x** $\oplus$ (11ef00)) for any e and f.

(S-6) S(**x**) $\neq$ S(**x** $\oplus$ (0abcd0)) for any a, b, c, and d, abcd $\neq$ 0000.

In addition to these criteria, Coppersmith [12] has recently published additional design criteria of DES S-boxes, such as

(S-7) For a given nonzero input XOR and output XOR, no more than 8 of the outputs may exhibit the given output XOR among the 32 pairs of inputs exhibiting the given input XOR.

(S-8) Similar to (S-7), but stronger restrictions in the case zero output XOR, for the case of 3 active S-boxes on round i.

(S-9) Other criteria dealt with other issue, such as ease of implementation.

He has described that *most of the criteria are aimed at increasing the number of active S-boxes (against DC) involved over the 12 or 16 rounds of the "probable pattern", say this total number is k. Then (S-7), along with the simplifying assumption of independence, puts an upper bound of* $(1/4)^k$ *on the overall probability of this "probable pattern".*

In other words, (S-7) means that the maximal entry in a XOR distribution table of any DES S-boxes is 16 and (S-8) means that nonzero input XOR with 3 active S-boxes resulting the same output always exists with some probability. From all criteria, we can see that the DES designers might initially put in mind the strength of DES against DC.

3 Resistance against DC

The important properties of DES-like S-boxes are derived through an analysis of tables showing a group of particular distributions—called the pairs XOR distribution—defined as follows:

Definition 1 (XOR distribution table) *A table that shows the distribution of input XORs and output XORs of all possible pairs of an S-box is called the pairs XOR distribution table of the S-box. In this table, each row corresponds to a particular input XOR, each column corresponds to a particular output XOR, and the entries in the table count the number of possible pairs with such an input XOR and an output XOR.*

Since the pairs XOR distribution in any DES S-box are used for DC, some intuitive definitions which measures the level of its resistance against DC are necessary. We can consider how many entries appear in the pairs XOR distribution–% of entry, denoted by μ_d.

$$\mu_d = \frac{nz}{64 \times 16} \times 100 \qquad (1)$$

where nz denotes the number of non-zero entries in a pairs XOR distribution table. In order to measure how well the values of all entries are distributed from the ideal (uniform) value of entry, the standard deviation, σ_d of all entries can be checked by

$$\sigma_d \quad = \quad \sqrt{\sum_{i=0}^{63}\sum_{j=0}^{15}(e_{ij}-4)^2/64 \times 16.} \qquad (2)$$

where e_{ij} is a measured number of entry in pairs XOR distribution table. We can also check the nontrivial[1] maximum value of entries, denoted by λ_d, in DES S-boxes since the relatively higher valued entries are directly employed for DC. We simply call these three parameters differential characteristics of an S-box. Differential characteristics of DES S-boxes are measured in Table 2.

Up to now, the 2-round iterative characteristics ($\Phi \longrightarrow 0$ with some prob. p) have ever been known to be the basic tool in breaking DES by DC. The 2-round iterative characteristic $\Phi = 1960000_x$ with probability $\frac{1}{234}$ is found to be close to the best characteristic for attacking the full 16-round DES. We discuss cases where 2-round iterative characteristics occur in DES and suggest an efficient design criterion of DES-like S-boxes which guarantees to exhibit 2-round iterative characteristics with very low probability, *i.e.*, the method to immunize DES against DC is to find any necessary condition such that 2-round iterative characteristics exist with more than 3 active S-boxes.

The careful examination of E-expansion of DES F-functions leads us to the following theorem.

Theorem 1 *For a given DES-like S-box satisfying 6 design criteria, the possibilities that nonzero input XOR with 3 active S-boxes results in the nonzero output XOR is one of the followings :*

(A-1) $S(\mathbf{x}) \neq S(\mathbf{x} \oplus (00ef11))$ *for any* e *and* f.

(A-2) $S(\mathbf{x}) \neq S(\mathbf{x} \oplus (10ef00))$ *for any* e *and* f.

(A-3) $S(\mathbf{x}) \neq S(\mathbf{x} \oplus (11ef10))$ *for any* e *anf* f.

If we combine 6 design criteria with A-3, we can obtain the following important theorem to immunize DES against DC.

Theorem 2 *If nonzero input XOR* $\underbrace{00a_1a_2a_3a_4}_{1}\ \underbrace{a_3a_4a_5a_6a_7a_8}_{2}\ \ldots\underbrace{a_{n-7}a_{n-6}a_{n-5}a_{n-4}a_{n-3}a_{n-2}}_{l-1}$ $\underbrace{a_{n-3}a_{n-2}a_{n-1}a_n00}_{l}$ *for each* $n = 4i+2, (i = 1, \ldots, l)$ *is given to* l *neighbouring DES S-boxes satisfying 6 criteria and (A-3), output XOR will never be zero.*

The above theorem means that we can only find 2-round iterative characteristics with 8 active S-boxes, *i.e.*, we cannot find any 2-round iterative characteristics with less than 7 active S-boxes. Combining (A-3) and (S-5), the necessary condition to immunize DES against DC is :

Condition 1 (D-1) $S(\mathbf{x}) \neq S(\mathbf{x} \oplus (11efg0))$ *for any* efg.

Note that if DES-like S-boxes meet **(D-1)**, DES whose S-boxes are replaced with new S-boxes can be resistant against Improved Davies' attack [13],[17].

[1] It is trivial that the entry always has a value of 64 when the input XOR and output XOR of any DES-like S-box are zero.

4 Resistance against LC

The following notations are used hereinafter.

- I_i : The input value of *i-th* round in DES F-function.
- O_i : The output value of *i-th* round in DES F-function.
- K_i : The key value of *i-th* round in DES F-function.
- $X[Z] = \oplus_{k \in Z} X[k]$, where $Z \subset \{0, 1, \ldots, 47\}$ and $X[k]$ is the *k-th* bit of X which is one of I_i, O_i and K_i.
- a_x : The hexadecimal value of a.
- $W(\alpha)$: The Hamming weight of α.
- For $x, y \in GF(2)^n$, $x \bullet y$ denotes the dot product of x and y.

Kim et al. [8],[9] have already suggested the necessary condition to design DES-like S-boxes which can be resistant to LC. In this section, we will summarize those conditions and revise them by the computer experiments.

4.1 Uniformity of a Linear Distribution Table

It is necessary to precompute the linear distribution table of DES-like S-boxes defined as below like DC in order to break DES by LC.

Definition 2 (Linear distribution table) *For a given DES S-box S, we define $NS(\alpha, \beta)$ as the number of times minus 32 out of 64 input patterns of S, such that an XORed value of the input bits masked by α coincides with an XORed value of the output bits masked by β, i.e.,*

$$NS(\alpha, \beta) \quad = \quad \#\{x \in GF(2)^6 | x \bullet \alpha = S(x) \bullet \beta\} - 32$$

where $\alpha \in GF(2)^6$ and $\beta \in GF(2)^4$. We refer the complete table for every α and β to the linear distribution table.For a specific S-box, S_i $(i = 1, \ldots, 8)$, we denote its linear distribution table as $NS_i(\alpha, \beta)$.

In order to check the uniformity of a linear distribution table of any S-box, let us denote % of the entry by μ_l and the maximal entry by λ_l as the same way to check the uniformity of XOR distribution table.

Based on the computer experiments and theoretical analysis, the first necessary condition [9] to make DES-like S-boxes to be immunized against LC is :

Condition 2 (L-1) *The λ_l of any S-box should be less than 16.*

4.2 Iterative Linear Approximation

The following definitions are necessary to understand the concept of LC.

Definition 3 (Linear approximation) *For a given expression $I[Z_1] \oplus O[Z_2] = K[Z_3]$ with probability $p + 1/2$, this linear approximation is denoted as $A\colon I[Z_1], K[Z_3] \longrightarrow O[Z_2]$ with p. We denote this expression as $A, B, C, \cdots$ and sometimes omit the term $K[Z_3]$. Also $\delta(A)$ denotes the set of S-boxes necessary to express A and $\#\delta(A) = |\delta(A)|$.*

Definition 4 (nR iterative linear approximation) *The n-round (simply, nR) iterative linear approximation is defined by* $I_1[Z_1] \oplus I_n[Z_n] = K_2[Z_2] \oplus \cdots \oplus K_{n-1}[Z_{n-1}]$. *For the consecutive n-rounds, the XORed values of n-2 keys in an (n-2)-round can be expressed by its input and output XORed values. When this expression holds with probability* $q = p+1/2$, *the probability of this linear approximation is to be* p. *Also, we denote nR iterative linear approximation as* $-A_1 \cdots A_{n-2}-$ *and its concatenated expression as* $-A_1 \cdots A_{n-2} - A_{n-2} \cdots A_1 -$.

To cryptanalyze n-round DES by LC in general, we need to find an useful linear approximation of $(n-1)$-round DES. When the linear approximation of $(n-1)$-round DES holds with probability $q = p + \frac{1}{2}$, the number of plaintexts which the attacker needs are about $|p|^{-2}$ by Lemma 2 in [6]. Thus, a linear approximation of 15-round DES is necessary to break the full 16-round DES. When this approximation holds with probability p_{15}, the necessary condition that LC is no more efficient than key-exhaustive search is $p_{15}^{-2} \geq 2^{56}$, *i.e.*, $|p_{15}| \leq 2^{-28}$. Thus, in order that DES can be resistant to LC, the main idea is to find some necessary conditions such that DES-like S-boxes should have any linear approximation with the small probability.

By Lemma 3 in [6], if we find an nR iterative linear approximation with probability p, then we can also obtain $(k \cdot (n-1) + 1)$R linear approximation with probability $2^{k-1}p^k$ when nR iterative linear approximation is applied k times. We discuss the necessary condition how to lower the probability of 3R, 4R, and 5R iterative linear approximations to prevent DES from being broken by a successful LC.

4.2.1 3R Iterative Linear Approximation

The 3R iterative linear approximation has a form of $I_1[Z_1] \oplus I_3[Z_1] = K_1[Z_2]$, *i.e.*, there exists a linear correlation between key and output subblocks without input subblock as $O_i[Z_1] = K_i[Z_2]$. This case always occurs when two outer bits of a DES S-box are given to two neighbouring S-boxes. Thus, we can build the 3R iterative linear approximation from this case.

Theorem 3 *There exists a 3R iterative linear approximation if and only if the input of Si-box and the input of S(i+1)-box are* 3_x *and* 30_x*, respectively.*

If $NS_i(3_x, \beta_1)$ and $NS_{i+1}(30_x, \beta_2)$ are not equal to zero, we can build some 3R iterative linear approximations. From the 3R iterative linear approximation $-A-$ with probability p, we can build the 15-round linear approximation like $-A-A-A-A-A-A-A-$ and the total probability for this approximation to hold is 2^6p^7. Thus, the necessary condition that this attack is no more efficient than key-exhaustive search is $2^6|p|^7 \leq 2^{-28}$, *i.e.*, $|p| \leq 2^{-4.9}$. In other words,

$$|2 \cdot \frac{NS_i(3_x, \beta_1)}{64} \cdot \frac{NS_{i+1}(30_x, \beta_2)}{64}| \leq 2^{-4.9}. \tag{3}$$

If any DES-like S-box satisfies **(D-1)**, the values of $NS(30_x, \beta)$ and $NS(31_x, \beta)$ are always to be zero for any β. Thus the LHS of Eq. (3) is always equal to zero.

