Non-Linear Approximations in Linear Cryptanalysis

Lars R. Knudsen[1] and M.J.B. Robshaw[2]

[1] K.U. Leuven, ESAT, Kardinaal Mercierlaan 94, B-3001 Heverlee
email:knudsen@esat.kuleuven.ac.be
[2] RSA Laboratories, 100 Marine Parkway, Redwood City, CA 94065, USA
email:matt@rsa.com

Abstract. By considering the role of non-linear approximations in linear cryptanalysis we obtain a generalization of Matsui's linear cryptanalytic techniques. This approach allows the cryptanalyst greater flexibility in mounting a linear cryptanalytic attack and we demonstrate the effectiveness of our non-linear techniques with some simple attacks on LOKI91. These attacks potentially allow for the recovery of seven additional bits of key information with less than 1/4 of the plaintext that is required using current linear cryptanalytic methods.

1 Introduction

The technique of linear cryptanalysis [7] is now well known. Most dramatically it has provided the first experimental (though barely practical) compromise [8] of the Data Encryption Standard DES [9].

In addition to some theoretical and practical enhancements or extensions to linear cryptanalysis [4, 6, 11] it is natural to consider whether the linear approximations on which linear cryptanalysis relies can be replaced with non-linear approximations. Since there are far more non-linear approximations than linear approximations, it seems fair to say that by opening ourselves to their use, we might obtain a much improved attack on some cipher. As a motivational example, the best *linear* approximation to a DES S-box is to S5, and this approximation holds with an absolute valued bias of 20/64, yet there is a relatively simple *non-linear* approximation to S8 involving four input bits[3], which holds with absolute bias 28/64. While previous work [3] has already demonstrated insurmountable problems in the general use of non-linear approximations, we will show that they should not be abandoned and that non-linear approximations can offer effective additions to the basic techniques in use today.

In the following sections we describe the essential issues in linear cryptanalysis and the use of non-linear approximations within such an attack. We show that current linear cryptanalytic techniques are essentially special instances of

[3] Labeling the input bits to DES S-box S8 as $x_5 \dots x_0$ and the output bits as $y_3 \dots y_0$ the approximation $1 \oplus x_3 \oplus x_4 \oplus x_2x_3 \oplus x_3x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4 = y_0 \oplus y_1 \oplus y_2 \oplus y_3$ holds with probability 60/64.

U. Maurer (Ed.): Advances in Cryptology - EUROCRYPT '96, LNCS 1070, pp. 224-236, 1996.

our more general approach and we demonstrate that our techniques have implications for both the design and cryptanalysis of block ciphers. In particular, our techniques pose a threat even when Matsui's advanced linear cryptanalytic attacks (in which the cryptanalyst guesses the key bits used to evaluate some S-box) are rendered impractical due to the use of large S-boxes. While we will motivate our discussion with examples that involve DES (since this is the cipher with which most people are familiar) we note that our current techniques do not seem to offer any significant advantage over existing attacks on DES. There are however some open questions in this regard and future research alone will determine if this is indeed the case. Instead, our techniques have been most useful in improving existing attacks [16] on LOKI91 [1] where it is straightforward to recover seven additional bits of the user-defined key while using less than 1/4 of the plaintext that is currently required. We also expect our techniques to be applicable to many other block ciphers.

2 Linear Cryptanalysis

In a linear cryptanalytic attack the cryptanalyst identifies a linear relation between some bits of the plaintext, some bits of the ciphertext and some bits of the user-provided key. While a relation between single bits of information which holds all the time (or none of the time) would be especially useful to a cryptanalyst, Matsui [7] showed that provided the relation does not hold exactly half the time, there are ways to extract key information by analyzing a large enough set of known plaintext and ciphertext pairs.

There are two basic approaches. The first is to use an approximation which relates bits of the user-defined key and the plaintext/ciphertext data in a linear way thereby providing one bit of key information when sufficient data is available. The second is to identify, by analysis of the block cipher, some bits of a linear approximation that depend for their value on a small subset of bits in the user-defined key. It is then assumed that only by making a correct guess for these key bits will the anticipated bias in certain bits of the plaintext/ciphertext data be detectable. Matsui showed how to use these key-bit guessing techniques in what are sometimes referred to as the *1R-* and *2R-methods*. The block ciphers we are concerned with are iterative and repeatedly use a round transformation during encryption. With the 1R-method the cryptanalyst guesses the value of part of the user-provided key in either the first or the last rounds, while in the 2R-method the guess is for part of the user-provided key from both the first and the last rounds simultaneously.

