Specification of *Camellia* — a 128-bit Block Cipher

Kazumaro AOKI †, Tetsuya ICHIKAWA ‡, Masayuki KANDA †,
Mitsuru MATSUI ‡, Shiho MORIAI †, Junko NAKAJIMA ‡, Toshio TOKITA ‡

†Nippon Telegraph and Telephone Corporation, ‡Mitsubishi Electric Corporation

March 10, 2000

Contents

1 Introduction

This document shows a complete description of the encryption algorithm *Camellia*, which is a secret key cipher with 128-bit data block and 128/192/256-bit secret key.

2 Notations and Conventions

2.1 Radix

We use the prefix **0x** to indicate **hexadecimal** numbers.

2.2 Notations

Throughout this document, the following notations are used.

- $\mathbf{B}$ denotes a vector space of 8-bit (byte) elements; that is, $\mathbf{B} := \mathrm{GF}(2)^8$.
- $\mathbf{W}$ denotes a vector space of 32-bit (word) elements; that is, $\mathbf{W} := \mathbf{B}^4$.
- $\mathbf{L}$ denotes a vector space of 64-bit (double word) elements; that is, $\mathbf{L} := \mathbf{B}^8$.
- $\mathbf{Q}$ denotes a vector space of 128-bit (quad word) elements; that is, $\mathbf{Q} := \mathbf{B}^{16}$.
- An element with the suffix $_{(n)}$ (e.g. $x_{(n)}$) shows that the element is n-bit long.
- An element with the suffix $_L$ (e.g. x_L) denotes left-half part of x.
- An element with the suffix $_R$ (e.g. x_R) denotes right-half part of x.

The suffix $_{(n)}$ will be omitted if no ambiguity is expected. See section 2.4 for numerical examples of "left" and "right".

2.3 List of Symbols

$\oplus$	The bitwise exclusive-OR operation.
$\|\|$	The concatenation of the two operands.
$\lll_n$	The left circular rotation of the operand by n bits.
$\cap$	The bitwise AND operation.
$\cup$	The bitwise OR operation.
$\overline{x}$	The bitwise complement of x.

2.4 Bit/Byte Ordering

We adopt big endian ordering. The following example shows how to compose a 128-bit value $Q_{(128)}$ of two 64-bit values $L_{i(64)}$ $(i = 1, 2)$, four 32-bit values $W_{i(32)}$ $(i = 1, 2, 3, 4)$, sixteen 8-bit values $B_{i(8)}$ $(i = 1, 2, \ldots, 16)$, or 128 1-bit values $E_{i(1)}$ $(i = 1, 2, \ldots, 128)$, respectively.

$$
\begin{aligned}
Q_{(128)} &= L_{1(64)}||L_{2(64)} \\
&= W_{1(32)}||W_{2(32)}||W_{3(32)}||W_{4(32)} \\
&= B_{1(8)}||B_{2(8)}||B_{3(8)}||B_{4(8)}||\ldots\ldots||B_{13(8)}||B_{14(8)}||B_{15(8)}||B_{16(8)} \\
&= E_{1(1)}||E_{2(1)}||E_{3(1)}||E_{4(1)}||\ldots\ldots\ldots\ldots\ldots\ldots||E_{125(1)}||E_{126(1)}||E_{127(1)}||E_{128(1)}
\end{aligned}
$$

Numerical examples:

$$
\begin{array}{lclclclcl}
Q_{(128)} & = & \multicolumn{7}{l}{\texttt{0x0123456789ABCDEF0011223344556677}_{(128)}} \\
L_{1(64)} & = & Q_{L(64)} & = & \multicolumn{5}{l}{\texttt{0x0123456789ABCDEF}_{(64)}} \\
L_{2(64)} & = & Q_{R(64)} & = & \multicolumn{5}{l}{\texttt{0x0011223344556677}_{(64)}} \\
W_{1(32)} & = & L_{1L(32)} & = & \multicolumn{5}{l}{\texttt{0x01234567}_{(32)}} \\
W_{2(32)} & = & L_{1R(32)} & = & \multicolumn{5}{l}{\texttt{0x89ABCDEF}_{(32)}} \\
W_{3(32)} & = & L_{2L(32)} & = & \multicolumn{5}{l}{\texttt{0x00112233}_{(32)}} \\
W_{4(32)} & = & L_{2R(32)} & = & \multicolumn{5}{l}{\texttt{0x44556677}_{(32)}} \\
B_{1(8)} & = & \texttt{0x01}_{(8)}, & B_{2(8)} & = & \texttt{0x23}_{(8)}, & B_{3(8)} & = & \texttt{0x45}_{(8)}, \quad B_{4(8)} = \texttt{0x67}_{(8)}, \\
B_{5(8)} & = & \texttt{0x89}_{(8)}, & B_{6(8)} & = & \texttt{0xAB}_{(8)}, & B_{7(8)} & = & \texttt{0xCD}_{(8)}, \quad B_{8(8)} = \texttt{0xEF}_{(8)}, \\
B_{9(8)} & = & \texttt{0x00}_{(8)}, & B_{10(8)} & = & \texttt{0x11}_{(8)}, & B_{11(8)} & = & \texttt{0x22}_{(8)}, \quad B_{12(8)} = \texttt{0x33}_{(8)}, \\
B_{13(8)} & = & \texttt{0x44}_{(8)}, & B_{14(8)} & = & \texttt{0x55}_{(8)}, & B_{15(8)} & = & \texttt{0x66}_{(8)}, \quad B_{16(8)} = \texttt{0x77}_{(8)}, \\
E_{1(1)} & = & \texttt{0}_{(1)}, & E_{2(1)} & = & \texttt{0}_{(1)}, & E_{3(1)} & = & \texttt{0}_{(1)}, \quad E_{4(1)} = \texttt{0}_{(1)}, \\
E_{5(1)} & = & \texttt{0}_{(1)}, & E_{6(1)} & = & \texttt{0}_{(1)}, & E_{7(1)} & = & \texttt{0}_{(1)}, \quad E_{8(1)} = \texttt{1}_{(1)}, \\
 & \vdots & & & & & & & \\
E_{121(1)} & = & \texttt{0}_{(1)}, & E_{122(1)} & = & \texttt{1}_{(1)}, & E_{123(1)} & = & \texttt{1}_{(1)}, \quad E_{124(1)} = \texttt{1}_{(1)}, \\
E_{125(1)} & = & \texttt{0}_{(1)}, & E_{126(1)} & = & \texttt{1}_{(1)}, & E_{127(1)} & = & \texttt{1}_{(1)}, \quad E_{128(1)} = \texttt{1}_{(1)}.
\end{array}
$$

$$
\begin{aligned}
Q_{(128)} \lll_1 &= E_{2(1)}||E_{3(1)}||E_{4(1)}||E_{5(1)}||\ldots\ldots\ldots\ldots\ldots\ldots||E_{125(1)}||E_{126(1)}||E_{127(1)}||E_{128(1)}||E_{1(1)} \\
&= \texttt{0x02468ACF13579BDE0022446688AACCEE}_{(128)}
\end{aligned}
$$