Condition 3 (L-2) $S(\mathrm{x}) \neq S(\mathrm{x} \oplus (11efg0))$ *for any* efg.

Note that **(L-2)** is a simple and common design criterion to prevent DES from being broken by successful DC, Improved Davies' attack, and special case of LC.

4.2.2 4R Iterative Linear Approximation

We discuss cases when a 4R iterative linear approximation occurs from two given linear approximations such as,

$$A \ : \ I_2[Z_1], K_2[Z_3] \longrightarrow O_2[Z_2] \qquad (4)$$

$$B \ : \ I_3[Y_1], K_3[Y_3] \longrightarrow O_3[Y_2] \qquad (5)$$

If we linearly approximate the 2nd round and 3rd round function of DES to A and B, respectively, $I_2[Z_1]$ should be equal to the XORed value between the 3rd round output and 4th round input, and $I_3[Y_1]$ should be equal to the XORed value between the 1st round input and 2nd round output in order to get an useful 4R iterative linear approximation.

Theorem 4 *By concatenating two linear approximations Eqns. (4) and (5) with probability p_1 and p_2, respectively, the condition for building a 4R iterative linear approximation is $Z_1 = Y_2, Z_2 = Y_1$. Also, the 4R iterative linear approximation is of the form*

$$I_1[Z_2] \oplus I_4[Z_1] \ = \ K_2[Z_3] \oplus K_3[Y_3] \qquad (6)$$

with probability $2p_1p_2$.

If the 4R iterative linear approximation $-AB-$ with probability p is given, we can build the 15-round linear approximation as $-AB - BA - AB - BA - AB$. The necessary condition that this attack is no more efficient than key-exhaustive search is $|2^4 p^5| \leq 2^{-28}, i.e., |p| \leq 2^{-6.4}$.

Condition 4 (L-3) *The followings (18 cases in total) are necessary so that the 4R iterative linear approximation will not occur.*

- *S1-box* : $NS_1(4,4) = NS_1(2,2) = 0$, *S2-box* : $NS_2(4,4) = NS_2(2,1) = 0$
- *S3-box* : $NS_3(8,4) = NS_3(4,8) = 0$, *S4-box* : $NS_4(8,4) = NS_4(2,2) = 0$
- *S5-box* : $NS_5(16,1) = NS_5(8,8) = NS_5(2,4) = 0$
- *S6-box* : $NS_6(16,4) = NS_6(4,8) = NS_6(2,2) = 0$
- *S7-box* : $NS_7(4,8) = NS_7(2,1) = 0$, *S8-box* : $NS_8(16,1) = NS_8(2,4) = 0$

When DES-like S-boxes satisfying **(L-3)** are given, the necessary condition not to find 4R linear approximation is as follows:

Condition 5 (L-4) *For $W(\alpha), W(\beta) \leq 2$, $|NS(\alpha,\beta)| \leq 8$ where $\alpha \in GF(2)^6$ and $\beta \in GF(2)^4$.*

We, however, have found that it was very difficult to find a set of 8 DES-like S-boxes satisfying **(L-4)** by computer experiments. Thus we loose the maximum allowable values of $|NS(\alpha,\beta)|$ upto 10 in an empirical way but this would not disturb the strength of new DES-like S-boxes.

Condition 6 (Revised L-4) *For $W(\alpha), W(\beta) \leq 2$, $|NS(\alpha,\beta)| \leq 10$ where $\alpha \in GF(2)^6$ and $\beta \in GF(2)^4$.*

4.2.3 5R Iterative Linear Approximation

The following theorem points us cases where the possible 5R iterative linear approximations occur.

Theorem 5 *When three linear approximations are given by*

$$\begin{aligned} A &: I_2[Z_1], K_2[Z_3] \longrightarrow O_2[Z_2] \\ B &: I_3[Y_1], K_3[Y_3] \longrightarrow O_3[Y_2] \\ C &: I_4[X_1], K_4[X_3] \longrightarrow O_4[X_2], \end{aligned}$$

we can obtain a 5R iterative linear approximation only if $Z_1 = Y_2 = X_1, Y_1 = Z_2 \cup X_2 - Z_2 \cap X_2$, *and the 5R iterative linear expression* $-ABC-$ *is of the form*

$$I_1[Z_2] \oplus I_5[X_2] = K_2[Z_3] \oplus K_3[Y_3] \oplus K_4[X_3].$$

Using the 5R iterative linear approximation $-ABC-$, we can build 15-round linear approximation as $-ABC-CBA-ABC-DE$. The probabilities of D and E is less than 2^{-2} by **L-1**. If the probability of $-ABC-$ is p, $|p|^3$ should be less than 2^{-28}, *i.e.*, $|p| \leq 2^{-9.4}$. Some computation leads us that

Condition 7 (L-5) *For* $\alpha \in GF(2)^6$, β_1 *and* $\beta_2 \in GF(2)^4$,
$W(\alpha) = 1$ *and* $W(\beta_1 \oplus \beta_2) = 1 \Longrightarrow |NS(\alpha, \beta_1) \cdot NS(\alpha, \beta_2)| \leq 48$.

We guess that **L-2** causes $|NS(10_x, \beta_1) \cdot NS(10_x, f_x)| = 16 \times 16 = 256$ for some β_1, $W(\beta_1 \oplus f_x) = 1$. Thus we could not find any DES-like S-boxes satisfying both **(L-2)** and **(L-5)**. In order to revise **(L-5)**, we utilized the property of P-permutation in DES F-function such that :

Condition 8 (Revised L-5) *If* $\alpha \neq 10_x$,

$$|NS(\alpha, \beta_1) \cdot NS(\alpha, \beta_2)| \leq 48$$

for $W(\alpha) = 1$ *and* $W(\beta_1 \oplus \beta_2) = 1$
If $\alpha = 10_x$,

$$\begin{aligned} |NS_k(\alpha, \beta_1) \cdot NS_k(\alpha, \beta_2)| &\leq 48 \text{ for } \beta_1 \oplus \beta_2 = 1 \\ |NS_l(\alpha, \beta_1) \cdot NS_l(\alpha, \beta_2)| &\leq 48 \text{ for } \beta_1 \oplus \beta_2 = 4 \end{aligned}$$

for $k = 5, 8$ *and* $l = 6$.

5 Comparison

Using the same method in [3], we have successfully found a set of DES-like S-boxes satisfying additional 5 conditions described in Section 4. In the appendix, we listed a set of DES-like S-boxes for s^5DES [2] which took about three monthes to generate by Hyundai Axil HWS310 Sparc workstation (22MIPS, 33MHz).

In this section, we measure the cryptographic strength of S-boxes themselves and evaluate the breaking complexity of s^5DES.

[2] The name of s^4DES is skipped with intention since it was distributed in an informal way.

5.1 Local Properties

We compare the quantitative characteristics of S-boxes in DES and s^iDES in various points of cryptographic view and evaluate the goodness-of-fit of them.

We checked the nonlinearity of 4 Boolean functions: $Z_2^6 \rightarrow Z_2$ consisting of an S-box as shown in Table 1. In the output bit column of this table, 4 denotes the most significant location of an output vector and 1 denotes the least significant location of an output vector.

Table 1: Nonlinearity of S-boxes

Box	DES				s^2DES				s^3DES				s^5DES			
	output bit				output bit				output bit				output bit			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
S1	18	20	22	18	22	20	20	22	16	20	22	20	20	18	20	22
S2	22	20	18	18	24	22	22	22	22	22	20	22	18	20	20	20
S3	18	22	20	18	20	24	22	22	18	22	20	20	22	22	20	18
S4	22	22	22	22	20	22	22	22	18	24	20	20	20	22	18	22
S5	22	20	18	20	22	24	22	24	20	18	18	22	20	22	20	22
S6	20	20	20	20	22	22	20	22	22	20	18	12	20	22	18	20
S7	18	22	14	20	22	20	22	18	22	18	18	16	22	22	20	20
S8	22	20	20	22	22	22	22	22	20	22	22	20	22	20	18	22

It is clear that the nonlinearity of DES S-boxes ranges from 14 to 22, the nonlinearity of s^2DES and s^5DES ranges from 18 to 24, and the nonlinearity of s^3DES varies from 12 to 24. From these observation, we could say that the nonlinearity of DES-like S-boxes is required to be over 18.

We measured the differential characteristics of a S-box as shown in Table 2.

Table 2: Differential characteristics of S-boxes

Box	DES			s^2DES			s^3DES			s^5DES		
	μ_d	σ_d	λ_d	μ_d	σ_d	λ_d	μ_d	σ_d	λ_d	μ_d	σ_d	λ_d
S1	79.49	3.76	16	84.38	3.44	14	73.54	4.09	20	74.12	4.07	18
S2	78.61	3.83	16	85.25	3.39	14	75.49	3.97	18	73.63	4.07	20
S3	79.69	3.78	16	84.38	3.34	14	73.44	4.12	18	72.85	4.06	20
S4	68.55	4.18	16	83.40	3.54	16	73.44	4.25	20	74.41	4.09	20
S5	76.56	3.86	16	82.91	3.57	16	70.61	4.41	20	74.61	4.07	20
S6	80.47	3.69	16	83.98	3.48	16	71.19	4.30	20	75.98	3.92	18
S7	77.25	3.95	16	81.93	3.62	16	75.39	3.99	20	73.54	4.08	18
S8	77.15	3.82	16	82.81	3.54	16	75.20	4.01	20	75.0	4.04	18

The uniformity of s^3DES and s^5DES in XOR distribution tables could be said to be worse than that of DES and s^2DES but s^3DES was verified to be stronger than DES and s^2DES from DC. In [7], Seberry et al. proposed a new measure of checking the robustness of S-box against DC. As defined in Section 3, let λ_d denote the largest value in the XOR distribution table of DES-like S-box and N denote the number of nonzero entries in the first column of the table. In either case the value 2^6 in the first row is not counted. R-robustness against differential cryptanalysis can be defined by $R = (1 - \frac{N}{2^6})(1 - \frac{\lambda_d}{2^6})$. We have checked the R-robustness of DES and s^iDES as shown in Table 3. All DES-like cryptosystems have

Table 3: R-robustness of S-boxes

	DES		s^2DES		s^3DES		s^5DES	
Box	N	R	N	R	N	R	N	R
S1	37	0.316	37	0.329	36	0.301	32	0.359
S2	33	0.363	42	0.269	34	0.337	34	0.322
S3	37	0.316	41	0.280	31	0.371	36	0.301
S4	24	0.469	41	0.270	35	0.312	31	0.354
S5	31	0.387	41	0.270	31	0.355	35	0.312
S6	33	0.363	40	0.281	26	0.408	32	0.359
S7	35	0.340	37	0.316	34	0.322	32	0.359
S8	36	0.328	36	0.328	35	0.312	36	0.314

the similar value of R-robustness. Thus it is difficult to decide which DES-like cryptosystem can resist harder against DC from the uniformity of XOR distribution tables and R-robustness.