3 Linear and Non-Linear Approximations

Linear approximations are built up by analyzing individual rounds of the block cipher. For ease of exposition, we will consider a Feistel cipher [2] where at round i of the cipher we denote the partially encrypted data input to the round

as C_h^{i-1} and C_l^{i-1}; the high-order half of the data and the low-order half of the data respectively. We shall denote the action of the round function with subkey k_i by $f(\cdot, k_i)$ and the output from the i^{th} round of the cipher will be written as $C_h^i = C_l^{i-1}$ and $C_l^i = C_h^{i-1} \oplus f(C_l^{i-1}, k_i)$. Note that in this notation C_h^0, C_l^0 constitutes the plaintext and C_l^r, C_h^r constitutes the ciphertext produced by the r-round cipher (since there is no swap in the last round).

3.1 Joining Approximations

The notation $C_l^{i-1}[\alpha]$ (and later β, γ and δ) is used to denote a general and unspecified linear sum of bits of the data block C_l^{i-1}. An approximation to the action of a single round of the cipher might be written as

$$C_h^{i-1}[\alpha] \oplus C_l^{i-1}[\beta] = C_h^i[\gamma] \oplus C_l^i[\alpha] \oplus k_i[\delta] \tag{1}$$

where $k_i[\delta]$ is a linear combination of subkey bits (the exact form of which will depend on the block cipher in question). By writing the approximation in this way, we are tacitly approximating the action of the round function by

$$\begin{aligned} C_l^{i-1}[\beta \oplus \gamma] \oplus k_i[\delta] &= C_h^{i-1}[\alpha] \oplus C_l^i[\alpha] \\ &= (C_h^{i-1} \oplus C_l^i)[\alpha]. \end{aligned} \tag{2}$$

Suppose now that we have some partially encrypted data C_h^{i-1} and C_l^{i-1} and let us consider an approximation which involves a non-linear function of bits in C_h^{i-1}. We shall use the notation $C_h^{i-1}[p(\alpha)]$ where α is used to identify some set of bits and $p(\cdot)$ is, in this case, a non-linear polynomial involving these bits. Now forming a one round approximation as we had in (1) is difficult because, as we can see from (2), it requires that

$$(C_h^{i-1} \oplus C_l^i)[p(\alpha)] = C_h^{i-1}[p(\alpha)] \oplus C_l^i[p(\alpha)].$$

and for non-linear $p(\cdot)$ this will not, in general, hold. But while one-round approximations that are non-linear in the output bits from $f(C_l^{i-1}, k_i)$ cannot be joined together (when bitwise exclusive-or is used to combine this output with C_h^{i-1}) non-linear approximations can still be used in a variety of ways.

First note that the input to an approximation to the first and last rounds of some cipher need not be combined with any other approximations. Consequently approximations to these rounds can equally be linear or non-linear expressions in bits of the data input. Second, and more interestingly, we note that the 1R- and 2R-methods of linear cryptanalysis make certain bits of the input to the second (or penultimate) round available to the cryptanalysis. We can use this to our advantage and non-linear approximations can potentially be used in the *second* and the *penultimate* rounds of an attack on some block cipher. We will demonstrate this practically with an improved attack on LOKI91.

There is however one major problem that we have yet to overcome, and that is to identify and use non-linear approximations to a single round of a cipher.

3.2 Non-Linear Approximations to a Single Round

To illustrate our approach to using non-linear approximations in a single round of a cipher, we shall use as our example the round function used in DES. Consider the input to the i^{th} round which we have denoted as C_h^{i-1} and C_l^{i-1}. The data C_l^{i-1} used as input to the round function $f(\cdot, k_i)$ is expanded from 32 to 48 bits and combined using exclusive-or with the subkey k_i for the round. The resultant 48 bits are then used as input to the non-linear transformation affected by eight S-boxes. The 32 bits produced as output are permuted and the result is combined using bitwise exclusive-or with C_h^{i-1}. The two data halves are then swapped.