3 Structure of *Camellia*

3.1 List of Functions and Variables

$M_{(128)}$	The plaintext block.
$C_{(128)}$	The ciphertext block.
K	The secret key, whose length is 128, 192, or 256 bits.
$kw_{t(64)}, k_{u(64)}, kl_{v(64)}$	The subkeys. $(t = 1, 2, 3, 4)$ $(u = 1, 2, \ldots, 18)$ $(v = 1, 2, 3, 4)$ for 128-bit secret key. $(t = 1, 2, 3, 4)$ $(u = 1, 2, \ldots, 24)$ $(v = 1, 2, \ldots, 6)$ for 192-bit and 256-bit secret key.
$Y_{(64)} = F(X_{(64)}, k_{(64)})$	The F-function that transforms a 64-bit input $X_{(64)}$ to a 64-bit output $Y_{(64)}$ using a 64-bit subkey $k_{(64)}$.
$Y_{(64)} = FL(X_{(64)}, k_{(64)})$	The FL-function that transforms a 64-bit input $X_{(64)}$ to a 64-bit output $Y_{(64)}$ using a 64-bit subkey $k_{(64)}$.
$Y_{(64)} = FL^{-1}(X_{(64)}, k_{(64)})$	The FL^{-1}-function that transforms a 64-bit input $X_{(64)}$ to a 64-bit output $Y_{(64)}$ using a 64-bit subkey $k_{(64)}$.
$Y_{(64)} = S(X_{(64)})$	The S-function that transforms a 64-bit input $X_{(64)}$ to a 64-bit output $Y_{(64)}$.
$Y_{(64)} = P(X_{(64)})$	The P-function that transforms a 64-bit input $X_{(64)}$ to a 64-bit output $Y_{(64)}$.
$y_{(8)} = s_i(x_{(8)})$	The s-boxes that transform an 8-bit input to an 8-bit output $(i = 1, 2, 3, 4)$.

3.2 Encryption Procedure

3.2.1 128-bit key

Figure 1 shows the encryption procedure for a 128-bit key. The data randomizing part has an 18-round Feistel structure with two FL/FL^{-1}-function layers after the 6-th and 12-th rounds, and 128-bit XOR operations before the first round and after the last round. The key schedule part generates subkeys $kw_{t(64)}$ $(t = 1, 2, 3, 4)$, $k_{u(64)}$ $(u = 1, 2, \ldots, 18)$ and $kl_{v(64)}$ $(v = 1, 2, 3, 4)$ from the secret key K; see section 3.4 for details of the key schedule part.

In the data randomizing part, first the plaintext $M_{(128)}$ is XORed with $kw_{1(64)}||kw_{2(64)}$ and separated into $L_{0(64)}$ and $R_{0(64)}$ of equal length, i.e., $M_{(128)} \oplus (kw_{1(64)}||kw_{2(64)}) = L_{0(64)}||R_{0(64)}$. Then, the following operations are perfomed from $r = 1$ to 18, except for $r = 6$ and 12;

$$
\begin{aligned}
L_r &= R_{r-1} \oplus F(L_{r-1}, k_r), \\
R_r &= L_{r-1}.
\end{aligned}
$$

For $r = 6$ and 12, the following is carried out;

$$\begin{aligned} L'_r &= R_{r-1} \oplus F(L_{r-1}, k_r), \\ R'_r &= L_{r-1}, \\ L_r &= FL(L'_r, kl_{2r/6-1}), \\ R_r &= FL^{-1}(R'_r, kl_{2r/6}). \end{aligned}$$

Lastly, $R_{18(64)}$ and $L_{18(64)}$ are concatenated and XORed with $kw_{3(64)}||kw_{4(64)}$. The resultant value is the ciphertext, i.e., $C_{(128)} = (R_{18(64)}||L_{18(64)}) \oplus (kw_{3(64)}||kw_{4(64)})$.

3.2.2 192-bit and 256-bit key

Figure 2 shows the encryption procedure for a 192-bit or 256-bit key. The data randomizing part has a 24-round Feistel structure with three FL/FL^{-1}-function layers after the 6-th, 12-th, and 18-th rounds, and 128-bit XOR operations before the first round and after the last round. The key schedule part generates subkeys $kw_{t(64)}$ $(t = 1, 2, 3, 4)$, $k_{u(64)}$ $(u = 1, 2, \ldots, 24)$, and $kl_{v(64)}$ $(v = 1, 2, \ldots, 6)$ from the secret key K.

In the data randomizing part, first the plaintext $M_{(128)}$ is XORed with $kw_{1(64)}||kw_{2(64)}$ and separated into $L_{0(64)}$ and $R_{0(64)}$ of equal length, i.e., $M_{(128)} \oplus (kw_{1(64)}||kw_{2(64)}) = L_{0(64)}||R_{0(64)}$. Then, perform the following operations from $r = 1$ to 24, except for $r = 6$, 12, and 18;

$$\begin{aligned} L_r &= R_{r-1} \oplus F(L_{r-1}, k_r), \\ R_r &= L_{r-1}. \end{aligned}$$

For $r = 6$, 12, and 18, perform the following;

$$\begin{aligned} L'_r &= R_{r-1} \oplus F(L_{r-1}, k_r), \\ R'_r &= L_{r-1}, \\ L_r &= FL(L'_r, kl_{2r/6-1}), \\ R_r &= FL^{-1}(R'_r, kl_{2r/6}). \end{aligned}$$

Lastly, $R_{24(64)}$ and $L_{24(64)}$ are concatenated and XORed with $kw_{3(64)}||kw_{4(64)}$. The resultant value is the ciphertext, i.e., $C_{(128)} = (R_{24(64)}||L_{24(64)}) \oplus (kw_{3(64)}||kw_{4(64)})$.

