Analysis of Cipherunicorn-A

January 12, 2001

Executive Summary

This report presents the results of a limited time evaluation of the block cipher Cipherunicorn-A.

The round function of Cipherunicorn-A is very complex, which makes it hard to analyse. We have outlined a few surprising properties of the round function. First we discovered a differential which specifies 32 bits in the 64-bit outputs with a probability orders of magnitudes higher than for a random function. Also, it holds with a high probability that up to four rounds of the Feistel network inside the round function can be ignored. Our findings are not sufficient to establish cryptanalytic attacks faster than an exhaustive search for the key. But it is felt that there are potential in the round function of Cipherunicorn-A for cryptanalysis of a substantial number of rounds. However, we believe that with respect to the state-of-the-art a cryptanalytic attack on all 16 rounds of Cipherunicorn-A is likely not to exist or to be of a very high complexity.

Finally we mention that this report is the result of a limited time of review, and the analysis was performed without access to computer code implementing the block cipher. A longer, concentrated analysis might reveal properties of Cipherunicorn-A which we were not able to detect.

Contents

1 Structural features and characteristics

Cipherunicorn-A is an iterated block cipher with 128-bit blocks and allows for three different key sizes to be compliant with the AES [30]. The structure of Cipherunicorn-A is the well-known Feistel network and it runs in 16 rounds. The Feistel round function consists itself of two smaller (64-bit) Feistel networks whose outputs are combined by exclusive-ors to form the output of the round function. The two smaller Feistel networks run in respectively ten and six rounds together with several other components, e.g., multiplications by some fixed constants. We shall call these networks the main stream and the secondary stream, respectively. It is claimed by the designers that each one of these two smaller Feistel networks by themselves provides enough "shuffling" of the data. It is shown in this report that this claim is not fully justified.

The complexity of the Feistel round function is relatively big compared to other known Feistel ciphers. In Cipherunicorn-A there is a total of 16 T-functions, which consists of an evaluation of four S-boxes on the same input byte, six multiplications modulo 2^{32}, four additions modulo 2^{32}, two 32-bit rotations and three other bitwise logical operations. The four S-box evaluations can be performed as one table lookup, if a table of size about 1 Kbytes is precomputed.

An unusual property

Next we report of an unusual property in the both the main stream and the secondary stream of the round function. Consider the four rounds of the main stream starting with a round after the second round containing T_0. Then it holds that the 64-bit texts are unaltered with probability 2^{-16}. Consider Figure 3.4 in [2] and the following figure where an eight-tuple gives the values of the eight bytes in the inputs and outputs to one Feistel round in the round function of Cipherunicorn-A. $\overset{T_k}{\rightarrow}$ specifies which round function is considered and in which direction the output of T_k is computed.

$$\begin{aligned}
(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) &\overset{T_k}{\rightarrow} (x_0, x_1, x_2, x_3, y_4, y_5, y_6, y_7) \\
(z_0, z_1, z_2, z_3, y_4, y_5, y_6, y_7) &\overset{T_\ell}{\leftarrow} (x_0, x_1, x_2, x_3, y_4, y_5, y_6, y_7) \\
(z_0, z_1, z_2, z_3, y_4, y_5, y_6, y_7) &\overset{T_m}{\rightarrow} (z_0, z_1, z_2, z_3, x_4, x_5, x_6, x_7) \\
(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) &\overset{T_n}{\leftarrow} (z_0, z_1, z_2, z_3, x_4, x_5, x_6, x_7).
\end{aligned}$$

As seen the ciphertext after four rounds equals the plaintext. The fact follows from the observations that if in the first and third rounds the inputs to T_k and T_m are equal, and if in the second and fourth rounds the inputs to T_ℓ and T_n are equal, then the quantities output from the round functions in every second round cancel out, and the output of the four rounds equal the input of the four rounds. These two events have a probability of about 2^{-8} each. This is an approximation only, since the events in the four rounds are not independent. Thus, for any of such four rounds, there are many fixed points, approximately 2^{48}. Since the

S-boxes were designed not to have any fixed points, this observation could be in contradiction to the design principles of the designers. Note that for a general 64-bit Feistel cipher the above phenomenon would hold with a probability of only 2^{-64}.

In the secondary stream a similar phenomenon holds for the last three rounds in the six-round Feistel construction. Assume the byte input to T_0 in the third-last round equals the byte input to T_1 in the last round. In this case the outputs of the secondary stream which is exclusive-ored to the outputs of the main stream come from the first three rounds in the secondary stream. Thus with probability 2^{-8} the main output from the secondary stream is computed in only three Feistel rounds using T_0 in every round.

2 Differential cryptanalysis

In this section we evaluate Cipherunicorn-A with respect to differential cryptanalysis. A difference of two bit-strings of equal lengths is defined via the exclusive-or operation. The Feistel round function consists itself of two smaller (64-bit) Feistel networks with respectively ten and six rounds.

In the following we examine the different components of the Feistel round function with respect to differential cryptanalysis.