Let us suppose that analysis of the S-boxes has revealed that an approximation consisting of a non-linear combination of some input bits to an S-box and a linear combination of the output bits is strongly biased. To exploit this in an attack, we need to transform the approximation across a DES S-box into an approximation across the entire round function. The output of the S-boxes can be easily related via the bit-wise permutation into an expression in the data bits output from the round function. For the non-linear combination of bits that are used as input to the S-boxes, it is harder to get an expression in terms of the bits that are used as input to the round function. This is because the key k_i is combined with the expanded C_l^{i-1} using bitwise exclusive-or. Denote the expansion of the data block C_l^{i-1} by $z_{47} \ldots z_0$. Combined with the key $k_{47} \ldots k_0$ this forms the input to the S-boxes $x_{47} \ldots x_0$ where $x_i = z_i \oplus k_i$ for $0 \leq i \leq 47$. Let us suppose, by way of illustration, that a non-linear approximation to the eighth S-box S8 involves $x_0 x_1$. Then depending on the actual values of k_0 and k_1 we can express $x_0 x_1$ in terms of z_0 and z_1. More explicitly, when $(k_0, k_1) = (0, 0)$ it is clear that $x_0 x_1 = z_0 z_1$ and when $(k_0, k_1) = (1, 1)$ we have $x_0 x_1 = z_0 z_1 \oplus z_0 \oplus z_1 \oplus 1$. For $(k_0, k_1) = (0, 1)$ we have that $x_0 x_1 = z_0 z_1 \oplus z_0$ and when $(k_0, k_1) = (1, 0)$ it follows that $x_0 x_1 = z_0 z_1 \oplus z_1$. Note that the key is *fixed* for all the data we collect. When a non-linear approximation is used in the first and/or last round of the cipher, the input to the round function can be directly observed in the plaintext or the ciphertext respectively. We might then assume that the value of the key bits involved in the non-linear terms of the approximation are fixed to some value and with a certain proportion of the keys, we will be correct in our analysis. We illustrate this phenomenon with a simple example using DES.

Example with DES. The following approximations to S-boxes S5 and S1 (`A`, `C` and `D` appear in [7, 8]) will be useful in attacking five-round DES. The input to an S-box will be denoted $x_5 \ldots x_0$ and the output $y_3 \ldots y_0$.

	box	*input*	*output*	$\lvert bias \rvert$
`A`	S5	x_4	$y_0 \oplus y_1 \oplus y_2 \oplus y_3$	20/64
`D`	S5	x_4	$y_1 \oplus y_2 \oplus y_3$	10/64
`C`	S1	x_2	y_2	2/64
`A'`	S5	$x_1 \oplus x_0x_1 \oplus x_0x_4 \oplus x_1x_5 \oplus x_4x_5 \oplus x_0x_1x_5 \oplus x_0x_4x_5$	$y_0 \oplus y_1 \oplus y_2 \oplus y_3$	24/64
`D'`	S5	$x_1 \oplus x_3 \oplus x_0x_3 \oplus x_0x_5 \oplus x_1x_3 \oplus x_1x_5 \oplus x_0x_1x_3 \oplus x_0x_1x_5$	$y_1 \oplus y_2 \oplus y_3$	18/64

Using the five-round linear approximation DCA-A, which holds with probability p where $(p - 1/2)^{-2} = 68,720$, we can recover one bit of key information with the following success rates over 50 trials:

plaintexts	17,180	34,360	68,720
success rate	74%	88%	98%

Alternatively, one bit of key information can be recovered using the non-linear approximation D'CA-A' which holds with probability p' where $(p' - 1/2)^{-2} = 14,728$. Note the reduced data requirements. Again, success rates are quoted for 50 trials.

plaintexts	3,682	7,364	14,728
success rate	86%	92%	100%

For this experiment, the key bits directly involved in the non-linear approximation in the outer rounds were fixed and known.

3.3 Recovering More Key Bits

In certain circumstances non-linear approximations can be used to give a mechanism which allows the recovery of more bits of key information with less plaintext. As might already be apparent, specific instances of our general approach are equivalent to Matsui's 1R- and 2R-methods.

