Improving Resistance to Differential Cryptanalysis and the Redesign of LOKI

Lawrence BROWN Matthew KWAN Josef PIEPRZYK
Jennifer SEBERRY

Technical Report CS38/91

Department of Computer Science,
University College, UNSW, Australian Defence Force Academy,
Canberra ACT 2600. Australia.

Abstract

Differential Cryptanalysis is currently the most powerful tool available for analysing block ciphers, and new block ciphers need to be designed to resist it. It has been suggested that the use of S-boxes based on bent functions, with a flat XOR profile, would be immune. However our studies of differential cryptanalysis, particularly applied to the LOKI cipher, have shown that this is not the case. In fact, this results in a relatively easily broken scheme. We show that an XOR profile with carefully placed zeroes is required. We also show that in order to avoid some variant forms of differential cryptanalysis, permutation P needs to be chosen to prevent easy propagation of a constant XOR value back into the same S-box. We redesign the LOKI cipher to form LOKI91, to illustrate these results, as well as to correct the key schedule to remove the formation of equivalent keys. We conclude with an overview of the security of the new cipher.

1 Introduction

Cryptographic research is currently a very active field, with the need for new encryption algorithms having spurred the design of several new block ciphers [1]. The most powerful tool for analysing such block ciphers currently known is differential cryptanalysis. It has been used to find design deficiencies in many of the new ciphers. Some new design criteria have been proposed which are claimed to provide immunity to differential cryptanalysis. These involve the use of S-boxes based on bent functions, selected so the resultant box has a flat XOR profile.

Figure 1: An XOR Profile

In this paper, after presenting a brief introduction to the key concepts in differential cryptanalysis, we will show that these new criteria do not provide the immunity claimed, but rather can result in the design of a scheme which may be relatively easily broken. What we believe is required, is an S-box with carefully placed zeroes, which significantly hinder the differential cryptanalysis process.

We continue by documenting our analysis of the LOKI cipher. We report on a previously discovered differential cryptanalysis attack, faster than exhaustive search up to 11 rounds, as well as on a new attack, using an alternate form of analysis, which is faster than exhaustive search up to 14 rounds. We also briefly note some design deficiencies in the key schedule which resulted in the generation of equivalent keys. This process highlighted some necessary additional design criteria needed to strengthen block ciphers against these attacks.

We conclude by describing the redesign of the LOKI cipher, using it to illustrate the application of these additional design principles, and make some comments on what we believe is the security of the new scheme.

2 Differential Cryptanalysis

2.1 Overview

Differential Cryptanalysis is a dynamic attack against a cipher, using a very large number of chosen plaintext pairs, which through a statistical analysis of the resulting ciphertext pairs, can be used to determine the key in use. In general, differential cryptanalysis is much faster than exhaustive search for a certain number of rounds in the cipher, however there is a breakeven point where it becomes slower than exhaustive search. The lower the number of rounds this is, the greater the security of the cipher. Differential Cryptanalysis was first described by Biham and Shamir in [2], and in greater detail in [3]. These described the general technique, and its application to the analysis of the DES and the Generalised DES. Subsequent papers by them have detailed its

Figure 2: A 2-round Iterative Characteristic

application to FEAL and N-Hash [4], and to Snefru, Khafre, Redoc-II, LOKI, and Lucifer [5].

In Differential Cryptanalysis, a table showing the distribution of the XOR of input pairs against the XOR of output pairs is used to determine probabilities of a particular observed output pair being the result of some input pair. The general form of such a table is shown in Fig 1.

To attack a multi-round block cipher, the XOR profile is used to build n round characteristics, which have a given probability of occurring. These characteristics specify a particular input XOR, a possible output XOR, the necessary intermediate XOR's, and the probability of this occurring. In their original paper [3], Biham and Shamir describe 1,2,3 and 5 round characteristics which may be used to directly attack versions of DES up to 7 rounds. Knowing a characteristic, it is possible to infer information about the outputs for the next two rounds. To utilise this attack, a number of pairs of inputs, having the nominated input XOR, are tried, until an output XOR results which indicates that the pattern specified in the characteristic has occurred. Since an n round characteristic has a probability of occurrence, for most keys we can state on average, how many pairs of inputs need to be trialed before the characteristic is successfully matched. Once a suitable pair, known as a right pair, has been found, information on possible keys which could have been used, is deduced. Once this is done we have two plaintext-ciphertext pairs. We know from the ciphertext, the input to the last round. Knowing the input XOR and output XOR for this round, we can thus restrict the possible key bits used in this round, by considering those outputs with an XOR of zero, providing information on the outputs of some of the S-boxes. By then locating additional right pairs we can eventually either uniquely determine the key, or deduce sufficient bits of it

Figure 3: Flat XOR Profile

that an exhaustive search of the rest may be done.