The addition of subkeys modulo 2^{32} is not a linear operation with respect to the differences defined. However it is well-known that differences of small Hamming weights stay differences of small Hamming weights with a high probability through modular additions [20]. For example, two texts different in one bit only will remain different in only one bit after a modular addition with a probability of 1/2 in general, and with probability 1 if the difference is in the most significant bit. Let us examine this in more detail. Consider two 32-bit words A and B and a subkey K. Integer addition of a constant word K to words A and B which only differ in few bits does not necessarily lead to an increase of bit differences in the sums $A + K$ and $B + K$. This may be illustrated by the following special cases: Suppose the words A and B only differ in the most significant bit. Then it follows that $A + K$ and $B + K$ also differ in only the most significant bit. Suppose next that the words A and B only differ in the i-th bit, $i < 31$. Then it can be shown that with probability $\frac{1}{2}$, $A + K$ and $B + K$ also differ in only the i-th bit. If we use the binary representation of words, i.e., $A = a_{w-1}2^{w-1} + \cdots + a_1 2 + a_0$, and similarly for B and K, the binary representation of the sum $Z = A + K$ may be obtained by the formulae

$$z_j = a_j + k_j + \sigma_{j-1} \text{ and } \sigma_j = a_j k_j + a_j \sigma_{j-1} + k_j \sigma_{j-1}, \tag{1}$$

where σ_{j-1} denotes the carry bit and $\sigma_{-1} = 0$ (cf. [32]). Using these formulae one sees that $A + K$ and $B + K$ with probability $\frac{1}{4}$ differ in exactly two (consecutive) bits. Suppose now the words A and B already differ in exactly two consecutive bits. Then again using the formulae (1) one can see that with probability $\frac{1}{4}$, $A + K$ and $B + K$ differ in exactly one bit and that with probability $\frac{3}{8}$, $A + K$ and $B + K$ differ in exactly two (not necessarily consecutive) bits.

Thus with probability $\frac{5}{8}$ the words $A + K$ and $B + K$ differ again in at most two bits if A and B differ in two consecutive bits. Using the formulae (1) one could discuss relations between integer addition and bit differences in a more general setting. However the above suggests that addition of subkeys can only moderately contribute to an avalanche effect of bit differences. Therefore, let us first replace the modular addition of the keys by an exclusive-or of the keys.

The $A3$ function in the main stream is linear with respect to the exclusive-or operation. Therefore, we shall ignore the $A3$ function in the search for good differentials. For any differential found by ignoring the $A3$ function one can easily compute the differential input difference before the $A3$ function.

We use the following notation

$$(x_0, x_1, x_2, x_3) \xrightarrow{G} (y_0, y_1, y_2, y_3)$$

if texts of differences (x_0, x_1, x_2, x_3) can result texts of differences (y_0, y_1, y_2, y_3) after one application of a function G, where each x_i and y_i are byte values. Let $\otimes$ denote a multiplication of one of the two constants modulo 2^{32} used in Cipherunicorn-A. Then the following differentials hold with probability one.

$$\begin{aligned}
(a, 0, 0, 0) &\xrightarrow{\otimes} (A, 0, 0, 0) \\
(a, b, 0, 0) &\xrightarrow{\otimes} (A, B, 0, 0) \\
(a, b, c, 0) &\xrightarrow{\otimes} (A, B, C, 0).
\end{aligned}$$

Here a, b, c, A, B, and C denote some nonzero values of a byte. In other words, if a pair of texts are equal in the lower s bytes, $1 \leq s \leq 4$, then the texts after a multipication of a constant modulo 2^{32} have the same property. There is an even stronger result, namely if a pair of texts are equal in the lower s bits, then the texts after a multipication of a constant modulo 2^{32} have the same property. In particular, the following characteristic (where h_x is hexadecimal notation)

$$(80_x, 0, 0, 0) \xrightarrow{\otimes} (80_x, 0, 0, 0) \tag{2}$$

holds with probability one. In plain words, two texts different in only the most significant bit are also different in only the most significant bit after the multiplication of an odd constant modulo 2^{32}.

Next let us consider the smaller Feistel rounds inside the Cipherunicorn-A Feistel round function.

Let us first consider the main stream. This is a ten-round Feistel network, consisting of first a modular addition of two subkeys to the round input, then one application of the $A3$ function, then two Feistel rounds using the nonlinear function T_0, then two Feistel rounds using the nonlinear function T_1, then two Feistel rounds using the nonlinear function T_2, then two Feistel rounds using the nonlinear function T_3, and finally two Feistel rounds using the nonlinear function T_k, where k is a two-bit value specified by data in the secondary stream.

By

$$(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) \xrightarrow{T_k} (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)$$

we shall denote one round of the Feistel network (with 64-bit inputs and outputs) in Cipherunicorn-A, where T_k is used as the nonlinear function. Also, when considering two rounds of the Feistel network we shall denote the differentials as follows

$$\begin{aligned}(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) &\stackrel{T_k}{\rightarrow} (y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7)\\ (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) &\stackrel{T_k}{\leftarrow} (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7),\end{aligned}$$

so that the differential is depicted in analogue to the figure of [1].

Also, when considering two consecutive rounds with T_0 as the nonlinear function we shall assume that the multiplications of the constants are included, cf. [1].

