Analysis of Camellia

January 12, 2001

Executive Summary

This report presents the results of a limited evaluation of the block cipher Camellia. We have found no important flaws nor weaknesses in Camellia.

Camellia is an iterated cipher which runs in at least 18 rounds. The cipher has a simple and conservative design, which facilitates an easy analysis. It is relatively easy to be convinced about the resistance of Camellia with respect to differential and linear cryptanalysis.

It is further believed that any practical attacks against Camellia is not possible with respect to the state of the art, and it would require a major breakthrough in the area of cryptanalysis of encryption systems.

Finally we mention that a concentrated, longer analysis might reveal properties which we did not detect in the limited-time review. Also, the analysis was performed without access to computer code implementing the block cipher.

Contents

1 Structural features and characteristics

Camellia is an iterated block cipher with 128-bit blocks and allows for three different key sizes to be compliant with the AES [29].

The structure is the classical Feistel network, but where special operations are inserted after each six Feistel rounds. These operations seems to have a good effect in complicating certain attacks, but also destroy the classical Feistel structure, from which a cipher can benefit. However, since these operations are inserted only after every six rounds and since Camellia runs in at least 18 rounds (and at most 24 rounds), the negative effects of the special operations may be hard to spot. The Feistel round function is reminiscent of, but simpler than, the round function in the block cipher E2 [31]. The eight input bytes are first exclusive-ored with some key bytes, and then input to one of in total four S-boxes. Before output the bytes are mixed such that each output byte depends on at least five and at most six input bytes. This allows for certain structures to pass one round of Camellia with probability one, however when iterated it seems that such (dis)advantages disappear. The S-boxes are all constructed from the inverse function over a Galois field, which is known to have excellent properties for increasing resistance against differential and linear cryptanalysis.

2 Differential and linear cryptanalysis

In this section we evaluate Camellia with respect to differential and linear cryptanalysis. A difference of two bit-strings of equal lengths is defined via the exclusive-or operation. The maximum probability of a differential through one S-box is 2^{-6}. The maximum bias of a linear approximation through one S-box is 2^{-4}, which gives a "linear probability", see Appendix, of 2^{-6}. The question is how we can use these in combination to obtain differentials and linear approximations for several or many rounds. One possible tool to make such multi-round structures is to estimate or measure the total number of *active S-boxes* in an analysis. For a few rounds of Camellia it is easy to find the minimum number of active S-boxes. For differential attacks with equal inputs to the round function in one round the number of active S-boxes is zero, and similarly for linear approximations the minimum number of active S-boxes is zero. However, when moving to more rounds, the number of active S-boxes will increase. It follows from a simple inspection of the structure in the round function of Camellia that for two rounds of Camellia the minimum total number of active S-boxes will be one, for three rounds the minimum total number of active S-boxes will be six. This number will increase rapidly for more than 3 rounds. A conservative estimate for both differential and linear attacks would be to expect at least 3 active S-boxes in each round on the average. This would mean a probability of 2^{-18} per round for differential attacks and a bias of 2^{-10} per round for linear attack. Iterated to six rounds this gives a probability of 2^{-108} for differential attacks and a bias of 2^{-55} for linear attacks. All in all, in an attack this would mean a requirement of 2^{108} pairs of chosen plaintexts for the differential attack, and 2^{110} known plaintexts for the linear attack. These estimates are very conservative. First of all, in the estimates we have used the maximum probabilities and biases for every active S-box. Furthermore we have estimated a maximum number of three active S-boxes for every round, which may be quite optimistic. For both attacks, there is a sufficient level of security for all versions of Camellia.

3 Truncated Differentials

In this section we report on several truncated differentials for Camellia. We use the following notation. Let $(x_1 \cdots x_s)$ denote a vector of s bytes. With $s = 16$ this will be a plaintext block or a ciphertext block after i rounds of encryption. With $s = 8$ this will typically be the input or the output of the function F. The *difference* between two s-byte vectors, $s \in \{8, 16\}$ will be the exclusive-or of the individual bytes in the vectors. Also, we let

$$x = (x_1 \cdots x_8) \xrightarrow{F} (y_1 \cdots y_8) = y,$$

denote that a difference of x in two input vectors to F can result in a difference of y in the two output vectors with probability p. A one-round differential will be denoted

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	$\xrightarrow{F}$	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8		z_9	z_{10}	z_{11}	z_{12}	z_{13}	z_{14}	z_{15}	z_{16}

Here $x_1, \ldots, x_{16}$ denote the difference in the two 128-bit texts before the round, and $z_1, \ldots, z_{16}$ denote the difference in the texts after the round. Also, $z_{j+8} = x_j$ for $j = 1, \ldots, 8$, and $z_i = x_{i+8} \oplus y_i$ for $i = 1, \ldots, 8$. Note that the halves after one application of F are swapped.

