A Cryptographic Review of Cipherunicorn-E

M.J.B. Robshaw
16d Stowe Rd.
London
W12 8BN
mrobshaw@supanet.com

December 14, 2001

Executive summary

This report describes a brief cryptographic review of Cipherunicorn-E. While a broad range of attacks were considered, our attention was particularly focused on differential and linear cryptanalysis as requested. Cipherunicorn-E is a complicated cipher which hinders accurate analysis. This should be constrasted with other ciphers that permit a reasonably close assessment. However in this report we draw the following conclusions.

With regards to differential cryptanalysis, the techniques of the designers appear to be reasonable, given the complexity of the cipher. Some issues appear to have been overlooked in the self-evaluation report [16] and this suggests that the bound of 2^{-84} for the probability of an exploitable differential might be better replaced with 2^{-72}. With regards to linear cryptanalysis the situation is less clear. It seems that compromising even a limited number of rounds of Cipherunicorn-E with linear cryptanalytic techniques would be unlikely. However there might be good grounds to question some of the techniques used in establishing a bound for a linear cryptanalytic attack. Without considerable additional and very detailed analysis, it is difficult to comment more on the true state of the cipher. Nevertheless, no new attacks have been identified. So with the current state of knowledge, it seems unlikely that practical differential and linear cryptanalytic attacks can be easily mounted against Cipherunicorn-E.

This review took place over a limited time and with limited resources. It should be anticipated that additional analysis may well find improved results for the cryptanalysis of this cipher and provide a greater understanding of the true security offered.

1 Introduction

In this report we present the results of a brief cryptographic review of the block cipher CIPHERUNICORN-E. This cipher has been submitted to the *Cryptrec Evaluation* process and has already received considerable study by the designers. CIPHERUNICORN-E is a companion to CIPHERUNICORN-A and they share some functional components. However the specific details of the ciphers are sufficiently different that little of the analysis from one cipher is of immediate relevance to the other.

While some effort has been made to consider a broad range of possible attacks on the cipher, most effort was concentrated on considering the effectiveness of differential and linear cryptanalysis. The materials provided for this evaluation were

- Cryptographic techniques specifications: CIPHERUNICORN-E, FY 2000 submission, NEC Corporation. (Undated.)
- Notice of updates to the above report. NEC Corporation. (Undated.)
- Cryptographic techniques specifications: CIPHERUNICORN-E, Version 2, NEC Corporation. (Undated.) [15].
- Self Evaluation Report: CIPHERUNICORN-E, Version 2, FY 2000 submission, NEC Corporation. (Undated.)
- Notice of updates to the above report. NEC Corporation. (Undated.)
- Self Evaluation Report: CIPHERUNICORN-E, Version 3, NEC Corporation. (Undated.) [16].
- Copy of overhead slides: "64-bit Block Cipher CIPHERUNICORN-E (UNI-E)", NEC Corporation. (Undated.)

2 Terminology, definitions, and notation

Throughout this report we will assume that the reader has a basic familiarity with many different aspects of block cipher design and analysis particularly differential [1] and linear [11] cryptanalysis.

With differential cryptanalysis, we will typically consider a notion of difference that is given by bitwise exclusive-or. While other notions of difference might be considered, the design of CIPHERUNICORN-E is such that this particular measure is likely to be the most useful. In general, differences will be denoted by Δ. For linear cryptanalysis, we will require the use of so-called *parity masks* which will typically be denoted by Γ. Any specific values to either differences or parity masks will be presented in hexadecimal notation prefixed by `0x`.

CIPHERUNICORN-E relies on several structural components. These include integer addition modulo 2^{32}, denoted by $+$, and bitwise exclusive-or of both 8 and 32 bit data units denoted by $\oplus$. Four 8-bit to 8-bit substitution boxes S_0–S_3 are required and will typically be referred to as S-boxes. The cipher requires the use of a bitwise shift to the left. The shift of a to the left by r bit positions will be denoted by $a \ll r$. The bitwise `and` of two words a and b will be denoted by $a \wedge b$ and the Hamming weight of a word a is the number of ones in the binary representation of a.

3 Existing analysis of CIPHERUNICORN-E

The designers of CIPHERUNICORN-E have provided the results of their own evaluation of the cipher [16]. The bulk of this analysis appears to be concentrated on the results of extensive statistical testing. While this is not entirely without some merit, it is very unlikely that such testing, no matter how extensive, will uncover problems with the cipher. A cipher must pass such tests, but a cipher that has passed these statistical tests is not necessarily secure. In addition to statistical tests, the designers also discuss the resistance of the cipher to a wide-range of sophisticated cryptanalytic attacks. In particular, bounds on the effectiveness of differential and linear cryptanalysis were derived [16].

4 Description of CIPHERUNICORN-E

In this section we give an overview of the important features of CIPHERUNICORN-E. More details of the cipher can be found in the cipher documentation [15]. CIPHERUNICORN-E is built around the well-established Feistel design used in DES [13]. The cipher operates on 64-bit blocks with a 128-bit key.

4.1 The L function

The most obvious divergence from the Feistel approach at a structural level, is the use of a key-dependent mixing function $L(a, b) \rightarrow (x, y)$. This is used after every two rounds in the Feistel structure and takes as input two 32-bit words a and b giving as output two words x and y. Between each occurrence of the L function there are two rounds of the Feistel network. The effectiveness and impact of the L function will be assessed in Section 5.

4.2 The round function

CIPHERUNICORN-E uses a very complicated round function which is illustrated in Figure 1. One of the distinguishing features of both CIPHERUNICORN-A and

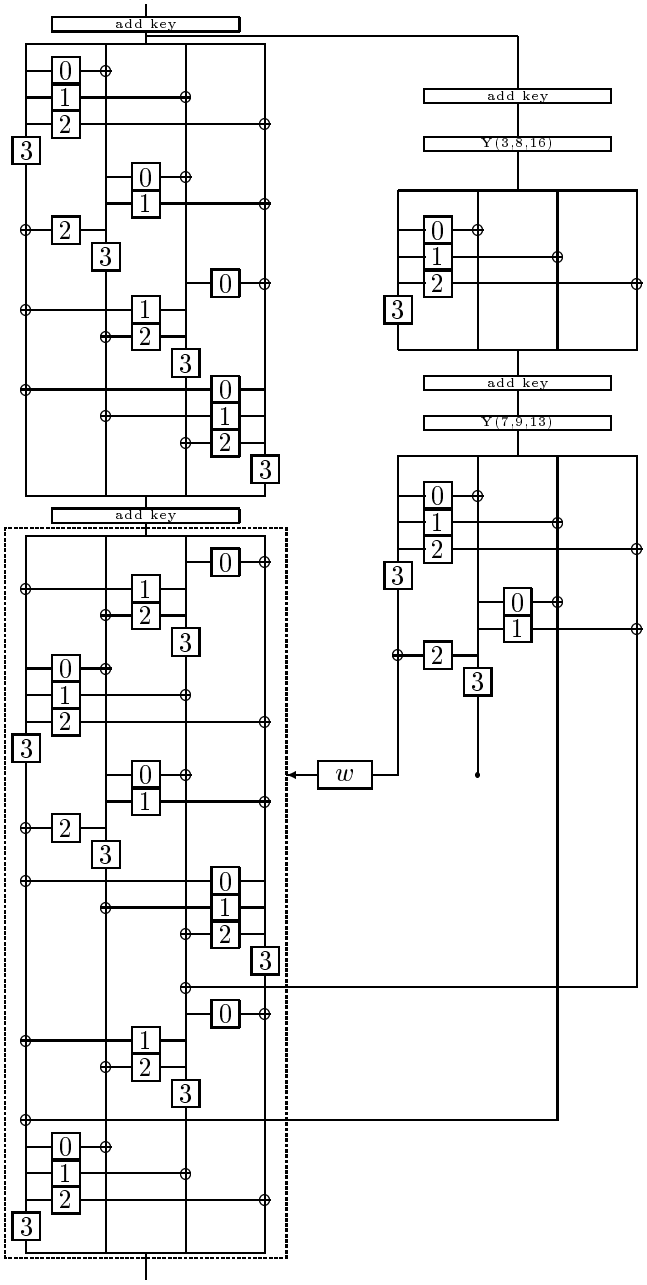
CIPHERUNICORN-E is that the round function consists of two parallel computations. We will often describe our analysis of the round function in terms of two processes which we refer to as Computation I and Computation II.