Then the following general types of differentials with two rounds are possible

$$\begin{aligned}(a, 0, c, d, e, 0, g, h) &\stackrel{T_1}{\rightarrow} (a, 0, c, d, e, 0, g, h)\\ (a, 0, c, d, e, 0, g, h) &\stackrel{T_1}{\leftarrow} (a, 0, c, d, e, 0, g, h)\end{aligned}$$

$$\begin{aligned}(a, b, 0, d, e, f, 0, h) &\stackrel{T_2}{\rightarrow} (a, b, 0, d, e, f, 0, h)\\ (a, b, 0, d, e, f, 0, h) &\stackrel{T_2}{\leftarrow} (a, b, 0, d, e, f, 0, h)\end{aligned}$$

$$\begin{aligned}(a, b, c, 0, e, f, g, 0) &\stackrel{T_3}{\rightarrow} (a, b, c, 0, e, f, g, 0)\\ (a, b, c, 0, e, f, g, 0) &\stackrel{T_3}{\leftarrow} (a, b, c, 0, e, f, g, 0).\end{aligned}$$

All these two-round differentials have probability one, where a, b, c, d, e, f, g, and h denote arbitrary byte values. Also, the following differentials over six rounds have probability one.

$$\begin{aligned}(a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_1}{\rightarrow} (a, 0, 0, 0, b, 0, 0, 0)\\ (a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_1}{\leftarrow} (a, 0, 0, 0, b, 0, 0, 0)\\ (a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_2}{\rightarrow} (a, 0, 0, 0, b, 0, 0, 0)\\ (a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_2}{\leftarrow} (a, 0, 0, 0, b, 0, 0, 0)\\ (a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_3}{\rightarrow} (a, 0, 0, 0, b, 0, 0, 0)\\ (a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_3}{\leftarrow} (a, 0, 0, 0, b, 0, 0, 0),\end{aligned}$$

where a and b are any byte values. At the end of the main stream in the round function of Cipherunicorn-A two rounds of encryption is performed using T_k in the smaller Feistel rounds, where k is specified by the secondary stream. Clearly, the two-round differential

$$\begin{aligned}(a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_k}{\rightarrow} (a, 0, 0, 0, b, 0, 0, 0)\\ (a, 0, 0, 0, b, 0, 0, 0) &\stackrel{T_k}{\leftarrow} (a, 0, 0, 0, b, 0, 0, 0),\end{aligned}$$

holds with probability $(1/4 * 3/4)^2 \simeq 1/32$. Note that the probability that k has one of the values $1, 2$, or 3 in one round for one text is $3/4$, and the probability that for the other text one gets the same value of k is $1/4$.

The above differentials can be concatenated and the following differential over eight (8) rounds holds with probability $1/32$.

$$\begin{aligned}
(a,0,0,0,b,0,0,0) &\overset{T_1}{\rightarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_1}{\leftarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_2}{\rightarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_2}{\leftarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_3}{\rightarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_3}{\leftarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_k}{\rightarrow} (a,0,0,0,b,0,0,0)\\
(a,0,0,0,b,0,0,0) &\overset{T_k}{\leftarrow} (a,0,0,0,b,0,0,0),
\end{aligned}$$

where a and b are any byte values.

Next we consider the case where T_0 is used as the nonlinear function. Consider the following two-round differential.

$$\begin{aligned}
(80_x,0,0,0,a,b,c,d) &\overset{T_0}{\rightarrow} (80_x,0,0,0,0,0,0,0)\\
(80_x,0,0,0,0,0,0,0) &\overset{T_0}{\leftarrow} (80_x,0,0,0,0,0,0,0)
\end{aligned}$$

of probability 2^{-8}. To see why this differential has probability 2^{-8} note that the inputs to the function T_0 in the first round are of difference 80_x because of (2). For any values of these two 8-bit inputs, the corresponding 32-bit outputs can be determined and therefore also the exclusive-or of the two 32-bit outputs. Thus, for any two 8-bit inputs to T_0 one can compute the values of a, b, c, and d, such that the differential above has probability one in the first round. The differential has probability one in the second round, since there will be equal inputs to the function T_0 because of (2). Clearly, one cannot know with certainty the inputs to T_0 in the first round, since in the Feistel round function the initial operation is the addition of some unknown keys. However, one can guess one of the 8-bit inputs with probability 2^{-8}, the other 8-bit input can then be easily calculated.

By combining the above results we find that the following differential over ten rounds of the main stream holds with probability 2^{-13}.

$$\begin{aligned}
(80_x,0,0,0,a,b,c,d) &\overset{T_0}{\rightarrow} (80_x,0,0,0,0,0,0,0)\\
(80_x,0,0,0,0,0,0,0) &\overset{T_0}{\leftarrow} (80_x,0,0,0,0,0,0,0)\\
(80_x,0,0,0,0,0,0,0) &\overset{T_1}{\rightarrow} (80_x,0,0,0,0,0,0,0)\\
(80_x,0,0,0,0,0,0,0) &\overset{T_1}{\leftarrow} (80_x,0,0,0,0,0,0,0)\\
(80_x,0,0,0,0,0,0,0) &\overset{T_2}{\rightarrow} (80_x,0,0,0,0,0,0,0)
\end{aligned}$$

$$
\begin{aligned}
(80_x, 0, 0, 0, 0, 0, 0, 0) &\overset{T_2}{\leftarrow} (80_x, 0, 0, 0, 0, 0, 0, 0) \\
(80_x, 0, 0, 0, 0, 0, 0, 0) &\overset{T_3}{\rightarrow} (80_x, 0, 0, 0, 0, 0, 0, 0) \\
(80_x, 0, 0, 0, 0, 0, 0, 0) &\overset{T_3}{\leftarrow} (80_x, 0, 0, 0, 0, 0, 0, 0) \\
(80_x, 0, 0, 0, 0, 0, 0, 0) &\overset{T_k}{\rightarrow} (80_x, 0, 0, 0, 0, 0, 0, 0) \\
(80_x, 0, 0, 0, 0, 0, 0, 0) &\overset{T_k}{\leftarrow} (80_x, 0, 0, 0, 0, 0, 0, 0),
\end{aligned}
$$