The dependence matrix of a DES-like S-box are examined to check if DES-like S-boxes satisfy the SAC (Strict Avalanche Criterion). The dependence matrix $\mathbf{P} = (p_{i,j})$ of the S-box is defined as follows: The element $p_{i,j}$ of $\mathbf{P}$ is the probability that the output variable $\mathbf{y_j}$ of the S-box changes when the input variable $\mathbf{x_i}$ is complemented. The average values, *i.e.*, $(p_{i,1} + p_{i,2} + p_{i,3} + p_{i,4})/4$ of $(p_{i,j})$ of S-boxes in DES and s^iDES are compared in Table 4. Whenever one bit of input in s^2DES is complemented, every output bits can be said to change with close to the probability $\frac{1}{2}$. But, the probability of dependence matrix in DES, s^3DES and s^5DES is found to have greater value than 0.6. Therefore we can infer that Boolean functions consisting of DES, s^3DES and s^5DES do not satisfy the SAC.

Table 4: Average $(p_{i,j})$ values of S-boxes

Box	DES	s^2DES	s^3DES	s^5DES
S1	0.620	0.495	0.609	0.633
S2	0.633	0.510	0.609	0.625
S3	0.661	0.505	0.617	0.620
S4	0.615	0.521	0.617	0.617
S5	0.633	0.516	0.617	0.641
S6	0.651	0.516	0.620	0.628
S7	0.656	0.516	0.638	0.630
S7	0.625	0.508	0.625	0.625

Finally we checked the uniformity of linear distribution table of a S-box as shown in Table 5. We can see that there was no design criteria of DES against LC but the maximal entry in s^5DES is restricted being under 16.

Thus we cannot tell which measure is the best one to check the immunity of DES-like S-boxes against DC and LC.

Table 5: Uniformity of Linear distribution table

	DES		s^2DES		s^3DES		s^5DES	
Box	μ_l	λ_l	μ_l	λ_l	μ_l	λ_l	μ_l	λ_l
S1	75.87	18	77.46	14	62.22	16	68.15	16
S2	69.95	16	77.67	14	68.57	16	68.68	16
S3	77.14	16	79.15	14	68.78	16	71.43	16
S4	58.84	16	75.45	14	65.08	24	71.64	16
S5	75.77	20	74.39	18	71.01	24	72.91	16
S6	76.51	14	78.62	14	69.95	20	72.49	16
S7	75.56	18	76.83	16	73.33	20	71.32	16
S8	73.12	16	77.57	14	70.69	16	71.32	16

5.2 Global Properties

5.2.1 Breaking Complexity by DC

Since the 2-round iterative differential characteristics are directly employed for the successful DC, we compare the probability of 2-round differential characteristics and the breaking complexity of 13-round DES-like cryptosystems by using them as shown in Table 6. This table tells us that the breaking complexity of DES and s^2DES by DC is more efficient than by key-exhaustive search. However, the attacking by DC is not useful to break s^3DES and s^5DES. Also the Improved Davies' attack cannot be applicable to break s^3DES and s^5DES due to their design criteria.

Table 6: Breaking complexity by DC

	Best Char.	Complexity
DES	19600000_x with 1/234	2^{47}
s^2DES	00000580_x with 1/51	2^{33}
	$07e00000_x$ with 1/68	2^{35}
	$5c000000_x$ with 1/68	2^{35}
s^3DES	11173737_x with 1.42576×10^{-5}	2^{96}
s^5DES	75175117_x with 1.15513×10^{-5}	2^{96}
	75175317_x with 1.15513×10^{-5}	2^{96}
	75177117_x with 1.02678×10^{-5}	2^{96}
	75377317_x with 1.02678×10^{-5}	2^{96}

5.2.2 Breaking Complexity by LC

In [18], Lee et al. proposed an efficient method to find the linear approximation of DES-like cryptosystems by LC. By their method, we have checked the best probability of linear approximation of 4 DES-like cryptosystems as shown in Table 7.

From this table, we can see that the breaking s^5DES by LC needs $(1.94 \times 10^{-9})^{-2} \simeq 2^{57.88}$ complexity which is greater than the breaking complexity of key exhaustive search.

Table 7: Best linear approximation

Round	10	12	14	16
DES	4.66×10^{-5}	9.07×10^{-6}	5.67×10^{-7}	8.88×10^{-8}
s^2 DES	3.62×10^{-6}	2.09×10^{-7}	1.59×10^{-8}	9.17×10^{-10}
s^3 DES	5.15×10^{-5}	1.20×10^{-5}	6.03×10^{-7}	7.07×10^{-8}
s^5 DES	8.89×10^{-7}	6.94×10^{-8}	1.94×10^{-9}	1.82×10^{-10}

6 Concluding Remarks

Our systematic approach to immunize DES against three robust attacks is not only verified to enhance the security of DES, but also is very simple since the current DES S-boxes can be substituted with new S-boxes without changing other components of DES. We can conclude that three attacks to s^5DES are no more efficient than the key exhaustive search attack. In [17], Biham and Biryukov proposed some methods to strengthen the power of DES replaced by DES-like S-boxes of s^3DES in a switched order against the key exhaustive search attack. Their methods can also be applicable directly without changing the location of 8 DES-like of s^5DES to enhance the security of s^5DES against the key exhaustuve search attack.

Finally, further works are left as open problems to evaluate that s^5DES is resistant against differential-linear attack [15] and multiple linear attack [16].

References

[1] "Data Encryption Standard", FIPS-Pub. 46, National Bureau of Standards (former NIST), 1977.

[2] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", J. of Cryptology, Vol.4, pp.3–72, 1991.

[3] K. Kim, "Construction of DES-like S-boxes Based on Boolean Functions Satisfying the SAC", Advances in Cryptology-Asiacrypt'91, Springer-Verlag, pp.59–72, Fujiyoshida, Japan, 1991.

[4] E. Biham and A. Shamir, "Differential Cryptanalysis of the full 16-round DES", Advances in Cryptology-Crypto'92, Springer-Verlag, pp.487–496, 1992.

[5] K. Kim, S. Park and S. Lee, "Reconstruction of s^2DES S-boxes and their Immunity to Differential Cryptanalysis", Proc. of 1993 Korea-Japan Joint Workshop on Information Security and Cryptology (JW-ISC'93), Oct.24-26, Seoul, 1993.

[6] M. Matsui, "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology-Eurocrypt'93, Springer-Verlag, pp.386–397, 1993.

[7] J. Seberry, X. Zhang, and Y. Zheng, "Systematic Generation of Cryptographically Robust S-boxes", Proc. of the 1st ACM Conf. on Comp. and Comm. Security, pp.172–182, ACM, 1993.

[8] K. Kim, S. Lee, and S. Park, "Necessary Conditions to Strengthen DES S-boxes against Linear Cryptanalysis", Proc. of SCIS'94, Biwako, Japan, pp.15D.1-11, Jan.27–29, 1994.

[9] K. Kim, S. Lee, S. Park, and D. Lee, "DES can be Immune to Linear Cryptanalysis", Workshop Record of SAC'94 (Selected Areas in Cryptography), May 5–6, Queen's Univ., Canada, 1994.

[10] T. Sorimachi, T. Tokita, and M. Matsui, "On a Cipher Evaluation Method Based on Differential Cryptanalysis (in Japanese) ", Proc.of SCIS'94, SCIS94-4C, Lake Biwa, Japan, Jan.27–29, 1994.

[11] T. Tokita, T. Sorimachi, and M. Matsui, "An Efficient Search Algorithm for the Best Expression on Linear Cryptanalysis (in Japanese) ", Technical Report on Information SECurity of IEICE, ISEC93-97, Mar., 1994.

[12] D. Coppersmith, "The Data Encryption Stanadard (DES) and its strength against attacks", IBM J. , Vol.38, No.3, pp.243–250,1994

[13] E. Biham and A. Biryukov, "An Improvement of Davies' Attack on DES", Proc. of Eurocrypt'94, to apprear,

[14] M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard", Advances in Cryptology-Crypto'94, Springer-Verlag, pp.1–11, 1994.

[15] S.K. Langford and M.E. Hellman, "Differential-Linear Cryptanalysis", Advances in Cryptology-Crypto'94, Springer-Verlag, pp.17–25, 1994.

[16] B.S. Kaliski and M.J.B. Robshaw, "Linear Cryptanalysis Using Multiple Approximations", Advances in Cryptology-Crypto'94, Springer-Verlag, pp.26–39, 1994.

[17] E. Biham and A. Biryukov, "How to Strength DES Using Existing Hardware", Pre-Proc. of Asiacrypt'94, pp.339–353, Univ. of Wollongong, Australia, 1994.

[18] S. Lee, S. Sung, and K. Kim "An Efficient Method to Find the Linear Expressions for Linear Cryptanalysis", Proc. of JW-ISC'95. Jan.24-27, Inuyama, Japan, 1995.

Appendix: 8 DES-like S-boxes of s^5DES

S1-box

9	10	15	1	4	7	2	12	6	5	3	14	8	11	13	0
2	13	8	4	11	1	14	7	12	3	15	9	5	6	0	10
10	12	4	7	9	2	15	1	3	6	13	8	14	5	0	11
4	11	1	13	14	7	8	2	10	0	6	3	9	12	15	5

S2-box

6	3	5	0	8	14	11	13	9	10	12	7	15	4	2	1
9	6	10	12	15	0	5	3	4	1	7	11	2	13	14	8
5	8	3	14	6	13	0	11	10	15	9	2	12	1	7	4
6	3	15	9	0	10	12	5	13	8	2	4	11	7	1	14

S3-box

11	5	8	2	6	12	1	15	7	14	13	4	0	9	10	3
7	8	1	14	11	2	13	4	12	3	6	9	5	15	0	10
8	11	1	12	15	6	2	5	4	7	10	9	3	0	13	14
13	2	4	7	1	11	14	8	10	9	15	0	12	6	3	5

S4-box

13	11	8	14	3	0	6	5	4	7	2	9	15	12	1	10
10	0	3	5	15	6	12	9	1	13	4	14	8	11	2	7
6	5	11	8	0	14	13	3	9	12	7	2	10	1	4	15
9	12	5	15	6	3	0	10	7	11	2	8	13	4	14	1

S5-box

12	6	2	11	5	8	15	1	3	13	9	14	0	7	10	4
15	0	12	5	3	6	9	10	4	11	2	8	14	1	7	13
1	12	15	5	6	11	8	2	4	7	10	9	13	0	3	14
6	3	10	0	9	12	5	15	13	4	1	14	7	11	8	2

S6-box

14	8	2	5	9	15	4	3	7	1	12	6	0	10	11	13
1	13	11	8	2	4	7	14	10	6	0	15	5	9	12	3
4	2	9	15	14	8	3	5	10	7	0	12	13	1	6	11
8	11	7	4	13	1	14	2	5	0	9	10	6	15	3	12

S7-box

4	13	10	3	7	0	9	14	2	1	15	6	12	11	5	8
9	0	15	10	12	6	5	3	14	7	1	13	11	8	2	4
13	10	3	9	0	7	14	4	8	6	5	12	11	1	2	15
10	3	12	6	5	9	0	15	4	8	11	1	14	7	13	2

S8-box

1	10	2	12	15	9	4	7	14	3	5	0	8	6	11	13
14	13	7	11	2	4	1	8	0	10	9	6	5	15	12	3
10	15	12	1	9	2	7	4	13	0	6	11	3	5	8	14
4	8	1	2	7	11	13	14	10	5	15	12	0	6	3	9

DESV: A Latin Square Variation of DES

G.Carter[‡],E.Dawson[†] and L.Nielsen[†]

[†] Information Security Research Centre
[‡] School of Mathematics
Queensland University of Technology
GPO Box 2434 Brisbane 4001
Queensland Australia

Abstract.