See section 4 for details of the F-function and FL/FL^{-1}-functions.

3.3 Decryption Procedure

3.3.1 128-bit key

The decryption procedure of *Camellia* can be done in the same way as the encryption procedure by reversing the order of the subkeys.

Figure 3 shows the decryption procedure for a 128-bit key. The data randomizing part has an 18-round Feistel structure with two FL/FL^{-1}-function layers after the 6-th and 12-th rounds, and 128-bit XOR operations before the first round and after the last round. The key schedule part generates subkeys $kw_{t(64)}$ $(t = 1, 2, 3, 4)$, $k_{u(64)}$ $(u = 1, 2, \ldots, 18)$, and $kl_{v(64)}$ $(v = 1, 2, 3, 4)$ from the secret key K; see section 3.4 for details of the key schedule part.

In the data randomizing part, first the ciphertext $C_{(128)}$ is XORed with $kw_{3(64)}||kw_{4(64)}$ and separated into $R_{18(64)}$ and $L_{18(64)}$ of equal length, i.e., $C_{(128)} \oplus (kw_{3(64)}||kw_{4(64)}) = R_{18(64)}||L_{18(64)}$. Then, the following operations are perfomed from $r = 18$ down to 1, except for $r = 13$ and 7;

$$\begin{aligned} R_{r-1} &= L_r \oplus F(R_r, k_r), \\ L_{r-1} &= R_r. \end{aligned}$$

For $r = 13$ and 7, the following is carried out;

$$\begin{aligned} R'_{r-1} &= L_r \oplus F(R_r, k_r), \\ L'_{r-1} &= R_r. \\ R_{r-1} &= FL(R'_{r-1}, kl_{2(r-1)/6}), \\ L_{r-1} &= FL^{-1}(L'_{r-1}, kl_{2(r-1)/6-1}). \end{aligned}$$

Lastly, $L_{0(64)}$ and $R_{0(64)}$ are concatenated and XORed with $kw_{1(64)}||kw_{2(64)}$. The resultant value is the plaintext, i.e., $M_{(128)} = (L_{0(64)}||R_{0(64)}) \oplus (kw_{1(64)}||kw_{2(64)})$.

3.3.2 192-bit and 256-bit key

Figure 4 shows the decryption procedure for a 192-bit or 256-bit key. The data randomizing part has a 24-round Feistel structure with three FL/FL^{-1}-function layers after the 6-th, 12-th, and 18-th rounds, and 128-bit XOR operations before the first round and after the last round. The key schedule part generates subkeys $kw_{t(64)}$ $(t = 1, 2, 3, 4)$, $k_{u(64)}$ $(u = 1, 2, \ldots, 24)$, and $kl_{v(64)}$ $(v = 1, 2, \ldots, 6)$ from the secret key K.

In the data randomizing part, first the ciphertext $C_{(128)}$ is XORed with $kw_{3(64)}||kw_{4(64)}$ and separated into $R_{24(64)}$ and $L_{24(64)}$ of equal length, i.e., $C_{(128)} \oplus (kw_{3(64)}||kw_{4(64)}) = R_{24(64)}||L_{24(64)}$. Then, perform the following operations from $r = 24$ down to 1, except for $r = 19$, 13, and 7;

$$\begin{aligned} R_{r-1} &= L_r \oplus F(R_r, k_r), \\ L_{r-1} &= R_r. \end{aligned}$$

For $r = 19$, 13, and 7, perform the following.

$$\begin{aligned} R'_{r-1} &= L_r \oplus F(R_r, k_r), \\ L'_{r-1} &= R_r. \\ R_{r-1} &= FL(R'_{r-1}, kl_{2(r-1)/6}), \\ L_{r-1} &= FL^{-1}(L'_{r-1}, kl_{2(r-1)/6-1}). \end{aligned}$$

Lastly, $L_{0(64)}$ and $R_{0(64)}$ are concatenated and XORed with $kw_{1(64)}||kw_{2(64)}$. The resultant value is the plaintext, i.e., $M_{(128)} = (L_{0(64)}||R_{0(64)}) \oplus (kw_{1(64)}||kw_{2(64)})$.

3.4 Key Schedule

In the key schedule part of *Camellia*, we introduce two 128-bit variables $K_{L(128)}$, $K_{R(128)}$ and four 64-bit variables $K_{LL(64)}$, $K_{LR(64)}$, $K_{RL(64)}$ and $K_{RR(64)}$, which are defined so that the following relations are satisfied:

$$
\begin{array}{lllllll}
K_{(128)} & = & K_{L(128)}, & K_{R(128)} & = & 0; & \text{for 128-bit key,} \\
K_{(192)} & = & K_{L(128)}||K_{RL(64)}, & K_{RR(64)} & = & \overline{K_{RL(64)}}; & \text{for 192-bit key,} \\
K_{(256)} & = & K_{L(128)}||K_{R(128)}; & & & & \text{for 256-bit key.}
\end{array}
$$

$$
\left.\begin{array}{lll}
K_{L(128)} & = & K_{LL(64)}||K_{LR(64)}, \\
K_{R(128)} & = & K_{RL(64)}||K_{RR(64)};
\end{array}\right. \quad \text{for any size of key.}
$$

Using these variables, we generate two 128-bit variables $K_{A(128)}$ and $K_{B(128)}$, as shown in figure 8, where $K_{B(128)}$ is used only if the length of the secret key is 192 or 256 bits. First $K = K_{L(128)}$ is XORed with $K_{R(128)}$ and "encrypted" by two rounds using the constant values $\Sigma_{1(64)}$ and $\Sigma_{2(64)}$ as "keys". The result is XORed with $K_{L(128)}$ and again encrypted by two rounds using the constant values $\Sigma_{3(64)}$ and $\Sigma_{4(64)}$; the resultant value is $K_{A(128)}$. Lastly $K_{A(128)}$ is XORed with $K_{R(128)}$ and encrypted by two rounds using the constant values $\Sigma_{5(64)}$ and $\Sigma_{6(64)}$; the resultant value is $K_{B(128)}$. Σ_i is defined as the continuous values from the second hexadecimal place to the seventeenth hexadecimal place of the hexadecimal representation of the square root of the i-th prime. These constant values are listed in table 1.