Clearly, this differential holds also if we incorporate the $A3$ function, since this is linear, cf. the above discussion. The probability of the differential through the complete main stream depends on the Hamming weights of the byte values a, b, c, and d. Thus, according to the above considerations, one could optimise the probability by finding values of a, b, c, of d of minimum Hamming weights. In the best cases each of these are of Hamming weight 1, in which case the probability drops by a factor of 2^{-4}.

Let us next consider the secondary stream. This is a six-round Feistel network, consisting of first a modular addition of two subkeys to the round input, then four Feistel rounds using the nonlinear function T_0, and finally two Feistel rounds using the nonlinear function T_1. The left half of the secondary stream is exclusive-ored to both output halves of the main stream. The right half of the secondary stream is used to specify two indices for a choice of a T_k function in the main stream, thus only four bits of the right half is used. This immediately gives rise to a differential for the whole round function but where the initial key additions are replaced with exclusive-ors. It holds with probability 2^{-13} that two inputs of difference $(80_x, 0, 0, 0, a, b, c, d)$ where a, b, c, d have some precomputed fixed values, result in texts of difference $(e', f, g, h, e, f, g, h)$ where e and e' differ only in the most significant bits. Thus, this still gives 32 bits of information about the output differences at the cost of a probability of only 2^{-13}. This follows from the fact that the same 32 bits of the outputs of the secondary stream are exclusive-ored to both 32-bit halves in the output of the main stream. We did assume that the keys were added bitwise modulo 2 instead of modulo 2^{32} in the main stream. However, as discussed above, if the differential has a low Hamming weight in the input differences this will only decrease the overall probability with a small factor.

Moreover, there are nontrivial cases for the secondary stream. Consider the following six-round differential.

$$
\begin{aligned}
(0, 0, 0, 0, a, b, c, d) &\overset{T_0}{\rightarrow} (0, 0, 0, 0, a, b, c, d) \\
(0, 0, 0, 0, 0, e, f, g) &\overset{T_0}{\leftarrow} (0, 0, 0, 0, 0, e, f, g) \\
(0, 0, 0, 0, 0, e, f, g) &\overset{T_0}{\rightarrow} (0, 0, 0, 0, 0, e, f, g) \\
(l, m, n, p, h, i, j, k) &\overset{T_0}{\leftarrow} (0, 0, 0, 0, h, i, j, k) \\
(l, m, n, p, h, i, j, k) &\overset{T_1}{\rightarrow} (l, m, n, p, q, r, s, t) \\
(0, 0, 0, 0, q, r, s, t) &\overset{T_1}{\leftarrow} (l, m, n, p, q, r, s, t)
\end{aligned}
$$

The probability in the first round is one, since the inputs to T_0 are equal. In the second round after the multiplication with the constant, there is a probability of about 2^{-8} that the inputs to T_0 are equal. Consequently, the probability in the third round is one again. The values in the differential follows now from the fact that if the pair of bytes input to T_0 in the fourth round and the pair of bytes input to T_1 in the sixth round are equal, then differences cancel out and the left halves of the secondary stream are equal. If we let α and β be the two byte values input to T_0 in the fourth round, then the probability that the inputs to T_1 in the sixth round are α, β or β, α is in total 2^{-15}. Thus, in total this six-round differential has a probability of 2^{-23}. Note that the right halves are not specified any further in this differential. Only four bits are used of the right halves, and in this case it can be assumed that these bits are random. This differential could have some good applications in a differential for the whole round function, since the contribution from the secondary stream with respect to differences amount to the four bits used to select two times one of four functions in the main stream.

Finally, one should try to look for differentials which go through both the main stream and the secondary stream with a high probability. Although we have not been able to make much progress on this in the time frame set for this work, it is believed that there exist such differentials of high probabilities. However, it is also believed that the probabilities for such differentials are low enough, such that attacks on 16 rounds of Cipherunicorn-A will be of very high complexity, if possible at all.

3 Linear cryptanalysis

In this section we consider attacks based on linear cryptanalysis. In the following we examine the different components of the Feistel round function with respect to linear cryptanalysis.

First we note that the $A3$ function helps complicate linear attacks, since each output is a sum of three input bits. Thus in linear characteristics considering one or a few bits, $A3$ has the effect of involving additional bits which complicate analysis.

The addition of subkeys modulo 2^{32} is not a linear operation respect to the exclusive-or operation. However it is well-known that there is a linear relation of maximum bias in the least significant bits.

We use the following notation

$$(x_0, x_1, x_2, x_3) \xrightarrow{G} (y_0, y_1, y_2, y_3)$$

to denote that texts with bit masks (x_0, x_1, x_2, x_3) are correlated to outputs of the function G with bit masks (y_0, y_1, y_2, y_3). As before, let $\otimes$ be denote a multiplication of one of the two constants modulo 2^{32} used in Cipherunicorn-A. Then the following linear characteristic holds with probability one.

$$(0, 0, 0, 0x1) \quad \xrightarrow{\otimes} \quad (0, 0, 0, 0x1)$$

Thus, there is a linear approximation of the least significant bits of maximum bias for both modular addition and modular multiplication modulo 2^{32}.