Many variations of DES which do not alter the key length have been proposed. Biham and Shamir have shown that many of these proposed variations are more susceptible to differential cryptanalysis than DES. In this paper we propose another variation which replaces the exclusive-or (XOR) operation of DES with a latin square. We will show that the resulting cipher is more resistant to differential and linear cryptanalysis.

1 Introduction

Recently, Biham and Shamir [1] and Matsui [4] have shown that the DES algorithm may be susceptible to attacks by the methods of differential and linear cryptanalysis. Many changes to DES which do not alter the key length have been proposed to prevent these attacks. Biham and Shamir [1] have shown that many of these variations are more susceptible to differential cryptanalysis than DES, itself.

We propose a replacement for DES called DESV which has the same structure and key size of DES. Our implementation involves replacing the XOR operation of DES with the operation $*$ as defined by a latin square. In DESV, the inputs to $*$ are two 32-bit blocks. Each is considered to consist of k successive subblocks (k = 4,8,16) each consisting of $\frac{32}{k}$ bits. The corresponding latin square is of order $2^{32\;k}$. Corresponding subblocks of the two input blocks are acted upon by the operation $*$ as per the defining latin square. This is done for each subblock and the results are concatenated to form the 32-bit input block to the next round.

The operation $*$ is performed by means of a look-up table. We evaluated the efficiency of the $*$ operation in relation to encryption/decryption speeds in software for latin squares of order 4, 16 and 256. The results are given in Table 1, showing comparable times for a 20 megabyte datafile. DES is also included for comparison.

k	4	8	16	DES
Time (mins)	8.9	8.7	10.8	6.7

Table 1: Encryption speed for a 20 megabyte file

A brief review of latin squares is provided in Section 2 where it is shown that the $*$ operation needs to be a group in order to encrypt and decrypt. In Section 3 an overview of the methods used to attack the DESV cipher in relation to linear and differential cryptanalysis is given.

The results of the analysis in relation to differential and linear cryptanalysis are concentrated on latin squares of order 16 since, as Table 1 demonstrates, latin squares of order 16 seem to offer the *best* size in terms of encryption and decryption speeds. We selected a latin square of order 16 defined by modulo 17 multiplication, denoted $\otimes$, (using the same operation as performed in the IDEA algorithm [3]). We called the resulting cipher DESV-1. In Sections 4 and 5 it is demonstrated that DESV-1 is more resistant to differential and linear cryptanalysis, respectively, than DES.

The group structure of $*$ in DESV-1 is abelian and we conjecture that perhaps. as far as differential and linear cryptanalysis are concerned, a non-abelian group structure would be better. There are 14 non-isomorphic latin squares of order 16, 9 of which have an underlying group structure that is non-abelian [2]. We generated each of the latin squares and we present results in relation to their immunity to differential and linear cryptanalysis in Section 5.

2 Latin Squares

For our purposes, a *latin square of order n* is an $n \times n$ array where elements are the integers 0 to $n-1$ inclusive, written $[0, n-1]$ and arranged in such a way that each integer appears once only in every row and column, e.g.

$$\begin{array}{c|cccc} & 0 & 1 & 2 & 3 \\ \hline 0 & 1 & 0 & 2 & 3 \\ 1 & 0 & 1 & 3 & 2 \\ 2 & 3 & 2 & 0 & 1 \\ 3 & 2 & 3 & 1 & 0 \end{array}$$

is a latin square of order 4. Note that the bordering row and column is not considered part of the square but will be included for counting purposes.

Using the latin square we can define an operation $*$ on the integers $[0, n-1]$ as follows. If x and y are two such integers then $x * y$ is defined to be the entry in the latin square at the conjunction of the row numbered x and the column numbered y. Since this operation is not necessarily commutative the order is important. If the latin square defined by $*$ exhibits group properties then we can decrypt the ciphertext as shown below.

In DESV, the final round of output is $R_{16} = R_{15}$ and $L_{16} = L_{15} * f(R_{15}, K_{16})$. where L_i and R_i stand for the left and right 32-bit output blocks in round i, and f is the round function of DESV, i.e. the DES S-boxes. Let ${}_jR_{15}, {}_jL_{16}, {}_jL_{15}, {}_jK_{16}$ be the jth block ($1 \leq j \leq 8$) of $\frac{32}{k}$ bits of $R_{15}, L_{16}, L_{15}, L_{16}$ respectively. In the first round of decryption we have

$$\begin{aligned}
{}_jf(R_{15},K_{16}) *_j L_{16}^{-1} &= {}_jf(R_{15},K_{16}) * \langle {}_jL_{15} *_j f(R_{15}, K_{16})\rangle^{-1} \\
&= {}_jf(R_{15},K_{16}) * \langle {}_jf^{-1}(R_{15}, K_{16}) *_j L_{15}^{-1}\rangle \\
&= \langle {}_jf(R_{15},K_{16}) *_j f^{-1}(R_{15}, K_{16})\rangle *_j L_{15}^{-1} \\
&= {}_jL_{15}^{-1}
\end{aligned}$$

${}_jL_{16}^{-1}$ and ${}_jL_{15}^{-1}$ are the inverses of ${}_jL_{16}$ and ${}_jL_{15}$ respectively. We now determine ${}_jL_{15} ={}_j R_{14}$ and repeat for round 2. This procedure can be repeated until all 16 rounds are decrypted.

It should be noted that the decryption procedure outlined above required all four properties of a group in relation to $*$ operation. Hence, if the $*$ operation does not define a group then decryption is not possible.

3 Analysis of DESV

3.1 Differential Cryptanalysis of DESV

The notion of difference will be as for the DES; that is, given X and $\bar{X}$, the difference, $X' = X \ominus \bar{X}$ where $\ominus$ indicates the XOR operation [1]. The initial attempt to define the difference X' to be $X * (\bar{X})^{-1}$ failed, as the principles of differential cryptanalysis require expressions of the type $(X \ominus K) * (\bar{X} \ominus K)^{-1}$ to eliminate the key, leaving $X * (\bar{X})^{-1}$. This does not happen, hence the original notion of difference as it is used in the attack on the DES, is applied to the attack on DESV. Let a_i and $\bar{a}_i$ be 32-bit blocks which, after expanding and XORing with key K_i, are inputs to the F-function in round i of DESV. Let $a_i' = a_i \ominus \bar{a}_i$. Let A_i and $\bar{A}_i$ be the respective outputs of F (after the permutation) for inputs $a_i \ominus K_i$ and $\bar{a}_i \ominus K_i$. Let $A_i' = A_i \ominus \bar{A}_i$. Thus, in round i, the r-bit input differences to the operation $*$ are ${}_jA'_i$ and ${}_ja'_{i-1}, 1 \leq j \leq k$. For given differences, a'_i, Biham and Shamir determined A'_i with certain probability as per the so-called XOR tables [1]. In the DES, ${}_ja'_{i-1} = {}_jA'_i \ominus {}_ja'_{i-1}$ for *all* pairs $({}_jA_i, {}_j\bar{A}_i)$ and $({}_ja_{i-1}, {}_j\bar{a}_{i-1})$. However, in DESV, in most cases, ${}_ja'_{i-1} \neq {}_jA'_i * {}_ja'_{i-1}$. In fact, ${}_ja'_{i+1}$ will vary depending on the pairs $({}_jA_i, {}_j\bar{A}_i)$ and $({}_ja_{i-1}, {}_j\bar{a}_{i-1})$. Thus, given ${}_jA'_i$ and ${}_ja'_{i-1}$, ${}_ja'_{i+1}$ can be determined only probabilistically. Such probabilities can be determined and can be tabulated in an XOR table which we will refer to as the XMOD table. Table 2 contains an example of the XMOD Table for $r = 4$ and the $*$ operation represents modulo 17 multiplication. Entries in the first two columns indicate the input differences to operation $*$ in round i, one from the previous round and the other from the F-function in the ith round. The top row is the output difference of operation $*$. The entries in the main body of the table represent the number of times out of 256, given input differences yield a given output difference. It follows that the input difference to round $i + 1$ can only be determined probabilistically from a knowledge of A'_i and a'_{i-1}. This probability is called a *cross probability*. For example, in Table 2 the largest cross probability is $\frac{72}{256}$. If we define a characteristic in a similar manner to Biham and Shamir, its probability must incorporate such cross probabilities.

3.2 Linear Cryptanalysis of DESV

In linear cryptanalysis of DES each round is approximated by an expression of the kind $X[a_1, a_2, \cdots, a_n] \ominus F(X, K)[b_1, b_2, \cdots, b_n] = K[c_1, c_2, \cdots, c_k]$, where, in any given round, $X[a_1, a_2, \cdots, a_n]$ indicates the XOR sum of the plaintext bits entering the S-boxes of DES, $F(X, K)[b_1, b_2, \cdots, b_n]$ is the XOR sum of the output bits of the DES S-boxes after the permutation, and $K[c_1, c_2, \cdots, c_k]$ is the XOR sum of the key bits. Such expressions hold

X1'	X2'	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	256	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	40	12	12	8	16	16	16	0	24	16	16	16	40	12	12
0	2	0	12	20	18	20	18	18	18	20	16	16	18	16	10	18	18
0	3	0	12	18	24	18	14	16	22	20	16	22	16	10	6	24	18
0	4	0	8	20	18	24	18	18	18	24	12	16	18	16	10	18	18
0	5	0	16	18	14	18	32	18	12	16	12	14	18	22	12	14	20
0	6	0	16	18	16	18	18	20	18	16	12	18	20	22	10	16	18
0	7	0	16	18	22	18	12	18	24	16	12	22	18	14	8	22	16
0	8	0	0	20	20	24	16	16	16	32	16	16	16	16	8	20	20
0	9	0	24	16	16	12	12	12	12	16	40	12	12	4	36	16	16
0	10	0	16	16	22	16	14	18	22	16	12	24	18	16	6	22	18
0	11	0	16	18	16	18	18	20	18	16	12	18	20	22	10	16	18
0	12	0	16	16	10	16	22	22	14	16	4	16	22	40	14	10	18
0	13	0	40	10	6	10	12	10	8	8	36	6	10	14	72	6	8
0	14	0	12	18	24	18	14	16	22	20	16	22	16	10	6	24	18
0	15	0	12	18	18	18	20	18	16	20	16	18	18	18	8	18	20

