Security Analysis of the Block Cipher Camellia

VERSION 1.1

FINAL REPORT

Alex Biryukov
University of Luxembourg, Luxembourg

Ivica Nikolić
Nanyang Technological University, Singapore

Contents

Chapter 1

Introduction

In this chapter is given a short description of *Camellia* along with known cryptanalytical results. A preliminary analysis of the operations used in *Camellia* is provided.

1.1 Description

Camellia is 128-bit Feistel block cipher with a number of rounds that depends on the key size: 18 rounds for 128-bit key size, 24 rounds for 192-bit and 256-bit keys. All the transformations in the cipher can be divided into two parts:

1. **State transforms.** *Camellia* follows the balanced Feistel design and in each round half of the state (i.e. 64 bits or 8 bytes) is updated by the round function F which is a **byte**-oriented SP network: after the XOR of the round key, the S-layer with 8 S-boxes in parallel is applied, followed by the P-layer which is a matrix multiplication. There are additional **word**-oriented transforms FL, FL^{-1} after the rounds 6, 12, and 18.

2. **Key schedule transforms.** The round keys are produced from the master key by the same Feistel transform with an additional **word** rotations.

The mixture of the byte and word oriented transforms in the key schedule makes the related-key analysis of *Camellia* complex.

1.2 Known Analysis

Camellia has attracted plenty of attention among the cryptographic community and the amount of published analysis is quite large. Most of the attacks are in the framework of impossible differentials on up to 12 rounds with the non-linear layers and 15 rounds without the layers [28, 34, 33, 22, 23, 20, 1, 18, 17, 5, 2, 21]. Analysis against truncated and higher order differentials was given in

[14, 12, 28, 8]. Square attacks were proposed on up to 12 rounds of *Camellia* without the non-linear layers[16, 35, 9], and collision attacks on up to 6 rounds [25, 32, 31, 11]. Various attacks and analysis on the original and modified versions of *Camellia* were also presented in [27, 15, 30, 13, 4, 19, 10, 6, 26].

1.3 Analysis of the Transformations

To estimate the resistance of *Camellia* against various attack, first we focus on each of the transformations used in the cipher. In particular, we analyze the S-boxes and the linear transformations.

1.3.1 S-box Analysis

The complexity of the differential attacks on a cipher is tightly related to the differential properties of the S-boxes used in the cipher. In *Camellia* there are 4 different types of 8x8 non-linear bijective S-boxes. The maximal differential propagation probability for each of them is 2^{-6}, which is optimal.

1.3.2 Analysis of the Linear Transformations

The branch number of the multiplication by the matrix L, i.e. the minimal number of active input and output bytes, is 5. This is sub-optimal, as the maximal branch number for this type of transformations is 9. Low branch number could lead to potential (impossible) differential attacks on higher number of rounds.

1.3.3 Analysis of the Non-Linear Transformations

The functions FL, FL^{-1} are 2-round Feistels and use round keys to produce the output. There exist many good differential characteristics for them, including those that have zero output difference for specific difference in the input state and the input round keys, hence FL, FL^{-1} can cancel the difference in the state, resulting in a related-key differential attack on extended number of rounds. However, these two functions play an important role in increasing the diffusion among the bytes.

Chapter 2

Cryptanalysis of the Full-Round *Camellia-128* without FL, FL^{-1} in the Hashing Mode

2.1 Cryptanalysis of Modified *Camellia* in the Hashing Mode

In this chapter we present a distinguisher for the Davies-Meyer[1] hash function mode of *Camellia* with 128-bit keys and without the non-linear layers FL, FL^{-1}. The attack exploits the complementation property of Feistel constructions which leads to producing differential q-multicollisions which are non-trivial distinguisher for the hashing mode of *Camellia-128*.

2.2 Complementation Property of the Classical Feistel Construction

The complementation property was first observed in DES. It is based on the observation that if one flips all of the bits of the master key and the plaintext, then all of the bits of the ciphertext will flip as well. The foundation of this observations for Feistel ciphers is given below. Without loss of generality we assume that the Feistel is balanced as the case for unbalanced Feistels can be examined similarly.

[1] Indeed the attack can be mounted on other modes as well.

A balanced Feistel with r rounds is defined as:

$$\begin{aligned} L_{n+1} &= F(L_n, K_n) \oplus R_n \\ R_{n+1} &= L_n, \end{aligned}$$

where K_n is the n-th round key, $P = L_0 || R_0$ is the plaintext, and $C = L_r || R_r$ is the ciphertext. In the vast majority of Feistel ciphers, the round function $F(L, K)$ can be decomposed as:

$$F(L, K) = G(L \oplus K),$$

i.e. first the round key is bitwise added to the state L, followed by some additional non-linear and linear transformations (G is usually a Substitution-Permutation network). We use the term *classical Feistels* for the ciphers that have such F function.

Let $KS(K)$ be the key schedule function of the cipher, i.e. given the master key K, the function produces $K_i, i = 1, \ldots, r$ round keys:

$$KS(K) = (K_1, \ldots, K_r)$$

Further assume that all of the round keys K_i are obtained by (possibly different) bit permutations of the master key K (as in the case of DES). If one has two related master keys K^1, K^2 such that $K^1 \oplus K^2 = -1$ (with -1 we denote the difference in all of the bits) then for all i holds $K_i^1 \oplus K_i^2 = -1$. Let P^1, P^2 be two related plaintexts such that $P^1 \oplus P^2 = -1$, i.e. $L_0^1 \oplus L_0^2 = -1$ and $R_0^1 \oplus R_0^2 = -1$. Then by induction for each i we get:

$$\begin{aligned} L_{i+1}^1 \oplus L_{i+1}^2 =& F(L_i^1, K_i^1) \oplus R_i^1 \oplus F(L_i^1, K_i^1) \oplus R_i^1 = \\ & G(L_i^1 \oplus K_i^1) \oplus R_i^1 \oplus G(L_i^1 \oplus -1 \oplus K_i^1 \oplus -1) \oplus R_i^1 = R_i^1 \oplus R_i^2 = -1 \\ R_{i+1}^1 \oplus R_{i+1}^2 =& L_i^1 \oplus L_i^2 = -1 \end{aligned}$$

Therefore $L_r^1 \oplus L_r^2 = -1, R_r^1 \oplus R_r^2 = -1$ and hence there is a difference in all of the bits of the ciphertext.