One of the best ways to push information about differences through several rounds in an iterated cipher is to use what is called *iterative* differentials. These are differentials which can be concatenated with themselves any number of times. For Camellia the following 1-round differential of probability 2^{-16} exists, called Ω_1.

x_1	0	x_3	0	x_5	0	x_7	0		x_9	0	x_{11}	0	x_{13}	0	x_{15}	0
x_1	0	x_3	0	x_5	0	x_7	0	$\xrightarrow{F}$	y_1	0	y_3	0	y_3	0	y_1	0
z_1	0	z_3	0	z_5	0	z_7	0		z_9	0	z_{11}	0	z_{13}	0	z_{15}	0

where it is assumed that $x_i \neq 0$ for $i \in \{1, 3, 5, 7\}$. The differential is iterative, which means that it can be concatenated with itself r times, yielding an r-round differential of probability 2^{-16r}. The probability is calculated as follows. A difference $x_5 \neq 0$ in two input bytes to the S-box 4 results in some difference y_3, and a difference $x_7 \neq 0$ in two input bytes to the S-box 2 results in some difference y_1. Then with an average probability of 2^{-8} a difference $x_1 \neq 0$ in two input bytes to the S-box 1 results in the difference $y_1 \oplus y_3$. Similarly, with an average probability of 2^{-8} a difference $x_3 \neq 0$ in two input bytes to the S-box 3 results in the difference $y_1 \oplus y_3$. Both these events will happen with probability 2^{-16}. The mixing of bytes at the end of the round function results in the difference indicated above. Note that in this report we use a top-down numbering of the bytes in the round function of one Feistel-round, opposite to the designers' numbering from [1], e.g., in Figure 5.

In [24] similar differentials are exploited in cryptanalytic attacks on the block cipher E2. As noted in [24] the above individual probabilities are in fact $\frac{1}{255}$ rather than 2^{-8}, however since we are going to iterate this differential there will be a dependency between the different rounds and the exact probability will be hard to calculate. However, it seems plausible in the analysis of Camellia to assume independence between the rounds, just as the authors assumed in the analysis of E2 [24]. Therefore, for convenience, we shall use 2^{-8} as an approximation of the individual probabilities.

The differential Ω_1 iterated to five rounds looks as follows, where we specify the difference in all sixteen bytes in the input to and in the output of the five rounds, but only the combinations through the function F in every intermediate round.

x_1	0	x_3	0	x_5	0	x_7	0		x_9	0	x_{11}	0	x_{13}	0	x_{15}	0
a_1	0	a_3	0	a_5	0	a_7	0	$\xrightarrow{F}$	A_1	0	A_3	0	A_3	0	A_1	0
b_1	0	b_3	0	b_5	0	b_7	0	$\xrightarrow{F}$	B_1	0	B_3	0	B_3	0	B_1	0
c_1	0	c_3	0	c_5	0	c_7	0	$\xrightarrow{F}$	C_1	0	C_3	0	C_3	0	C_1	0
d_1	0	d_3	0	d_5	0	d_7	0	$\xrightarrow{F}$	D_1	0	D_3	0	D_3	0	D_1	0
e_1	0	e_3	0	e_5	0	e_7	0	$\xrightarrow{F}$	E_1	0	E_3	0	E_3	0	E_1	0
y_1	0	y_3	0	y_5	0	y_7	0		y_9	0	y_{11}	0	y_{13}	0	y_{15}	0

Note that $a_i = x_i$ for $i \in \{1, 3, 5, 7\}$. Here it is assumed that $a_i \neq 0$, $b_i \neq 0$, $c_i \neq 0$, $d_i \neq 0$, $e_i \neq 0$, for $i \in \{1, 3, 5, 7\}$. In a chosen plaintext attack one can choose the plaintexts such that $a_i = x_i \neq 0$ for $i \in \{1, 3, 5, 7\}$. The probability that the difference in the four input words is nonzero in one round is $(\frac{255}{256})^4 \approx 0.984$. The total probability is therefore $(0.984)^4 2^{-80} \approx 2^{-80}$. In the following we shall ignore the factors 0.984.