Table 2: Extract of XMOD table

with certain probabilities and can be determined for each S-box. The aim is to link equations of this type together for several rounds and obtain a linear expression which involves only plaintext bits and ciphertext bits. The links between the rounds are provided by expressions of the kind $X_n[a_1, a_2, \cdots, a_t] \ominus F_{n-1}(X_{n-1}, K_{n+1})[b_1, b_2, \cdots, b_s] = X_{n+2}[c_1, c_2, \cdots, c_p]$ which hold with probability 1 in DES. In 3-round DES. for example: In round 1, $P_L[15] \ominus F_1[P_L, K_1)[7, 18, 24, 29] = K_1[22]$ holds with probability $\frac{12}{64}$ and $P_H[7, 18, 24, 29] \ominus F_1(P_L, K_1)[7, 18, 24, 29] = X_2[7, 18, 24, 29]$ holds with probability 1. In round 3, $C_L[15] \ominus F_3(C_L, K_3)[7, 18, 24, 29] = K_3[22]$ holds with probability $\frac{12}{64}$ and $X_2[7, 18, 24, 29] \ominus F_3(C_L, K_3)[7, 18, 24, 29] = C_H[7, 18, 24, 29]$ holds with probability 1. Therefore, summing these four equations we get a linear approximation for three rounds of DES as follows: $P_L[15] \ominus P_H[7, 18, 24, 29] \ominus C_L[15] \ominus C_H[7, 18, 24, 29] = K_1[22] \ominus K_3[22]$. Its probability can be determined and is 0.6953. Note that the notation P_L, P_H, C_L, C_H is used as in Matsui [4].

The critical difference in DESV is that the linking expression $X_n[a_1, a_2, \cdots, a_t] \oplus F_{n+1}(X_{n-1}, K_{n+1})[b_1, b_2, \cdots, b_s] = X_{n-2}[c_1, c_2, \cdots, c_p]$ does not hold with probability 1. It is actually a linear approximation of the operation $*$, and hence holds with probability less than 1. Thus, the probabilities of the linear approximations of DESV must incorporate the probabilities of the linking expressions. We call these probabilities *linking probabilities.*

We can determine the linking probabilities p and tabulate $|p - \frac{1}{2}|$ as Matsui did. An extract of such a table for the latin square defined by mod 17 multiplication is presented in Table 3. The first column of the table represents which bits of input are to be XOR'd, e.g. 18_x indicates that the least significant bit of one input is to be XOR'd with the most significant bit of the other. The top row represents which bits of the output should be XOR'd. The entries in the main body of the table represent the difference from $\frac{1}{2}$ of the probability that the XOR sum of input bits is equal to the XOR sum of output bits. The largest probability in the table is 0.625.

hex	1	2	3	$\cdots$	14	15
18_x	0.0625	0.0000	0.0312	$\cdots$	0.0312	-0.0312
19_x	-0.0625	0.0000	0.0000	$\cdots$	-0.0156	-0.0156
$1a_x$	-0.0312	0.0000	-0.0625	$\cdots$	0.0000	0.0938
$1b_x$	-0.0625	0.0000	0.0000	$\cdots$	-0.0156	-0.0156
$1c_x$	-0.0156	0.0000	0.0156	$\cdots$	-0.0156	-0.0312
$1d_x$	-0.0781	0.0000	0.0469	$\cdots$	-0.0312	0.0156
$1e_x$	-0.0156	0.0000	0.0156	$\cdots$	-0.0156	-0.0312
$1f_x$	-0.0469	0.0000	-0.0469	$\cdots$	0.0000	-0.0469

Table 3: $|p - \frac{1}{2}|$ for linking probabilities p

4 The Differential Attack on DESV-1

We have developed a theoretical attack that will work on any number of rounds of DESV-1 provided a characteristic of suitably high probability can be found. In the XMOD table, as shown in Table 2, we exploit the fact that, if the input differences to the operation $*$ are 0 and 13, then the output difference will be 1, 9 or 13 with probability $\frac{148}{256} > 0.5$. Consider the last two rounds of an n-round version of DESV-1. The attack requires some of the 4-bit subblocks ${}_ja'_{n-1}$, $1 \leq j \leq 8$ to be zero blocks. This, generally, can be arranged with good probability by choosing a suitable characteristic. So we may suppose that such a zero block exists. Generate enough pairs so that the left ciphertext output difference, ${}_jC'_L$ equals thirteen. By examining all input pairs $({}_ja_{n-1}, {}_j\bar{a}_{n-1})$ to the operation $*$ such that ${}_ja'_{n-1}$ equals zero, and all the output pairs $({}_jC_L, {}_j\bar{C}_L)$ such that ${}_jC'_L$ equals thirteen, we can determine the other input difference to $*$, ${}_jA'_n$. Note that if the pair $({}_jC_L, {}_j\bar{C}_L) \in S$, where $S = \{(2,15),(3,14),(6,11),(7,10)\}$ then ${}_jA'_n$ equals 1, 9 or 13 with probability one rather than $\frac{148}{256}$; that is, ${}_jA'_n$ equals 0001, 1001 or 1101 always. Thus we can say that the first bit of ${}_jA'_n$ is one with probability $\frac{12}{16}$, the second bit is one with probability one half, the third bit is zero always, and the fourth bit is one always. The task is to generate enough pairs such that $({}_jC_L, {}_j\bar{C}_L) \in S$. If this can be done, then certain bits of A'_n are known with high probability and the principles of differential cryptanalysis can be applied to find those keys which are involved in generating the first, third and fourth bits of ${}_jA'_n$. This attack has been successfully used on a four-round version of DESV-1, the characteristic being the one used by Biham and Shamir to break six rounds of the DES, $4008\,0000\,0400\,0000_x$, [1]. Generate enough pairs so that $({}_1C_L, {}_1\bar{C}_L)$ and $({}_6C_L, {}_6\bar{C}_L) \in S$. Thus, for such pairs, ${}_1A'_4$ and ${}_6A'_4$ will be 0001, 1001 or 1101 since ${}_1a'_3$ and ${}_1a'_6$ equals zero. Now, allowing for the permutation, this implies (using the bit numbering system 1,2,3,..,32) that bits 16 and

32 of the output difference of the F-function will each be one with probability $\frac{12}{16}$, bits 3 and 20 will have a difference of zero always, and bits 9 and 21 will have a difference of one always. Applying the differential cryptanalysis method, we find the keys to S-boxes 1,3,4,5,6 and 8 in the last round; that is thirty-six bits of the last-round key. If a minimum of 100,000 pairs are used then in all our trials all thirty-six bits of the key were found correctly. The characteristic with the highest probability is a compromise between the F-function probabilities and the cross probabilites. We have not yet found a method which yields the best compromise. The probability of this characteristic in the attack on DESV-1 is made up as follows: $\frac{1}{4} \times (\frac{24}{256} \times \frac{32}{256}) \times \frac{1}{32} \approx 2^{-13}$. The first factor in the above expression is the F-function probability from the XOR tables. The second factor, $(\frac{24}{256} \times \frac{32}{256})$, is the cross probability from Round 1 to Round 2 from the XMOD table. The last factor is the probability that, given ${}_ja'_3$ equals zero then $({}_jC_L, {}_j\bar{C}_L) \epsilon S$. Note that the cross probability from Round 2 to Round 3 is not required as it does not matter what the non-zero component of a'_3 is. We only require some of the ${}_ja'_3$ to be zero.

If we use Biham and Shamir's iterative characteristic, $1960\,0000\,0000\,0000_x$ [1] to attack any number of rounds, we can make some comparisons with the DES in regards to the complexity of the attack. To break n rounds of the DES this characteristic probability is given by $(\frac{14\times8\times10}{64^3})^{\frac{n-2}{2}}$ while for DESV-1 the corresponding probability is given by $\frac{1}{32}(\frac{14\times8\times10}{64^3})^{\frac{n-2}{2}}(\frac{40^2\times20}{256^3})^{\frac{n-2}{2}}$. These results are produced in Table 4 for $n = 4.\ 8, 12, 16$ rounds. The iterative characteristic may not be the one with highest probability, so one cannot draw the conclusion that eight rounds of DESV-1 is as secure from a differential attack as fifteen rounds of the DES. However, it appears certain that fewer than sixteen rounds of DESV-1 will be as secure from a differential attack as the full sixteen round DES. This helps offset the loss of speed in encryption and decryption.

$n =$	4	8	12	16
DES	2^{-8}	2^{-24}	2^{-40}	2^{-56}
DESV-1	2^{-22}	2^{-56}	2^{-90}	2^{-124}

Table 4:Probabilities of the Iterative Characteristics

The iterative characteristic may not be the one with the best 16-round probability, but we can show that the probability of the best 16-round characteristic is at most 2^{-57} and, hence differential cryptanalysis is worse than exhaustive search. The figure of 2^{-57} is obtained by realising that the best possible scenario for a characteristic path through sixteen rounds of DESV-1 is that in any round, the middle two bits of the input difference ${}_j a_i'$ should be non-zero and the output difference ${}_k A_i'$ should contain only one non-zero bit. In this case, the largest crossing probability is $\frac{38}{256}$ and the largest S-box probability is $\frac{1}{4}$, hence the probability associated with each round is at best $(\frac{1}{4} \times \frac{38}{256})$. Using a 13-round characteristic in a 3R-attack, the best characteristic probability is given by $(\frac{1}{4} \times \frac{38}{256})^{12} \sim 2^{-57}$.

5 Linear Cryptanalysis of DESV-1

Linear cryptanalysis of DES involves summing round approximations to get an approximate expression for the cipher. The probability of this expression holding depends on the number of round approximations needed and their probability. However, an extra round approximation increases dramatically the number of pairs required to determine key bits. Thus, in the linear cryptanalysis of DESV-1, we aim to minimise the number of round approximations and linking expressions required. To do this we must severely restrict allowable round approximations. Given the round approximation $X[a_1, a_2, \cdots, a_m] \ominus F(X,K)[b_1, b_2, \cdots, b_n] = K[c_1, c_2, \cdots, c_k]$ and allowing for the DES permutation, then if $n = 2$, 3 or 4 then bits

b_i in the expression $F(X, K)[b_1, b_2, \cdots, b_n]$ appear in 2, 3 or 4 different subblocks of the input to the operation $*$ from the S-boxes. Thus, to eliminate $F(X, K)[b_1, b_2, \cdots, b_n]$ in the summing process would require 2, 3 or 4 linking expressions, one for each subblock in which the b_i's appear. Thus, if our goal is to minimise the number of round approximations and linking expressions to be summed, we must restrict round approximations to the form $X[a_1, a_2, \cdots, a_m] \oplus F(X, K)[b_1] = K[c_1, c_2, \cdots, c_k]$. This will ensure only one linking expression is required per round. From Matsui's tables [4], the best probability of a round function of the above type is $\frac{36}{64}$ or $\frac{28}{64}$. We have applied this technique to successfully break three and four round versions of DESV-1 as follows.