The subkeys $kw_{t(64)}, k_{u(64)}$, and $kl_{v(64)}$ are generated from (left-half or right-half part of) rotate shifted values of $K_{L(128)}$, $K_{R(128)}$, $K_{A(128)}$, and $K_{B(128)}$. The exact details are shown in table 2 and table 3, respectively.

Table 1: The key schedule constants

$\Sigma_{1(64)}$	0xA09E667F3BCC908B
$\Sigma_{2(64)}$	0xB67AE8584CAA73B2
$\Sigma_{3(64)}$	0xC6EF372FE94F82BE
$\Sigma_{4(64)}$	0x54FF53A5F1D36F1C
$\Sigma_{5(64)}$	0x10E527FADE682D1D
$\Sigma_{6(64)}$	0xB05688C2B3E6C1FD

Table 2: Subkeys for 128-bit secret key

	subkey	value
Prewhitening	$kw_{1(64)}$	$(K_L \lll_0)_{L(64)}$
	$kw_{2(64)}$	$(K_L \lll_0)_{R(64)}$
F (Round1)	$k_{1(64)}$	$(K_A \lll_0)_{L(64)}$
F (Round2)	$k_{2(64)}$	$(K_A \lll_0)_{R(64)}$
F (Round3)	$k_{3(64)}$	$(K_L \lll_{15})_{L(64)}$
F (Round4)	$k_{4(64)}$	$(K_L \lll_{15})_{R(64)}$
F (Round5)	$k_{5(64)}$	$(K_A \lll_{15})_{L(64)}$
F (Round6)	$k_{6(64)}$	$(K_A \lll_{15})_{R(64)}$
FL	$kl_{1(64)}$	$(K_A \lll_{30})_{L(64)}$
FL^{-1}	$kl_{2(64)}$	$(K_A \lll_{30})_{R(64)}$
F (Round7)	$k_{7(64)}$	$(K_L \lll_{45})_{L(64)}$
F (Round8)	$k_{8(64)}$	$(K_L \lll_{45})_{R(64)}$
F (Round9)	$k_{9(64)}$	$(K_A \lll_{45})_{L(64)}$
F (Round10)	$k_{10(64)}$	$(K_L \lll_{60})_{R(64)}$
F (Round11)	$k_{11(64)}$	$(K_A \lll_{60})_{L(64)}$
F (Round12)	$k_{12(64)}$	$(K_A \lll_{60})_{R(64)}$
FL	$kl_{3(64)}$	$(K_L \lll_{77})_{L(64)}$
FL^{-1}	$kl_{4(64)}$	$(K_L \lll_{77})_{R(64)}$
F (Round13)	$k_{13(64)}$	$(K_L \lll_{94})_{L(64)}$
F (Round14)	$k_{14(64)}$	$(K_L \lll_{94})_{R(64)}$
F (Round15)	$k_{15(64)}$	$(K_A \lll_{94})_{L(64)}$
F (Round16)	$k_{16(64)}$	$(K_A \lll_{94})_{R(64)}$
F (Round17)	$k_{17(64)}$	$(K_L \lll_{111})_{L(64)}$
F (Round18)	$k_{18(64)}$	$(K_L \lll_{111})_{R(64)}$
Postwhitening	$kw_{3(64)}$	$(K_A \lll_{111})_{L(64)}$
	$kw_{4(64)}$	$(K_A \lll_{111})_{R(64)}$

Table 3: Subkeys for 192/256-bit secret key

	subkey	value
Prewhitening	$kw_{1(64)}$	$(K_L \lll_0)_{L(64)}$
	$kw_{2(64)}$	$(K_L \lll_0)_{R(64)}$
F (Round1)	$k_{1(64)}$	$(K_B \lll_0)_{L(64)}$
F (Round2)	$k_{2(64)}$	$(K_B \lll_0)_{R(64)}$
F (Round3)	$k_{3(64)}$	$(K_R \lll_{15})_{L(64)}$
F (Round4)	$k_{4(64)}$	$(K_R \lll_{15})_{R(64)}$
F (Round5)	$k_{5(64)}$	$(K_A \lll_{15})_{L(64)}$
F (Round6)	$k_{6(64)}$	$(K_A \lll_{15})_{R(64)}$
FL	$kl_{1(64)}$	$(K_R \lll_{30})_{L(64)}$
FL^{-1}	$kl_{2(64)}$	$(K_R \lll_{30})_{R(64)}$
F (Round7)	$k_{7(64)}$	$(K_B \lll_{30})_{L(64)}$
F (Round8)	$k_{8(64)}$	$(K_B \lll_{30})_{R(64)}$
F (Round9)	$k_{9(64)}$	$(K_L \lll_{45})_{L(64)}$
F (Round10)	$k_{10(64)}$	$(K_L \lll_{45})_{R(64)}$
F (Round11)	$k_{11(64)}$	$(K_A \lll_{45})_{L(64)}$
F (Round12)	$k_{12(64)}$	$(K_A \lll_{45})_{R(64)}$
FL	$kl_{3(64)}$	$(K_L \lll_{60})_{L(64)}$
FL^{-1}	$kl_{4(64)}$	$(K_L \lll_{60})_{R(64)}$
F (Round13)	$k_{13(64)}$	$(K_R \lll_{60})_{L(64)}$
F (Round14)	$k_{14(64)}$	$(K_R \lll_{60})_{R(64)}$
F (Round15)	$k_{15(64)}$	$(K_B \lll_{60})_{L(64)}$
F (Round16)	$k_{16(64)}$	$(K_B \lll_{60})_{R(64)}$
F (Round17)	$k_{17(64)}$	$(K_L \lll_{77})_{L(64)}$
F (Round18)	$k_{18(64)}$	$(K_L \lll_{77})_{R(64)}$
FL	$kl_{5(64)}$	$(K_A \lll_{77})_{L(64)}$
FL^{-1}	$kl_{6(64)}$	$(K_A \lll_{77})_{R(64)}$
F (Round19)	$k_{19(64)}$	$(K_R \lll_{94})_{L(64)}$
F (Round20)	$k_{20(64)}$	$(K_R \lll_{94})_{R(64)}$
F (Round21)	$k_{21(64)}$	$(K_A \lll_{94})_{L(64)}$
F (Round22)	$k_{22(64)}$	$(K_A \lll_{94})_{R(64)}$
F (Round23)	$k_{23(64)}$	$(K_L \lll_{111})_{L(64)}$
F (Round24)	$k_{24(64)}$	$(K_L \lll_{111})_{R(64)}$
Postwhitening	$kw_{3(64)}$	$(K_B \lll_{111})_{L(64)}$
	$kw_{4(64)}$	$(K_B \lll_{111})_{R(64)}$

4 Components of *Camellia*

4.1 *F*-function

The F-function is shown in figure 5, which is defined as follows:

$$\begin{array}{rcl} F : \mathbf{L} \times \mathbf{L} & \longrightarrow & \mathbf{L} \\ (X_{(64)}, k_{(64)}) & \longmapsto & Y_{(64)} = P(S(X_{(64)} \oplus k_{(64)})). \end{array}$$

See sections 4.4 and 4.6 for the S-function and the P-function, respectively.