N round characteristics can be concatenated to form longer characteristics if the output of the first supplies the input to the second, with probabilities multiplied together. A particularly useful characteristic is one whose output is a swapped version of its input, and which hence may be iterated with itself. This may be used to analyse an arbitrary number of rounds of the cipher, with a steadily increasing work factor. A particularly useful form is one where a non-zero input XOR to the F function results in a zero output XOR. Such a characteristic is illustrated in Fig 2, and may be denoted as:

```
A. (x, 0) -> (0, x) always (ie Pr=1)
B. (0, x) -> (x, 0) with some probability p
```

It may be iterated as A B A B A B A B to 8 rounds for example, with characteristic probability p^4. This form of characteristic is then used in the analysis of arbitrary n round forms of a cipher, until the work factor exceeds exhaustive search. These techniques are described in detail in [3].

2.2 Why Flat XOR profiles Don't Work

Given the success of differential cryptanalysis in the analysis of block ciphers, it has become important to develop design criteria to improve the resistance of block ciphers to it, especially with several candidates having performed poorly. Dawson and Tavares [6] have proposed that the selection of S-boxes with equal probabilities for each output XOR given an input XOR (except input 0) would result in a cipher that was immune to differential cryptanalysis (see Fig. 3). However a careful study of Biham and Shamir's attack on the 8 round version of DES [3], confirmed by our own analyses of LOKI, have shown that this is not the case.

Indeed the selection of such S-boxes results in a cipher which is significantly easier to cryptanalyse than normal. This is done by constructing a 2 round iterative characteristic of the form in Fig. 2, with an input XOR that changes bits to one S-box only. We know we can do this, since the flat XOR profile implies that an output XOR of 0 for a specified input XOR will occur with $Pr(1/2^n)$. When iterated over 2k rounds, this will have a probability of $Pr(1/2^{(k-1)n})$, since you get the last round for free. Consider a 16 round DES style cryptosystem, but with S-boxes having a flat XOR profile of the form in Fig. 3 with $m = 6$, $n = 4$, and $k = 8$. This may be attacked by a 15-round characteristic, chosen to alter inputs to a single S-box only. This gives a probability to break with a given test pair of $Pr(1/2^{28})$, implying that about 2^{28} pairs need to be tried to break the cipher, far easier than by exhaustive search.

2.3 Significance of Permutation P

Although differential cryptanalysis may be done independent of permutation P, knowledge of a particular P may be used to construct some other useful n round characteristics for cryptanalysing a particular cipher. The most useful of these take the form of a 2 or 3 round characteristic which generate an output XOR identical to the input XOR, either directly in 2 rounds, or by oscillating between two alternate XOR values over 3 rounds. The 3 round characteristic is sensitive to the form of permutation P. This form of characteristic has been found in the original version of LOKI, as detailed below. It thus indicates that care is needed in the design of not just the S-boxes, but of all elements in function F, in order to reduce susceptibility to differential cryptanalysis.

3 Analysis of LOKI

3.1 Overview

LOKI is one of several recently proposed new block ciphers. It was originally detailed by Brown, Pieprzyk and Seberry in [7]. Its overall structure is shown in Fig. 4. We will refer to this version of the cipher as LOKI89 in the remainder of this paper. It is a 16 round Feistel style cipher, with a relatively straightforward key schedule, a non-linear function $F = P(S(E(R \oplus K)))$, and four identical 12-to-8 bit S-boxes. Permutation P in LOKI has a regular structure which distributes the 8 output bits from each S-box to boxes [+3 +2 +1 0 +3 +2 +1 0] in the next round. Its S-box consists of 16 1-1 functions, based on exponentiation in a Galois field $GF(2^8)$ (see [8]), with the general structure shown in Fig. 4.

3.2 Security of LOKI89

Initial testing of the statistical and complexity properties of LOKI89 indicated that its properties were very similar to those exhibited by DES, FEAL8 and Lucifer [9], results that were very encouraging. Initial examination of the XOR profile of the LOKI89 S-box also suggested that it should be more resistant than DES to differential cryptanalysis.