In the three round version we only use linking equations of the form $X_q[4n] \oplus F_{q-1}(X, K)[4n] = X_{q+2}[4n]$ as follows

$$\begin{aligned} P_H[4n] &\oplus F_1(P_L, K_1)[4n] &= X_2[4n] \\ X_2[4n] &\oplus F_3(C_L, K_3)[4n] &= C_H[4n], n = 0, 1, 2. \cdots, 7. \end{aligned}$$

Summing these we have

$$P_H[4n] \oplus F_1(P_L, K_1)[4n] \oplus C_H[4n] \oplus F_3(C_L, K_3)[4n] = 0.$$

With 12000 pairs we were able to determine bits of the key as shown in Table 5. Overall this gives us a total of 48 bits of the key.

For four rounds, we used three linking expressions and one round approximation as follows:

$$\begin{aligned} P_H[4n] &\oplus F_1(P_L, K_1)[4n] &= X_2[4n] \\ X_2[4n] &\oplus F_3(X_3, K_3)[4n] &= C_L[4n] \\ X_3[4n] &\oplus F_3(C_3, K_3)[4n] &= K_3[t] \\ X_3[4n] &\oplus F_4(C_L, K_4)[4n] &= C_H[4n]. n = 0, 1, 2. 3, \cdots, 7. \end{aligned}$$

On summarising, we have

$$P_H[4n] \oplus F_1(P_L, K_1)[4n] \oplus C_L[4n] \oplus C_H[4n] \oplus F_3(C_L, K_3)[4n] = K_3[t], n = 3, 5, 6.$$

With 6×10^7 pairs we were able to determine k bits of the key according to Table 5. Overall the three approximations yield a total of twenty-eight different bits of the key.

n	0	1	2	3	4	5	6	7
3 rounds (number of bits)	11	10	11	11	11	11	11	9
4 rounds (number of bits)	-	-	-	11	-	12	12	-

Table 5: Summary of attack on 3 rounds and 4 rounds of DESV-1

In a sense we can iterate the four-round expression to obtain probabilistic expressions for 8, 12 and 16 rounds of DESV-1. The round approximations all have the form $X_z[4n] \ominus F_z(X_z, K_z)[4n] = K_z[t]$ and we will call this expression A. The linking expressions are all of the form $X_m[4n] \oplus F_{m+1}(X_{m+1}, K_{m+1})[4n] = X_{m+2}[4n]$ and we will call this expression B. The 8, 12 and 16 round expressions are constructed from expressions A and B as in Table 6. The bracketed expression in the construction column tells which of the expressions A and B are used in each round. The symbol - indicates that no approximation is needed in that round.

Rounds	*Expression*	*Construction*
8	$P_H \ominus F_1 \ominus C_L \ominus C_H \ominus F_8$ $= K_2 \ominus K_4 \ominus K_5 \ominus K_7$	$(B)(AB) - (AB)(AB) - (AB)(B)$
12	$P_H \ominus F_1 \ominus C_H \ominus F_{12} =$ $K_3 \ominus K_4 \ominus K_6 \ominus K_7 \ominus$ $K_9 \ominus K_{10}$	$(B) - (AB)(AB) - (AB)(AB) - (AB)(AB) - (B)$
16	$P_H \oplus F_1 \ominus C_H \ominus F_{16} =$ $K_2 \ominus K_4 \ominus K_5 \ominus K_7 \ominus$ $K_8 \ominus K_{10} \ominus K_{11} \ominus K_{13} \ominus K_{14}$	$(B)(AB) - (AB)(AB) - (AB)(AB) - (AB)(AB)$ $-(AB)(AB) - (B)$

Table 6: Linear approximations for 8, 12, and 16 rounds of DESV-1

The probabilities of these expressions are so close to one half that linear cryptanalysis is much worse than exhaustive search. In Table 7 we have produced some results which

$n =$	4	8	12	16
DES	$2^{-1.7}$	2^{-8}	$2^{-14.4}$	2^{-21}
DESV-1	2^{-11}	2^{-27}	2^{-38}	2^{-54}

Table 7: Linear attack on DES and DESV-1

compare the linear cryptanalysis of DES to that of DESV-1 in the cases where the number of rounds n equals 4, 8, 12, 16. The figures displayed are actually the value of $|p - \frac{1}{2}|$ where p is the probability of the approximate expression. Since the number of plaintext-ciphertext pairs required is a multiple of the square of the inverse of the figures in the table, it is clear that for 8, 12 and 16 rounds linear cryptanalysis is indeed much worse than exhaustive search.

The method we have used to determine the 8, 12 and 16 round linear expressions yields the best possible expressions in the sense that it minimises the number of round approximations and linking expressions required. To date, we have not found a combination of a round approximation and a linking expression that has a higher probability than the one used.

6 Extensions

As shown in Section 2 the latin square used in DESV-1 could be replaced by any of the other thirteen non-isomorphic latin squares of order 16 defining a group. For each of these we have generated the difference table and the linear approximate table.

In the case of the difference tables, four of the latin squares are improvements on the one used in DESV-1, in the sense that the best possible 16-round characteristic, as defined in Section 4, has a largest crossing probability $< \frac{38}{256}$. The best improvement has greatest crossing probability of $\frac{26}{256}$ and this occurs in three of the four latin squares. In these cases, using a 3R differential attack, the best characteristic probability is given by $(\frac{1}{4} \times \frac{26}{256})^{12} \sim 2^{-64}$. One of these three latin squares also has a largest entry of 48, much less than the 72 for the square used in DESV-1. This would also reduce the likelihood of a successful attack on a 4-round version as carried out on DESV-1 in Section 4.

In the case of the linear approximate tables, three latin squares are improvements on what is used in DESV-1, in the sense that the best linear expressions, as outlined in Section 5, has a best linking expression table entry less than that which appears for DESV-1. One of these latin squares is the same one described as being best in the previous paragraph. It is the dicyclic group $< 2, 2, 4 >$ [2] whose generators S, T satisfy the relations $S^8 = T^4 = E, S^4 = T^2 = (ST)^2$, where E is the identity. However, it is third best as far as linear cryptanalysis is concerned. This latin square yields 2^{-80} for the value of $|p - \frac{1}{2}|$ for the 16-round approximation as outlined in Section 5, Table 6, making linear cryptanalysis much worse than exhaustive search.

All the latin squares which are improvements on the one used in DESV-1 either in the differential or linear sense are non-abelian. Thus, it appears our conjecture that a non-abelian latin square would improve DESV-1 is true.

7 Conclusion

All indications are that eight rounds of DESV-1 or its variant DESV-2, which has the dicyclic group latin square as its operation $*$, are as secure as 16 rounds of DES. To improve the security still further, one could use an extended key to select latin squares. Altogether there are $14 \times 16!$ latin squares of order 16 which define a group with respect to its $*$ operation.

If, for example, there was an extra eight bits of key one could pre-select 256 of these latin squares which provide a security level similar or better than DESV-1 and store the * table for each of these 256 latin squares. The eight extra bits of the key will define in the natural way which of the latin squares to select. In this fashion this adaption should be secure from the machine proposed by Wiener in [6] to perform an exhaustive key attack on DES.

Acknowledgement: The research on this paper was partially supported by a 1995 QUT research grant.

References

[1] E.Biham and A.Shamir.*Differential Cryptanalysis of DES-like Cryptosystems* Journal of Cryptology, 4(1):3-72, 1991.

[2] H.S.M.Coxeter and W.O.J.Moser *Generators and Relations for Discrete Groups*, Springer-Verlag, Berlin, 1972.

[3] X.Lai and J.Massey.*A Proposal for a New Block Encryption Standard*, Advances in Cryptology: Proceedings of EUROCRYPT '90, Springer-Verlag, Berlin, pp 389-404, 1991.

[4] M.Matsui.*Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology: Proceedings of EUROCRYPT '93, Springer-Verlag, Berlin, pp 386-397, 1994.

[5] National Bureau of Standards (U.S.).*Data Encryption Standard (DES)*, tech.rep., Federal Information Processing Standards, 1977. Publication 46.

[6] M.J.Wiener.*Efficient DES Key Search*, Workshop on Selected Areas in Cryptography (SAC '94), Queen's University, Canada, p 1, 1994.

A Low-Power CMOS Private Key Cipher for PCS Terminals

A.R. McKinney and **S.E. Tavares**
Department of Electrical and Computer Engineering
Queen's University, Kingston, Ontario K7L 3N6
email: mckinney@eleceng.ee.queensu.ca
tavares@eleceng.ee.queensu.ca

Abstract

One of the fastest growing areas for encryption services is expected to be Personal Communication Systems (PCS). With the continued growth of information exchange there is a real need for privacy and authentication. PCS battery operated handheld devices require very low power circuitry. To address these needs a 64-bit Substitution-Permutation Network (SPN) block cipher is being designed in 1.2 μm CMOS and which consumes only 15 μW of power.

1 Introduction

Customer demand for wireless telecommunications is anticipated to grow considerably in the foreseeable future. Already there are over 30 million cellular users worldwide utilizing the power, and convenience of cellular telephones. Millions more use cordless phones and tetherless computing daily. By the year 2005, 50 percent of the communications is estimated to be wireless[1]. However the basis for this mobility is communication over an insecure medium - public airspace. Thus there is the increased opportunity for undetectable eavesdropping, and consequently telecommunication-related crimes. Last year alone it was estimated there was $500 million (US) in fraud to just the cellular companies[2].

Originally the wireline network had a natural deterrent to tapping. However there is no real deterrent for wireless networks. Anybody with a suitable receiver can intercept data being transmitted. To ensure privacy and authentication we need encryption algorithms to encode voice and data traffic traveling across this medium, and secure protocols to ensure that the data is not compromised in other ways by an intruder. These algorithms can be implemented in Application Specific ICs (ASIC) or as code for DSPs or microprocessors. In most designs DSPs and ASICs are used for repetitive functions, with ASICs being used when the lowest power and highest data rate are required. Often after prototyping, DSP algorithms are often re-engineered in silicon to reduce costs through a large production run. The increased availability of high-quality VLSI tools and the use of Multi-Chip Modules (MCM) to reduce inter-chip capacitances and thus power has helped in this progression towards ASICs[3].

Public key ciphers such as RSA[4] are well suited for authentication and digital signatures. However the circuitry for implementing multiplications and exponentiations of numbers several hundred bits in length is complex. This results in a considerably higher power consumption for a given data rate compared to block ciphers, which predominantly only use table lookups and the bitwise exclusive-ORs.

The most well known private key block cipher is the Data Encryption Standard (DES)[5]. This cipher encodes 64-bits of data at a time using a 56-bit key. Ever since this standard was released there have been criticisms that the 56-bit key was too small and that eventually an exhaustive key search type attack would be feasible. Recently, Wiener[6] has designed an exhaustive key-search machine for $1 million (US) that can on average find a DES key in a mere 3.5 hours. To counter this threat triple-DES is often used, but with the penalty of increased power consumption and lower data rates.