So far we have dealt with the non-linearity in the input bits to some S-box by assuming that the key bits involved have a certain value and that for some proportion of the user-defined keys we are correct. There is however another approach. Whenever a product appears in a nonlinear approximation, the possible values for the key bits involved force us to consider alternative approximations. For instance, let us suppose that the product of the two least significant input bits x_0 and x_1 to S-box $S8$ in DES is equal to the linear sum of all the output bits from S-box $S8$ with some probability p. Define the absolute value of the bias of this approximation to be ϵ where $\epsilon = |p - 1/2|$. Suppose in our attack, that we know the corresponding bits z_0 and z_1 before transformation with the user-defined key k_0 and k_1 which gives $x_0 = z_0 \oplus k_0$ and $x_1 = z_1 \oplus k_1$. Since k_0 and k_1 are fixed, we can try each guess for k_0 and k_1 in turn with the data we have. When we make the correct key guess, we correctly reconstruct x_0 and x_1 the actual inputs to the S-box and hence the correct product x_0x_1. Suppose we guess incorrectly and choose $k_0 \oplus 1$ and k_1. Then we erroneously construct the values $x_0 \oplus 1$ and x_1 instead of x_0 and x_1. Now $(x_0 \oplus 1)x_1 = x_0x_1 \oplus x_1$ and this expression in the input bits will equal the sum of the output bits with probability p_1 say. Define $\epsilon_1 = |p_1 - 1/2|$. If $\epsilon_1 < \epsilon$ then by taking sufficient data the correct guess k_0, k_1 can be distinguished from $k_0 \oplus 1, k_1$. If $\epsilon_1 > \epsilon$ then the incorrect key guess will dominate, though in a practical attack we would use the approximation with the greater bias anyway (and in so doing we would recover the correct key guess). If $\epsilon_1 = \epsilon$ then the two guesses cannot be distinguished.

Example with DES. We can use the approximation D'CA-A' as defined previously to recover key bits used in the non-linear approximation. Denote the

key bits used in S5 in round one as $k_5^1 \ldots k_0^1$ and the key bits used in S5 in round five as $k_5^5 \ldots k_0^5$. Analysis of `A'` and `D'` reveals that with `A'` we can reliably recover k_0^5, $k_1^5 \oplus k_4^5$ and k_5^5 and with `D'` we can recover k_0^1, k_1^1 and $k_3^1 \oplus k_5^1$. We obtained the following success rate over 50 trials when using the non-linear approximation `D'CA-A'` (which holds with probability p where $(p-1/2)^{-2} = 14,728$) to recover six bits of key material:

plaintexts	14,728	29,456	58,912	117,824
success rate	18%	38%	60%	82%

A Special Case. The 1R- and 2R-methods of Matsui, and even the basic technique of linear cryptanalysis which recovers one bit of key information, are all special instances of this more general technique.

With DES we might imagine using non-linear expressions of an S-box using all six input bits, see e.g. [15]. These 'approximations' would hold with probability 1 and we would expect to recover six bits of user-defined key. Of course, we could simply represent the action of these polynomials by means of the look-up table for the S-box. This gives us precisely the 1R- and 2R-methods. By choosing an incorrect key guess we are in effect deriving a different approximation to the S-box which holds with a reduced bias. With sufficient data the correct key guess can be distinguished. Note that when we have the polynomial expressions at our disposal we can actually evaluate which of the incorrect guesses are most likely to occur. In this way we can improve our basic attacks by allowing for certain, predicted incorrect answers and adjusting them accordingly. In experiments on DES this gives us an improvement in our attacks, but not by a significant margin.

4 Implications

4.1 Greater Cryptanalytic Flexibility

In practice, we recover key bits using non-linear techniques by first counting the number of plaintext/ciphertext pairs that fall into a variety of classes. These classes are defined according to the text involved in the nonlinear approximation (the *effective text* bits). We then process this data by guessing each possible value for the key bits involved in the non-linear approximation (the *effective key* bits) and combine this guess with the effective text. In this way scores can be kept for the number of times the bit identified by the linear approximation to the rest of the cipher is either 0 or 1. A guess can be made for the value of the effective key bits depending on these final scores. Thus the basic work effort in processing the data once it has been initially sorted is 2^{k+t} where k is the number of effective key bits and t is the number of effective text bits.

It is now clear that our more general approach to the use of non-linear approximations has numerous practical implications beyond the use of approximations with greater absolute biases. Using Matsui's 1R- and 2R-methods, the cryptanalyst is unnecessarily restricted to using a number of effective key and text bits that is a multiple of the number of bits involved in the input to some S-box. When larger S-boxes are used, and this is a common recommendation [13],

Matsui's 2R- and even the 1R-methods can become impractical just because the number of effective text and key bits becomes excessive. Existing examples of ciphers where the 2R-method is impractical include FEAL [14] and LOKI91 [16]. When the S-boxes are so large that the 1R-method itself becomes impractical then it might previously have been argued that the cryptanalyst would be reduced to recovering just a single bit of user-defined key. Instead the cryptanalyst can use non-linear approximations, in the fashion we have described here, to recover additional bits from the user-defined key. These techniques can be used to supplement the 2R-method, they can be used to supplement the 1R-method when the 2R-method is impractical and they can be used even if both the 2R- and 1R-methods are infeasible.