Note that the exclusive-or of the difference in the plaintext vectors and the difference in the ciphertext vectors (after five rounds) has the following form.

$$z_1 \; 0 \; z_3 \; 0 \; z_3 \; 0 \; z_1 \; 0 \quad z_9 \; 0 \; z_{11} \; 0 \; z_{11} \; 0 \; z_9 \; 0 \tag{1}$$

This follows from the observation that the exclusive-or of the right halves of the plaintext and the left halves of the ciphertext equals the exclusive-or of the outputs of the F-function in the fourth and second rounds. And similarly, the exclusive-or of the left halves of the plaintext and the right halves of the ciphertext equals the exclusive-or of the outputs of the F-function in the fifth, third and first rounds. For a randomly chosen permutation the exclusive-or of the pairs of plaintexts and pairs of ciphertexts will have the form of (1) with a probability of 2^{-96}. For five rounds of Camellia this happens with a probability of at least 2^{-80}, since if the texts follow the differential we will have the form (1).

Also, the following three-round iterative differential, called Ω_2, is possible, but less useful as we shall see.

x_1	0	x_3	0	x_3	0	x_1	0		x_9	0	x_{11}	0	x_{11}	0	x_9	0
a_1	0	a_3	0	a_3	0	a_1	0	$\xrightarrow{F}$	A_1	0	A_3	0	A_3	0	A_1	0
b_1	0	b_3	0	b_3	0	b_1	0	$\xrightarrow{F}$	B_1	0	B_3	0	B_3	0	B_1	0
0	0	0	0	0	0	0	0	$\xrightarrow{F}$	0	0	0	0	0	0	0	0
y_1	0	y_3	0	y_3	0	y_1	0		0	0	0	0	0	0	0	0

Here the differences in the inputs are equal in the third and fifth bytes, in the first and seventh bytes, in the ninth and fifteenth bytes, and in the eleventh and thirteenth bytes. The first round of the differential is similar to the rounds of the previously shown differentials and has a probability of approximately 2^{-16}. The second round is also similar to before, but the important difference to the first-round differential is that the outputs when exclusive-ored to the outputs of the first round, will cancel the differences, and consequently there will be equal inputs to the F function in the third round. The probability in the second round is 2^{-16} as before plus an additional factor of 2^{-16} to make the inputs to the third round equal. Thus, the second round has a probability of approximately 2^{-32}, and in total the three round differential has a probability of 2^{-48}.

In the following we shall apply these differentials in distinguishing attacks on Camellia.

3.1 Distinguishing attacks

The differential consisting of iterations of Ω_1 can be enhanced by letting the first round be a trivial round with probability one.

0	0	0	0	0	0	0	0		x_9	0	x_{11}	0	x_{13}	0	x_{15}	0
0	0	0	0	0	0	0	0	$\xrightarrow{F}$	0	0	0	0	0	0	0	0
b_1	0	b_3	0	b_5	0	b_7	0	$\xrightarrow{F}$	B_1	0	B_3	0	B_3	0	B_1	0
c_1	0	c_3	0	c_3	0	c_1	0	$\xrightarrow{F}$	C_1	0	C_3	0	C_3	0	C_1	0
d_1	0	d_3	0	d_5	0	d_7	0	$\xrightarrow{F}$	D_1	0	D_3	0	D_3	0	D_1	0
e_1	0	e_3	0	e_3	0	e_1	0	$\xrightarrow{F}$	E_1	0	E_3	0	E_3	0	E_1	0
y_1	0	y_3	0	y_5	0	y_7	0		y_9	0	y_{11}	0	y_{11}	0	y_9	0

The probability of this differential is 2^{-64}. Note that in the right halves of the ciphertext differences, the ninth and fifteenth bytes and the thirteenth and fifteenth bytes are equal.

There are four nonzero bytes in the plaintext difference. From one *structure* of 2^{32} plaintexts different in these four bytes, one can form about 2^{63} pairs of plaintexts with the desired difference. There are twelve zero bytes in the input difference, so in theory it is possible to form 2^{96} such structures. For

Camellia reduced to 5 rounds, with four structures one would get 2^{65} pairs of plaintexts, and consequently one would get approximately two ciphertext pairs for which (1) is satisfied. For a randomly chosen permutation one would get approximately 2^{-31} ciphertext pairs with the desired relationship between the plaintexts and ciphertexts differences. Thus with a high probability one can distinguish Camellia with five rounds from a randomly chosen permutation with 2^{34} chosen plaintexts.