Figure 4: LOKI89 Overall Structure and S-box Detail

LOKI89 was then analysed in detail using differential cryptanalysis. Biham [5] describes an attack, using a 2 round iterative characteristic with $Pr(118/2^{20}) \approx Pr(2^{-13.12})$, with non-zero inputs to 2 S-boxes resulting in the same output. There are four related variants (by rotation) of the form:

```
A. (00000510,00000000) -> (00000000,00000510)   always
B. (00000000,00000510) -> (00000510,00000000)   Pr(118/2^20)
```

This characteristic is iterated to 8 rounds with $Pr(2^{-52.48})$, allowing up to 10 rounds to be broken faster than by exhaustive key space search. This is a significantly better result than for the DES. The authors have verified this attack.

The authors have subsequently found an alternate 3 round iterative characteristic, attacking S-box 3, with an output XOR identical to the input XOR. Since permutation P in LOKI89 permutes some output bits back to the same S-box in the next round, this allows the characteristic to be iterated. It has the form:

```
A. (00400000,00000000) -> (00000000,00400000)   always
B. (00000000,00400000) -> (00400000,00400000)   Pr(28/4096)
C. (00400000,00400000) -> (00400000,00000000)   Pr(28/4096)
```

Since $28/4096 \approx 2^{-7.2}$ if we concatenate these characteristics, we get a 13 round characteristic in the order A B C A B C A B C A B C A with a probability of $Pr(2^{-7.2*8}) \approx Pr(2^{-57.6})$. This may be used to attack a 14 round version of LOKI89, and requires $O(2^{59})$ pairs to succeed. This is of the same order as exhaustive search (which is of $O(2^{60})$, as detailed below), and is thus a more successful attack than that reported previously. It has been verified by Biham. This still leaves the full 16 round version of LOKI89 secure, but with a reduced margin against that originally believed.

Independently, the authors [10], Biham [5], and the members of the RIPE consortium have discovered a weakness in the LOKI89 key schedule. It results in the generation of 15 equivalent keys for any given key, effectively reducing the key-space to 2^{60} keys. A complementation property also exists which results in 256 (key, plain, cipher) triples being formed, related by $LOKI(P,K) \oplus pppppppppppppppp = LOKI(P \oplus pppppppppppppppp, K \oplus mmmmmmmmnnnnnnnn)$, where $p = m \oplus n$ for arbitrary hex values m, n. This may be used to reduce the complexity of a chosen plaintext attack by an additional factor of 16. These results were found by analysing the key schedule by regarding each S-box input as a linear function of the key and plaintext, and solving to form (key, plaintext) pairs which result in identical S-box input values. This lead to solving the following equations:

$$RD \oplus KRD \oplus n.ROT12(KLD) = 0 \tag{1}$$

$$LD \oplus KLD \oplus n.ROT12(KRD) = 0 \tag{2}$$

where $LD = L' \oplus L$, $RD = R' \oplus R$, $KLD = KL' \oplus KL$, and $KRD = KR' \oplus KR$ describe the difference between the related (key, plaintext) pairs. This method is detailed by Kwan in [10].

In the light of these results, the authors have devised some additional design guidelines to those originally used in the design of LOKI, and have applied them in the development of a new version, LOKI91.

4 Redesign of LOKI

4.1 Some Additional Design Guidelines

To improve the resistance of a cipher to differential cryptanalysis, and to remove problems with the key schedule, the following guidelines were used:

- analyse the key schedule to minimize the generation of equivalent key, or related (key, plaintext) pairs.
- minimise the probability that a non-zero input XOR results in a zero output XOR, or in an identical output XOR, particularly for inputs that differ in only 1 or 2 S-boxes.
- ensure the cipher has sufficient rounds so that exhaustive search is the optimal attack (ie have insufficient pairs to do differential cryptanalysis).
- ensure that there is no way to make all S-boxes give 0 outputs, to increase the ciphers security when used in hashing modes.

These criteria were used when selecting the changes made to the LOKI structure, detailed below.

4.2 Design of LOKI91

LOKI91 is the revised version, developed to address the results detailed above. Changes have been made to two aspects of the LOKI structure. Firstly the key schedule has been amended in several places to significantly reduce the number of weak keys. Secondly the function used in the S-boxes has been altered to improve its utility in hashing applications, and to improve its immunity to differential cryptanalysis. In more detail, the four changes made to the original design were:

1. change key schedule to swap halves after every second round
2. change key rotations to alternate between ROT13 and ROT12
3. remove initial and final XORs of key with plaintext and ciphertext

Figure 5: LOKI91 Overall Structure

4. alter the S-box function used in the LOKI S-box (Fig. 4) to

$$Sfn(row, col) = (col + ((row * 17) \oplus ff_{16})\&ff_{16})^{31} mod\, g_{row} \qquad (3)$$

   where $+$ and $*$ refer to arithmetic addition and multiplication, $\oplus$ is addition modulo 2, and the exponentiation is performed in $GF(2^8)$. The generator polynomials used (g_{row}) are as for LOKI89 [7].