Computation I is quite traditional. The 32-bit data input is split into four bytes and processed via a network consisting of ten mini-rounds of exclusive-ors and S-box look-ups. There are four types of mini-round, each using a different byte as input but effecting all four bytes. Meanwhile Computation II is a parallel process that uses the same inputs as were used in Computation I. The results of this computation are then used to provide a limited interaction with the data in Computation I. Computation II indicates the choice and ordering of operations in mini-rounds five to ten and further provides two bytes of data for use in Computation I prior to mini-rounds nine and ten.

The design of the round function is such that each type of mini-round is used once (in a fixed order) within the first four mini-rounds. Then each type of mini-round is also used in mini-rounds five to eight, though the order they appear is dependent on information derived in the parallel Computation II. Mini-rounds nine and ten are a repeat of mini-rounds five and six.

We have yet to describe the introduction of key material to the round function. This happens in several places. First, the 32-bit input to the round function is added modulo 2^{32} to key material and the results are used as inputs to Computation I and Computation II. Second, key material is introduced between mini-rounds four and five where the four bytes of intermediate data are considered as a 32-bit quantity and combined using integer addition modulo 2^{32} with 32 bits of key material. Third, a four-bit value and two bytes of material are derived in a key and data-dependent manner via Computation II for use in Computation I. The four-bit index is used to choose the type and order of transformations in mini-rounds five to ten (as described above), the order being determined via a pre-defined [15] array $Sh[\cdot]$ which does not significantly concern us here. The two bytes of data are exclusive-ored with some of the intermediate data in the latter part of Computation I. This is illustrated in Figure 1.

Figure 1: CIPHERUNICORN-E round function with $w = 9$



4.3 The Y function

The function $Y_{r,s,t}(x)$ takes a 32-bit argument as input and returns a 32-bit output. It is described by the following equations where addition is carried out modulo 2^{32}

$$\begin{aligned} a &= x + (x \ll r), \\ b &= a + (a \ll s), \\ Y_{r,s,t}(x) &= b + (b \ll t). \end{aligned}$$

The function $Y_{r,s,t}(x)$ appears twice (with different values to r, s, and t) in Computation II of the round function. The particular values for the rotation constants are chosen according to given design principles [15]. Due to the limited time available it is unknown whether there are any particular weaknesses arising from the chosen set of values. Analysis in Sections 6 and 7.2 suggests that the function $Y_{r,s,t}(x)$ can make a tangible contribution to the security of the cipher.

4.4 The S-boxes

Four different 8-bit to 8-bit S-boxes are used in CIPHERUNICORN-E. They have been designed according to similar principles used in the AES [3, 14]. The construction and the properties of the S-boxes have not been checked and it is assumed that they have the properties claimed in the cipher documentation.

4.5 The key schedule

CIPHERUNICORN-E has a rather complicated key schedule requiring the iterated use of a nested byte-wise Feistel structure [15]. The key schedule has not been examined closely here. Further work might pay close attention to the implications of choosing keys with certain bytes values differing in the most significant bit; particularly since the boundaries between bytes is well-respected throughout. Provisional analysis failed to find an exploitable weakness, but further work might be profitable for the cryptanalyst. For the purposes of this report we will not pay any further attention to the key schedule. We will instead make the typical assumption that all subkey material throughout the cipher is determined independently of the rest.

4.6 Initial comments

CIPHERUNICORN-E is built around the well-established Feistel design [13]. The function L is perhaps intended to provide some moderate key-dependent mixing between the two Feistel strands. We will discuss this function in more detail in Section 5. However any additional benefit from L is unclear. Indeed, as we will show in Section 5, it appears that the function L might potentially *reduce* the security of the cipher! As a certificational weakness of this structure, we

will also observe that if this function L were to be used in every other round (for which it might be argued that this would provide even more complicated mixing) then the whole cipher would be trivially weak.

The round function itself is very complicated. The fact that there are two strands of computation running concurrently with only limited interaction between them raises several questions. At first sight it seemed that the bulk of the cryptographic strength of CIPHERUNICORN-E might be derived from the use of the S-boxes in Computation I. Since Computation II is used to vary the flow of operations in Computation I (of which there are only 16 possibilities) and to provide 16 bits of data that are combined with Computation I using exclusive-or (a questionable advantage with regards to a differential attack) then the value of Computation II was unclear. However more advanced analysis in Section 7.2 suggests that Computation II is vital for the security of the cipher. More particularly, the function Y appears to make a tangible contribution to the security of the cipher. It is not clear whether this property should be seen as a positive attribute of Computation II or a negative attribute of Computation I. However, on balance, it seems that a less complicated but more robust design to the round function might have been preferable.

4.7 Some simplifications to CIPHERUNICORN-E

The round function of CIPHERUNICORN-E is too complicated to allow a complete and accurate analysis. Indeed, it appears to have been a design principle that the cipher achieve high security goals by being difficult to analyze. This is an unusual approach. It is more common for cipher designers to aim to provide as complete an understanding of the behavior of the cipher as possible so as to fully appreciate the true security level offered.

However we still need to develop an understanding of the cipher. To do this, we will consider some possible simplifications to the round function.

1. The designers of CIPHERUNICORN-E consider a variant of the round function that is illustrated in Figure 2. This is identical to the round function used in the full cipher, except that the integer addition of key material has been omitted and the function Y has been replaced by a modified function Y^{rep} where

$$Y^{\mathrm{rep}}(x) = x \oplus (x \ll 24) \oplus (x \wedge \mathtt{0xff00}) \ll 16) \oplus (x \wedge \mathtt{0xff0000}) \ll 8).$$

   Replacing the function Y with the function Y^{rep} appears to be a reasonable cryptanalytic tool. Apart from a few degenerate examples, it seems unlikely that the analysis resulting from this simplified version will be catastrophically different to that attained in the full cipher (see Section 6). We will denote this variant of the cipher UNI-E-REP-Y thereby indicating that the function Y has been replaced.

2. A second useful simplification of the round function is derived by omitting the function Y entirely but retaining the integer addition of key material. This is illustrated in Figure 3. We will denote this variant of the cipher UNI-E-NO-Y.

3. A third simplification might be derived by omitting both the function Y and the integer addition of the key material. However there seems to be little advantage in considering this variant.