Example. In [12] "almost perfect non-linear functions" were studied. For ciphers constructed using these functions, linear approximations will have low absolute biases. Examples of such functions are $f(x) = x^{2^k+1}$ in $GF(2^n)$ for odd n [12]. The output bits of f are quadratic in the input bits and any linear approximation for f will have an absolute bias at most $2^{\frac{n+s}{2}-1}/2^n$, where $s = gcd(k, n)$ [10]. For a Feistel cipher with round function $F(x, k) = f(x \oplus k)$ with $n = 33$, $k = s = 1$ (given as an example in [12]) this yields a maximum bias for one round of 2^{-17}. Clearly, the 2R-method is impossible for this cipher, and the 1R-method requires many effective text and key bits. However the functions f are only quadratic, so non-linear approximations which involve only a few input bits might provide improved opportunities for attack. Experiments on the functions f defined above for small values of n confirm this. For $n = 7$, $k = s = 1$, the absolute value of the bias of a linear approximation is at most 8/128. With just two input bits, there exist non-linear approximations with absolute biases 16/128. For $n = 9$ and $k = s = 1$, the bias of a linear approximation is at most 16/512 yet with three input bits there exist non-linear approximations with biases 32/512. It is immediately clear that by using our non-linear techniques in the outer rounds of the cipher, the basic linear cryptanalytic attack can be readily improved.

4.2 The Non-Linear Approximation of Inner Rounds

While we might be familiar with the use of non-linear techniques in the outer rounds of a cipher it is interesting to observe that non-linear approximations can also be used in the second and penultimate round of a cipher. To illustrate this, suppose for some cipher that n bits from an S-box in round one are mapped to the same S-box in round two and that by using t effective text bits we can replicate the output from the S-box in round one. When this output is correct, n input bits to a single S-box in round two will be correct and we can use a non-linear function of these input bits in an approximation of the second round instead of the linear function that techniques currently demand. In practice we would increase the number of effective text bits to $t + n$ by additionally considering certain bits of C_h^0 to be effective. There would also be an increase in the number of effective key bits, to accommodate those used in the second round, but these

might well be recoverable during the attack anyway. We provide experimental verification of this approach in our attack on LOKI91.

While it might appear that we are only able to improve attacks on a round by round basis, such improvements should not be overlooked. The plaintext requirements in a linear cryptanalytic attack are considered to be proportional to ϵ^{-2} where ϵ is the bias of the approximation [8] and increases in the bias of just two rounds of a cipher by a factor of $\sqrt{2}$ will give a reduction in plaintext requirements by a factor of 4.

5 LOKI91

LOKI91 is a DES-like block cipher that operates on 64-bit blocks and uses a 64-bit key [1]. The most interesting feature of LOKI91 for our purposes is that the cipher uses four identical S-boxes which map 12 bits to 8. Evidence for the resistance of LOKI91 to linear cryptanalysis was recently provided by Tokita et al. [16]. In this section we provide experimental verification of our new techniques. While we mounted our attacks on four-round LOKI91 (for reasons of practicality) the approximation we chose matched the outer rounds of the best linear approximation [16] that would be used to attack $(4 + 3r)$-round LOKI91 for $r > 0$. Clearly the plaintext requirements N for a linear cryptanalytic attack increase substantially as we add more rounds and we note that 16-round LOKI91 (when $r = 4$) remains immune to these attacks[4]. We will show that it is straightforward to use non-linear approximations in the first two rounds and in the last round of LOKI91 simultaneously, thereby improving the basic linear cryptanalytic attack. The polynomials we will use in our attack are given in Table 1, where we denote the input to the 12-bit S-box by $x_{11} \ldots x_0$ and the output by $y_7 \ldots y_0$.

Tokita et al. [16] point out that the S-boxes in LOKI91 are too large to allow the cryptanalyst to use the 2R-method and they restrict themselves to considering only the 1R-method as an alternative. This allows the recovery of 13 bits of user-defined key (with a work-effort proportional to 2^{24} operations). By reversing the role of the plaintext and ciphertext, potentially another 13 bits can be recovered leaving 38 bits to be discovered by exhaustive search (with a 2^{38} work effort).