4.2 *FL*-function

The FL-function is shown in figure 6, which is defined as follows:

$$\begin{array}{rcl} FL : \mathbf{L} \times \mathbf{L} & \longrightarrow & \mathbf{L} \\ (X_{L(32)} || X_{R(32)}, kl_{L(32)} || kl_{R(32)}) & \longmapsto & Y_{L(32)} || Y_{R(32)}, \end{array}$$

where

$$\begin{array}{rcl} Y_{R(32)} & = & ((X_{L(32)} \cap kl_{L(32)}) \lll_1) \oplus X_{R(32)}, \\ Y_{L(32)} & = & (Y_{R(32)} \cup kl_{R(32)}) \oplus X_{L(32)}. \end{array}$$

4.3 *FL*$^{-1}$-function

The FL^{-1}-function is shown in figure 7, which is defined as follows:

$$\begin{array}{rcl} FL^{-1} : \mathbf{L} \times \mathbf{L} & \longrightarrow & \mathbf{L} \\ (Y_{L(32)} || Y_{R(32)}, kl_{L(32)} || kl_{R(32)}) & \longmapsto & X_{L(32)} || X_{R(32)}, \end{array}$$

where

$$\begin{array}{rcl} X_{L(32)} & = & (Y_{R(32)} \cup kl_{R(32)}) \oplus Y_{L(32)}, \\ X_{R(32)} & = & ((X_{L(32)} \cap kl_{L(32)}) \lll_1) \oplus Y_{R(32)}. \end{array}$$

4.4 *S*-function

The S-function is a part of F-function, which is defined as follows:

$$\begin{array}{rcl} S : \mathbf{L} & \longrightarrow & \mathbf{L} \\ l_{1(8)} || l_{2(8)} || l_{3(8)} || l_{4(8)} || l_{5(8)} || l_{6(8)} || l_{7(8)} || l_{8(8)} & \longmapsto & l'_{1(8)} || l'_{2(8)} || l'_{3(8)} || l'_{4(8)} || l'_{5(8)} || l'_{6(8)} || l'_{7(8)} || l'_{8(8)} \end{array}$$

$$\begin{aligned}
l'_{1(8)} &= s_1(l_{1(8)}),\\
l'_{2(8)} &= s_2(l_{2(8)}),\\
l'_{3(8)} &= s_3(l_{3(8)}),\\
l'_{4(8)} &= s_4(l_{4(8)}),\\
l'_{5(8)} &= s_2(l_{5(8)}),\\
l'_{6(8)} &= s_3(l_{6(8)}),\\
l'_{7(8)} &= s_4(l_{7(8)}),\\
l'_{8(8)} &= s_1(l_{8(8)}),
\end{aligned}$$

where the four s-boxes, s_1, s_2, s_3, and s_4, are described in section 4.5.

4.5 s-boxes

The four s-boxes of *Camellia* are affine equivalent to an inversion function over GF(2^8), which are shown in tables 4, 5, 6, and 7. An algebraic representation of the s-boxes is shown below:

$$\begin{aligned}
s_1 &: \mathbf{B} \longrightarrow \mathbf{B}\\
x_{(8)} &\longmapsto \mathbf{h}(\mathbf{g}(\mathbf{f}(\mathtt{0xc5} \oplus x_{(8)}))) \oplus \mathtt{0x6e},\\
s_2 &: \mathbf{B} \longrightarrow \mathbf{B}\\
x_{(8)} &\longmapsto s_1(x_{(8)}) \lll_1,\\
s_3 &: \mathbf{B} \longrightarrow \mathbf{B}\\
x_{(8)} &\longmapsto s_1(x_{(8)}) \ggg_1,\\
s_4 &: \mathbf{B} \longrightarrow \mathbf{B}\\
x_{(8)} &\longmapsto s_1(x_{(8)} \lll_1),
\end{aligned}$$

where the functions $\mathbf{f}, \mathbf{g}$, and $\mathbf{h}$ are given as follows:

$$\begin{aligned}
&\mathbf{f} : \mathbf{B} \longrightarrow \mathbf{B}\\
&a_{1(1)}||a_{2(1)}||a_{3(1)}||a_{4(1)}||a_{5(1)}||a_{6(1)}||a_{7(1)}||a_{8(1)}\\
&\longmapsto b_{1(1)}||b_{2(1)}||b_{3(1)}||b_{4(1)}||b_{5(1)}||b_{6(1)}||b_{7(1)}||b_{8(1)},
\end{aligned}$$

where

$$\begin{aligned}
b_1 &= a_6 \oplus a_2,\\
b_2 &= a_7 \oplus a_1,\\
b_3 &= a_8 \oplus a_5 \oplus a_3,\\
b_4 &= a_8 \oplus a_3,\\
b_5 &= a_7 \oplus a_4,\\
b_6 &= a_5 \oplus a_2,\\
b_7 &= a_8 \oplus a_1,\\
b_8 &= a_6 \oplus a_4.
\end{aligned}$$

$$
\begin{aligned}
&\mathbf{g} : \mathbf{B} \longrightarrow \mathbf{B} \\
&a_{1(1)}||a_{2(1)}||a_{3(1)}||a_{4(1)}||a_{5(1)}||a_{6(1)}||a_{7(1)}||a_{8(1)} \\
&\longmapsto b_{1(1)}||b_{2(1)}||b_{3(1)}||b_{4(1)}||b_{5(1)}||b_{6(1)}||b_{7(1)}||b_{8(1)},
\end{aligned}
$$

where

$$
\begin{aligned}
&(b_8 + b_7\alpha + b_6\alpha^2 + b_5\alpha^3) + (b_4 + b_3\alpha + b_2\alpha^2 + b_1\alpha^3)\beta \\
&= 1/((a_8 + a_7\alpha + a_6\alpha^2 + a_5\alpha^3) + (a_4 + a_3\alpha + a_2\alpha^2 + a_1\alpha^3)\beta).
\end{aligned}
$$