5 The Function L

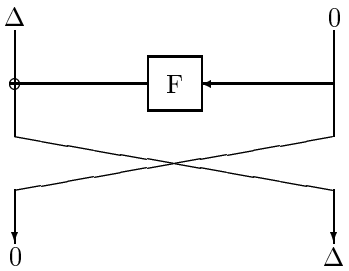
CIPHERUNICORN-E is essentially a Feistel cipher with some additional key-dependent mixing performed every two rounds. This mixing is accomplished by a function L which maps, under the influence of key material k_0 and k_1, two 32-bit words (a, b) to two 32-bit words (x, y) according to the equations

$$\begin{aligned} x &= a \oplus (b \wedge k_1) \oplus (a \wedge k_0 \wedge k_1), \text{ and} \\ y &= b \oplus (a \wedge k_0) \oplus (b \wedge k_0 \wedge k_1). \end{aligned}$$

It might be presumed that the purpose of this operation is to provide additional mixing between the two strands of computation within the Feistel structure. Yet, it appears that the cryptographic significance of this operation is very limited. The essential cryptographic strength of CIPHERUNICORN-E comes from the round functions. Indeed, it is not clear whether the mixing function L might not, under certain exceptional circumstances, make matters easier for the cryptanalyst.

Some of this has already been observed by the designers of CIPHERUNICORN-E. In the self-evaluation report [16] it was observed that if $k_0 =$ `0xffffffff` and $k_1 =$ `0xffffffff` then $L(a, b) = (b, a)$. That is the two 32-bit words of the Feistel structure are swapped. However it is reasonably observed, that this is so unlikely that this particular issue should not be viewed as a problem [16].

What might be a problem, though, is a similar phenomenon on a more limited scale. Suppose that an attacker is interested in mounting a differential attack with a 32-bit difference given by $\Delta =$ `0x00000001` (say). Further, suppose that we have a 64-bit input difference given by $(\Delta, 0)$. Then for one in four keys we have that $L(\Delta, 0) = (0, \Delta)$. Only two out of the 64 key bits needs to take the value one for this to occur since we don't care about the other bit positions. Now consider a single round of a Feistel structure.



In a differential attack a round is said to be *active* if there is non-zero input difference to the round function F. Typically an attacker will try to reduce the number of active rounds in a differential attack so as to increase its effectiveness. Three types of rounds are interesting to us for CIPHERUNICORN-E.

Δ	0	$\rightarrow$	0	Δ	(illustrated)	(1)
0	Δ	$\rightarrow$	Δ	Δ'		(2)
0	Δ	$\rightarrow$	Δ	0		(3)

The first differential is a trivial differential and the round is not active. The second and third differentials both provide active rounds. The third one is very interesting and it is not typically possible for a Feistel cipher. However, the structure of the round function of CIPHERUNICORN-E is such that this type of differential could be a (very remote) possibility.

One way to estimate the resistance of a cipher to differential cryptanalysis (and the approach that is adopted by the designers of CIPHERUNICORN-E [16]) is to provide a lower bound on the probability of an active round, and then to provide a lower bound on the number of active rounds required for a differential attack.

In making such estimates it is typical to assume that the attacker can mount what is referred to as a 2R-attack. That is, the outer two rounds of the cipher can be removed by the cryptanalyst. (We will indicate such a round by U for "unwind".) In fact, these outer rounds are usually required for the recovery of key material but that does not concern us here. Instead, we merely remark that for the purposes of a conservative analysis, we will assume that a differential need only extend over 14 of the 16 rounds of the cipher.

Given this, and our earlier observation about the function L swapping differentials, we can now make the following observation. If differentials of type (3) are possible, then the function L might allow the following pattern to the rounds in a differential attack where A denotes an active round, U an "unwound" round, and "$-$" an inactive round.

$$U\ -\ -\ A\ A\ -\ -\ A\ A\ -\ -\ A\ A\ -\ -\ U.$$

This is instead of what would otherwise be one of the optimal attacks for a version of the cipher when the function L is not used.

$$U\ -\ A\ -\ A\ -\ A\ -\ A\ -\ A\ -\ A\ -\ A\ U$$

We can immediately see that the use of the L function can reduce the number of active rounds from seven to six. Further, depending on the Hamming weight of characteristics within the differential, the proportion of keys for which this happens need not be that significant.

It is now trivial to see (but perhaps still worth observing) that a variant of CIPHERUNICORN-E where the function L is used after every round of the Feistel computation would be trivially weak. This might be a little counter-intuitive since L is intended to provide key-dependent mixing between the strands of the Feistel structure and it might be argued that including this function after every round would make matters harder for the cryptanalyst. Yet starting the cipher with an input exclusive-or difference of $(\Delta, 0)$ where $\Delta = \texttt{0x00000001}$ (say), one round of the Feistel network would give the difference $(0, \Delta)$ as input to L. For one in four keys, the output from L would be $(\Delta, 0)$ to be used as input for the start of the next Feistel round. So on, and so forth, throughout as many (inactive) rounds of the cipher as we care to go. Over 14 rounds of the cipher this would give a characteristic holding with probability 1 for a fraction of 2^{-14} of all possible keys. However this does not apply to CIPHERUNICORN-E.

At this level of analysis, one impact of the function L seems to be in possibly reducing the minimum number of active rounds in a differential attack from seven to six. This will have an effect on the bounds for the effectiveness of differential cryptanalysis on CIPHERUNICORN-E.

6 The Function Y

The function Y is a particularly useful function in CIPHERUNICORN-E. It is described by the following equations where addition is carried out modulo 2^{32}

$$\begin{aligned} a &= x + (x \ll r), \\ b &= a + (a \ll s), \\ Y_{r,s,t}(x) &= b + (b \ll t). \end{aligned}$$

The typical result of an application of the function Y is to amplify any small difference between two input words. The intention seems to be to improve the avalanche of change in the cipher and in this it can be expected to be reasonably effective. This is particularly the case since change tends to be introduced towards the arithmetically more significant bits of the word, and it is the most significant byte of the output from Y that is immediately used as input to the S-boxes.

It should be observed that fixed characteristics for Y are possible. For example, $Y(\Delta) = \Delta = \texttt{0x80000000}$. There are also some degenerate cases where the function Y can actually *reduce* the Hamming weight of a difference between two inputs. For example, choosing $a = \texttt{0x90000000}$ and $b = \texttt{0x00000000}$ we have an input exclusive-or difference $a \oplus b = \texttt{0x90000000}$ yet for $Y_{3,8,16}$, the

output difference $Y_{3,8,16}(a) \oplus Y_{3,8,16}(b) = \texttt{0x10000000}$. However these, and any other such cases, might be viewed as exceptional.

Unfortunately the function Y is rather difficult to analyze. Certainly it is difficult to incorporate the role of Y into a broader analysis of the CIPHERUNICORN-E round function. It might be argued that despite the use of integer addition, exclusive-or differences between the inputs to Y will be modified by the function in a reasonably predictable way. Thus it is interesting to consider the resistance of CIPHERUNICORN-E to differential cryptanalysis when this function is not present. However, it does appear that the modification Y makes to an input difference Δ can be significant. This is particularly the case since it is the only operation that effectively operates across byte boundaries.

As mentioned in Section 4.7 one simplification to the cipher is to replace the function Y with $Y^{\text{rep}}(\cdot)$ described by

$$Y^{\text{rep}}(x) = x \oplus ((x \wedge \texttt{0xff}) \ll 24) \oplus (x \wedge \texttt{0xff00}) \ll 16) \oplus (x \wedge \texttt{0xff0000}) \ll 8).$$

As an approximation this seems to be reasonable. Small exclusive-or differences are magnified somewhat and changes are propagated towards the most significant bits in a word. At this level of analysis, it is difficult to imagine a circumstance where cryptanalysis of the round function used in CIPHERUNICORN-E will be significantly easier than cryptanalysis of the same round function when Y is replaced with Y^{rep}.