With the non-linear approximations we have identified however, we can mount a range of attacks that are quite different from the typical approach of `?X-Y` where we use `?` to denote Matsui's 1R-method in the first round. These attacks have different work efforts and recover different numbers of key bits. By allowing for more work during the analysis of the data, more key bits might be recovered or alternatively less plaintext might be required for a successful attack.

The results of a series of experiments can be found in the attached Appendix. While we have obtained direct empirical evidence for the effectiveness of some

[4] For 4-, 7- and 10-round LOKI91 the known plaintext requirements are 2^{23}, 2^{40} and 2^{58} respectively. For 13- and 16-round LOKI91, linear cryptanalytic techniques are infeasible.

Table 1. Some linear and non-linear approximations for LOKI91.

	box	*input*	*output*	$\|bias\|$
X	S2	$x_2 \oplus x_6 \oplus x_{10}$	$y_4 \oplus y_5 \oplus y_6$	88/4096
Y	S2	$x_2 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_8$	$y_4 \oplus y_5 \oplus y_6$	108/4096
X'	S2	$x_2 \oplus x_{10} \oplus x_{10}x_6$	$y_4 \oplus y_5 \oplus y_6$	116/4096
Y.1	S2	$x_2 \oplus x_3 \oplus x_5 \oplus x_5x_7 \oplus x_3x_8 \oplus$ $x_5x_8 \oplus x_3x_5x_8 \oplus x_3x_7x_8 \oplus x_3x_5x_7$	$y_4 \oplus y_5 \oplus y_6$	136/4096
Y.2	S2	$x_2 \oplus x_3 \oplus x_5 \oplus x_8 \oplus x_5x_7 \oplus x_7x_8 \oplus$ $x_2x_3x_5 \oplus x_5x_7x_8 \oplus x_8x_5x_2$	$y_4 \oplus y_5 \oplus y_6$	130/4096
Y.3	S2	$x_2 \oplus x_3 \oplus x_5 \oplus x_7 \oplus x_3x_8 \oplus x_3x_7 \oplus$ $x_3x_5x_7 \oplus x_5x_7x_8 \oplus x_3x_5x_8$	$y_8 \oplus y_5 \oplus y_6$	110/4096

Table 2. The complexity of conventional and various new linear cryptanalytic attacks on LOKI91.

Attacks on $(4+3r)$-round LOKI91				
approximation	*# plaintexts*	*success rate*	*# key bits recovered*	*work effort*
?X-Y *current methods [16]*	$6,232,416$ $(r=0)$ N $(r>0)$	94%	13	2^{24}
?X'-Y	$3,586,800$ $(r=0)$ $0.58 \times N$ $(r>0)$	90%	15	2^{29}
?X'-Y.1	$2,261,912$ $(r=0)$ $0.36 \times N$ $(r>0)$	84%*	15	2^{29}
?X'-Y.1	$2,261,912$ $(r=0)$ $0.36 \times N$ $(r>0)$	74%+	18	2^{37}
?X'-Y.1	$2,261,912$ $(r=0)$ $0.36 \times N$ $(r>0)$	68%+	19	2^{37}
?X'-Y.2 *and* ?X'-Y.3 *simultaneously*	$1,442,632$ $(r=0)$ $0.23 \times N$ $(r>0)$	86%+	20	2^{38}

* this applies to 1/16 keys but is an empirical result

\+ a prediction derived from results presented in the Appendix

of our attacks, we have used experimental evidence to predict the success rate of others.

The results of our work have been summarized in Table 2. The number of key bits recovered refers to this single phase of the attack alone and further gains by reversing the role of plaintext and ciphertext have not been considered. When considering the work effort involved, recall that current methods already require a work effort proportional to 2^{38} encryptions in exhaustive search for the key bits not recovered via linear cryptanalysis.

We have also used multiple non-linear approximations in much the same way we might use multiple linear approximations [5]. The use of multiple non-linear approximations is much more complicated than the use of multiple linear

approximations and considerable care has to be taken in deciding which non-linear approximations should be used together and exactly which bits of key information can be reliably recovered. Results given in the Appendix and in Table 2 demonstrate that additional substantial savings in the plaintext requirements can be expected in this way.