The overall structure of LOKI91 that results from these changes is shown in Fig. 5.

The key schedule changes remove all but a single bit complementation property, leaving an effective search space of 2^{63} keys under a chosen-plaintext attack. It reduces key equivalences to a single bit complementation property, similar to that of the DES where $LOKI(\overline{P},\overline{K}) = \overline{LOKI(P,K)}$, by reducing the solutions to each of Eq. 1 and Eq. 2 to one, and removing the independence between them by altering the swapping to every two rounds. It also greatly reduces the number of weak and semi-weak keys to those shown in Table 1 (where an * denotes weak keys). The DES also has 16 weak and semi-weak keys.

Encrypt Key	Decrypt Key
0000000000000000	0000000000000000 *
00000000aaaaaaaa	aaaaaaaa00000000
0000000055555555	5555555500000000
00000000ffffffff	ffffffff00000000
aaaaaaaa00000000	00000000aaaaaaaa
aaaaaaaaaaaaaaaa	aaaaaaaaaaaaaaaa *
aaaaaaaa55555555	55555555aaaaaaaa
aaaaaaaaffffffff	ffffffffaaaaaaaa
5555555500000000	0000000055555555
55555555aaaaaaaa	aaaaaaaa55555555
5555555555555555	5555555555555555 *
55555555ffffffff	ffffffff55555555
ffffffff00000000	00000000ffffffff
ffffffffaaaaaaaa	aaaaaaaaffffffff
ffffffff55555555	55555555ffffffff
ffffffffffffffff	ffffffffffffffff *

Table 1: LOKI91 Weak and semi-weak key pairs

The removal of the initial and final XORs became necessary with the change in the swap locations, since otherwise it would have resulted in cancellation of the keys bits at the input to E in some rounds. This change does affect the growth of ciphertext dependence on keys bits (see [11]), increasing it from 3 to 5 rounds. This still compares favorably with the DES which takes either 5 or 7 rounds, dependent on the type of dependency analysed. This change also greatly assisted in the reduction of the number of equivalent keys.

The new Sfn uses arithmetic addition and multiplication, as these are non-linear when used in $GF(2^8)$. The addition modulo two of the row with ff_{16} ensures that an all zero input gives a non-zero output, thus removing the major deficiency of LOKI89 when used as a hash function. The result of the addition and multiplication is masked with ff_{16} to restrict the value to lie with $GF(2^8)$, prior to the exponentiation in that field. The new function reduces the probabilities of occurance of n round iterative characteristics, useful for differential cryptanalysis, to be as low as possible. With the new function, LOKI91 is theoretically breakable faster than an exhaustive key space search in:

- up to 10 rounds using a 2 round characteristic with the $f(x')- > 0'$ mapping occuring with $Pr(122/1048576)$
- up to 12 rounds using a 3 round characteristic with the $f(x')- > x'$ mapping occuring with $Pr(16/4096)$ (used twice)

At 16 rounds, cryptanalysis is generally impossible, as insufficient pairs are available to complete the analysis. It would require:

- 2^{80} pairs using the 3 round characterisitic, or
- 2^{92} pairs using the 2 round characteristic

compared to a total of 2^{63} possible plaintext pairs.

5 Conclusion

In this paper, we have shown that a flat XOR profile does not provide immunity to differential cryptanalysis, but in fact leads to a very insecure scheme. Instead a carefully chosen XOR profile, with suitably placed 0 entries is required to satisfy the new design guidelines we have identified. We also note an analysis of key schedules, which can be used to determine the number of equivalent keys. We conclude with the application of these results to the design of LOKI91.

6 Acknowledgements

Thank you to the members of the crypt group for their support and suggestions. This work has been supported by ARC grant A48830241, ATERB, and Telecom Australia research contract 7027.