Fact 1 *The first five rounds of Camellia can be distinguished from a randomly chosen permutation using about 2^{34} chosen plaintexts.*

Note that it is possible to use also the differential Ω_2. Let the first round be a trivial round where equal inputs lead to equal outputs in the F-function. Let the next three rounds be as specified in Ω_2, and let the fifth round be as in Ω_1. This differential also has a probability of 2^{-64}. However, since the input difference to this differential will require a special form of the nonzero bytes in the input difference, a structure similar to the above with 2^{32} plaintexts will contain less good pairs to be used in the analysis and in total more plaintexts would be required. This seems to be the case also in the examples to follow, so we shall concentrate on differentials constructed from Ω_1.

The above attack on five rounds can be adapted to an attack on six rounds of Camellia. The differential can be extended to six rounds by adding one round of probability 2^{-16}.

$$\begin{array}{cccccccccccccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & x_9 & 0 & x_{11} & 0 & x_{13} & 0 & x_{15} & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \xrightarrow{F} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
a_1 & 0 & a_3 & 0 & a_5 & 0 & a_7 & 0 & \xrightarrow{F} & A_1 & 0 & A_3 & 0 & A_3 & 0 & A_1 & 0 \\
b_1 & 0 & b_3 & 0 & b_5 & 0 & b_7 & 0 & \xrightarrow{F} & B_1 & 0 & B_3 & 0 & B_3 & 0 & B_1 & 0 \\
c_1 & 0 & c_3 & 0 & c_5 & 0 & c_7 & 0 & \xrightarrow{F} & C_1 & 0 & C_3 & 0 & C_3 & 0 & C_1 & 0 \\
d_1 & 0 & d_3 & 0 & d_5 & 0 & d_7 & 0 & \xrightarrow{F} & D_1 & 0 & D_3 & 0 & D_3 & 0 & D_1 & 0 \\
e_1 & 0 & e_3 & 0 & e_5 & 0 & e_7 & 0 & \xrightarrow{F} & E_1 & 0 & E_3 & 0 & E_3 & 0 & E_1 & 0 \\
\hline
y_1 & 0 & y_3 & 0 & y_3 & 0 & y_1 & 0 & & y_9 & 0 & y_{11} & 0 & y_{13} & 0 & y_{15} & 0
\end{array}$$

This differential has a probability of approximately 2^{-80}. For a pair of plaintexts following the differential one will have the form of (1) for the exclusive-or of the ciphertext and plaintext differences. To use this differential to distinguish Camellia from a randomly chosen permutation, one has to generate about 2^{81} pairs of plaintexts with the desired difference. So to generate 2^{81} pairs one would need about 2^{18} structures, totally about 2^{50} chosen plaintexts. The distinguishing technique is the same as above for five rounds.

Fact 2 *The first six rounds of Camellia can be distinguished from a randomly chosen permutation using about 2^{50} chosen plaintexts.*

If we ignore the functions FL and FL^{-1} we can extend the attack to seven rounds of Camellia. First we extend the above six-round differential by adding another 1-round differential of probability 2^{-16}.

0	0	0	0	0	0	0	0		x_9	0	x_{11}	0	x_{13}	0	x_{15}	0
0	0	0	0	0	0	0	0	$\xrightarrow{F}$	0	0	0	0	0	0	0	0
a_1	0	a_3	0	a_5	0	a_7	0	$\xrightarrow{F}$	A_1	0	A_3	0	A_3	0	A_1	0
b_1	0	b_3	0	b_5	0	b_7	0	$\xrightarrow{F}$	B_1	0	B_3	0	B_3	0	B_1	0
c_1	0	c_3	0	c_5	0	c_7	0	$\xrightarrow{F}$	C_1	0	C_3	0	C_3	0	C_1	0
d_1	0	d_3	0	d_5	0	d_7	0	$\xrightarrow{F}$	D_1	0	D_3	0	D_3	0	D_1	0
e_1	0	e_3	0	e_5	0	e_7	0	$\xrightarrow{F}$	E_1	0	E_3	0	E_3	0	E_1	0
f_1	0	f_3	0	f_5	0	f_7	0	$\xrightarrow{F}$	F_1	0	F_3	0	F_3	0	F_1	0
y_1	0	y_3	0	y_5	0	y_7	0		y_9	0	y_{11}	0	y_{11}	0	y_9	0