We see that numerous trade-offs are possible between the number of key bits recovered, the amount of plaintext required and the work effort the cryptanalyst might wish to invest in attacking some cipher. In short, the use of non-linear approximations offers greatly improved flexibility to the cryptanalyst.

6 Conclusions

We have presented a general approach to linear cryptanalysis which allows us to consider within the same framework all linear cryptanalytic techniques currently used. While this has opened numerous avenues for research it is already evident that there are several new developments.

When trying to accurately gauge the resistance of a block cipher to linear cryptanalysis, it is no longer sufficient to restrict attention to Matsui's 1R- and 2R-methods of linear cryptanalysis. There may well be circumstances where non-linear approximations, involving far fewer text and key bits than are required to describe an S-box, can be used to recover additional bits of the user-defined key with less plaintext than current linear techniques might suggest. Consequently our techniques offer the cryptanalyst much more flexibility in attacking a cipher than was previously appreciated. By adjusting the various requirements in an attack, the cryptanalyst can decide on the approach that is best suited to the resources available be they the amount of available data or the amount of computing power possessed by the cryptanalyst. These techniques will be a particular concern for ciphers that depend for their security on the fact that the 1R- and/or the 2R-methods are impractical due to reasons of work-effort rather than the amount of data required. We have also noted that some block cipher designs allow the use of non-linear approximations in the second and penultimate rounds of a cipher.

We have confirmed our techniques with attacks on reduced-round LOKI91 and we expect that seven additional bits of key information can be recovered with less than 1/4 of the plaintext than current techniques require. Further improvement may well be possible. In short, the additional flexibility available to a cryptanalyst has been demonstrated and linear cryptanalytic attacks on a wide variety of block ciphers may well be much improved with these new methods.

References

1. L. Brown and M. Kwan and J. Pieprzyk and J. Seberry. Improving resistance to differential cryptanalysis and the redesign of LOKI. In H. Imai and R.L. Rivest and

T. Matsumoto, editors, *Advances in Cryptology — AsiaCrypt '91*, Lecture Notes in Computer Science 453, Springer-Verlag (1993), 36–50.

2. H. Feistel. Cryptography and computer privacy. *Scientific American*, 228(5):15–23, 1973.
3. C. Harpes and G.G. Kramer and J.L. Massey. A generalization of linear cryptanalysis and the applicability of Matsui's piling-up lemma. In L.C. Guillou and J.J. Quisquater, editors, *Advances in Cryptology — Eurocrypt '95*, Lecture Notes in Computer Science 921, Springer-Verlag (1995), 24–38.
4. B.S. Kaliski and M.J.B. Robshaw. Linear cryptanalysis using multiple approximations. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, Lecture Notes in Computer Science 839, Springer-Verlag (1994), 26–39.
5. B.S. Kaliski and M.J.B. Robshaw. Linear cryptanalysis using multiple approximations and FEAL. In B. Preneel, editor, *Fast Software Encryption*, Lecture Notes in Computer Science 1008, Springer Verlag (1995), 249–264.
6. S.K. Langford and M.E. Hellman. Differential-linear cryptanalysis. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, Lecture Notes in Computer Science 839, Springer Verlag (1994), 17–25.
7. M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology — Eurocrypt '93*, Lecture Notes in Computer Science 765, Springer-Verlag (1994), 386–397.
8. M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, Lecture Notes in Computer Science 839, Springer-Verlag (1994), 1–11.
9. National Institute of Standards and Technology (NIST). *FIPS Publication 46-2: Data Encryption Standard.* December 30, 1993.
10. K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology — Eurocrypt '93*, Lecture Notes in Computer Science 765, Springer-Verlag (1994), 55–64.
11. K. Nyberg. Linear approximation of block ciphers. In A. De Santis, editor, *Advances in Cryptology — Eurocrypt '94*, Lecture Notes in Computer Science 950, Springer-Verlag (1995), 439–444.
12. K. Nyberg and L.R. Knudsen. Provable security against a differential attack. *The Journal of Cryptology*, 8(1):27–38, 1995.
13. L. O'Connor. Properties of linear approximation tables. In B. Preneel, editor, *Fast Software Encryption*, Lecture Notes in Computer Science 1008, Springer Verlag (1995), 131–136.
14. K. Ohta and K. Aoki. Linear cryptanalysis of the Fast Data Encipherment Algorithm. In Y. Desmedt, editor, *Advances in Cryptology — Crypto '94*, Lecture Notes in Computer Science 839, Springer-Verlag (1994) 12–16.
15. I. Schaumüller-Bichl. Cryptanalysis of the Data Encryption Standard by a method of formal coding. In T. Beth, editor, *Cryptography, Proc. Burg Feuerstein 1982*, Springer-Verlag (1983), 235–255.
16. T. Tokita and T. Sorimachi and M. Matsui. Linear Cryptanalysis of LOKI and s^2DES. In J. Pieprzyk and R. Safavi-Naini, editors, *Advances in Cryptology — Asiacrypt '94*, Lecture Notes in Computer Science 917, Springer-Verlag (1995), 293–303.