Appendix A - Specification of LOKI91

Encryption

The overall structure of LOKI91 is shown in Fig. 5, and is specified as follows. The 64-bit input block X is partitioned into two 32-bit blocks L and R. Similarly, the 64-bit key is partitioned into two 32-bit blocks KL and KR.

$$\begin{array}{ll} L_0 = L & KL_0 = KL \\ R_0 = R & KR_0 = KR \end{array} \tag{4}$$

The key-dependent computation consists (except for a final interchange of blocks) of 16 rounds (iterations) of a set of operations. Each iteration includes the calculation of the encryption function f. This is a concatenation of a modulo 2 addition and three functions E, S, and P. Function f takes as input the 32-bit right data half R_{i-1} and the 32-bit left key half KL_i produced by the key schedule KS (denoted K_i below), and which produces a 32-bit result which is added modulo 2 to the left data half L_{i-1}. The two data halves are then interchanged (except after the last round). Each round may thus be characterised

3	2	1	0	31	30	29	28	27	26	25	24
27	26	25	24	23	22	21	20	19	18	17	16
19	18	17	16	15	14	13	12	11	10	9	8
11	10	9	8	7	6	5	4	3	2	1	0

Table 2: LOKI Expansion Function E

as:

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1}, KL_i) \\ f(R_{i-1}, K_i) &= P(S(E(R_{i-1} \oplus K_i))) \end{aligned} \tag{5}$$

The component functions E, S, and P are described later.

The key schedule KS is responsible for deriving the sub-keys K_i, and is defined as follows: the 64-bit key K is partitioned into two 32-bit halves KL and KR. In each round i, the sub-key K_i is the current left half of the key KL_{i-1}. On odd numbered rounds (1, 3, 5, etc), this half is then rotated 12 bits to the left. On even numbered rounds (2, 4, 6, etc), this half is then rotated 13 bits to the left, and the key halves are interchanged.

This may be defined for odd numbered rounds as:

$$\begin{aligned} K_i &= KL_{i-1} \\ KL_i &= ROL(KL_{i-1}, 13) \\ KR_i &= KR_{i-1} \end{aligned} \tag{6}$$

This may be defined for even numbered rounds as:

$$\begin{aligned} K_i &= KL_{i-1} \\ KL_i &= KR_{i-1} \\ KR_i &= ROL(KL_{i-1}, 12) \end{aligned} \tag{7}$$

Finally after the 16 rounds, the two output block halves L_{16} and R_{16} are then concatenated together to form the output block Y. This is defined as (note the swap of data halves to undo the final interchange in Eq.5):

$$Y = R_{16} \,|\, L_{16} \tag{8}$$

Decryption

The decryption computation is identical to that used for encryption, save that the partial keys used as input to the function f in each round are calculated in reverse order. The rotations are to the right, and an initial pre-rotation of 8 places is needed to form the key pattern.

Row	gen_{row}	e_{row}
0	375	31
1	379	31
2	391	31
3	395	31
4	397	31
5	415	31
6	419	31
7	425	31
8	433	31
9	445	31
10	451	31
11	463	31
12	471	31
13	477	31
14	487	31
15	499	31

Table 3: LOKI S-box Irreducible Polynomials and Exponents

Function f

The encryption function f is a concatenation of a modulo 2 addition and three functions E, S, and P, which takes as input the 32-bit right data half R_{i-1} and the 32-bit left key half KL_i, and produces a 32-bit result which is added modulo 2 to the left data half L_{i-1}.

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1} \oplus K_i))) \tag{9}$$

The modulo 2 addition of the key and data halves ensures that the output of f will be a complex function of both of these values.

The expansion function E takes a 32-bit input and produces a 48-bit output block, composed of four 12-bit blocks which form the inputs to four S-boxes in function f. Function E selects consecutive blocks of twelve bits as inputs to S-boxes S(4), S(3), S(2), and S(1) respectively, as follows:
$[b_3\, b_2 \ldots b_0\, b_{31}\, b_{30} \ldots b_{24}]$
$[b_{27}\, b_{26} \ldots b_{16}]$
$[b_{19}\, b_{18} \ldots b_8]$
$[b_{11}\, b_{10} \ldots b_0]$

This is shown in full in Table 2 which specifies the source bit for outputs bits 47 to 0 respectively:

The substitution function S provides the confusion component in the LOKI cipher. It takes a 48-bit input and produces a 32-bit output. It is composed of

31	23	15	7	30	22	14	6
29	21	13	5	28	20	12	4
27	19	11	3	26	18	10	2
25	17	9	1	24	16	8	0