This inversion is performed in GF(2^8) assuming $\frac{1}{0} = 0$, where β is an element in GF(2^8) that satisfies $\beta^8 + \beta^6 + \beta^5 + \beta^3 + 1 = 0$, and $\alpha = \beta^{238} = \beta^6 + \beta^5 + \beta^3 + \beta^2$ is an element in GF(2^4) that satisfies $\alpha^4 + \alpha + 1 = 0$.

$$
\begin{aligned}
&\mathbf{h} : \mathbf{B} \longrightarrow \mathbf{B} \\
&a_{1(1)}||a_{2(1)}||a_{3(1)}||a_{4(1)}||a_{5(1)}||a_{6(1)}||a_{7(1)}||a_{8(1)} \\
&\longmapsto b_{1(1)}||b_{2(1)}||b_{3(1)}||b_{4(1)}||b_{5(1)}||b_{6(1)}||b_{7(1)}||b_{8(1)},
\end{aligned}
$$

where

$$
\begin{aligned}
b_1 &= a_5 \oplus a_6 \oplus a_2, \\
b_2 &= a_6 \oplus a_2, \\
b_3 &= a_7 \oplus a_4, \\
b_4 &= a_8 \oplus a_2, \\
b_5 &= a_7 \oplus a_3, \\
b_6 &= a_8 \oplus a_1, \\
b_7 &= a_5 \oplus a_1, \\
b_8 &= a_6 \oplus a_3.
\end{aligned}
$$

Table 4: The s-box s_1

This table below reads $s_1(\mathtt{0}) = \mathtt{112}, s_1(\mathtt{1}) = \mathtt{130}, \ldots, s_1(\mathtt{255}) = \mathtt{158}$.

112	130	44	236	179	39	192	229	228	133	87	53	234	12	174	65
35	239	107	147	69	25	165	33	237	14	79	78	29	101	146	189
134	184	175	143	124	235	31	206	62	48	220	95	94	197	11	26
166	225	57	202	213	71	93	61	217	1	90	214	81	86	108	77
139	13	154	102	251	204	176	45	116	18	43	32	240	177	132	153
223	76	203	194	52	126	118	5	109	183	169	49	209	23	4	215
20	88	58	97	222	27	17	28	50	15	156	22	83	24	242	34
254	68	207	178	195	181	122	145	36	8	232	168	96	252	105	80
170	208	160	125	161	137	98	151	84	91	30	149	224	255	100	210
16	196	0	72	163	247	117	219	138	3	230	218	9	63	221	148
135	92	131	2	205	74	144	51	115	103	246	243	157	127	191	226
82	155	216	38	200	55	198	59	129	150	111	75	19	190	99	46
233	121	167	140	159	110	188	142	41	245	249	182	47	253	180	89
120	152	6	106	231	70	113	186	212	37	171	66	136	162	141	250
114	7	185	85	248	238	172	10	54	73	42	104	60	56	241	164
64	40	211	123	187	201	67	193	21	227	173	244	119	199	128	158

Table 5: The s-box s_2

224	5	88	217	103	78	129	203	201	11	174	106	213	24	93	130
70	223	214	39	138	50	75	66	219	28	158	156	58	202	37	123
13	113	95	31	248	215	62	157	124	96	185	190	188	139	22	52
77	195	114	149	171	142	186	122	179	2	180	173	162	172	216	154
23	26	53	204	247	153	97	90	232	36	86	64	225	99	9	51
191	152	151	133	104	252	236	10	218	111	83	98	163	46	8	175
40	176	116	194	189	54	34	56	100	30	57	44	166	48	229	68
253	136	159	101	135	107	244	35	72	16	209	81	192	249	210	160
85	161	65	250	67	19	196	47	168	182	60	43	193	255	200	165
32	137	0	144	71	239	234	183	21	6	205	181	18	126	187	41
15	184	7	4	155	148	33	102	230	206	237	231	59	254	127	197
164	55	177	76	145	110	141	118	3	45	222	150	38	125	198	92
211	242	79	25	63	220	121	29	82	235	243	109	94	251	105	178
240	49	12	212	207	140	226	117	169	74	87	132	17	69	27	245
228	14	115	170	241	221	89	20	108	146	84	208	120	112	227	73
128	80	167	246	119	147	134	131	42	199	91	233	238	143	1	61

Table 6: The s-box s_3

56	65	22	118	217	147	96	242	114	194	171	154	117	6	87	160
145	247	181	201	162	140	210	144	246	7	167	39	142	178	73	222
67	92	215	199	62	245	143	103	31	24	110	175	47	226	133	13
83	240	156	101	234	163	174	158	236	128	45	107	168	43	54	166
197	134	77	51	253	102	88	150	58	9	149	16	120	216	66	204
239	38	229	97	26	63	59	130	182	219	212	152	232	139	2	235
10	44	29	176	111	141	136	14	25	135	78	11	169	12	121	17
127	34	231	89	225	218	61	200	18	4	116	84	48	126	180	40
85	104	80	190	208	196	49	203	42	173	15	202	112	255	50	105
8	98	0	36	209	251	186	237	69	129	115	109	132	159	238	74
195	46	193	1	230	37	72	153	185	179	123	249	206	191	223	113
41	205	108	19	100	155	99	157	192	75	183	165	137	95	177	23
244	188	211	70	207	55	94	71	148	250	252	91	151	254	90	172
60	76	3	53	243	35	184	93	106	146	213	33	68	81	198	125
57	131	220	170	124	119	86	5	27	164	21	52	30	28	248	82
32	20	233	189	221	228	161	224	138	241	214	122	187	227	64	79