After a brief review, it seems reasonable to assume that conservative estimates for the resistance of UNI-E-NO-Y to attack are likely to be conservative estimates for the strength of UNI-E-REP-Y. In turn, such estimates for UNI-E-REP-Y are likely to provide conservative estimates for the security of CIPHERUNICORN-E itself.

Figure 2: Modified CIPHERUNICORN-E round function with $w = 15$.

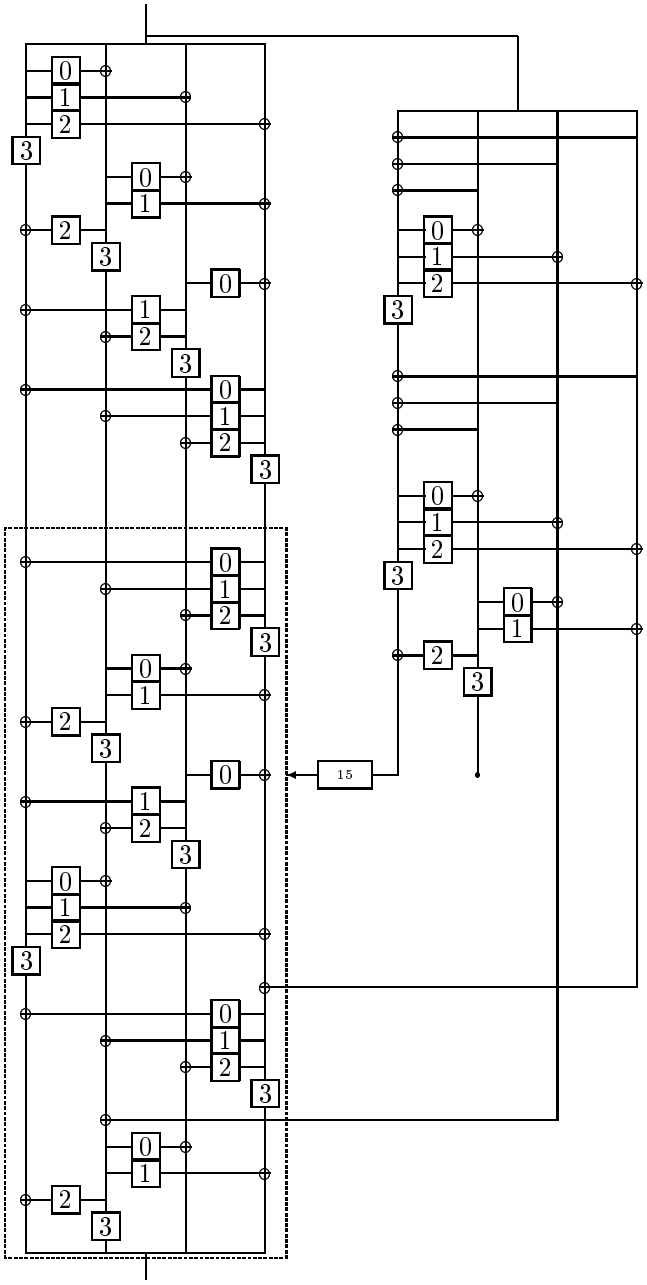
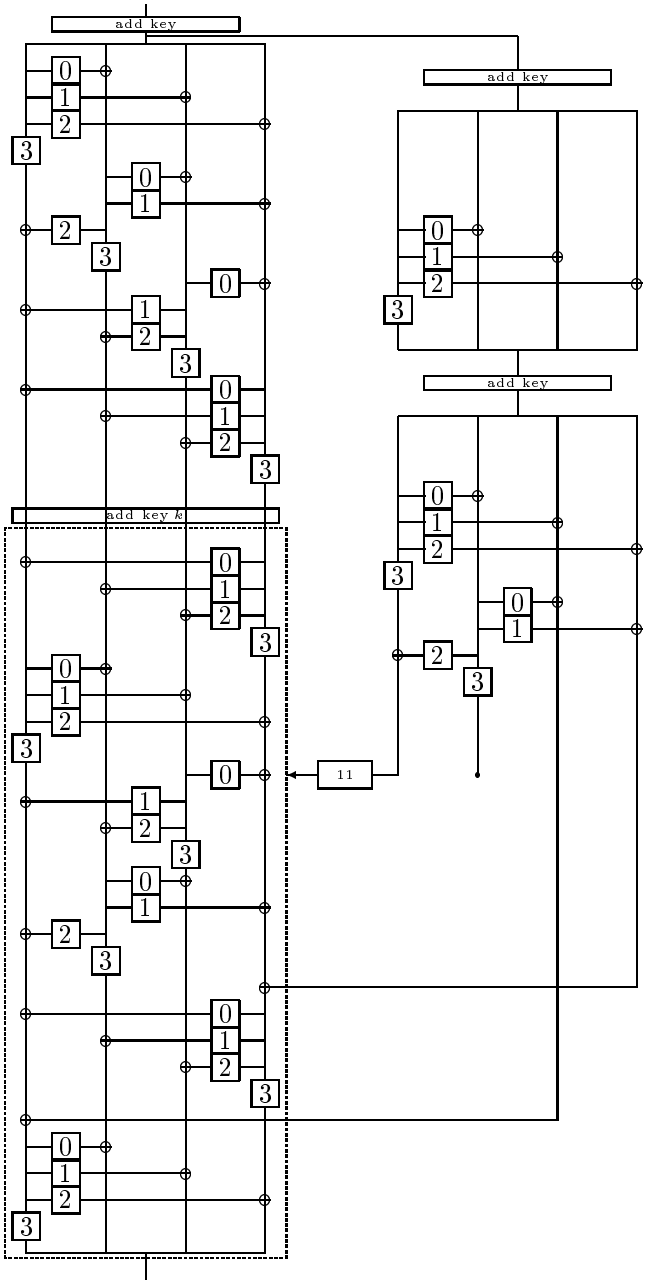


Figure 3: Second modified CIPHERUNICORN-E round function with $w = 11$.



7 Differential Cryptanalysis

Differential cryptanalysis [1] is a powerful technique. While some advanced variants have been proposed [6, 7, 10], these will not be our concern in this report. In this style of analysis the cryptanalyst attempts to predict (with some probability) the evolution of a difference between two inputs as they pass through the encryption process. The notion of difference can vary depending on the cipher, but it seems that bitwise exclusive-or would be most appropriate for CIPHERUNICORN-E.

The evolution of the difference can be expressed in different ways. It is typical to trace this evolution in an exact manner, defining an input and output for each operation in the encryption process. Under certain assumptions, the probability of this path (which is called a *characteristic*) is estimated by the product of the probabilities at each step in the process. It is typical to assume that a cryptanalyst is trying to identify a 14-round characteristic when attacking a 16-round cipher. Often the two outer rounds of the cipher can be removed in what is frequently called a 2R-attack (we have already observed this in Section 5).

The success of the attack is dependent on the probability of the identified characteristic. Actually, it is more accurate to say that the success of the attack depends on the accumulated probability of all possible characteristics that have the same starting and ending difference. Thus accumulation of all relevant characteristics is typically termed a *differential* [9]. Throughout this section we will switch between characteristics and differentials as the need arises.