Table 4: LOKI Permutation P

four S-boxes, each of which takes a 12-bit input and produces an 8-bit output, which are concatenated together to form the 32-bit output of S. The 8-bit output from S(4) becomes the most significant byte (bits [31...24]), then the outputs from S(3) (bits[23...16]), S(2) (bits[15...8]), and S(1) (bits [7...0]). In LOKI91 the four S-boxes are identical. The form of each S-box is shown in Fig 4. The 12-bit input is partitioned into two segments: a 4-bit row value row formed from bits $[b_{11}\, b_{10}\, b_1\, b_0]$, and an 8-bit column value col formed from bits $[b_9\, b_8 \ldots b_3\, b_2]$. The row value row is used to select one of 16 S-functions $Sfn_{row}(col)$, which then take as input the column value col and produce an 8-bit output value. This is defined as:

$$Sfn_{row}(col) = (col + ((row * 17) \oplus ff_{16})r\&ff_{16})^{e_{row}} mod\, g_{row} \qquad (10)$$

where gen_{row} is an irreducible polynomial in $GF(2^8)$, and e_{row} is the (constant 31) exponent used in forming $Sfn_{row}(col)$. The generators and exponents to be used in the 16 S-functions in LOKI91 are specified in Table 3. For ease of implementation in hardware, this function can also be written as:

$$Sfn_{row}(col) = (col + ((\overline{row})|(\overline{row} << 4))\&ff_{16})^{31} mod\, g_{row} \qquad (11)$$

The permutation function P provides diffusion of the outputs from the four S-boxes across the inputs of all S-boxes in the next round. It takes the 32-bit concatenated outputs from the S-boxes, and distributes them over all the inputs for the next round via a regular wire crossing which takes bits from the outputs of each S-box in turn, as defined in Table 4 which specifies the source bit for output bits 31 to 0 respectively.

Test Data

A single test triplet for the LOKI91 primitive is listed below.

```
#   Single LOKI91 Certification triplet
#   data is saved as (key, plaintext, ciphertext) hex triplets
#
3849674c2602319e 126898d55e911500 c86caec1e3b7b17e
```

References

[1] J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security.* Englewood Cliffs, NJ, Prentice Hall, 1989.

[2] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," *Journal of Cryptology*, 4, no. 1, pp. 3–72, 1991.

[3] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems," Weizmann Institute of Science, Rehovot, Israel, Technical Report, 19 July 1990.

[4] E. Biham and A. Shamir, "Differential Cryptanalysis of Feal and N-Hash," *Advances in Cryptology - Eurocrypt'91*, 547, pp. 1–16, 1991.

[5] E. Biham and A. Shamir, "Differential Cryptanalysis Snefru, Kharfe, REDOC-II, LOKI and Lucifer," *Advances in Cryptology - Crypto'91*, 576, pp. 156–171, 1991.

[6] M. H. Dawson and S. E. Tavares, "An Expanded Set of S-box Design Criteria Based On Information Theory and Its Relation to Differential-Like Attacks," *Advances in Cryptology - Eurocrypt'91*, 547, pp. 352–367, 1991.

[7] L. Brown, J. Pieprzyk and J. Seberry, "LOKI - A Cryptographic Primitive for Authentication and Secrecy Applications," in *Advances in Cryptology: Auscrypt '90* (Lecture Notes in Computer Science), vol. 453. Berlin: Springer Verlag, pp. 229–236, 1990.

[8] J. Pieprzyk, "Non-Linearity of Exponent Permutations," in *Advances in Cryptology - Eurocrypt'89* (Lecture Notes in Computer Science), vol. 434, J. J. Quisquater and J. Vanderwalle, Eds. Berlin: Springer Verlag, pp. 80–92, 1990.

[9] L. Brown, J. Pieprzyk, R. Safavi-Naini and J. Seberry, "A Generalised Testbed for Analysing Block and Stream Ciphers," in *Information Security*, W. Price and D. Lindsey, Eds. North-Holland, 1991.

[10] M. Kwan and J. Pieprzyk, "A General Purpose Technique for Locating Key Scheduling Weaknesses in DES-style Cryptosystems," in *Advances in Cryptology - Asiacrypt'91* (Lecture Notes in Computer Science), vol. 739. Berlin: Springer Verlag, pp. 237–246, 1993.

[11] L. Brown and J. Seberry, "Key Scheduling in DES Type Cryptosystems," in *Advances in Cryptology: Auscrypt '90* (Lecture Notes in Computer Science), vol. 453. Berlin: Springer Verlag, pp. 221–228, 1990.