1.1 Other Hardware Implementations

There are many hardware implementations of DES. A high-speed version presented by Eberle[7] uses gallium arsenide (GaAs). It has an encryption rate of 1 Gbit/s and an active power consumption of 8 W. Typical commercial versions can encode (ECB/CBC) at rates up to 15 Mbytes/s using a standard 5 V supply and an active power consumption of 10-15 mW/MHz[8]. Although these are advertised as being "low power" they are clearly not suitable for a battery operated environment.

A high-speed 512-bit RSA implementation described by Iwamura et al[9] encoded at a data rate of 64 kbit/s using 50 000 gates. A typical commercial public key processor can encrypt at a rate of 300 kbit/s using a clock frequency of 20 MHz with a maximum power dissipation of 300 mW[10].

1.2 Substitution-Permutation Networks

Since the release of DES there has been abundant research into the design and cryptographic strength of Substitution-Permutation Networks (SPNs) and Substitution boxes (S-boxes) in particular [11, 12, 13, 14, 15]. They are usually implemented as a table lookup.

This particular implementation (Figure 1) uses large 8x8 S-boxes proven to be secure by Heys and Tavares[16]. It encodes a message M into ciphertext C using key K by:

$$C = \mathcal{E}(M, K) \tag{1}$$

where C, M and K are all 64-bit quantities. Being a multiple of 8-bits eases interfacing with a microprocessor in addition to being very efficient for software implementations.

The function $\mathcal{E}$ consists of sixteen $(n = 1, \ldots, 16)$ identical rounds of:

$$C_n = \mathcal{F}(C_{n-1}, K_n) \tag{2}$$

where $C_0 = M$ and $C = C_{16}$. The K_n are the sub-keys used in each round, which are derived from the supplied key.

Each round divides the incoming 64 bits into 8 sub-blocks $(m = 1, \ldots, 8)$ each of 8 bits. Each sub-block is combined with a portion of the current sub-key before a new 8 bit pattern is substituted with the corresponding S-Box $\mathcal{S}_m$.

$$C_{n_m} = \mathcal{P}(\mathcal{S}_m(C_{n-1_m} \oplus K_{n_m})) \tag{3}$$

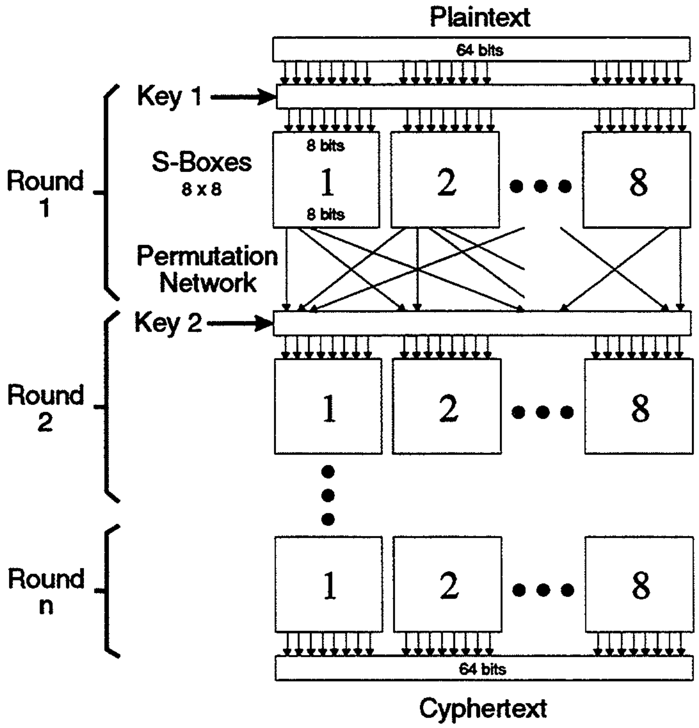

Figure 1: SPN Algorithm

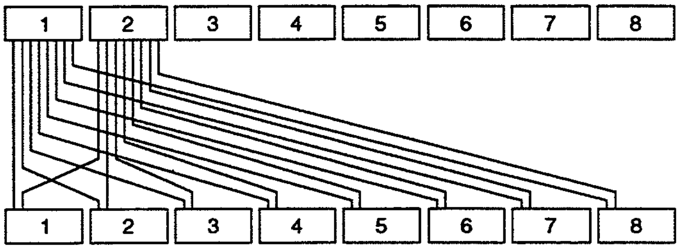

Figure 2: Permutation Network

The 64 bits of the 8 sub-blocks are then rearranged (permuted) with a one-to-one bit mapping as shown in Figure 2.

For decryption the sub-keys must be used in reverse order and each of the S-boxes inverted. In a typical full-duplex design one channel would contain the encryption S-boxes while the other channel has the decryption implementation.

1.3 Project Goals

The goal of this project has been to develop an encryption algorithm that could easily be included into a handheld wireless communications device to supply encryption. It must have a minimal power drain on the system, and easily operate at the required speed with minimum delay in the signal. Of a lesser concern is the total silicon area. The goal is to use this design as a small section of a larger ASIC, or perhaps as a separate chip in a Multi-Chip Module. These techniques can potentially reduce power consumption by a factor of three by avoiding the large interconnection capacitances that need to be driven in a standard single chip package on a printed circuit board approach[17].

1.3.1 Power

Estimates for the total power consumption of small handheld wireless devices varies depending on the required transmitter power. The DSP section of any portable device however is expected to consume on the order of a few milliwatts[18] regardless of the environment. Thus if encryption consumes 1% of this available power, this sets an upper limit of a few tens of microwatts.

The three major sources of power consumption in digital CMOS circuits are: switching, short-circuit current I_{sc}, and leakage current I_{leak}.

$$P_{total} = p_t(C_L\, f_{clk}\, V\, V_{dd}) + I_{sc}\, V_{dd} + I_{leak}\, V_{dd} \tag{4}$$

The switching term consists of the probability that a transition is going to take place p_t, the loading capacitance C_L, the clocking frequency f_{clk}, the voltage swing V and the supply voltage V_{dd}. Normally V is equal to V_{dd}, but can be smaller in some situations. The short circuit current occurs when both NMOS and PMOS transistors are conducting, thus offering a direct current path from the supply to ground. The leakage current is from the reverse-biased junctions of the substrate to the diffusion regions, and is process dependent.

For a well conceived low power design the switching term should be dominant. Thus the effort is on minimizing p_t, C_L, f_{clk} and V_{dd}. Since $P_{total} \propto V_{dd}^2$, reducing the supply voltage will have profound affect. Chandrakasan et al [19, 20] have demonstrated the advantages of trading switching speed for a reduction in the supply voltage.

1.3.2 Speed

The data rates anticipated for the various PCS services ranges from 1.2 kbit/s for remote terminals to nearly 2 Mbit/s for unrestricted digital information[21]. The encryption rate should handle data rates up to this speed.

Since the power consumption is directly proportional to the operating clock frequency, determining the minimum clocking speed required to encode the expected data rate is important. The electronic codebook (ECB) and cipher block chaining (CBC) modes both encode data at the full rate of the algorithm - 64 bits per encryption. A cipher feedback mode (CFB) mode implementation is usually arranged to encode only 8 new bits of data every encryption, thus requiring a clock operating 8 times faster to achieve the same throughput.

To determine the minimum clock speed we assume an ISDN data rate of 64 kbit/s. The design allows the previous ciphertext and the next plaintext to be unloaded and loaded in parallel with the current encryption, thus not affecting throughput. Each 64-bit encryption requires 16 rounds, each of two clock-cycles, thus requiring a 32.8 kHz minimum clock. If a cipher feedback mode is utilized then the minimum clock must be 262 kHz.

Transmission delay is the other speed consideration. Estimates for small picocell cellular installations of less than 10 ms have been suggested[18], with delays into the 300 ms range for geostationary satellite based systems.

2 Design

The overall design is broken into five areas: data I/O, substitution networks, permutation, key scheduling and control, as shown in Figure 3. All data and keys to and from this implementation are 8 bits wide. This was chosen to reduce the number of physical I/O pins required and keep the total silicon area low. Larger data widths up to 64-bits are

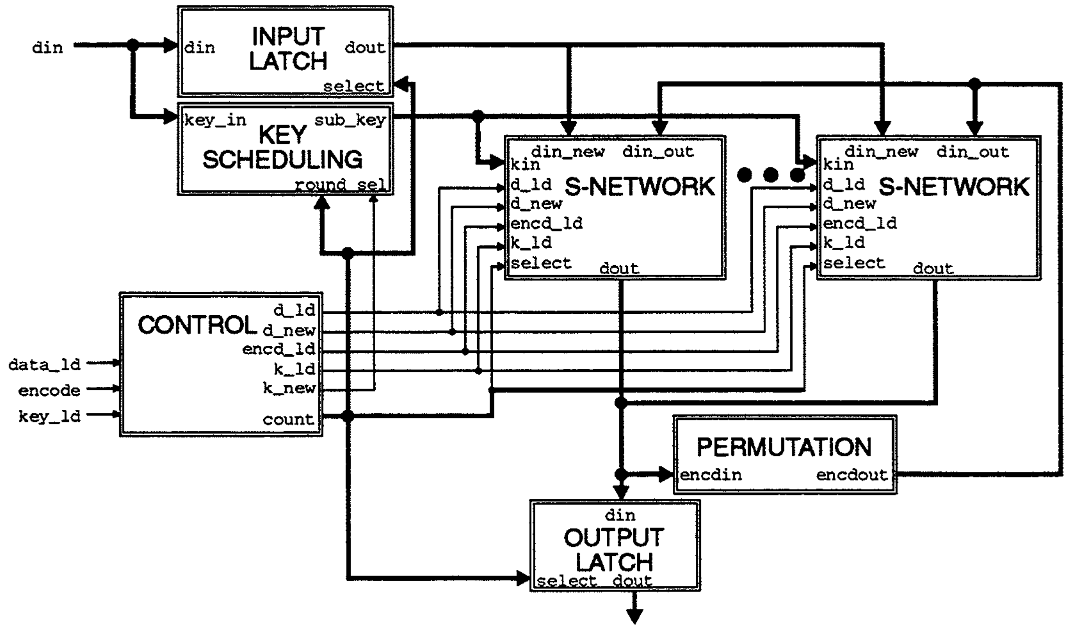

Figure 3: Hardware Organization

possible with only minor changes.

Before encrypting data, the 64-bit key is loaded serially 8-bits at a time and stored in the *Key Scheduling* block. As a security feature, it is impossible to read the key value back from the circuit. Other keys can be loaded and will overwrite the previous key.

To encode data eight bytes are loaded serially and stored in the *Input Latch.* It is then transferred in parallel to the Substitution Network (*S-Network*) where one round of encryption occurs. The data then passes through the *Permutation* block back to the *S-Network* for another round. Once the 16 rounds are completed the data is stored in the *Output Latch* where it can be serially unloaded 8-bits at a time.

The *Control* block orchestrates the entire process by counting the number of rounds performed and the data bytes being loaded and unloaded.