The complementation property of such ciphers allows reduction of the key space by one bit as for the brute force of the whole key space it is sufficient to try only one half of all possible keys – the other half will produce a compliment ciphertext under a compliment plaintext.

The complementation property can be observed for ciphers that not necessarily have a key schedule composed of permutations. Notice, the only requirement on the key schedule is to produce complemented round keys.

Lemma 1 *Let for an n-bit classical Feistel cipher $E_K(P)$ with k-bit keys and a key schedule $KS(K)$ exists a differential with probability p for $KS(K)$ with output difference in all of the bits in all of the round keys, i.e.*

$$\exists \Delta : KS(K \oplus \Delta) \oplus KS(K) \xrightarrow{p} (-1, \ldots, -1)$$

Then, if $p > 2^{-n}$, a weak-key class of size $p \cdot 2^k$ exists for the cipher $E_K(P)$.

Proof: Once the difference in all of the round keys is -1, the complementation property can be applied, i.e. the differential in the state holds with probability 1. Therefore if the attacker can build a differential with the input difference in the master keys Δ, and output difference -1 in all of the round keys, then the differential $(-1, \Delta) \rightarrow (-1)$ for the cipher $E_K(P)$ holds with probability p. To find the right key pair that follows the differential in the key schedule one has to try around $1/p$ pairs of master keys with input difference Δ, therefore the size of this weak key class is $2^k \cdot p$. For any cipher, to produce a pair of complemented plaintexts that result in complemented ciphertexts, one has to try around 2^n pairs, hence the probability of the differential has to be higher than 2^{-n}. $\square$

Remark 1 *If the attack can be converted into a key-recovery attack, the probability of the differential can be lower, nonetheless higher than* 2^{-k}.

Remark 2 *The complementation property holds regardless of the number of rounds in the cipher, by increasing the number of rounds one cannot expect to get a better resistance against this type of attacks.*

Remark 3 *The additional key whitenings at the beginning and at the end of the Feistel do not influence the attack complexities, but merely change the input difference in the plaintext and the output difference in the ciphertext.*

Remark 4 *The requirement of having the difference -1 in all of the round keys can be replaced with the requirement of having* Δ_1, Δ_2 *differences that alternate, i.e. the first round key has* Δ_1*, the second* Δ_2*, the third* Δ_1*, etc. Then, if the plaintext has input difference* $\Delta_1 || \Delta_2$*, the complementation property would still hold.*

2.3 Notations

We analyze full-round *Camellia-128* without the non-linear layers, i.e. we assume FL, FL^{-1} to be identity functions. To describe the attack we introduce a few notations.

Camellia is a classical Feistel cipher with a non-linear key schedule defined as follows. The 128-bit master key K_L is split into two keys L, R, i.e. $K_L = L||R$ – both L and R are seen as 8-byte vectors. Further, these keys are fed to a 4-round Feistel-like transformation with an additional keys feedback after the second round (see Fig. 2.3). Formally, the key schedule can be described as:

$$L_1||R_1 = K_L \tag{2.1}$$

$$L_2 = F(L_1 \oplus \Sigma_1) \oplus R_1; \qquad R_2 = L_1 \tag{2.2}$$

$$L_3 = F(L_2 \oplus \Sigma_2) \oplus R_2; \qquad R_3 = L_2 \tag{2.3}$$

$$\overline{L_3} = L_3 \oplus L_1; \qquad \overline{R_3} = R_3 \oplus R_1 \tag{2.4}$$

$$L_4 = F(\overline{L_3} \oplus \Sigma_3) \oplus \overline{R_3}; \qquad R_4 = \overline{L_3} \tag{2.5}$$

$$L_5 = F(L_4 \oplus \Sigma_4) \oplus R_4; \qquad R_5 = L_4 \tag{2.6}$$

$$K_A = L_5||R_5 \tag{2.7}$$

where Σ_i are word constants. In the sequel, we omit the addition of the constants as they play no role in our analysis. The function F is an SP network, with the S-layer defined as application of eight 8x8 S-boxes, and P-layer is a multiplication of the eight-byte input with 8x8 byte matrix P. All the round keys K_i used in the state are obtained from the two keys K_L and K_A with rotations on various amounts, e.g. $K_4 = K_L \lll_{15}, K_{15} = K_A \lll_{95}$, etc.

2.4 Complementing Camellia-128

The presented below distiguisher for the hash mode of *Camellia-128* without FL, FL^{-1} can be summarized as follow. As Camellia is a classical Feistel, by Lemma 1 we can apply the complementation property, if we can find a differential for the key schedule. We show that such differential exists, however its probability is too low for an attack on the block cipher. On the other hand, we show that a key pair following the differential can be found with a complexity 2^{112}, i.e. lower than than 2^{128}, hence this leads to a distinguisher for the hash mode of *Camellia-128*. We note that a large part of the analysis is focused on proving the existance of the differential and presenting an algorithm for obtaining a key pair that satisfies the differential.

From the description of Camellia-128 it follows that two different keys K_L, K_A are used, the first key being also the only input to the key schedule. Since the round keys are produced from these two keys with various rotations it follows that the differences in K_L, K_A have to be invariant of rotations and thus be -1. Therefore, we need the differential $\Delta K_L \to (\Delta K_L, \Delta K_A)$ to be $(-1) \to (-1, -1)$.