In the self-evaluation report [16] the designers provided conservative estimates for the resistance of this cipher to differential cryptanalysis. In this section we will look at their technique, consider our own separate independent approach and provide our conclusions on the resistance of CIPHERUNICORN-E to differential cryptanalysis.

7.1 Differential cryptanalysis of UNI-E-REP-Y

In this section we will consider the round function shown in Figure 2 and used by the designers to evaluate the resistance of CIPHERUNICORN-E to differential cryptanalysis.

In the limited time available it was not possible to identify a better differential path than the one outlined in Figure 3.3 of the designers' self-evaluation report [16]. An upper bound for the probability of the differential is estimated by 2^{-12}. This is likely to be conservative. Following our comments in Section 5, it might be advisable to use this in deriving a bound of $(2^{-12})^6 = 2^{-72}$ for the probability of a useful differential when attacking the cipher rather than the value $(2^{-12})^7 = 2^{-84}$ given in the self-evaluation report [16]. The fraction of the keys for which the bound of 2^{-72} might apply would depend on the Hamming weight of the difference. Taking a conservative approach we might assume the Hamming weight to be 1 for which the characteristic would apply to 2^{-14} of the

keyspace. However it should be noted that such estimates are conservative and overlook a large number of significant issues. It seems unlikely that a characteristic or differential could be identified which would actually hold with such probabilities for any reasonable fraction of the key space.

7.2 Differential cryptanalysis of UNI-E-NO-Y

In this variant of the round function we omit Y but keep the 32-bit integer addition of key material. This alternative simplification is illustrated in Figure 3. When we look at the round function in CIPHERUNICORN-E (see Figure 1) we see that between the first four mini-rounds of Computation I and the subsequent four mini-rounds, two things happen. First some key material is added to the intermediate data, and second, the ordering of the remaining mini-rounds is determined by information derived from Computation II.

One class of orderings is particularly interesting. Suppose that mini-round five is in fact identical to mini-round four. This happens with probability $\frac{1}{4}$. In this case, we potentially have the following characteristic over the first eight mini-rounds of Computation I holding with some probability p,

$$(0,0,0,\Delta) \xrightarrow{1-8} (0,0,0,\Delta).$$

For this to occur we would need the following set of characteristics to hold in mini-rounds four and five. Here δ_0, δ_1, δ_2, and δ_3 are intermediate non-zero differences whose specific values are not important to us since they are entirely internal to the round function. Note that these internal differences might be modified depending on the action of the integer addition modulo 2^{32}.

$$\begin{array}{l}
\Delta \xrightarrow{S_0} \delta_0 \quad \Delta \xrightarrow{S_1} \delta_1 \quad \Delta \xrightarrow{S_2} \delta_2 \quad \Delta \xrightarrow{S_3} \delta_3 \\
(\,\delta_0 \,||\, \delta_1 \,||\, \delta_2 \,||\, \delta_3\,) \xrightarrow{+} (\,\delta'_0 \,||\, \delta'_1 \,||\, \delta'_2 \,||\, \delta'_3\,) \\
\delta'_3 \xrightarrow{S_0} \delta'_0 \quad \delta'_3 \xrightarrow{S_1} \delta'_1 \quad \delta'_3 \xrightarrow{S_2} \delta'_2 \quad \delta'_3 \xrightarrow{S_3} \Delta
\end{array}$$

Let us suppose that the key in the integer addition has the value `0x00000002`, for example. Then there are values for Δ which will provide a differential $(0,0,0,\Delta) \xrightarrow{4,5} (0,0,0,\Delta)$ with non-zero probability. One such value is $\Delta =$ `0x10` and in this case

$$(0,0,0,\texttt{0x10}) \xrightarrow{4,5} (0,0,0,\texttt{0x10})$$

with probability $p = 2^{-7}$. Thus the differential

$$(0,0,0,\texttt{0x10}) \xrightarrow{1-8} (0,0,0,\texttt{0x10})$$

holds with probability 2^{-7} over the first eight mini-rounds of Computation I for one in four of the orderings of mini-rounds five to eight.

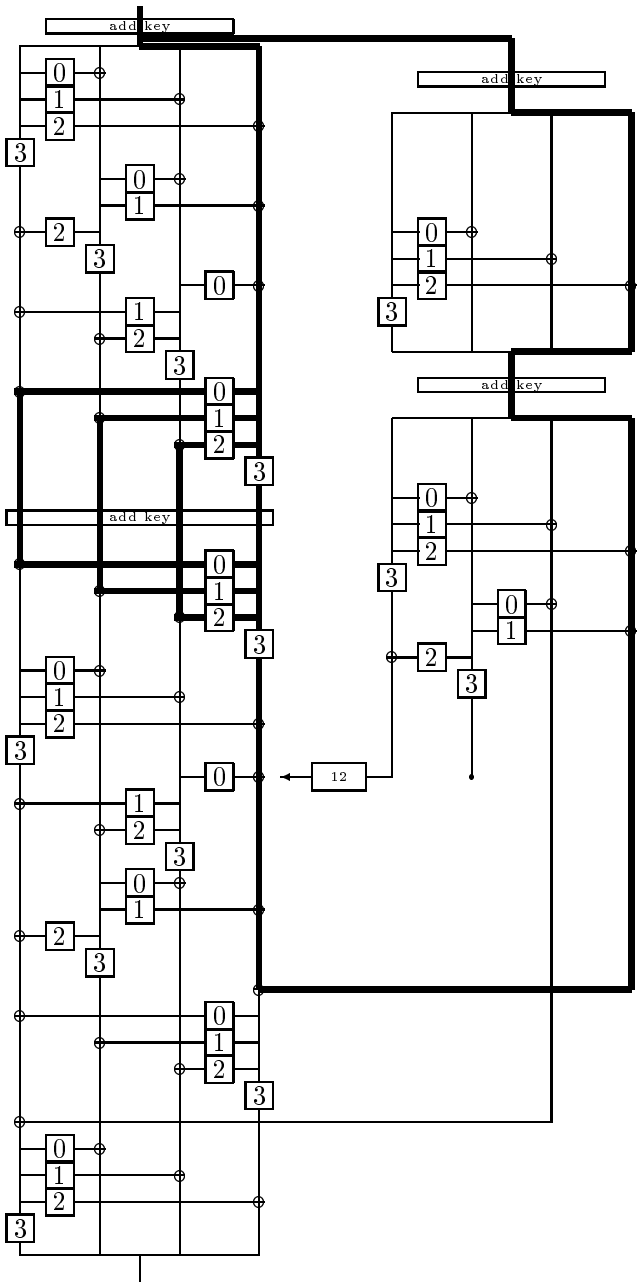
However the key 0x00000002 only occurs with probability 2^{-32}. So it is interesting to consider for what fraction of keys some exploitable effect might be manifested. Experiments suggest that if we consider randomly chosen keys of the form 0x******02, then the probability of the differential $(0,0,0,\texttt{0x10}) \xrightarrow{4,5} (0,0,0,\texttt{0x10})$ is at least 2^{-12} in roughly 10% of cases. In roughly 1% of the cases the probability of the differential is at least 2^{-10}. Of course we have already identified one value for which $p = 2^{-7}$ (namely 0x00000002).