This differential has a probability of approximately 2^{-96}. As before, we will generate many plaintext pairs and count the number of pairs for which (1) hold. However, note that the probability of the differential is the same as the probability that for a randomly chosen permutation (1) will hold. For Camellia with seven rounds, ignoring the functions FL and FL^{-1}, (1) will hold for all pairs which follow the differential. Clearly for all pairs the first round of the differential will hold. A pair of plaintexts will not follow the second round with probability $1 - 2^{-16}$. If we assume, which seems plausible, that the ciphertext differences in these cases look random, then (1) will hold also in these cases with a probability of 2^{-96}. All together, the probability for this seven-round version of Camellia that the plaintext and ciphertext difference will satisfy (1) is approximately $2^{-96} + (1 - 2^{-16})2^{-96} \simeq 2^{-95}$. (Actually, this probability also covers the cases where the pair of texts follow the differential in the first round, but not the second, and the cases where the pair of texts follow the differential in the first two rounds, but not in the third, and so on. However, the sum of the probabilities of all these cases are small compared to the above probabilities.) To distinguish this version of Camellia from a randomly chosen permutation we pick 2^{36} structures (as above) which gives us 2^{99} pairs of plaintexts with the desired differences. For the Camellia-version this yields about 16 pairs of texts for which (1) holds, while the expected number for a randomly chosen permutation is 8. Since the standard deviation in the first case is 4 and around 3 in the last case, with a high probability we will be able to distinguish between the two cases. Totally this attack needs 2^{68} chosen plaintexts and we have the following result.

Fact 3 *The first seven rounds of Camellia without the functions FL and FL^{-1} can be distinguished from a random permutation using about 2^{68} chosen plaintexts.*

We could try and extend this one round further by adding another round of probability 2^{-16} to the above differential. The probability will be 2^{-112}. However, in this case the difference in probabilities of obtaining texts on the form of (1) will be very small. For the Camellia-version the total probability of getting such pairs will be approximately $2^{-112} + (1 - 2^{-16})2^{-96} = 2^{-96}$ which is what one can expect also for a randomly chosen permutation.

It does not seem possible to extend the approach of distinguishing Camellia from a randomly chosen permutation further than to versions reduced to 7 rounds. However, there exist differentials for Camellia for even more number of rounds.

3.2 The existence of differentials

Consider the above differentials for five, six and seven rounds. There are $2^{16} - 2^8$ pairs of bytes with a nonzero exclusive-or difference, and 2^8 pairs of bytes with zero exclusive-or. The differences x_9, x_{11}, x_{13}, and x_{15} can take any value, thus totally 2^{32} values and consequently there are approximately 2^{63} possible pairs for all four bytes together. Thus, there are approximately $2^{63} \times (2^8)^{12} = 2^{159}$ pairs of plaintexts with the desired difference. If we ignore the keyed functions FL and FL^{-1} in Camellia we can iterate the above differential to any number of rounds, with a decrease in probability of a factor of 2^{-16} for every additional round. Iterated to eleven rounds the probability is approximately $(2^{-16})^{10} = 2^{-160}$. Therefore, for this version of Camellia for about one in every two keys, one can expect to find one pair of plaintexts, which will follow the expected (zero) values specified in the differential in every round.

However, here we assumed that the functions FL and FL^{-1} were not used. Let us analyse these functions with respect to the differences used above. If two 64-bit texts have a difference of $(a\ 0\ b\ 0\ b\ 0\ a\ 0)$ in the input to FL (or FL^{-1}) then the differences in the outputs will be of the form $(f\ 0\ e\ 0\ c\ 0\ d\ 0)$ with some high probability. However, the values of f and e will not be equal nor will the values of c and d. To see why this is the case, let us take a closer look at FL. The first operation is to bitwise 'AND' the left half of the input with a subkey, rotate the result by one position to the left, and exclusive-or this result to the right half of the input. Thus, if the left halves of two inputs has a difference $(a\ 0\ b\ 0)$, the difference after the 'AND' of a key will be $(a'\ 0\ b'\ 0)$, where a' and b' have Hamming weights at most those of a and b respectively. If the most significant bits of the leftmost bytes of the inputs (the bytes with difference a) are equal, they will be equal after the 'AND' operation; if the bits are different they will be equal after the 'AND' operation with a probability of 1/2, where the probability is taken over all keys. Similar observations for the most significant bits of the third leftmost bytes (the bytes with difference b). If the involved bits are equal, the one-bit rotation leaves the difference in the second and fourth bytes zero. Note that in that case the right halves of the output of FL will have the form $(c\ 0\ d\ 0)$. The 'OR' operation in the second "round" of FL means that the left halves of the outputs of FL will have the form $(f\ 0\ e\ 0)$. In total the differential

$$(a\ 0\ b\ 0\ b\ 0\ a\ 0) \overset{FL}{\to} (f\ 0\ e\ 0\ c\ 0\ d\ 0)$$

has probability at least 1/4 for all keys, probability at least 1/2 for three of four keys, and probability 1 for one in four keys, where the probability is taken over all input texts. A similar observation can be made about the function FL^{-1}.