The easiest way to build such differential is by providing a differential trail, i.e. besides specifying the input and output differences, fixing as well the intermediate differences after each transformation in the key schedule. Note that from the condition on the differential it follows that $\Delta L_1 = \Delta R_1 = \Delta L_5 = \Delta R_5 = -1$, i.e. each byte of these words has the fixed difference -1 (or ff in the hexadecimal representation). Therefore, in the first and the fourth round of the key schedule, the number of active bytes has to be maximal, i.e. eight active bytes will enter the S-layer. It is tempting to go with a trail that has no active bytes (or one active byte) in both the second and third round, hence obtain a

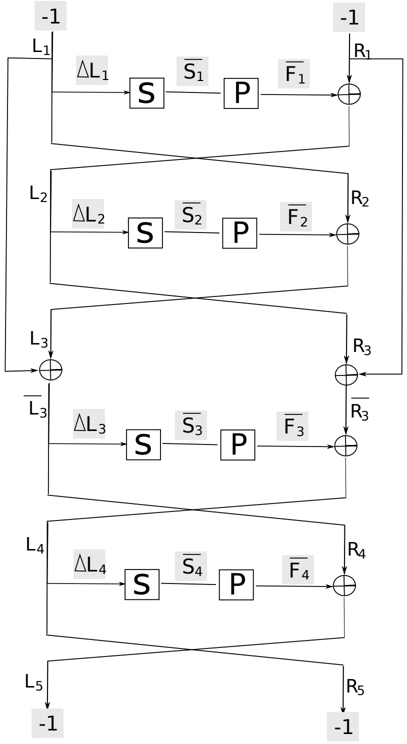

Figure 2.1: The key schedule of *Camellia-128* with the $(-1,-1) \rightarrow (-1,-1)$ differential. The gray values are the differences.

trail of the form (we write only the round-by-round active bytes entering the F function):

$$8 \rightarrow 0 \rightarrow 0 \rightarrow 8 \text{ or } 8 \rightarrow 1 \rightarrow 1 \rightarrow 8$$

However, these types of trails are not possible due to the matrix multiplication P, i.e. P-layer. For example, if we require no active bytes in the second round, then this means the output of the F function in the first round has canceled with the -1 difference in R_1, i.e. if we denote with $\tilde{a} = (a_1, \ldots, a_8)$ the output difference of the S-boxes in the function F of the first round, then the above condition can be expressed as:

$$P \cdot \tilde{a} \oplus (-1) = 0 \Rightarrow \tilde{a} = (0,0,0,0,-1,-1,-1,-1)$$

The solution vector $\tilde{a}$ has difference only in 4 bytes out of 8, while all the bijective S-boxes are active, i.e. we get a contradiction. Therefore, the second round of the key schedule cannot have zero active bytes. A similar situation can be observed when the second (or the third) round has only 1 active byte.

The above result suggests that the minimal number of active bytes in the key schedule is $8 + 2 + 2 + 8 = 20$. Theoretically, this can lead to a trail with probability $2^{-6\cdot 20} = 2^{-120} > 2^{-128}$ when all the active S-boxes hold with probability 2^{-6}. Due to the specific input and output differences in the active S-boxes in the first and the fourth rounds, this is not achievable – the differential probability of these S-boxes is 2^{-7}. Therefore if we assume the differential is composed of a single trail only, its probability would always be lower than 2^{-128}.

Further we try to find the actual probability of the differential taking into account all possible differential trails that compose it. All the trails can be divided into two groups: trails that have the same path (i.e. the have the same position of the active bytes, but different values for the differences), and trails that have different path.

Let $\tilde{S_i}$ be a possible output difference of the S-layer at round i, and $\tilde{F_i}$ be an output difference of the F function at round i. Note, both $\tilde{S_i}, \tilde{F_i}$ are 8 byte vectors – $\tilde{S_i} = (s_i^1, \ldots, s_i^8), \tilde{F_i} = (f_i^1, \ldots, f_i^8)$. Also, let F_i be the actual output of the F function at round i. We will use $S(x)$ to denote the S-layer, and ΔL_i to denote the difference of the left state at round i, hence $S(\Delta L_i) = \tilde{S_i}$. From the definition of the round function it holds $F(\Delta L_i) = P \cdot S(\Delta L_i) = P \cdot \tilde{S_i} = \tilde{F_i}$.

For $\tilde{S_1}, \tilde{S_2}, \tilde{S_3}, \tilde{S_3}$ the following conditions apply (see Fig.1):

- $\tilde{S_1}$ is produced when -1 difference in L_1 goes through the S-layer:

$$\tilde{S_1} = S(-1) \tag{2.8}$$

- $\tilde{S_2}$ is produced with an XOR of $\tilde{F_1}$ and the difference -1 in R_1, followed by the S-layer:

$$\tilde{S_2} = S(\tilde{F_1} \oplus (-1)) = S(P \cdot \tilde{S_1} \oplus (-1)) \tag{2.9}$$

- $\tilde{S_3}$ is produced with application of the S-layer to $\Delta\overline{L_3}$:

$$\tilde{S_3} = S(\Delta\overline{L_3})) = S(P \cdot S_2) \tag{2.10}$$

  Additionally, when $\tilde{F_3}$ is XOR-ed to $\Delta\overline{R_3}$, the output difference -1 is obtained in R_5:

$$\tilde{F_3} \oplus \Delta\overline{R_3} = P \cdot \tilde{S_3} \oplus P \cdot \tilde{S_1} = -1 \tag{2.11}$$

- $\tilde{S_4}$ is produced when -1 difference in R_5 goes through the S-layer:

$$\tilde{S_4} = S(-1) \tag{2.12}$$

  Additionally, when $\tilde{F_4}$ is XOR-ed to $\Delta\overline{L_3}$, the output difference -1 is obtained in L_5:

$$\tilde{F_4} \oplus \Delta\overline{L_3} = P \cdot \tilde{S_4} \oplus P \cdot \tilde{S_2} = -1 \tag{2.13}$$

The probability of the differential can be computed as the sum of probabilities of all differential trails defined with 4 intermediate differences:

$$\sum_{(\tilde{S_1},\tilde{S_2},\tilde{S_3},\tilde{S_4})|\ (2.8),(2.9),(2.10),(2.11),(2.12),(2.13) \text{ are satisfied}} 2^{-7(|\tilde{S_2}|+|\tilde{S_2}|+|\tilde{S_3}|+|\tilde{S_4}|)} \tag{2.14}$$

where $|\tilde{S_i}|$ denotes the number of active bytes in $\tilde{S_i}$. In the following, we try to simplify the conditions and to achieve formula for computing the above probability.