Table 7: The s-box s_4

112	44	179	192	228	87	234	174	35	107	69	165	237	79	29	146
134	175	124	31	62	220	94	11	166	57	213	93	217	90	81	108
139	154	251	176	116	43	240	132	223	203	52	118	109	169	209	4
20	58	222	17	50	156	83	242	254	207	195	122	36	232	96	105
170	160	161	98	84	30	224	100	16	0	163	117	138	230	9	221
135	131	205	144	115	246	157	191	82	216	200	198	129	111	19	99
233	167	159	188	41	249	47	180	120	6	231	113	212	171	136	141
114	185	248	172	54	42	60	241	64	211	187	67	21	173	119	128
130	236	39	229	133	53	12	65	239	147	25	33	14	78	101	189
184	143	235	206	48	95	197	26	225	202	71	61	1	214	86	77
13	102	204	45	18	32	177	153	76	194	126	5	183	49	23	215
88	97	27	28	15	22	24	34	68	178	181	145	8	168	252	80
208	125	137	151	91	149	255	210	196	72	247	219	3	218	63	148
92	2	74	51	103	243	127	226	155	38	55	59	150	75	190	46
121	140	110	142	245	182	253	89	152	106	70	186	37	66	162	250
7	85	238	10	73	104	56	164	40	123	201	193	227	244	199	158

4.6 *P*-function

The P-function is a part of F-function, which is defined as follows:

$$\begin{aligned} P : \mathbf{L} &\longrightarrow \mathbf{L} \\ z_{1(8)}||z_{2(8)}||z_{3(8)}||z_{4(8)}||z_{5(8)}||z_{6(8)}||z_{7(8)}||z_{8(8)} &\longmapsto z'_{1(8)}||z'_{2(8)}||z'_{3(8)}||z'_{4(8)}||z'_{5(8)}||z'_{6(8)}||z'_{7(8)}||z'_{8(8)}, \end{aligned}$$

where

$$\begin{aligned} z'_1 &= z_1 \oplus z_3 \oplus z_4 \oplus z_6 \oplus z_7 \oplus z_8, \\ z'_2 &= z_1 \oplus z_2 \oplus z_4 \oplus z_5 \oplus z_7 \oplus z_8, \\ z'_3 &= z_1 \oplus z_2 \oplus z_3 \oplus z_5 \oplus z_6 \oplus z_8, \\ z'_4 &= z_2 \oplus z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7, \\ z'_5 &= z_1 \oplus z_2 \oplus z_6 \oplus z_7 \oplus z_8, \\ z'_6 &= z_2 \oplus z_3 \oplus z_5 \oplus z_7 \oplus z_8, \\ z'_7 &= z_3 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_8, \\ z'_8 &= z_1 \oplus z_4 \oplus z_5 \oplus z_6 \oplus z_7. \end{aligned}$$

Equivalently, this transformation can be given in the following form:

$$\begin{pmatrix} z_8 \\ z_7 \\ \vdots \\ z_1 \end{pmatrix} \longmapsto \begin{pmatrix} z'_8 \\ z'_7 \\ \vdots \\ z'_1 \end{pmatrix} = P \begin{pmatrix} z_8 \\ z_7 \\ \vdots \\ z_1 \end{pmatrix},$$