In an attempt to specify differentials for more rounds of Camellia, one could allow both the left and the right halves of the plaintexts to take any values, and modify the differential accordingly. The differential would have the following form.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	$\xrightarrow{F}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8
b_1	0	b_3	0	b_5	0	b_7	0	$\xrightarrow{F}$	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8
c_1	0	c_3	0	c_5	0	c_7	0	$\xrightarrow{F}$	C_1	0	C_3	0	C_3	0	C_1	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
m_1	0	m_3	0	m_5	0	m_7	0	$\xrightarrow{F}$	M_1	0	M_3	0	M_3	0	M_1	0
y_1	0	y_3	0	y_5	0	y_7	0		y_9	0	y_{11}	0	y_{13}	0	y_{15}	0

Here it is assumed that $x_{i+8} = A_i$ for $i \in \{2, 4, 6, 8\}$ and that $a_i = B_i$ for $i \in \{2, 4, 6, 8\}$. The probability for r rounds, $r > 2$ is $2^{-16r-32}$ with an additional decrease in probability of a factor of 2^{-4} for every layer of FL and FL^{-1}. Such a differential would have probabilities 2^{-32} in the first two rounds, and as before, a probability of 2^{-16} in subsequent rounds. This would give approximately 2^{255} pairs of plaintexts available in the analysis. However, for such pairs of plaintexts and ciphertexts, pairs satisfying the differential can no longer be assumed to have the form of (1) for neither halves. However, the value of every second byte of the difference in the ciphertexts would still be expected to be zero.

This means that for Camellia (including FL and FL^{-1}) reduced to 13 rounds, the above differential has a probability of $2^{-16\cdot 13-32-4-4} = 2^{-248}$. Thus, there will exist pairs of plaintexts following the expected values in the differential for up to 13 rounds. However, it is very difficult to see the applications of such (good) pairs in cryptanalytic attacks.

Summing up, there exist truncated differentials for every fixed key for up to 13 rounds of Camellia, specifying 64 bits of information after every round. We showed that these differentials can be used to distinguish Camellia when reduced to 6 rounds from a randomly chosen permutation. We expect that this 6-round distinguisher can be extended to a key-recovery attack on Camellia reduced to 7 rounds. Also, we showed that for Camellia reduced to 7 rounds but without the functions FL and FL^{-1} there is a distinguishing attack requiring 2^{68} chosen plaintexts. Since Camellia operates in at least 18 Feistel rounds, these attacks have no impact of the security of Camellia.

4 The Key-schedule

The key-schedule takes a 128-bit, a 192-bit or a 256-bit key, K, as input. In the first phase the key-schedule defines two keys K_L and K_R each of 128 bits, and then computes two other keys of 128 bits each, K_A and K_B, as a function of the user-selected key. The key K_B is used only for the 192-bit and 256-bit key versions.

For the 128-bit key version $K_L = K$ and $K_R = 0$. For the 192-bit key version K_L is the leftmost 128 bits of K, and the remaining 64 bits are assigned the left half of K_R. The right half of K_R is the bitwise negated value of its left half. For the 256-bit key version, K_L is the leftmost 128 bits of K, and K_R the rightmost 128 bits of K. K_A is computed as follows. Let $C^i = C^i_L \mid C^i_R$ and let $C^0 = K_L \oplus K_R$. Then we compute

$$\begin{aligned}
C^1_L &= F(C^0_L, \sigma_1) \oplus C^0_R && (2)\\
C^1_R &= C^0_L && (3)\\
C^2_L &= F(C^1_L, \sigma_2) \oplus C^1_R && (4)\\
C^2_R &= C^1_L && (5)\\
C^2 &= C^2 \oplus K_L && (6)\\
C^3_L &= F(C^2_L, \sigma_3) \oplus C^2_R && (7)\\
C^3_R &= C^2_L && (8)\\
C^4_L &= F(C^3_L, \sigma_4) \oplus C^3_R && (9)\\
C^4_R &= C^3_L, && (10)
\end{aligned}$$

and define $K_A = C^4$. Set $C^4 = C^4 \oplus K_R$, compute

$$\begin{aligned}
C^5_L &= F(C^4_L, \sigma_5) \oplus C^4_R && (11)\\
C^5_R &= C^4_L && (12)\\
C^6_L &= F(C^5_L, \sigma_6) \oplus C^5_R && (13)\\
C^6_R &= C^5_L, && (14)
\end{aligned}$$

and define $K_B = C^6$. The values σ_i are fixed constants. For the 128-bit key version the 26 round keys of each 64 bits are computed from the keys K_L and K_A. For the 192-bit key and 256-bit versions the 34 round keys of each 64 bits are computed from the keys K_L, K_R, K_A and K_B.