Note that although both $\tilde{S_1} = S(-1)$ and $\tilde{S_4} = S(-1)$ are produced when (-1) goes through the S-layer, a randomly chosen difference $\tilde{S_1}$ and a difference $\tilde{S_4}$ are not necessarily the same (in fact they are different with a very high probability). To distinguish them we will use $S(-1)_{L_1}$ for the former and $S(-1)_{R_5}$ for the later.

Further we reduce the conditions on all $\tilde{S_i}$ to conditions only on $\tilde{S_2}, \tilde{S_3}$. From (2.11) and the linearity of the matrix multiplication P it follows that

$$P \cdot \tilde{S_3} \oplus P \cdot \tilde{S_1} = P \cdot (\tilde{S_3} \oplus \tilde{S_1}) = -1$$

This leads to:

$$\tilde{S_3} = P^{-1}(-1) \oplus \tilde{S_1} \tag{2.15}$$

Similarly, from (2.12) and (2.13) we get:

$$\tilde{S_2} = P^{-1}(-1) \oplus S(-1)_{R_5} \tag{2.16}$$

Taking into account (2.15), the condition (2.9) can be expressed as:

$$\tilde{S_2} = S(P \cdot \tilde{S_1} \oplus (-1)) = S(P \cdot (\tilde{S_3} \oplus P^{-1}(-1)) \oplus (-1)) = \tag{2.17}$$

$$= S(P \cdot \tilde{S_3} \oplus (-1) \oplus (-1)) = S(P \cdot \tilde{S_3}) \tag{2.18}$$

Let us summarize our findings. We get that for $\tilde{S_2}, \tilde{S_3}$ defined as:

$$\tilde{S_2} = P^{-1}(-1) \oplus S(-1)_{R_5} \tag{2.19}$$

$$\tilde{S_3} = P^{-1}(-1) \oplus S(-1)_{L_1} \tag{2.20}$$

two additional conditions have to hold:

$$\tilde{S_2} = S(P \cdot \tilde{S_3}) \tag{2.21}$$

$$\tilde{S_3} = S(P \cdot \tilde{S_2}) \tag{2.22}$$

In $\tilde{S_1}, \tilde{S_4}$ there are always 8 active S-boxes. The number of active S-boxes in $\tilde{S_2}, \tilde{S_3}$ is defined by the above conditions. As P is linear, we can compute the value of the vector $P^{-1}(-1)$, i.e.

$$P^{-1}(-1) = (0, 0, 0, 0, ff, ff, ff, ff) \tag{2.23}$$

Since the S-boxes in Camellia are bijective, the vector $S(-1)$ always has 8 active S-boxes. Therefore from (2.19),(2.20) we can conclude that the first 4 elements of $\tilde{S_2}, \tilde{S_3}$ have to be non-zero, thus the number of active S-boxes in round 2 and 3 is at least 4 (the first 4 bytes must be active). Additionally, regarding the number and position of the active S-boxes, since there are always at least 4 active S-boxes in $\tilde{S_2}$ and $\tilde{S_3}$, the conditions (2.21),(2.22) can always be satisfied (the branch number of P is 4).

Finally, we can give the probability of the differential $(-1,-1) \rightarrow (-1,-1)$:

$$\sum_{(\tilde{S_2},\tilde{S_3}) \text{ satisfy (2.19), (2.20), (2.21), (2.22)}} 2^{-7(8+|\tilde{S_2}|+|\tilde{S_3}|+8)} \tag{2.24}$$

Recall that a differential is a collection of trails that take the same path and trails that take different path. We group all trails that take the same path into one single *truncated trail*. Then a differential is a collection of truncated trails and hence its probability is the sum of probabilities of the truncated trails. To define a truncated trail we just have to fix the position of the actives S-boxes in the four rounds of the key schedule. With T_i we denote the truncated difference entering the round function of round i. Then a truncated trail can be defined as T_1, T_2, T_3, T_4. An actual trail with $\tilde{S_1}, \ldots, \tilde{S_4}$ belongs to a truncated trail if the position of the active S-boxes in S_i coincide with the position of the active S-boxes in T_i. Obviously $T_1 = T_4 = (1, 1, \ldots, 1)$ as all the S-boxes in the first and the fourth round are active. For the probability of the differential we get:

$$\sum_{(T_2,T_3)} 2^{-7(8+|T_2|+|T_3|+8)} \#\{(\tilde{S_2},\tilde{S_3}) | \tilde{S_2} \in T_2, \tilde{S_3} \in T_3, \tilde{S_1}, \tilde{S_2} \text{ satisfy (2.19), (2.20), (2.21), (2.22) } \} \tag{2.25}$$

Hence, to find the probability of the differential, we only have to count the number of possible differential trails (that satisfy a set of conditions) in all possible truncated trails T_2, T_3 of the form $(1, 1, 1, 1, x_5, x_6, x_7, x_8), x_i \in \{0, 1\}$. To proceed further we define the notion of compliance.

Definition 1 *Two differences Δ_1, Δ_2 comply through the function $f(x)$ if there exist x such that $f(x \oplus \Delta_1) \oplus f(x) = \Delta_2$.*

This notion is introduced to check if some input difference Δ_1 at function $f(x)$ can produce output difference Δ_2.

Observation 1 *Two randomly chosen differences Δ_1, Δ_2 comply through the S-boxes of Camellia with probability $\frac{127}{255} \approx 2^{-1}$.*

Every input difference to the S-box can go to 127 output differences or approximately to 2^7 out of $2^8 - 1$ possible, which is around 2^{-1}.