The S-boxes used in the T_k-functions are all constructed from the inverse function over $GF(2^8)$, which is known to be highly nonlinear [27].

The only approach we have been able to find is the traditional one of tracing bits round per round and combining these into relations over several rounds. In the round function of Cipherunicorn-A this approach would require to incorporate at least the output of eight or more S-boxes per round. Since the S-boxes are highly nonlinear this approach is not likely to enable attacks on more than a few rounds of Cipherunicorn-A and far less than the specified 16 rounds.

4 Other cryptanalysis

In this section we consider other attacks. First of all, there are trivial attacks which apply to all block ciphers. An exhaustive key search will take 2^k operations to succeed, where k is the key size. Also, the "matching ciphertext attack" applies in ECB and CBC mode, but requires about $2^{n/2}$ ciphertext blocks to succeed with good probability, where n is the block size. With $n = 128$ as in Cipherunicorn-A, 2^{64} ciphertext blocks are required after which an attacker would be able to deduce information about the plaintext blocks.

Higher order differentials. This attack applies to ciphers which uses nonlinear components of a low algebraic degree. Cipherunicorn-A uses S-boxes of a high nonlinear order in a relatively complex round function, and the probability that a higher order differential attack could be applicable is very small.

The slide attacks, the integral attacks, the non-surjective attacks and the "mod n" attacks do not seem applicable to Cipherunicorn-A .

The interpolation attacks apply to ciphers which use simple mathematical functions only. Cipherunicorn-A uses mathematical functions in the S-boxes, however the affine mappings in the S-boxes together with the $A3$ function seems to have a good effect in thwarting the interpolation attacks.

The key-schedule of Cipherunicorn-A does not seem to allow for related-key attacks. Since the round keys are encrypted in relatively many rounds using both modular multiplications and S-box lookups, it is unlikely that any easily identified weak keys or pairs of related keys exist.

5 Survey of previous results

The only previous results on Cipherunicorn-A that we are aware of are those of the designers themselves [2]. In the self evaluation report the designers spend most of the time discussing statistical tests of the round function of Cipherunicorn-A which were performed. However it is widely believed that such round function tests are not adequate as a tool for measuring the security of the full cipher. Resistance against statistical tests for the full cipher is on the other hand a necessary condition for a secure cipher although not a sufficient

condition. It is relatively easy to construct ciphers which pass these tests but for which there are efficient short-cut attacks. The comparison with the five AES finalist block ciphers is not fair. As an example, it is well-known that the Rijndael round function does not achieve full diffusion in one round, but the construction guarantees that this is achieved after two rounds.

A Block Ciphers in General

In the following we give a compressed overview of the state-of-the-art of block cipher cryptanalysis, and outline the following known attacks.

1. Exhaustive Key Search
2. Matching Ciphertext Attacks
3. Differential Cryptanalysis
4. Truncated Differential Attacks
5. Higher-order Differential Attacks
6. Linear Cryptanalysis
7. Related-key Attacks
8. Non-surjective Attacks
9. Interpolation Attacks
10. Mod-n Attacks
11. Slide Attacks
12. Integral Attacks

A.1 Exhaustive key search

This attack needs only a few known plaintext-ciphertext pairs. An attacker simply tries all keys, one by one, and checks whether the given plaintext encrypts to the given ciphertext. For a block cipher with a k-bit key and n-bit blocks the number of pairs of texts needed to determine the key uniquely is approximately $\lceil k/n \rceil$. Also, if the plaintext space is redundant, e.g., consists of English or Japanese text, the attack will work if only some ciphertext blocks is available. The number of ciphertext blocks needed depends on the redundancy of the language.

A.2 The matching ciphertext attack

The *matching ciphertext attack* is based on the fact that for block ciphers of m bits used in the modes of operations for the DES [29] after the encryption of $2^{m/2}$ blocks, equal ciphertext blocks can be expected and information is leaked about the plaintexts [7, 16, 26].

A.3 Differential cryptanalysis

The most well-known and general method of analysing conventional cryptosystems today is *differential cryptanalysis*, published by Biham and Shamir in 1990. Differential cryptanalysis is universal in the sense that it can be used against any cryptographic mapping which is constructed from iterating a fixed round function. One defines a **difference** between two bit strings, X and X' of equal length as

$$\Delta X = X \otimes (X')^{-1}, \tag{3}$$

where $\otimes$ is the group operation on the group of bit strings used to combine the key with the text input in the round function and where $(X)^{-1}$ is the inverse element of X with respect to $\otimes$. The idea behind this is, that the differences between the texts before and after the key is combined are equal, i.e., the difference is independent of the key. To see this, note that

$$(X \otimes K) \otimes (X' \otimes K)^{-1} = X \otimes K \otimes K^{-1} \otimes X'^{-1} = X \otimes (X')^{-1} = \Delta X.$$

In a differential attack one exploits that for certain input differences the distribution of output differences of the non-linear components is non-uniform.