First, we are convinced that the above key-schedule makes related-key attacks very difficult. In these attacks an attacker must be able to get encryptions under several related keys. If the relation between, say, two keys is known then if the corresponding relations between the round-keys can be predetermined, sometimes one can predict how the keys encrypt a pair of different plaintexts. However, since the round-keys depend on K_A and K_B, which are results of encryptions, these round-key relations will be very hard to control and predict. Also, the slide-attacks seems to be very unlikely to succeed for Camellia.

Our second observation is that the above description of the key-schedules can be simplified somewhat. Consider the version of Camellia with 128-bit keys. Let $K_L = K_{LL} \mid K_{LR}$ and set $C^0 = 0$. Then we compute

$$\begin{aligned}
C^1_L &= F(C^0_L, K_{LL} \oplus \sigma_1) \oplus C^0_R && (15)\\
C^1_R &= C^0_L && (16)
\end{aligned}$$

$$
\begin{aligned}
C_L^2 &= F(C_L^1, K_{LR} \oplus \sigma_2) \oplus C_R^1 && (17)\\
C_R^2 &= C_L^1 && (18)\\
C_L^3 &= F(C_L^2, \sigma_3) \oplus C_R^2 && (19)\\
C_R^3 &= C_L^2 && (20)\\
C_L^4 &= F(C_L^3, \sigma_4) \oplus C_R^3 && (21)\\
C_R^4 &= C_L^3. && (22)
\end{aligned}
$$

Then again $K_A = C^4$. Note that this definition saves one exclusive-or operation and further the structure is now of a classical Feistel-type. All in all, the key K_A is the ciphertext of the plaintext zero in a four-round Feistel encryption scheme using the Camellia round function. The round keys in this Feistel scheme are dependent on the halves of the key K_L in the first two rounds, but they are fixed in the final two rounds. This has the effect that if an attacker is able, by some means, to find the value of K_A, then he can compute the key K_L in a straightforward manner. Also, if an attacker is able to find K_{AR} and K_{LL} (totally 128 bits), then he can compute all of K_A and K_L. In both cases, the attacker can compute the values of all round-keys. Since K_A is the ciphertext of a 4-round Feistel cipher depending on K_L, the above properties are somewhat surprising.

Consider next the version of Camellia with 192-bit and 256-bit keys. Let $K_L = K_{LL} \mid K_{LR}$, $K_R = K_{RL} \mid K_{RR}$, and set $C^0 = 0$. Then we compute

$$
\begin{aligned}
C_L^1 &= F(C_L^0, K_{LL} \oplus K_{RL} \oplus \sigma_1) \oplus C_R^0 && (23)\\
C_R^1 &= C_L^0 && (24)\\
C_L^2 &= F(C_L^1, K_{LR} \oplus K_{RR} \oplus \sigma_2) \oplus C_R^1 && (25)\\
C_R^2 &= C_L^1 && (26)\\
C_L^3 &= F(C_L^2, \sigma_3 \oplus K_{RL}) \oplus C_R^2 && (27)\\
C_R^3 &= C_L^2 && (28)\\
C_L^4 &= F(C_L^3, \sigma_4 \oplus K_{RR}) \oplus C_R^3 && (29)\\
C_R^4 &= C_L^3 && (30)\\
C_L^5 &= F(C_L^4, \sigma_5) \oplus C_R^4 && (31)\\
C_R^5 &= C_L^4 && (32)\\
C_L^6 &= F(C_L^5, \sigma_6) \oplus C_R^5 && (33)\\
C_R^6 &= C_L^5 && (34)
\end{aligned}
$$

Then $K_A = C^4 \oplus K_R$ and $K_B = C^6$. Here one can compute $K_A \oplus K_R$ from K_B and similarly from K_B one can compute $K_A \oplus K_R$.

The above considerations will very likely not imply a speed-up of an exhaustive key search, they only say that knowledge of some round-keys gives immediate knowledge of other round-keys.

5 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 Camellia, 2^{64} ciphertext blocks are required after which an attacker would be able to deduce information about the plaintext blocks.