We can now extend this phenomenon to the full, modified round function shown in Figure 3. The path of this differential is illustrated in Figure 4 and there exist values to the additive key material in the round such that the simple differential 0x00000010 $\longrightarrow$ 0x00000000 holds over the entire round function of Uni-E-no-Y with probability $2^{-7} \times 2^{-2} = 2^{-9}$. The additional factor of 2^{-2} is due to the probability of having mini-round five identical to mini-round four. The key values that allow this differential are k = 0x00000002 with all other additive keys in the round set to zero.

Of course, there is considerable key dependence in this probability. Depending on the sophistication of the analysis much of this can be accounted for. However, for our purposes, we will adopt a worst-case analysis and assume that there exists a differential for a round of Uni-E-no-Y that holds with probability 2^{-9} for some portion of the keyspace. Note that this is exactly the style of differential we considered in Section 5 and it would lead to a bound (for some fraction of the key space) on the probability of the differential of around 2^{-54} over fourteen rounds of the cipher.

While it is difficult to gauge the effect of these findings on CipherUnicorn-E, it does suggest that the role of function Y is important. If we now include the function Y that we removed to facilitate this analysis, then the probability of the differential we have identified would likely fall by a factor of 2^{16} per round due to the exclusive-or of the byte material towards the end of Computation I. With the time available for this review, it was not immediately clear how the attacker might best try and control this effect.

Figure 4: Simple differential for the round function in Figure 3.



7.3 Unanticipated effects

Throughout this analysis, experimentation revealed little evidence of any substantially irregular effects when comparing a naive (yet typical) analysis of differential cryptanalysis and its performance in actuality. By this we mean that any experiments provided results that were broadly in line with analytic expectations. As an example, a full implementation of the differential described in Section 7.2 provided good confirmation of the probabilities predicted. Over 2^{24} randomly chosen texts, the differential $\texttt{0x00000010} \longrightarrow \texttt{0x00000000}$ for one round of UNI-E-NO-Y held with probability $\frac{32640}{2^{24}} \approx 2^{-9.0}$ when the keys had the values indicated in Section 7.2.

Due to the magnitude of the probabilities involved, extensive experimentation was out of the question. Further, it is not clear what value other experimentation might have. While the results of Section 8.2 imply that there can be unforeseen interactions within the cipher, limited experimentation here suggests that in the absence of results to the contrary, multiplying the probabilities of identified characteristics and differentials does not immediately seem to be unreasonable.

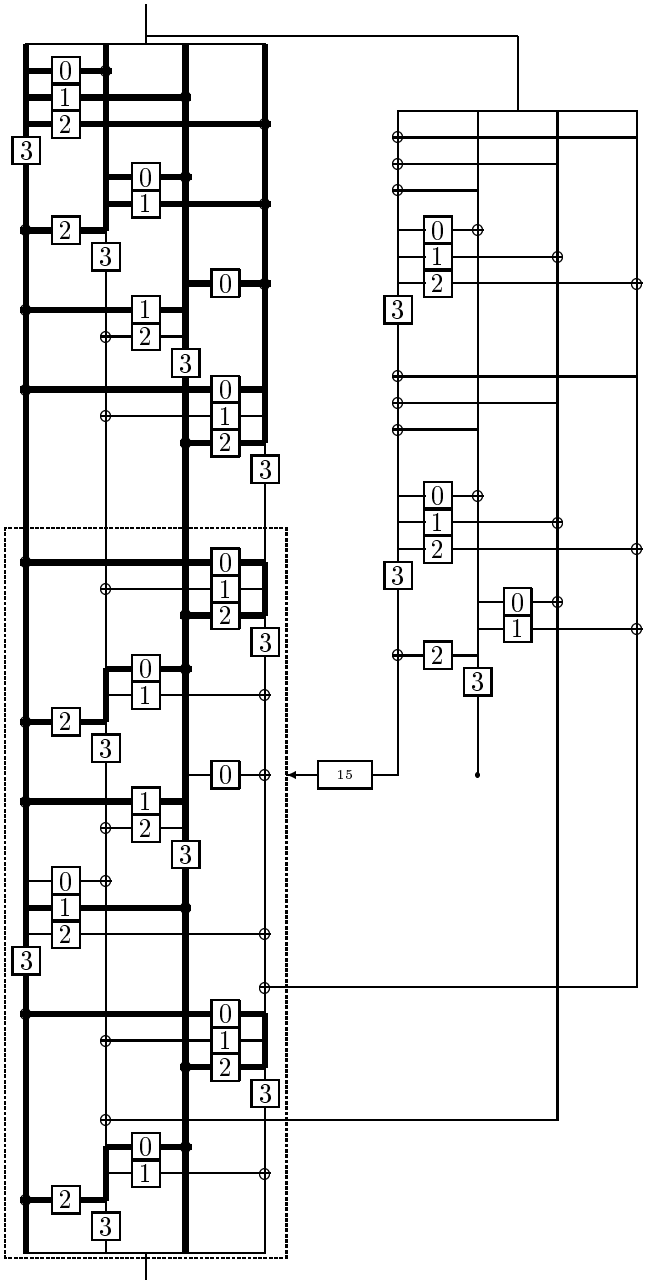
7.4 Implications for the full cipher

Given the complexity of CIPHERUNICORN-E we are left with little alternative but to study much simplified versions of the cipher. Yet if we make too many changes to allow for analysis, then it is hard to assess how close to the true behavior of the cipher the variant remains.

Two simplifications are natural ones to make. The first is to replace the function Y with a function that has similar properties yet is simpler to analyze. The second is to remove the function Y altogether. While some advanced analysis has revealed interesting behavior in the round function of CIPHERUNICORN-E, the upper bound on the probability of a differential across one round of the cipher provided by the designers might still be viewed as reasonable (though less conservative than might previously have been expected).

The function Y appears to be important for the security of the cipher. Without this function, a differential for one round of the modified cipher could be identified that holds with probability 2^{-9} for some fraction of the key space. Nevertheless, for CIPHERUNICORN-E it seems unlikely that a non-trivial differential for a single round could be readily identified holding with a probability much greater than 2^{-12} for even a small proportion of the keyspace. Best current estimates provide an upper bound on the probability of a differential for CIPHERUNICORN-E of 2^{-72}. So with current understanding, it would be reasonable to view CIPHERUNICORN-E as being practically resistant to differential cryptanalysis.

Figure 5: Improved linear characteristic for the round function in Figure 2.



8 Linear Cryptanalysis

While linear cryptanalysis [11] has been very effective in an analysis of DES [12], it is often less effective against other ciphers. There are several enhancements and more advanced considerations when we consider the resistance of a cipher to linear cryptanalysis [5, 8, 17]. However the resistance of a cipher to even the most basic techniques is often such that these enhancements have negligible effect.

In linear cryptanalysis we are concerned with predicting the value of a single bit of information. This bit is typically formed as the exclusive-or combination of different bits in a word. The bits from a word a, say, that contribute to the bit of information are indicated by a $(0,1)$-vector Γ and the value of the bit can be conveniently represented by the familiar dot product $a \cdot \Gamma$. This single bit value will have the values zero and one with a certain probability p. The effectiveness of a linear cryptanalytic attack can be measured in terms of the bias ϵ where $\epsilon = |1/2 - p|$. In the self-evaluation report [16] a measure we will refer to as the correlation coefficient LP is used for an assessment of linear cryptanalysis. The two notions are very closely related, but in this report we will continue to assess linear cryptanalysis using the bias directly.