As an example, let us compute the number of possible trails for the case when T_2, T_3 have all 8 active bytes. From the properties of the S-boxes used in *Camellia* we have that each input byte difference (including the difference ff) can go to 127 or approximately[2] 2^7 distinct output differences. Since we have 8 active input bytes in $S(-1)_{L_1}$ and in $S(-1)_{R_5}$, there are in total $2^{7 \cdot 8} = 2^{56}$

[2] We can approximate with 2^7 as one of the output differences happens twice, which means that although we increase the number from 127 to 128, on the other hand we decrease the probability for this difference from 2^{-6} to 2^{-7}, hence the trade off is compensated. This can easily be checked if one takes instead of bytes, 7-bit nibbles. Then the maximal differential probability of 7x7 S-box can be 2^6.

differences for $\tilde{S_2}$ and $\tilde{S_3}$ (see the definitions (2.19),(2.20)). As $\tilde{S_2}$ has 8 active bytes, the following condition has to hold:

$$(d_1, \dots, d_8) = P^{-1}(-1) \oplus (s^1_{R_5}, \dots, s^8_{R_5}) \tag{2.26}$$

$$= (0,0,0,0,ff,ff,ff,ff) \oplus (s^1_{R_5}, \dots, s^8_{R_5}), \tag{2.27}$$

where all d_i are non-zero. Hence, out of all 2^{56} this condition satisfy $2^{56} \cdot (1 - 127^4) \approx 2^{56}$ differences, or approximately all. A similar conclusion can be obtained regarding (2.20).

Now let us focus on (2.21),(2.22). The probability that $\tilde{S_2}$ comply with $\tilde{S_3}$ from (2.21) can be computed as:

1. the probability that $P \cdot \tilde{S_3}$ is 8 byte difference – it is approximately 1. In the general case, when $\tilde{S_2}$ has i active bytes, the probability is approximately $2^{-8\cdot(8-i)}$.
2. the probability that each of the differences in 8 bytes of $\tilde{S_2}$ and $P \cdot \tilde{S_3}$ comply. This is 2^{-8}, while in the general case it is 2^{-i} for differences in i bytes.

Therefore, for a randomly chosen differences the probability of (2.21) is 2^{-8}. A similar reasoning can be applied to (2.22). Hence, out of all possible $\tilde{S_2}, \tilde{S_3}$ there are $2^{56} \cdot 2^{56} \cdot 2^{-8} \cdot 2^{-8} = 2^{96}$ differences that satisfy all four conditions. Therefore, for $T_2 = T_3 = (1,1,1,1,1,1,1,1)$, the probability of the differential is at least:

$$2^{96} \cdot 2^{-7(8+8+8+8)} = 2^{96} \cdot 2^{-224} = 2^{-128} \tag{2.28}$$

If we take into account all possible T_2, T_3 for the probability of the differential we get:

$$\sum_{i,j} 2^{-7(8+i+j+8)} C_4^{i-4} \cdot C_4^{j-4} 2^{112-8\cdot(8-i)-8\cdot(8-j)} 2^{-8(8-i)-i} 2^{-8(8-j)-j} \approx \tag{2.29}$$

$$\approx 2^{-128} \tag{2.30}$$

Thus, by Lemma 1, the size of the weak key class is $2^{128} \cdot 2^{-128} = 1$. For this key K, the complementation property holds, i.e. $KS(K \oplus (-1)) \oplus KS(K) = -1$, and taking into account the whitening keys, we get that for any plaintext P, it holds

$$E_{K\oplus(-1)}(P) = E_K(P).$$

Note that the size of the weak key class is too small for any attack on the cipher, however it is sufficient for an attack on the hash function mode of the cipher. As a compression function, we can choose the standard Davies-Meyer compression mode:

$$C(H, M) = E_M(H) \oplus H \tag{2.31}$$

Let K be the key value for which the $(-1,-1) \to (-1,-1)$ differential in the key schedule holds. For the compression function we get that for any H the following holds:

$$C(H, K \oplus (-1)) \oplus C(H, K) = E_{K\oplus(-1)}(H) \oplus H \oplus E_K(H) \oplus H = 0 \tag{2.32}$$

Therefore if we can find the correct key K (which is indeed the correct message M, as $M = K$ in the hash mode), we can produce collisions for the compression function of *Camellia*. Note, as H can be arbitrary, this leads to collisions for the whole hash function. To find the exact value of the key K we use the conditions (2.19)-(2.22) combined into the algorithm:

1. Create a set $\tilde{S}$ of all possible differences $P^{-1}(-1) \oplus S(-1)$ – the size of the set is 2^{56}
2. Create a set S^R of pairs of differences $(\delta_2, \delta_3), \delta_2, \delta_3 \in \tilde{S}$ such that δ_2 complies with $P \cdot \delta_3$ and δ_3 complies with $P \cdot \delta_2$ - the size of this set is 2^{96}
3. Choose a random pair (δ_2, δ_3) from S^R
4. Produce the value of L_1 (and the corresponding F_1) that converts -1 into the $\delta_3 \oplus P^{-1}(-1)$, i.e. $S(L_1 \oplus (-1)) \oplus S(L_1) = \delta_3 \oplus P^{-1}(-1)$. As δ_3 has 8 active S-boxes, and for each active S-box there are 2 different values (A and $A \oplus (-1)$), for a fixed δ_3 there are 2^8 possible values of (L_1, F_1)
5. Produce similarly the values of (L_4, F_4) from δ_2
6. Produce $F_3 = L_4 \oplus \overline{F_3} = L_4 \oplus F_1$, and $\overline{L_3} = F^{-1}(F_3)$. Check if $F(\overline{L_3} \oplus P \cdot \delta_2) \oplus F(\overline{L_3}) = P \cdot \delta_3$. If not, go to step 3
7. Produce $F_2 = \overline{L_3}$, and $L_2 = F^{-1}(F_2)$. Check if $F(L_2 \oplus P \cdot \delta_3) \oplus F(L_2) = P \cdot \delta_2$. If not, go to step 3
8. Output the key $(L_1, R_1) = (L_1, F(L_1) \oplus L_2)$

The probability of steps 6,7 is 2^{-56} each and there are $2^{2(48+8)}$ possible (L_1, F_1) and (L_4, F_4). Hence, after repeating step 3 2^{96} times and steps 4,5 2^{112} times, one key candidate will be produced. Thus the complexity of the algorithm is 2^{112}.