Definition 1 *An s-round* characteristic *is a series of differences defined as an $s+1$-tuple $\{\alpha_0, \alpha_1, \ldots, \alpha_s\}$, where $\Delta P = \alpha_0$, $\Delta C_i = \alpha_i$ for $1 \leq i \leq s$.*

Here ΔP is the difference in the plaintexts and ΔC_i is the difference in the ciphertexts after i rounds of encryption. Thus, the characteristics are lists of expected differences in the intermediate ciphertexts for an encryption of a pair of plaintexts. In essence one specifies a characteristic for a number of rounds and searches for the correct key in the remaining few rounds. In some attacks it is not necessary to predict the values $\alpha_1, \ldots, \alpha_{s-1}$ in a characteristic. The pair (α_0, α_s) is called a *differential*. The complexity of a differential attack is approximately the inverse of the probability of the characteristic or differential used in the attack.

A.4 Truncated differentials

For some ciphers it is possible and advantageous to predict only the values of parts of the differences after each round of the cipher. The notion of truncated differentials was introduced by Knudsen [18]:

Definition 2 *A differential that predicts only parts of an n-bit value is called a* truncated differential. *More formally, let (a, b) be an i-round differential. If a' is a subsequence of a and b' is a subsequence of b, then (a', b') is called an i-round truncated differential.*

A truncated differential can be seen as a collection of differentials. As an example, consider an n-bit block cipher and the truncated differential (a', b), where a' specifies the least $n' < n$ significant bits of the plaintext difference and b specifies the ciphertext difference of length n. This differential is a collection of all $2^{n-n'}$ differentials (a, b), where a is any value, which truncated to the n' least significant bits is a'.

A.5 Impossible differentials

A special type of differentials are those of probability zero. The attack was first applied to the cipher DEAL [19] and later to Skipjack [4]. The main idea is to specify a differential of probability zero over some number of rounds in the attacked cipher. Then by guessing some keys in the rounds not covered by the differential one can discard a wrong value of the key if it would enable the cipher to take on the differences given in the differential.

A.6 Higher-order differentials

An sth-order differential is defined recursively as a (conventional) differential of the function specifying an $(s-1)$st order differential. In order words, an sth order differential consists of a collection of 2^s texts of certain pairwise, predetermined differences. We refer to [22, 18] for a more precise definition of higher order differentials.

In most cases one considers differences induced by the exclusive-or operation and the field of characteristic 2. The *nonlinear order* of a function $f : GF(2^n) \to GF(2^n)$ is defined as follows. Let the output bits y_j be expressed as multivariate polynomials $q_j(x) \in GF(2)[x_1, \ldots, x_n]$, where $x_1, \ldots, x_n$ are the input bits. The nonlinear order of f is then defined to be the minimum total degree of any linear combination of these polynomials. The higher order differential attacks exploit the following result.

Corollary 1 *Let $f : GF(2^n) \to GF(2^n)$ be a function of nonlinear order d. Then any dth order differential is a constant. Consequently, any $(d+1)$st order differential is zero.*

The boomerang attack [33] can be seen as a special type of a second-order differential attack. This variant applies particularly well to ciphers for which one particular (first-order) differential applies well to one half of the cipher, and where another particular (first-order) differential applies well to the other half of the cipher.

A.7 Linear cryptanalysis

Linear cryptanalysis was proposed by Matsui in 1993 [23]. A preliminary version of the attack on FEAL was described in 1992 [25]. Linear cryptanalysis [23] is a known plaintext attack in which the attacker exploits linear approximations of some bits of the plaintext, some bits of the ciphertext and some bits of the secret key. In the attack on the DES (or on DES-like iterated ciphers) the linear approximations are obtained by combining approximations for each round under the assumption of independent round keys. The attacker hopes in this way to find an expression

$$(P \cdot \alpha) \oplus (C \cdot \beta) = (K \cdot \gamma) \tag{4}$$

which holds with probability $p_L \neq \frac{1}{2}$ over all keys [23], such that $|p_L - \frac{1}{2}|$, called the bias, is maximal. In (4) $P, C, \alpha, \beta, \gamma$ are m-bit strings and '$\cdot$' denotes the dot product. The bit strings α, β, γ are called *masks*.

Definition 3 *An s-round* linear characteristic *is a series of masks defined as an $(s+1)$-tuple $\{\alpha_0, \alpha_1, \ldots, \alpha_s\}$, where α_0 is the mask of the plaintexts and α_i is the mask of the ciphertexts after i rounds of encryption for $1 \leq i \leq s$.*

As for differential cryptanalysis one specifies a linear characteristics for a number of rounds and searches for the keys in the remaining rounds, we refer to [23] for more details. A linear attack needs approximately about b^{-2} known plaintexts to succeed, where b is the bias of the linear characteristic used.

Also, the concepts of linear hulls, the analogue to differentials as opposed to characteristics in differentials cryptanalysis, has been defined in [28].

Finally, in [24] it has been shown that if one defines the quantity $q = (2p-1)^2$ where p is the probability of a linear characteristic or hull, then when combining several linear characteristics one can multiply their q values to get the q-value of the combination. Sometimes the q values are referred to as the "linear probability", which is somewhat misleading, but nevertheless seems to be widely used.

A.8 Mod n cryptanalysis

In [14] a generalisation of the linear attacks is considered. This attack is applicable to ciphers for which some words (in some intermediate ciphertext) are biased modulo n, where n typically is a small integer. It has been shown that ciphers which uses only bitwise rotations and additions modulo 2^{32} are vulnerable to these kinds of attacks.