8.1 Linear cryptanalysis of simplified variants

In this section we will consider the round function shown in Figure 2. This was used by the designers in evaluating the resistance of CIPHERUNICORN-E to linear cryptanalysis. Whether or not we believe this simplified round function to be sufficiently representative of the round function itself, and even though the results of Section 8.2 cast some doubt on the methodology used, there could be a slightly better linear approximation than that identified by the designers.

The designers identify a linear approximation that holds with an estimated correlation coefficient of $\mathrm{LP} = 2^{-63.90}$. Using exactly the same technique we can identify a linear approximation that appears to hold with correlation coefficient $\mathrm{LP} = 2^{-62}$. The active components of this linear approximation are illustrated in Figure 5. If we were to use the same methodology and terminology as was used in the self-evaluation report [16] we might estimate that

$$\begin{aligned}
\mathrm{LP} &= \{\text{input mask } \neq 0 \text{ for } (S_0||S_1||S_2)\}^2 \\
&= \{\text{input mask } \neq 0 \text{ for } (S_0||S_1)\} \\
&= \{\text{input mask } \neq 0 \text{ for } (S_0||S_2)\} \\
&= \{\text{input mask } = 0 \text{ for } (S_0||S_2)\}^4 \\
&= \{\text{input mask } \neq 0 \text{ for } S_1\}^2 \\
&= \{\text{input mask } \neq 0 \text{ for } S_3\}^4 \\
&\approx (2^{-2.6})^2 \times 2^{-3.8} \times 2^{-3.8} \times (2^{-3.66})^4 \times (2^{-6})^6 = 2^{-62}
\end{aligned}$$

Thus it is reasonably straightforward to use the designers' own techniques [16] to find slight improvements and we might be tempted to bound the correlation coefficient LP of a linear approximation to a round of CIPHERUNICORN-E by 2^{-62} instead of $2^{-63.9}$. This has no practical impact on the security of the cipher. Indeed, work by Chabaud and Vaudenay [2] and Selcuk [18] suggests that the low correlation values these per round estimates imply for the cipher as a whole, are not very useful. What is more important is that no avenue for mounting a practical linear cryptanalytic attack is evident from this short review.

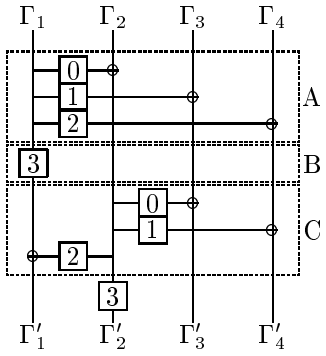
Since the linear approximation in Section 8.1 and Figure 5 did not involve Computation II, then the form of the function $Y(\cdot)$ was immaterial. Thus the linear approximation for UNI-E-REP-Y identified in Section 8.1 is also applicable to UNI-E-NO-Y. Computation I appears to be particularly resistant to linear cryptanalysis. It is not obvious how we might identify an interaction with Computation II that would make a linear cryptanalytic attack significantly easier.

8.2 Unanticipated effects

The cryptanalyst has few tools available in trying to assess the security of a cipher with regards to linear cryptanalysis. The typical approach is to consider the bias and correlations of sub-components in the cipher and then to combine these (using the so-called piling-up lemma [12] or multiplication of correlation coefficients) into an estimate for the magnitude of a bias or correlation for the cipher as a whole.

It is well-known that the piling-up lemma cannot be applied without considerable care. Due to unexpected interactions in the cipher, or unforeseen additional correlations, overall estimates derived in this way can end-up being in some considerable error. To try and gauge whether this might be the case with CIPHERUNICORN-E some limited experiments were completed. These were designed to try and assess whether simply composing the biases or linear correlations would be a reasonable way to estimate the security of a cipher.

It is very difficult to know where to look for such effects. The biases are expected to get small quickly, so if we are to experimentally assess the bias of a linear approximation the components must necessarily be very simple. This in turn provides reduced opportunities for significant dependencies to build. However, consider the very simple network described here.



We can test many different linear approximations across this network. There may well be some significant dependencies between the strands, but it is difficult to identify what they might be. We will illustrate the complications that take place by looking at two sets of simple approximations

$$(\texttt{0x8e}, -, \texttt{0x83}, \texttt{0x1f}) \longrightarrow (\texttt{0xd9}, -, \texttt{0x83}, \texttt{0x1f}) \quad (1)$$

$$(\texttt{0x30}, -, \texttt{0x83}, \texttt{0x1f}) \longrightarrow (\texttt{0xd9}, -, \texttt{0x83}, \texttt{0x1f}) \quad (2)$$

Off-line analysis might have suggested the following constructions for these approximations. Approximation (1) might have be composed as the concatenation of three approximations A, B, and C given by

$$\begin{aligned}
(-, -, \texttt{0x83}, \texttt{0x1f}) &\xrightarrow{S1,S2} (-, -, \texttt{0x83}, \texttt{0x1f}) \\
(\texttt{0x8e}, -, -, -) &\xrightarrow{S3} (\texttt{0xd9}, -, -, -) \\
(\texttt{0xd9}, -, \texttt{0x83}, \texttt{0x1f}) &\xrightarrow{S0,S1,S2} (\texttt{0xd9}, -, \texttt{0x83}, \texttt{0x1f})
\end{aligned}$$

Independently these approximations have biases $\frac{26}{256} \approx 2^{-3.3}$, $\frac{16}{256} = 2^{-4}$, and $\frac{26}{256} \approx 2^{-3.3}$ respectively. Meanwhile Approximation (2) might be composed as

$$\begin{aligned}
(-, -, \texttt{0x83}, \texttt{0x1f}) &\xrightarrow{S1,S2} (-, -, \texttt{0x83}, \texttt{0x1f}) \\
(\texttt{0x30}, -, -, -) &\xrightarrow{S3} (\texttt{0xd9}, -, -, -) \\
(\texttt{0xd9}, -, \texttt{0x83}, \texttt{0x1f}) &\xrightarrow{S0,S1,S2} (\texttt{0xd9}, -, \texttt{0x83}, \texttt{0x1f})
\end{aligned}$$

Independently these approximations have biases $\frac{26}{256} \approx 2^{-3.3}$, $\frac{16}{256} = 2^{-4}$, and $\frac{26}{256} \approx 2^{-3.3}$ respectively.

If we were to use the piling-up lemma, we would predict that the bias of both Approximations (1) and (2) would be

$$2^{-3.3} \times 2^{-4.0} \times 2^{-3.3} \times 2^{2.0} = 2^{-8.6}.$$

Yet when we come to experimentally measure these biases we find that out of 2^{24} random texts Approximation (1) gives a bias of $2^{-8.3}$ whereas there is no detectable bias for Approximation (2).

To see what is happening, we might consider the value of the three constituent approximations simultaneously since they are not independent. In the following table we consider all 2^{16} possible inputs to the two left-most strands (since these are what matter for this approximation) and we count the number of times the constituent approximations A, B, and C take the value zero or one for both Approximations (1) and (2).