The S-boxes used in Camellia have the highest possible nonlinear order of 8-bit S-boxes, namely seven. Therefore it can be expected that higher order differential attacks will have only limited applications.

Since the round function of Camellia is bijective, the non-surjective attack will not be applicable.

The interpolation attacks work particularly well for ciphers for which the nonlinear components have a simple mathematical description. For Camellia, the S-boxes used have a complex description. Furthermore the use of the functions FL and FL^{-1} are likely to destroy any mathematical structure from the S-boxes. In our opinion it is very unlikely that the interpolation attack will be of any threat to Camellia.

The mod-n cryptanalysis works particularly well for ciphers using a mix of rotations and modular additions. Since Camellia uses the exclusive-or operation and no modular additions, the mod-n analysis will not be applicable to Camellia.

The key-schedule of Camellia does not seem to allow for related-key attacks. The subkeys are generated by using the encryption function F. Therefore it is unlikely that there exist any easily detectable weak keys, and the slide attacks do not apply.

The integral cryptanalysis will apply to a few rounds of Camellia, but these attacks are expected to be less effective than the attacks based on truncated differentials.

Let us end this section with some comments on the S-boxes. Camellia uses four S-boxes, s_1, s_2, s_3, and, s_4. s_1 is derived from an inversion function in $GF(2^8)$ together with an affine transformation in the outputs and in the inputs. The other S-boxes are derived from s_1 by a simple rotation by one position of either the input or the output. In more detail, $s_2(x) = s_1(x) << 1$, $s_3(x) = s_1(x) >> 1$, and $s_4(x) = s_1(x << 1)$. Clearly, the four S-boxes are very related. The advantages of deriving all S-boxes from one S-box are clear for implementation reasons. The disadvantages are not as clear, however, if an attacker would find a weakness in one of the S-boxes, then there is a high probability that this weakness would appear in all S-boxes. We have not found any reason to suspect that cryptographic weaknesses are present nor will be detected in any of the S-boxes. Also, ciphers like the AES-candidate Rijndael make a more simple use of only one S-box. The S-box in Rijndael is derived in a manner similar to that of Camellia and is used throughout the cipher. So far, no reports have been published that this is a weakness for Rijndael. Thus there is no immediate indication that this poses any threat for the security of

Camellia.

6 Survey of previous results

The only previous results on Camellia that we are aware of are those of the designers themselves [2].

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 [28] 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

$$\Delta X = X \otimes (X')^{-1}, \tag{35}$$

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 [21, 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 $f : GF(2^n) \rightarrow GF(2^n)$ 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 $f : GF(2^n) \rightarrow GF(2^n)$ 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 [32] 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 [22]. A preliminary version of the attack on FEAL was described in 1992 [25]. Linear cryptanalysis [22] 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

$$(P \cdot \alpha) \oplus (C \cdot \beta) = (K \cdot \gamma) \tag{36}$$

which holds with probability $p_L \neq \frac{1}{2}$ over all keys [22], such that $|p_L - \frac{1}{2}|$, called the bias, is maximal. In (36) $P, C, \alpha, \beta, \gamma$ 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 [22] 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 [27].

Finally, in [23] 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

$$f(x) = \sum_{i=1}^{n} y_i \prod_{1 \leq j \leq n, j \neq i} \frac{x - x_j}{x_i - x_j}. \tag{37}$$

$f(x)$ 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 (37) 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 [30] 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 [20] 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] Aoki, Ichikawa, Kanda, Matsui, Moriai, Nakajima, Tokita. NTT and Mitsubishi Electric Corporation. Specification of Camellia - a 128-bit Block Cipher. July 13, 2000.

[2] Aoki, Ichikawa, Kanda, Matsui, Moriai, Nakajima, Tokita. NTT and Mitsubishi Electric Corporation. Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms. July 13, 2000.

[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 D. Wagner. Integral cryptanalysis. In preparation, 2001.

[21] 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.

[22] 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.

[23] 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.

[24] M. Matsui and T.Tokita. Cryptanalysis of a reduced version of the block cipher E2. In L. R. Knudsen, editor, *Fast Software Encryption, Sixth International Workshop, Rome, Italy, March 1999, LNCS 1636*, pages 71–79. Springer Verlag, 1999.

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

[28] 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.

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

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

[31] NTT Nippon Telegraph and Telephone Corporation. Specification of e2: A 128-bit block cipher. Submitted as candidate for AES. Available at `http://www.nist.gov/aes`.

[32] 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.