A.9 Related-key attacks

There are several variants of this attack depending on how powerful the attacker is assumed to be.

1. Attacker gets encryptions under one key.
2. Attacker gets encryptions under several keys.
   (a) Known relation between keys.
   (b) Chosen relation between keys.

Knudsen used the methods of 1 by giving a chosen plaintext attack of the first kind on LOKI'91 [15], reducing an exhaustive key search by almost a factor of four. The concept "related-key attack" was introduced by Biham [3], who also introduced the attack scenarios of 2, where the encryptions under several keys are requested. Knudsen later described a related key attack on SAFER K [17] and Kelsey, Schneier, and Wagner [13] applied the related key attacks to a wide range of block ciphers. It may be argued that the attacks with a chosen relation

between the keys are unrealistic. The attacker need to get encryptions under several keys, in some attacks even with chosen plaintexts. However there exist realistic settings, in which an attacker may succeed to obtain such encryptions. Also, there exists quite efficient methods to preclude the related key attacks [13, 11].

A.10 Interpolation attack

In [12] Jakobsen and Knudsen introduced the interpolation attack on block ciphers. The attack is based on the following well-known formula. Let R be a field. Given $2n$ elements $x_1, \ldots, x_n$, $y_1, \ldots, y_n \in R$, where the x_is are distinct. Define

$$f(x) = \sum_{i=1}^{n} y_i \prod_{1 \le j \le n, j \ne i} \frac{x - x_j}{x_i - x_j}. \tag{5}$$

$f(x)$ is the only polynomial over R of degree at most $n-1$ such that $f(x_i) = y_i$ for $i = 1, \ldots, n$. Equation (5) is known as the *Lagrange interpolation formula* (see e.g.,[6, page 185]). In the *interpolation attack* an attacker constructs polynomials using pairs of plaintexts and ciphertexts. This is particularly easy if the components in the cipher can be expressed as easily described mathematical functions. The idea of the attack is, that if the constructed polynomials have a small degree, only few plaintexts and their corresponding ciphertexts are necessary to solve for the (key-dependent) coefficients of the polynomial, e.g., using Lagrange's interpolation. To recover key bits one expresses the ciphertext before the last round as a polynomial of the plaintext.

A.11 Non-surjective attack

In [31] Rijmen-Preneel-De Win described the non-surjective attack on iterated ciphers. It is applicable to Feistel ciphers where the round function is not surjective and therefore statistical attacks become possible. In a Feistel cipher one can compute the exclusive-or of all outputs of the round functions from the plaintexts and the corresponding ciphertexts. Thus, if the round functions are not surjective this gives information about intermediate values in the encryptions, which can be used to get information about the secret keys.

A.12 Slide attacks

In [5] the "slide attacks" were introduced, based on earlier work in [3, 15]. In particular it was shown that iterated ciphers with identical round functions, that is, equal structures plus equal subkeys in the rounds, are susceptible to slide attacks. Let $F_r \circ F_{r-1} \circ \cdots \circ F_1$ denote an r-round iterated cipher, where all F_is are identical. The attacker tries to find pairs of plaintext P, P^* and their corresponding ciphertexts C, C^*, such that $F_1(P) = P^*$ and $F_r(C) = C^*$. Subsequently, an attacker has twice both the inputs and outputs of one round of the cipher. If the round function is simple enough, this can lead to very

efficient attacks. To find such pairs of texts, one can in the worst case apply the birthday paradox, such that one such pair is expected from a collection of $2^{n/2}$ texts, where n is the block size.

A.13 Integral Attacks

These attacks are sometimes referred to as the "Square attack", since it was first applied to the block cipher Square [9, 8]. The attack on Square slightly modified also applies to the block ciphers Crypton and Rijndael [10].

In [21] these attacks are generalised under the name of "integral cryptanalysis". In differential attacks one considers differences of texts, in integral cryptanalysis one considers sums of texts. In ciphers where all nonlinear functions are bijective, it is sometimes possible to predict a sum of texts, even in the cases where differential attacks are not applicable. The main observations are that in a collection of texts which in a particular word take all values exactly equally many times, the value of the words after a bijective function also take all values exactly equally many times. Also, assume that s words have this property and that in the cipher a linear combination of the s words are computed (with respect to the group operation considered). Then it is possible to determine also the sum of all linear combinations in a collection of texts. This attack is still today the best attack reported on Rijndael which has been the selected for the Advanced Encryption Standard.

References

[1] NEC Corporation. Cryptographic Techniques Specifications: Cipherunicorn-A.

[2] NEC Corporation. Self Evaluation Report: Cipherunicorn-A.

[3] E. Biham. New types of cryptanalytic attacks using related keys. In T. Helleseth, editor, *Advances in Cryptology: EUROCRYPT'93, LNCS 765*, pages 398–409. Springer Verlag, 1993.

[4] E. Biham, A. Biryukov, and A. Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In J. Stern, editor, *Advances in Cryptology: EUROCRYPT'99, LNCS 1592*, pages 12–23. Springer Verlag, 1999.

[5] A. Biryukov and D. Wagner. Slide attacks. In L. R. Knudsen, editor, *Fast Software Encryption, Sixth International Workshop, Rome, Italy, March 1999, LNCS 1636*, pages 245–259. Springer Verlag, 1999.

[6] P.M. Cohn. *Algebra, Volume 1.* John Wiley & Sons, 1982.