A standard cell library is being used for all digital portions of the design except for the ROMs. This library has been tested through HSPICE simulations to operate at supply voltages down to 1.5 V and is being used until another more suitable low-power library is obtained.

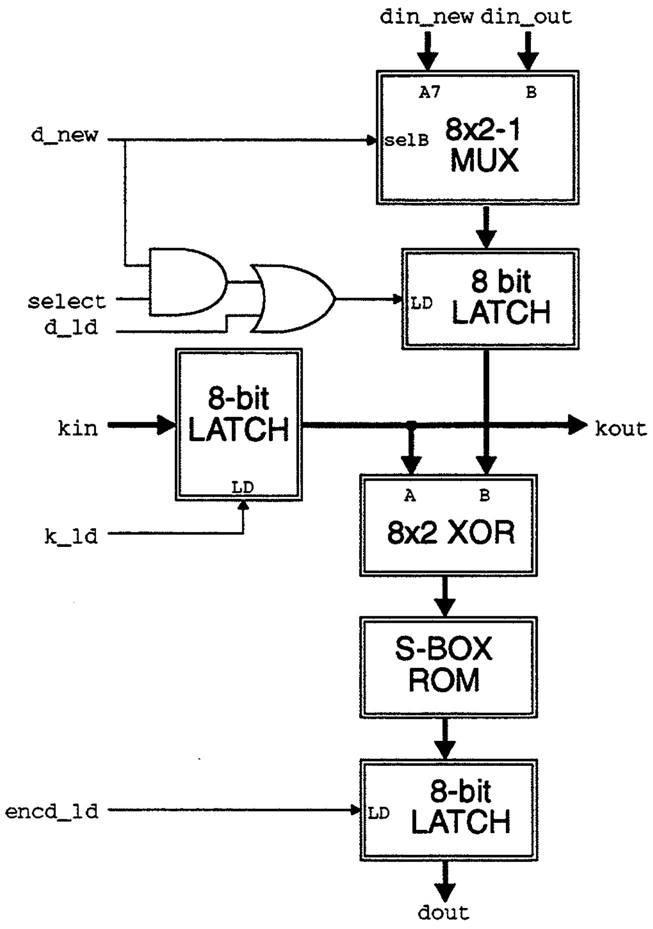

Figure 4: Single Substitution Network

2.1 Data I/O

Both the *Input Latch* and *Output Latch* blocks each contain 64 latches. A series of select lines control which bank of eight are currently being written to (*Input Latch*) or read from (*Output Latch*). To increase throughput data can be loaded and unloaded concurrently with the current block encryption.

2.2 Substitution Network

This block implements the inner portion of equation 3, which is graphically shown in Figure 4.

The ROM is organized into 4 blocks of 64 bytes each. Address pre-decoding is used to gate the clocks only to the block containing the required word thus reducing switching activity. Each block uses 5 bits to encode one of the 32 word lines, with the last bit used for column decoding.

A pre-charge dynamic ROM architecture is employed to greatly reduce static power consumption. The column decoding uses an NMOS switch from the charging PMOS to only one of two bit lines. Thus the bit line is charged up to $V_{dd} - V_T$, thus reducing the voltage swing to nearly 1/2 when using a 1.5 V supply. This NMOS transistor also acts as a cascode amplifier to the sense amplifier.

Each ROM block also contains an extra column for a 17th "bit". This column matches the worst case reading scenario, and is used for self-timing by deactivating the block once a stable value has been read. Thus the ROM can be operated with a variety of supply voltages without having to adjust the internal timing.

2.3 Key Scheduling

The key scheduling used in this implementation is a very simple rotate right by 8 bits for each round. This has been very easy to implement in hardware with the added luxury of the keys self-resetting after each block of 16 rounds has been encrypted, however it has not been proven to be secure. A different key scheduling algorithm is not expected to make a substantial impact on power consumption, but could have an impact on the area required for calculation and storage.

2.4 Permutation

A hardware implementation of the permutation step is simply the correct routing from the latches at the end of the basic network to the multiplexors at the top of the network. This permutation is selected from the CEP permutation class suggested by Ayoub[22]. By using 16 rounds in the algorithm, the Linear Transformation suggested by Heys and Tavares[11] is not needed to be secure. Future work could examine its implemenation and thus reducing the number of rounds required.

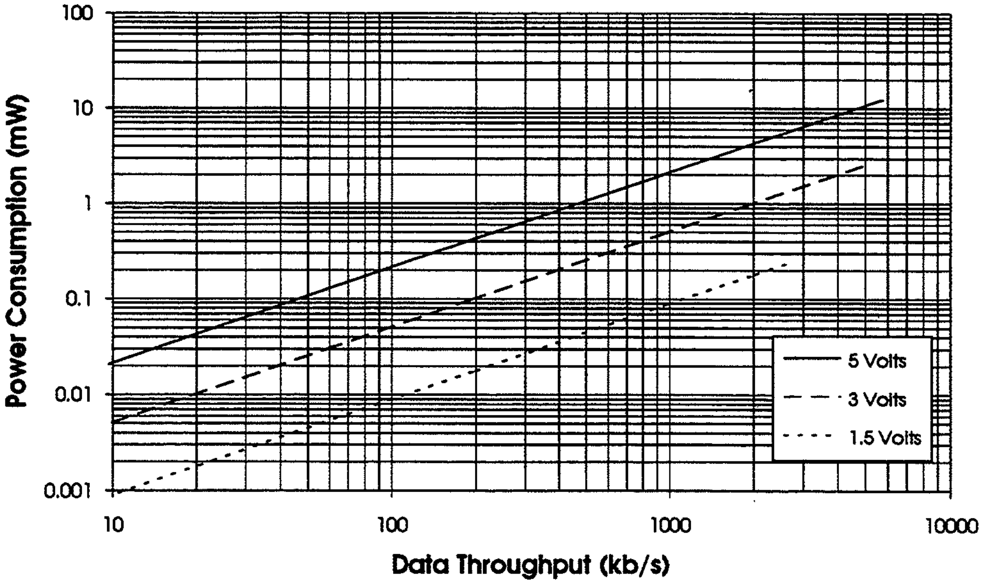

Figure 5: Power Consumption versus Data Throughput for Various Supply Voltages

3 Conclusions

With an overall design complete, work concentrated on creating low power S-boxes. Currently a single 64-byte ROM blocks and various parts of the surrounding digital circuitry have been fully simulated at supply voltages down to 1.5 V. Figure 5 shows these results versus the data throughput rate for the complete design. The final power figures will be higher since the control and latch circuitry has not been included. Thus power consumption for the complete design of 26 000 transistors is estimated to meet the 15 μW design goal while encoding data at a rate of 64 kbit/s with a propagation delay of 1.9 ms. The maximum data rate is estimated to be 5.7 Mbit/s at a supply voltage of 5 V.

Work is continuing on this design with fabrication of a smaller 32-bit design through the Canadian Microelectronics Corporation, using the Northern Telecom 1.2 μm process. This 32-bit version will test the logic of the larger 64-bit version, but fit into the die area available. Physical verification of the design is therefore expected by late summer of 1995.

4 Future

A hardware implementation will further research into SPNs. There are many uses outside the wireless environment which can use an efficient and easy to implement private key block cipher. For example, with a simple pipeline and 64-bit word architecture, this algorithm could be capable of encoding several hundred megabits per second.

Work in low-power VLSI circuits is an area that will continue to see growth as portable devices proliferate. The ROM designed for this project could have many other applications in battery operated environments, such as notebook computers. Further increases in operating speed and decreases in power consumption will occur by reducing capacitances with the migration toward smaller 0.5 μm CMOS design features.

References

[1] CEC/RACE Industrial Consortium. *IBC Common Functional Specifications: Mobile Network Subsystems*, RACE D730, B edition, December 1991.

[2] Cellular One's fraud protection for new customers. NewsBytes News Service, January 31 1995.

[3] H. Meyr and R. Subramanian. Advanced digital receiver principles and technologies for pcs. *IEEE Communications Magazine*, pp. 68–78, January 1995.

[4] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.

[5] National Bureau of Standards, Washington, D.C. *Data Encryption Standard*, FIPS PUB 46 edition, January 1977.

[6] M.J. Wiener. Efficient DES key search, August 1993. presented at CRYPTO '93.

[7] H. Eberle. A high-speed DES implementation for network applications. In *Lecture Notes in Computer Science: Advances in Cryptology - CRYPTO '92 Proceedings*, pp. 521–539. Springer-Verlag, 1992.

[8] Newbridge Microsystems. DES data ciphering processors - CA95C68/18/09, February 1994. Data Sheet.

[9] K. Iwamura, T. Matsumoto, and H. Imai. High-speed implementation methods for RSA. In *Advances in Cryptology: Proceedings of EUROCRYPT '92*, pp. 221–237, Berlin, 1992. Springer-Verlag.

[10] Newbridge Microsystems. Data encryption processor - CA34C168, August 1989. Data Sheet.

[11] H.M. Heys and S.E. Tavares. On the design of secure block ciphers. In *17th Biennial Symposium on Communications*, pp. 417–422, Kingston, Ontario, May 1994.

[12] K. Nyberg. Perfect nonlinear s-boxes. In *Advances in Cryptology: Proceedings of EUROCRYPT '91*, pp. 378–386, Berlin, 1991. Springer-Verlag.

[13] C.M. Adams and S.E. Tavares. The structured design of cryptographically good S-boxes. *Journal of Cryptology*, 3(1):27–41, 1990.

[14] A.F. Webster and S.E. Tavares. On the design of S-boxes. In *Advances in Cryptology: Proceedings of CRYPTO '85*, pp. 523–534, Berlin, 1985. Springer-Verlag.

[15] L. O'Connor. On the distribution of characteristics in bijective mappings. In *Advances in Cryptology: Proceedings of EUROCRYPT '93*, pp. 360–370, Berlin, 1993. Springer-Verlag.

[16] H.M. Heys and S.E. Tavares. Substitution-permutation networks resistant to differential and linear cryptanalysis. *Journal of Cryptology*. in press.

[17] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen. Design considerations for a future portable multimedia terminal. In S. Nanda and D.J. Goodman, editors, *Third*

Generation Wireless Information Networks, chapter 6, pp. 75–97. Kluwer Academic Publishers, Boston, 1992.

[18] D.C. Cox. Wireless network access for personal communications. *IEEE Communications Magazine*, pp. 96–115, December 1992.

[19] A.P. Chandrakasan, S. Sheng, and R.W. Brodersen. Low-power CMOS digital design. *IEEE Journal of Solid-State Circuits*, 27(4):473–483, April 1992.

[20] A.P. Chandrakasan, A. Burstein, and R.W. Brodersen. A low-power chipset for a portable multimedia I/O terminal. *IEEE Journal of Solid-State Circuits*, 29(12):1415–1428, December 1994.

[21] J.C.S. Cheung, M.A. Beach, and J.P. McGeehan. Network planning for third-generation mobile radio systems. *IEEE Communications Magazine*, pp. 54–59, November 1994.

[22] F. Ayoub. The design of complete encryption networks using cryptographically equivalent permutations. *Computers and Security*, 2:261–267, 1983.