Appendix

Experimental verification of the attack by Tokita et al. [16] on LOKI91 is provided in the following table with our experiments being carried out on a four-round version of the cipher. The results were obtained after 50 trials using Matsui's 1R-method and the approximation `?X-Y` as described previously. The approximation holds with probability p where $(p-1/2)^{-2} = 779,052$ and 13 bits of key information can be recovered.

plaintexts	1,558,104	3,116,208	6,232,416
success rate	20%	64%	94%

By substituting the non-linear approximation `X'` for the linear approximation `X` used in round two of the approximation, we can obtain improved attacks. First we provide the success rate over 50 trials in recovering 15 bits of key information; we use Matsui's 1R-method and the approximation `?X'-Y` which holds with probability p where $(p-1/2)^{-2} = 448,350$.

plaintexts	896,700	1,793,400	3,586,800
success rate	6%	38%	90%

To help in our later analysis, we will compare these success rates with those obtained over 50 trials when using the same approximation to recover just three bits of key information. Here we assume that the 12 bits of effective key used in S-box S2 in the first round remain fixed and known. This allows us to estimate how the success rate might degrade when we have to recover these additional 12 bits of key.

plaintexts	896,700	1,793,400	3,586,800
success rate	76%	92%	100%

Now consider using non-linear approximations in the last round by replacing `Y` with a non-linear approximation `Y.1` described previously. For one in 16 keys we obtain the following success rates over 50 trials when using Matsui's 1R-method and the approximation `?X'-Y.1` which holds with probability p where $(p-1/2)^{-2} = 282,739$. For one in 16 keys, 15 bits of key information can be recovered with the following success rates:

plaintexts	565,478	1,130,956	2,261,912
success rate	8%	22%	84%

Instead of assuming the value of four key bits in the last round and being correct some proportion of the time we can recover these four key bits. To estimate the success rate of this approach, we will use Matsui's 1R-method with the correct guess to S2 in round one and the approximation `?X'-Y.1` (which holds with probability p where $(p-1/2)^{-2} = 282,739$). In this way seven bits of key information can be recovered with the following success rates (over 50 trials):

plaintexts	565,478	1,130,956	2,261,912
success rate	40%	50%	76%

Alternatively, since recovery of one of the seven bits is somewhat unreliable we might recover just six bits. Then the success rates over 50 trials become:

plaintexts	565,478	1,130,956	2,261,912
success rate	55%	62%	82%

Using this information, we can now make predictions for the expected success rate in attacking LOKI91. We saw earlier that by deriving the 12 key bits of S-box 2 in the first round instead of fixing them as correct, our success rate with 3,586,800 plaintexts fell from 100% to 90%. From this we might estimate that by using `?X'-Y.1` we can recover 19 bits of key information (instead of 13) with a little more than one third the plaintext (2,261,912 instead of 6,232,416 plaintexts) with a slightly reduced success rate of 68% = 76% × 90% (from 94% previously). We could of course, suffice with recovering 18 bits of key information, in which case we might expect a success rate of 74%.

We might also consider the use of multiple non-linear approximations. Despite the additional complications of using multiple non-linear approximations, we note that more bits of user-defined key might be recovered with less plaintext. In the following table we give the success rates achieved in 50 trials with two non-linear approximations `?X'-Y.2` and `?X'-Y.3` defined previously. In these experiments we assume the correct key bits used in S2 in round one and we recover eight bits of key information.

plaintexts	360,658	721,316	1,442,632
success rate	26%	66%	96%

Using these results we might predict that we can use the two approximations `?X'-Y.1` and `?X'-Y.2` to recover 20 bits of key information instead of `?X-Y` to recover 13 bits of key information with essentially the same success rate (86% instead of 94%) but with much less than one quarter the plaintext (1,442,632 plaintexts instead of 6,232,416).