[7] D. Coppersmith, D.B. Johnson, and S.M. Matyas. Triple DES cipher block chaining with output feedback masking. Technical Report RC 20591, IBM, October 1996. Presented at the rump session of CRYPTO'96.

[8] J. Daemen, L. Knudsen, and V. Rijmen. Linear frameworks for block ciphers. *Design, Codes, and Cryptography*. To appear.

[9] J. Daemen, L. Knudsen, and V. Rijmen. The block cipher Square. In E. Biham, editor, *Fast Software Encryption, Fourth International Workshop, Haifa, Israel, January 1997, LNCS 1267*, pages 149–165. Springer Verlag, 1997.

[10] J. Daemen and V .Rijmen. AES proposal: Rijndael. Submitted as an AES Candidate Algorithm. Description available from NIST, see `http://www.nist.gov/aes`.

[11] I.B. Damgård and L.R. Knudsen. Two-key triple encryption. *The Journal of Cryptology*, 11(3):209–218, 1998.

[12] T. Jakobsen and L. Knudsen. The interpolation attack on block ciphers. In E. Biham, editor, *Fast Software Encryption, Fourth International Workshop, Haifa, Israel, January 1997, LNCS 1267*, pages 28–40. Springer Verlag, 1997.

[13] J. Kelsey, B. Schneier, and D. Wagner. Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and triple-DES. In Neal Koblitz, editor, *Advances in Cryptology: CRYPTO'96, LNCS 1109*, pages 237–251. Springer Verlag, 1996.

[14] J. Kelsey, B. Schneier, and D. Wagner. Mod n cryptanalysis, with applications against RC5P and M6. In L. Knudsen, editor, *Fast Software Encryption, Sixth International Workshop, Rome, Italy, March 1999, LNCS 1636*, pages 139–155. Springer Verlag, 1999.

[15] L.R. Knudsen. Cryptanalysis of LOKI'91. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology, AusCrypt 92, LNCS 718*, pages 196–208. Springer Verlag, 1993.

[16] L.R. Knudsen. *Block Ciphers – Analysis, Design and Applications*. PhD thesis, Aarhus University, Denmark, 1994.

[17] L.R. Knudsen. A key-schedule weakness in SAFER K-64. In Don Coppersmith, editor, *Advances in Cryptology - CRYPTO'95, LNCS 963*, pages 274–286. Springer Verlag, 1995.

[18] L.R. Knudsen. Truncated and higher order differentials. In B. Preneel, editor, *Fast Software Encryption - Second International Workshop, Leuven, Belgium, LNCS 1008*, pages 196–211. Springer Verlag, 1995.

[19] L.R. Knudsen. DEAL - a 128-bit block cipher. Technical Report 151, Department of Informatics,University of Bergen, Norway, February 1998. Submitted as an AES candidate by Richard Outerbridge.

[20] L.R. Knudsen and W. Meier. Improved differential attack on RC5. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO'96, LNCS 1109*, pages 216–228. Springer Verlag, 1996.

[21] L.R. Knudsen and D. Wagner. Integral cryptanalysis. In preparation, 2001.

[22] X. Lai. Higher order derivatives and differential cryptanalysis. In R. Blahut, editor, *Communication and Cryptography, Two Sides of One Tapestry*. Kluwer Academic Publishers, 1994. ISBN 0-7923-9469-0.

[23] M. Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93, LNCS 765*, pages 386–397. Springer Verlag, 1993.

[24] M. Matsui. New structure of block ciphers with provable security against differential and linear cryptanalysis. In D. Gollman, editor, *Fast Software Encryption, Third International Workshop, Cambridge, UK, February 1996, LNCS 1039*, pages 205–218. Springer Verlag, 1996.

[25] M. Matsui and A. Yamagishi. A new method for known plaintext attack of FEAL cipher. In R. Rueppel, editor, *Advances in Cryptology - EUROCRYPT'92, LNCS 658*, pages 81–91. Springer Verlag, 1992.

[26] U.M. Maurer. New approaches to the design of self-synchronizing stream ciphers. In D.W. Davies, editor, *Advances in Cryptology - EUROCRYPT'91, LNCS 547*, pages 458–471. Springer Verlag, 1991.

[27] K. Nyberg. Differentially uniform mappings for cryptography. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93, LNCS 765*, pages 55–64. Springer Verlag, 1993.

[28] K. Nyberg. Linear approximations of block ciphers. In A. De Santis, editor, *Advances in Cryptology - EUROCRYPT'94, LNCS 950*, pages 439–444. Springer Verlag, 1995.

[29] National Bureau of Standards. DES modes of operation. Federal Information Processing Standard (FIPS), Publication 81, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., December 1980.

[30] National Institute of Standards and Technology. Advanced encryption algorithm (AES) development effort. `http://www.nist.gov/aes`.

[31] V. Rijmen, B. Preneel, and E. De Win. On weaknesses of non-surjective round functions. *Designs, Codes, and Cryptography*, 12(3):253–266, 1997.

[32] R.A. Rueppel. *Analysis and Design of Stream Ciphers*. Springer Verlag, 1986.

[33] D. Wagner. The boomerang attack. In L. R. Knudsen, editor, *Fast Software Encryption, Sixth International Workshop, Rome, Italy, March 1999, LNCS 1636*, pages 156–170. Springer Verlag, 1999.