Note, with an effort of 2^{112} we can produce one collision for the compression function of *Camellia-128* (without FL, FL^{-1}). As once we have the correct message M, we can produce collision for any input chaining value, it means that for any messages M_1, M_3 (the M_3 block is used as message padding), we can produce a collision for the hash function of *Camellia-128*. The colliding pairs are $(M_1 || M \oplus (-1) || M_3)$ and $(M_1 || M || M_3)$. Therefore, to produce q collisions with the same fixed difference between the message words (the difference is $(0|| - 1||0)$ we need 2^{112} calls to the hash function[3]. On the other hand, for the generic case, producing such collisions (they are indeed called differential q multicollisions, see [3]), one needs around $q 2^{\frac{q-2}{q+2}128}$ calls to the hash function. Hence, producing 256 differential multicollisions requires $2^8 \cdot 2^{\frac{254}{258}128} \approx 2^{134}$ encryptions whereas for the hash function of *Camellia-128* without the non-linear layers FL, FL^{-1} in the Davies-Meyer mode, they can be produced with 2^{112} calls to the hash function.

[3] Actually, the number is smaller, as one hash function call requires much larger number of operations compared to the steps of our algorithm.

2.5 Applications to *Camellia-192* and *Camellia-256*

The key schedule for the cipher with 192 and 256 bit keys requires generation of additional key K_B. Also note that the initial difference -1 in K_L and K_R would cancel at the beginning and will be introduced only after the feedback at the beginning of the third round. This however leads to a situation where -1 cannot be obtained in K_A, i.e. it is trivial to see that two rounds of the Feistel cannot produce output difference of -1 if the input difference is -1. Hence, the hash functions based on *Camellia-192* and *Camellia-256* are resistant against this type of differential q-multicollisions.

Chapter 3

Analysis Against Various Attacks

In this chapter an analysis of the resistance of *Camellia* against different single-key and related-key attacks is given. In particular, we focus on:

- Classical Differential Cryptanalysis
- (Amplified) Boomerang Cryptanalysis
- Truncated Differential Cryptanalysis
- Slide Attack
- Rotational Attacks

3.1 Differential Cryptanalysis

Differential attacks are the most popular form of cryptanalysis for block ciphers. A widely accepted approach for designing a byte-oriented cipher resistant against differential attacks is to ensure that each differential characteristic has a certain number of active S-boxes. Besides on the differential properties of the S-boxes, this number also depends on the size of the state in the single key scenario, and on the size of the key in the related-key scenario. In the sequel we give the probabilities of the best differential trails based on the number of active S-boxes.

3.1.1 Single-key Differentials

In the single-key scenario we assume there is no difference in the key, and there is some initial difference in the plaintext. We use an advanced brute-force approach, based on Matsui's technique used to find the best characteristics in DES (see [24]), to find the probabilities and the number of active S-boxes

Table 3.1: The number of active S-boxes in the best round-reduced single-key differential characteristics for *Camellia*.

Rounds	Active S-boxes
1	0
2	1
3	2
4	7
5	9
6	11
7	13
8	15
9	18
10	21
11	22
12	25
13	26
14	30
15	32
16	34

in the best round-reduced single-key characteristics. We use the term "best" with regards to the characteristics with the highest probability. As mentioned previously, *Camellia* uses four different type of S-boxes: the maximal differential probability of each of them is 2^{-6}. We use these probabilities as estimations for the active S-boxes. The best single-key characteristics, in terms of the minimal number of active S-boxes, are presented in Table 3.1.

As the probability of each active S-box is 2^{-6} and the block size is 128 bits, to be valid a characteristic can have at most $\lfloor \frac{128}{6} \rfloor = 21$ active S-box. Given the table, we can easily give upper bounds on the best differential attacks based on differential characteristics:

Observation 2 *For all key sizes of Camellia, no single-key differential characteristic on more than 10 rounds can have a probability higher than* 2^{-128}.

The theory of computation of differential characteristics is far ahead of computing the same probabilities of differentials. Hence, we cannot give a precise bound on the number of rounds sufficient for resistance of *Camellia* against differential attacks. However under the standard assumptions that the number of characteristics within a differential is low and the relatively high margin for the best characteristic on more than 10 rounds, we expect *Camellia* to be resistant against single-key differential attacks.

3.1.2 Related-key Differentials

The non-byte oriented rotations used for the generations of the subkeys make the analysis of *Camellia* against related-key attacks infeasible. We have previously analyzed *Camellia* against related-key attacks, by modifying the rotations and making them all multiple of 8 (see [4]). The analysis has shown that no related-key attacks exist for this version of the cipher. Modifying our search algorithm and applying it to the original *Camellia* is possible but again infeasible. The true problem lies in the fact that if one fixes the position of the active bytes in the keys K_L, K_R, K_A, K_B then there are too many possible configurations for the active bytes in the subkeys. For example, let in K_L only the least significant byte (i.e. 0 byte) is active. Then $k_3 = K_L \lll 15$ can have active only the second byte if the difference in the least significant bit of the active byte was zero, only the first byte, if the difference was only in the least significant bit, or both the first and the second if the difference was in all bits of the active byte. Hence, for a single active byte we get three possibilities, i.e. the branching is three. Therefore, we get that the branching is exponential in the number of active bytes *per subkey*. Taking into account that there are 18 round subkeys in *Camellia-128*, we get that the search is infeasible as there are too many possible subkeys differences for a single difference in the keys K_L, K_R, K_A, K_B.

Note, we cannot conjecture if the rotations in the subkeys make the cipher more resistant against related-key differential attacks as unlike for the byte-oriented version, the search for the original version is impractical. What is interesting is the fact that due to the high branching of the active bytes in the subkeys, indeed one can expect to get better related-key trails as the number of possible related-key trails is much larger. Again let us reuse our previous example with a difference in a single active byte of K_L. In the (modified) byte-oriented version, the difference in the subkey k_3 is uniquely determined (it is in the second byte). However, in the original version the difference can be in 1, or 1 and 2 or 2. Hence, when canceling the active bytes, one gets more possibilities, i.e. the subkey can cancel the state difference of the byte 1, 1 and 2 or 2. On the other hand, finding the exact difference for the active bytes in the keys (and the subkeys) that follow a certain related-key differential path might not always be possible.

The complementation property mentioned previously can also be seen as a related-key differential attack. However, due to the low probability of the differential in the key schedule, it poses no threat to the security of the cipher in terms of related-key differential attack.