Approximation	value of A	value of B	value of C	count
(1)	0	0	0	13398
(1)	0	0	1	8874
(1)	0	1	0	10318
(1)	0	1	1	6834
(1)	1	0	0	8778
(1)	1	0	1	5814
(1)	1	1	0	6930
(1)	1	1	1	4590
(2)	0	0	0	13090
(2)	0	0	1	8670
(2)	0	1	0	10626
(2)	0	1	1	7038
(2)	1	0	0	9086
(2)	1	0	1	6018
(2)	1	1	0	6622
(2)	1	1	1	4386

For both approximations we can look at these counts and observe different features. For instance, to see the bias of constituent approximation B in both cases we can add the counts in those rows for which B takes the value 0. We have that

$$(13398 + 8874 + 8778 + 5814) = (13090 + 8670 + 9086 + 6018) = 36864$$

and approximation B (in isolation) in both cases has bias

$$\frac{(36864 - 32768)}{2^{16}} = 2^{-4}.$$

However consider the bias of Approximations (1) and (2) in their entirety. If we count the number of times that Approximation 1 takes the value 0, this can occur when all constituent approximations A, B, and C take the value 0, or when exactly one of them does. Thus the bias of Approximation 1 is given by

$$\frac{(13398 + 6834 + 5814 + 6930) - 32768}{2^{16}} \approx 2^{-8.3}$$

whereas the bias of Approximation 2 is given by

$$\frac{(13090 + 7038 + 6018 + 6622) - 32768}{2^{16}} = 0.$$

This is a dramatic illustration of the fact that there are considerable dependencies between the different components of the cipher. Using exactly the same techniques, we find that two approximations that we might expect to have the same bias, in fact behave very differently.

There has not been sufficient time in this short review to consider these issues in much greater depth. It is clear however, that the routine use of the piling-up lemma, or the equivalent process of multiplying correlation coefficients, can lead to misleading results in estimating the security of the cipher.

8.3 Implications for the full cipher

Given the exceptional complexity of CIPHERUNICORN-E an accurate assessment of the effectiveness of linear cryptanalysis is not easy. Nevertheless, advanced but limited analysis has revealed the potential for unforeseen effects within the cipher. While it is very unclear what implications these effects might have, it would still be a surprise if a practical linear cryptanalytic attack could be mounted on the cipher. It seems that the complexity of Computation I alone is such that compromising even a limited number of rounds of CIPHERUNICORN-E with linear cryptanalytic techniques seems unlikely. While there might be good grounds to question the typical approach of multiplying the correlations of different components in estimating the correlations over substantial portions of the cipher, the full implications of this cannot be gauged at the moment. So with our current state of knowledge, the absence of practical attacks means that we might still view CIPHERUNICORN-E as being practically resistant to linear cryptanalysis.

9 Conclusions

In this report we have presented the results of a brief cryptographic review of the block cipher CIPHERUNICORN-E. In particular we focused on the applicability of differential and linear cryptanalytic techniques.

Best current estimates provide an upper bound on the probability of a differential for CIPHERUNICORN-E of 2^{-72}. With our current understanding it would be reasonable to view CIPHERUNICORN-E as being practically resistant to differential cryptanalysis.

The function Y appears to provide a tangible contribution to the security of CIPHERUNICORN-E. Without this function, a differential holding with probability 2^{-54} over 14 rounds of the cipher could be identified. For CIPHERUNICORN-E itself, however, it seems unlikely that an active differential for a single round

could be readily identified with a probability much greater than 2^{-12} for even a small proportion of the keyspace.

The function L does not seem to have been fully accounted for in the designers' self-evaluation report. As a result, the bound of 2^{-84} for the probability of an exploitable differential given in the self-evaluation report [16] might be better replaced with the 2^{-72} stated above.

With regards to linear cryptanalysis, the situation is less clear. The complexity of the round function alone is such that compromising even a limited number of rounds of CIPHERUNICORN-E with linear cryptanalytic techniques seems unlikely. However, there might be good grounds to question some of the techniques used in establishing a bound for a linear cryptanalytic attack. Without considerable additional and very detailed analysis, it is impossible to comment further. Nevertheless, no new attacks have been identified. So while the current state of knowledge suggests that the status of CIPHERUNICORN-E is open, on current evidence a practical linear cryptanalytic attack seems unlikely.

This review took place over a limited time and with limited resources. It should be anticipated that additional analysis with increased resources may well find improved results in the cryptanalysis of this cipher and provide a greater understanding of the true security that is offered.

References

[1] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard.* Springer-Verlag, New York, 1993.

[2] F. Chabaud and S. Vaudenay. Links between differential and linear cryptanalysis. In A.D. Santis, editor, *Advances in Cryptology — Eurocrypt '94*, volume 950 of *Lecture Notes in Computer Science*, pages 356-365, 1994. Springer Verlag.

[3] J. Daemen and V. Rijmen. *AES Proposal: Rijndael.* June 11, 1998.

[4] T. Jakobsen and L.R. Knudsen. The interpolation attacks on block ciphers. In E. Biham, editor, *Fast Software Encryption*, volume 1267 of *Lecture Notes in Computer Science*, pages 28–40, 1997. Springer Verlag.

[5] B.S. Kaliski and M.J.B. Robshaw. Linear cryptanalysis using multiple approximations. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, volume 839 of *Lecture Notes in Computer Science*, pages 26–39, New York, 1994. Springer Verlag.

[6] L.R. Knudsen. Applications of higher order differentials and partial differentials. In B. Preneel, editor, *Fast Software Encryption*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211, 1995. Springer Verlag.

[7] L.R. Knudsen and T. Berson. Truncated differentials of SAFER. In D. Gollmann, editor, *Fast Software Encryption*, volume 1039 of *Lecture Notes in Computer Science*, pages 15–25, 1996. Springer Verlag.

[8] L.R. Knudsen and M.J.B. Robshaw. Non-linear approximations in linear cryptanalysis. In U. Maurer, editor, *Advances in Cryptology — Eurocrypt '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 224-236, 1996. Springer Verlag.

[9] X. Lai, J.L. Massey, and S. Murphy. Markov ciphers and differential cryptanalysis. In D.W. Davies, editor, *Advances in Cryptology — Eurocrypt '91*, volume 547 of *Lecture Notes in Computer Science*, pages 17–38, Berlin, 1992. Springer-Verlag.

[10] S.K. Langford and M.E. Hellman. Differential-linear cryptanalysis. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, volume 839 of *Lecture Notes in Computer Science*, pages 17–25, 1994. Springer Verlag.

[11] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology — Eurocrypt '93*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397, 1994. Springer-Verlag.

[12] M. Matsui. The first experimental cryptanalysis of the Data Encryption Standard. In Y.G. Desmedt, editor, *Advances in Cryptology — Crypto '94*, volume 839 of *Lecture Notes in Computer Science*, pages 1–11, New York, 1994. Springer-Verlag.

[13] National Institute of Standards and Technology (NIST). *FIPS Publication 46-2: Data Encryption Standard*, December 30, 1993.

[14] National Institute of Standards and Technology (NIST). *FIPS Publication 197: Advanced Encryption Standard*, November 26, 2001.

[15] NEC Corporation. *Cryptographic techniques specifications: Cipherunicorn-E*, Version 2. Undated.

[16] NEC Corporation. *Self Evaluation Report: Cipherunicorn-E*, Version 3. Undated.

[17] K. Nyberg. Linear approximation of block ciphers. In A.D. Santis, editor, *Advances in Cryptology — Eurocrypt '94*, volume 950 of *Lecture Notes in Computer Science*, pages 439–444, 1994. Springer-Verlag.

[18] A. Selçuk. On bias estimation in linear cryptanalysis. In Proceedings of *Indocrypt 2000*, volume 1977 of *Lecture Notes in Computer Science*, pages 52–66, 2000. Springer-Verlag.