3.2 Boomerang Cryptanalysis

In boomerang attacks, the characteristics do not have to cover the full cipher. Indeed, the number of rounds they cover should be chosen such that the probability of the boomerang is maximal. As we already have the probabilities of the best round-reduced characteristics, we can easily find the probability (and respectively the complexities) of the best boomerang attacks.

To find the best single-key boomerang we should take into account the results from Table 3.1. When the top characteristic is on 3 rounds and the bottom on 4 rounds, i.e. 7 rounds in total, the probability of the boomerang might be higher than 2^{-128} (only under the assumption that all of the active S-boxes hold with maximal probability of 2^{-128}). For any other choice of rounds (with sum greater than 7), the probability of the boomerang is lower than 2^{-128}. Hence, we can conclude that no boomerang exists for 8 rounds. Using some advanced techniques, an attacker might be able to skip round at the beginning, middle and at the end of the boomerang. However, the security margin is very high, and therefore we can conclude that approximately 11 rounds of *Camellia* are resistant against boomerang attacks. The case of amplified boomerangs gives no advantage to the attacker, over the classical boomerangs.

3.3 Truncated Differential Cryptanalysis

In truncated differential attacks, instead of following the propagation of certain difference through the rounds of the cipher and specifying how the initial difference changes after each transformation, the attacker only examines the position of the bytes with differences (i.e. active bytes) through the rounds. Hence the linear transformations in the cipher have the main and only impact on the probability of a characteristic. To find the best round-reduced truncated differentials we have used again Matsui's approach combined with the following standard assumptions:

1. S-boxes have no effect on the probability, i.e. they cannot change active byte into non-active and vice versa;
2. XOR can cancel two active bytes with probability 2^{-8};
3. The matrix multiplication can produce output column with t active bytes with probability $2^{-8(8-t)}$, unless there is only one single active input byte – then the probability is 1

We have implemented a brute force on the space of all possible truncated differentials and our findings are presented in Table 3.2. Based on the results, we can conclude that:

Observation 3 *For Camellia, no truncated differentials exist on more than 7 rounds.*

Note that the truncated differentials presented in the table have exactly specified positions of active bytes. By relaxing some of the positions, it might be possible to construct differentials for higher number of rounds. However, we expect that if such differentials are achievable, then the number of rounds they cover should not be significantly higher than 7. Taking into account the high security margin of *Camellia*, we can conclude that the full round cipher is resistant against truncated differential attacks.

Table 3.2: The probabilities of the best round-reduced truncated differentials for *Camellia*.

Rounds	$-log_2$ probability
1	0
2	24
3	48
4	64
5	80
6	104
7	120
8	136

3.4 Slide Attacks

Slide attacks are applicable to ciphers that have similar rounds. This is not the case for *Camellia* due to the key schedule – each round of the Feistel in the key schedule uses different 64-bit constant. Hence, we can conclude that *Camellia* is resistant against single-key and related-key slide attacks.

3.5 Rotational Attacks

So far rotational attacks have been applied only to addition-rotation-XOR primitives. To apply this type of attacks to substitution-permutations ciphers with byte-oriented structure the rotational input pairs have to differ by multiple of 8. Though *Camellia* satisfies this requirement, it uses high number of round constants in the key schedule that are not rotational. Therefore we believe rotational attacks cannot be applied to *Camellia*.

Chapter 4

Conclusion

The analysis presented in chapters 2 and 3 allows us to deduce a few conclusions regarding the security of *Camellia*.

- **Single-key differentials.** Both standard differential trails and truncated trails cover only around one half of the total number of rounds. Even with some advanced techniques, when the attacker can pass a few more rounds, the security margin of *Camellia* is high, hence the cipher is resistant against attacks based on differential trails. We note that *Camellia* applies standard cryptographic design techniques, and as there is no known analysis on such ciphers showing a significant advantage of differential attacks over the attacks based on differential trails, we believe that *Camellia* is secure against differential attacks as well. A similar conclusion applies to the case of boomerang attacks as they are differential-based attacks.

- **Related-key differentials.** Our analysis shows that finding the best related-key trails is infeasible due to the rotations in the round keys, hence we cannot give a precise bound on the related-key trails. We do note that good differentials only for the key schedule of *Camellia* exist, however it is unclear if they can be combined with differentials in the state to result in a related-key attack. The complementation property of Chapter 2 is a related-key differential, however it applies only to a modified version of *Camellia*, and only in the hash functions setting.

- **Other attacks.** *Camellia* is resistant against slide attacks and rotational attacks, as this type of cryptanalysis in the best scenario is applicable to the cipher with a few rounds only.

Bibliography

[1] D. Bai and L. Li. New impossible differential attacks on Camellia. *IACR Cryptology ePrint Archive*, 2011:661, 2011.

[2] D. Bai and L. Li. New impossible differential attacks on Camellia. In M. D. Ryan, B. Smyth, and G. Wang, editors, *ISPEC*, volume 7232 of *Lecture Notes in Computer Science*, pages 80–96. Springer, 2012.

[3] A. Biryukov, D. Khovratovich, and I. Nikolic. Distinguisher and related-key attack on the full AES-256. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 231–249. Springer, 2009.

[4] A. Biryukov and I. Nikolic. Automatic search for related-key differential characteristics in byte-oriented block ciphers: Application to AES, Camellia, Khazad and Others. In H. Gilbert, editor, *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 322–344. Springer, 2010.

[5] J. Chen, K. Jia, H. Yu, and X. Wang. New impossible differential attacks of reduced-round Camellia-192 and Camellia-256. In U. Parampalli and P. Hawkes, editors, *ACISP*, volume 6812 of *Lecture Notes in Computer Science*, pages 16–33. Springer, 2011.

[6] J. Chen and L. Li. Low data complexity attack on reduced Camellia-256. In Susilo et al. [29], pages 101–114.

[7] J. Daemen and V. Rijmen, editors. *Fast Software Encryption, 9th International Workshop, FSE 2002, Leuven, Belgium, February 4-6, 2002, Revised Papers*, volume 2365 of *Lecture Notes in Computer Science*. Springer, 2002.

[8] Y. Hatano, H. Sekine, and T. Kaneko. Higher order differential attack of Camellia (ii). In K. Nyberg and H. M. Heys, editors, *Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 129–146. Springer, 2002.

[9] Y. He and S. Qing. Square attack on reduced Camellia cipher. In Qing et al. [25], pages 238–245.

[10] D. Hong, B. Koo, and D.-C. Kim. Preimage and second-preimage attacks on pgv hashing modes of round-reduced ARIA, Camellia, and Serpent. *IEICE Transactions*, 95-A(1):372–380, 2012.

[11] G. Jie and Z. Zhongya. Improved collision attack on reduced round Camellia. In D. Pointcheval, Y. Mu, and K. Chen, editors, *CANS*, volume 4301 of *Lecture Notes in Computer Science*, pages 182–190. Springer, 2006.

[12] M. Kanda and T. Matsumoto. Security of Camellia against truncated differential cryptanalysis. In M. Matsui, editor, *FSE*, volume 2355 of *Lecture Notes in Computer Science*, pages 286–299. Springer, 2001.

[13] L. Keliher. Toward provable security against differential and linear cryptanalysis for camellia and related ciphers. *I. J. Network Security*, 5(2):167–175, 2007.

[14] S. Lee, S. Hong, S. Lee, J. Lim, and S. Yoon. Truncated differential cryptanalysis of Camellia. In K. Kim, editor, *ICISC*, volume 2288 of *Lecture Notes in Computer Science*, pages 32–38. Springer, 2001.

[15] D. Lei, C. Li, and K. Feng. New observation on Camellia. In B. Preneel and S. E. Tavares, editors, *Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 51–64. Springer, 2005.

[16] D. Lei, C. Li, and K. Feng. Square like attack on Camellia. In S. Qing, H. Imai, and G. Wang, editors, *ICICS*, volume 4861 of *Lecture Notes in Computer Science*, pages 269–283. Springer, 2007.

[17] L. Li, J. Chen, and K. Jia. New impossible differential cryptanalysis of reduced-round Camellia. In D. Lin, G. Tsudik, and X. Wang, editors, *CANS*, volume 7092 of *Lecture Notes in Computer Science*, pages 26–39. Springer, 2011.

[18] L. Li, J. Chen, and X. Wang. Multiplied conditional impossible differential attack on reduced-round camellia. *IACR Cryptology ePrint Archive*, 2011:524, 2011.

[19] Y. Li, W. Wu, L. Zhang, and L. Zhang. Improved integral attacks on reduced round Camellia. *IACR Cryptology ePrint Archive*, 2011:163, 2011.

[20] Y. Liu, D. Gu, Z. Liu, W. Li, and Y. Man. Improved results on impossible differential cryptanalysis of reduced-round Camellia-192/256. *IACR Cryptology ePrint Archive*, 2011:671, 2011.

[21] Y. Liu, L. Li, D. Gu, X. Wang, Z. Liu, J. Chen, and W. Li. New observations on impossible differential cryptanalysis of reduced-round Camellia. In A. Canteaut, editor, *FSE*, volume 7549 of *Lecture Notes in Computer Science*, pages 90–109. Springer, 2012.

[22] J. Lu, J. Kim, N. Keller, and O. Dunkelman. Improving the efficiency of impossible differential cryptanalysis of reduced 6Camellia and MISTY1. In T. Malkin, editor, *CT-RSA*, volume 4964 of *Lecture Notes in Computer Science*, pages 370–386. Springer, 2008.

[23] H. Mala, M. Shakiba, M. Dakhilalian, and G. Bagherikaram. New results on impossible differential cryptanalysis of reduced-round Camellia-128. In M. J. J. Jr., V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 281–294. Springer, 2009.

[24] M. Matsui. On correlation between the order of s-boxes and the strength of DES. In A. D. Santis, editor, *EUROCRYPT*, volume 950 of *Lecture Notes in Computer Science*, pages 366–375. Springer, 1994.

[25] S. Qing, T. Okamoto, and J. Zhou, editors. *Information and Communications Security, Third International Conference, ICICS 2001, Xian, China, November 13-16, 2001*, volume 2229 of *Lecture Notes in Computer Science*. Springer, 2001.

[26] Y. Sasaki, S. Emami, D. Hong, and A. Kumar. Improved known-key distinguishers on feistel-sp ciphers and application to Camellia. In Susilo et al. [29], pages 87–100.

[27] T. Shirai, S. Kanamaru, and G. Abe. Improved upper bounds of differential and linear characteristic probability for Camellia. In Daemen and Rijmen [7], pages 128–142.

[28] M. Sugita, K. Kobara, and H. Imai. Security of reduced version of the block cipher Camellia against truncated and impossible differential cryptanalysis. In C. Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 193–207. Springer, 2001.

[29] W. Susilo, Y. Mu, and J. Seberry, editors. *Information Security and Privacy - 17th Australasian Conference, ACISP 2012, Wollongong, NSW, Australia, July 9-11, 2012. Proceedings*, volume 7372 of *Lecture Notes in Computer Science*. Springer, 2012.

[30] W. Wu. Pseudorandomness of Camellia-like scheme. *J. Comput. Sci. Technol.*, 21(1):82–88, 2006.

[31] W. Wu and D. Feng. Collision attack on reduced-round Camellia. *Science in China Series F: Information Sciences*, 48(1):78–90, 2005.

[32] W. Wu, D. Feng, and H. Chen. Collision attack and pseudorandomness of reduced-round Camellia. In H. Handschuh and M. A. Hasan, editors, *Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 252–266. Springer, 2004.

[33] W. Wu, L. Zhang, and W. Zhang. Improved impossible differential cryptanalysis of reduced-round Camellia. In R. M. Avanzi, L. Keliher, and F. Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *Lecture Notes in Computer Science*, pages 442–456. Springer, 2008.

[34] W. Wu, W. Zhang, and D. Feng. Impossible differential cryptanalysis of reduced-round ARIA and Camellia. *J. Comput. Sci. Technol.*, 22(3):449–456, 2007.

[35] Y. Yeom, S. Park, and I. Kim. On the security of CAMELLIA against the square attack. In Daemen and Rijmen [7], pages 89–99.