where

$$P = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

A Figures of the *Camellia* Algorithm

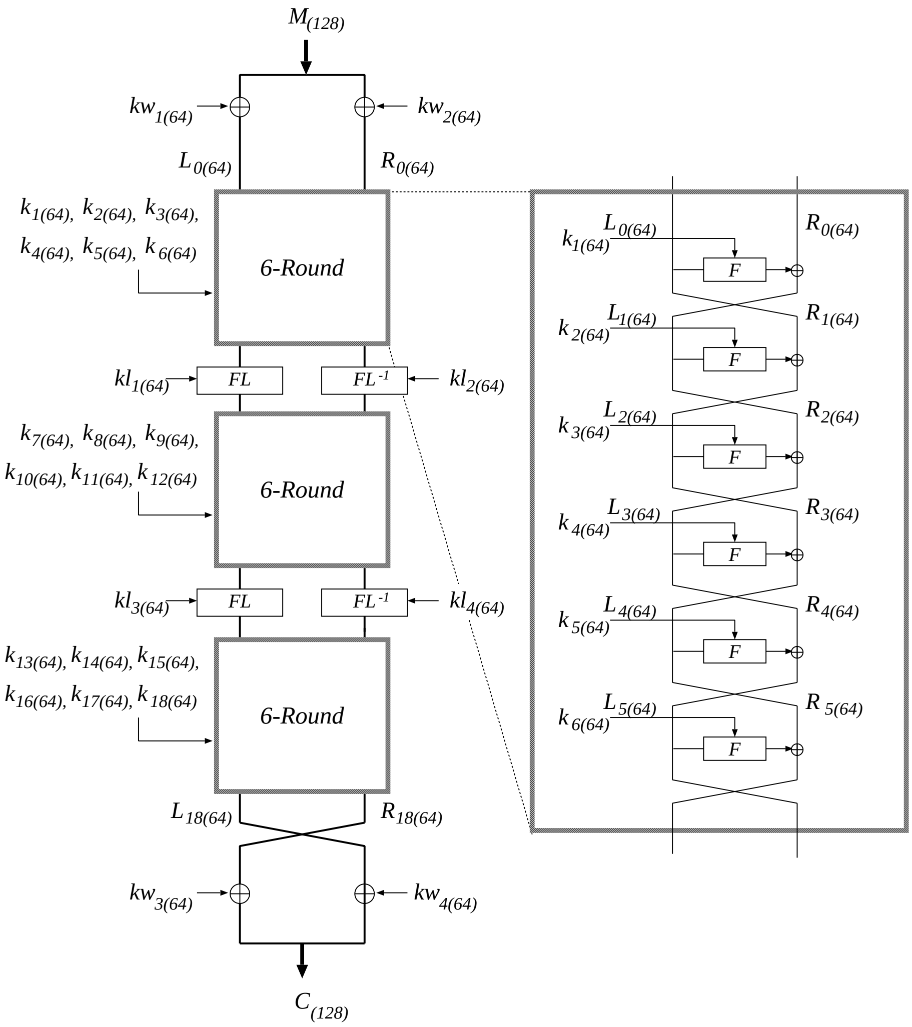

Figure 1: Encryption Procedure of *Camellia* for 128-bit key

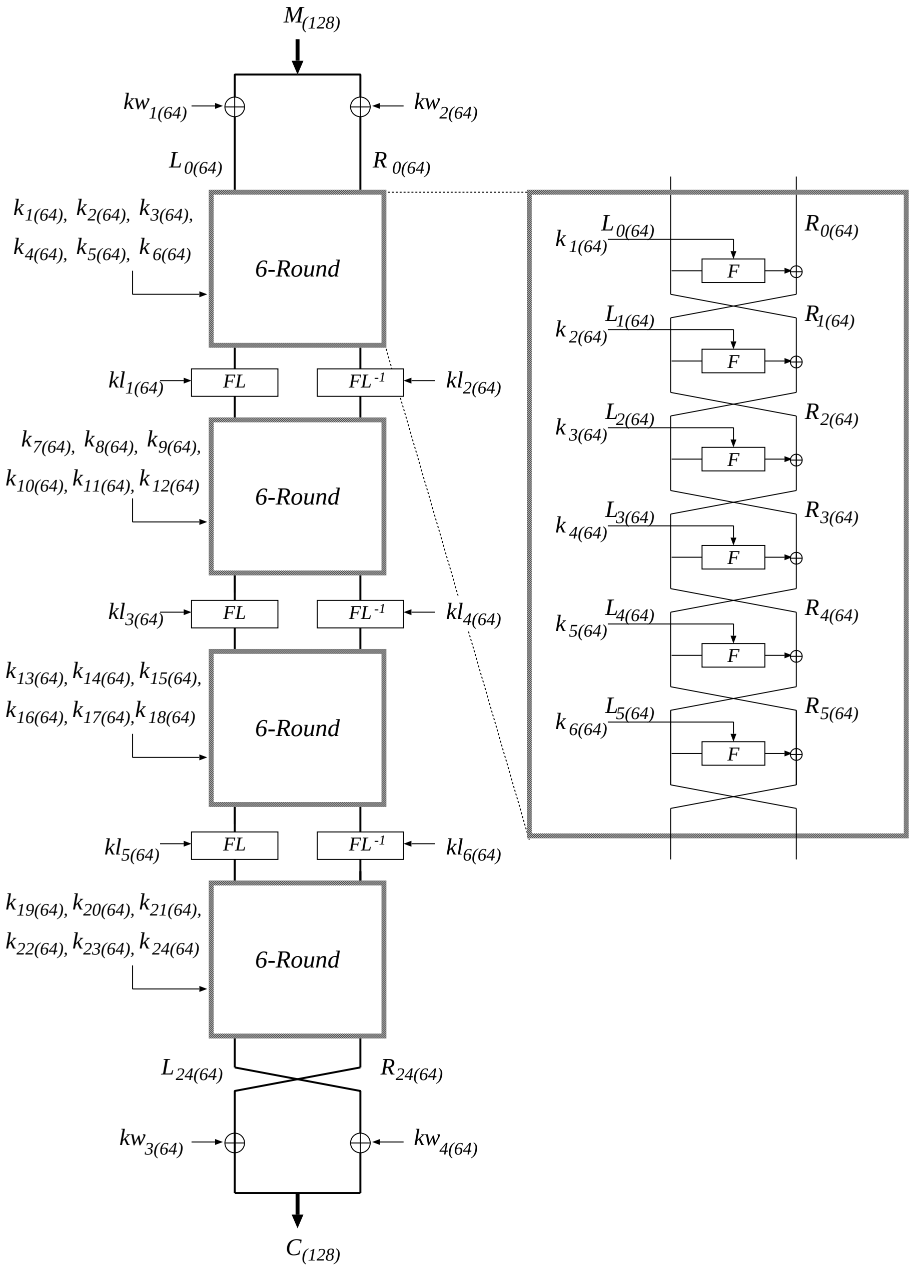

Figure 2: Encryption Procedure of *Camellia* for 192-bit and 256-bit key

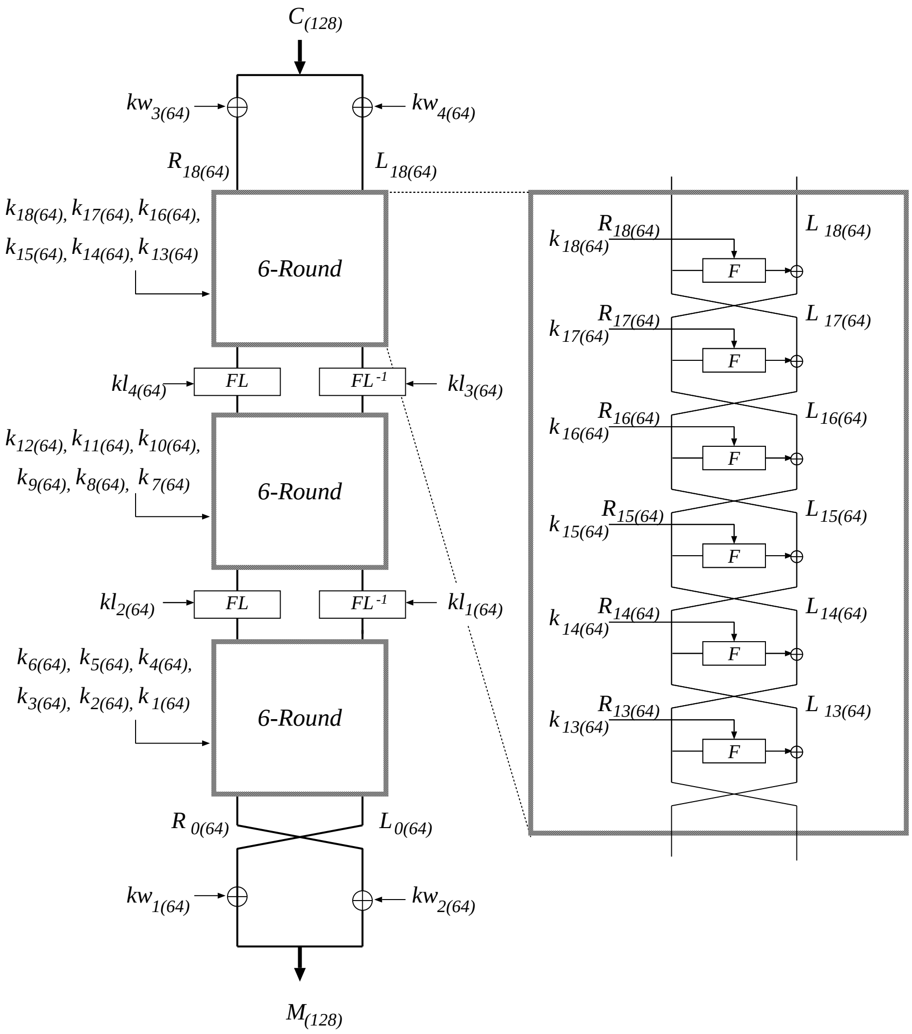

Figure 3: Decryption Procedure of *Camellia* for 128-bit key

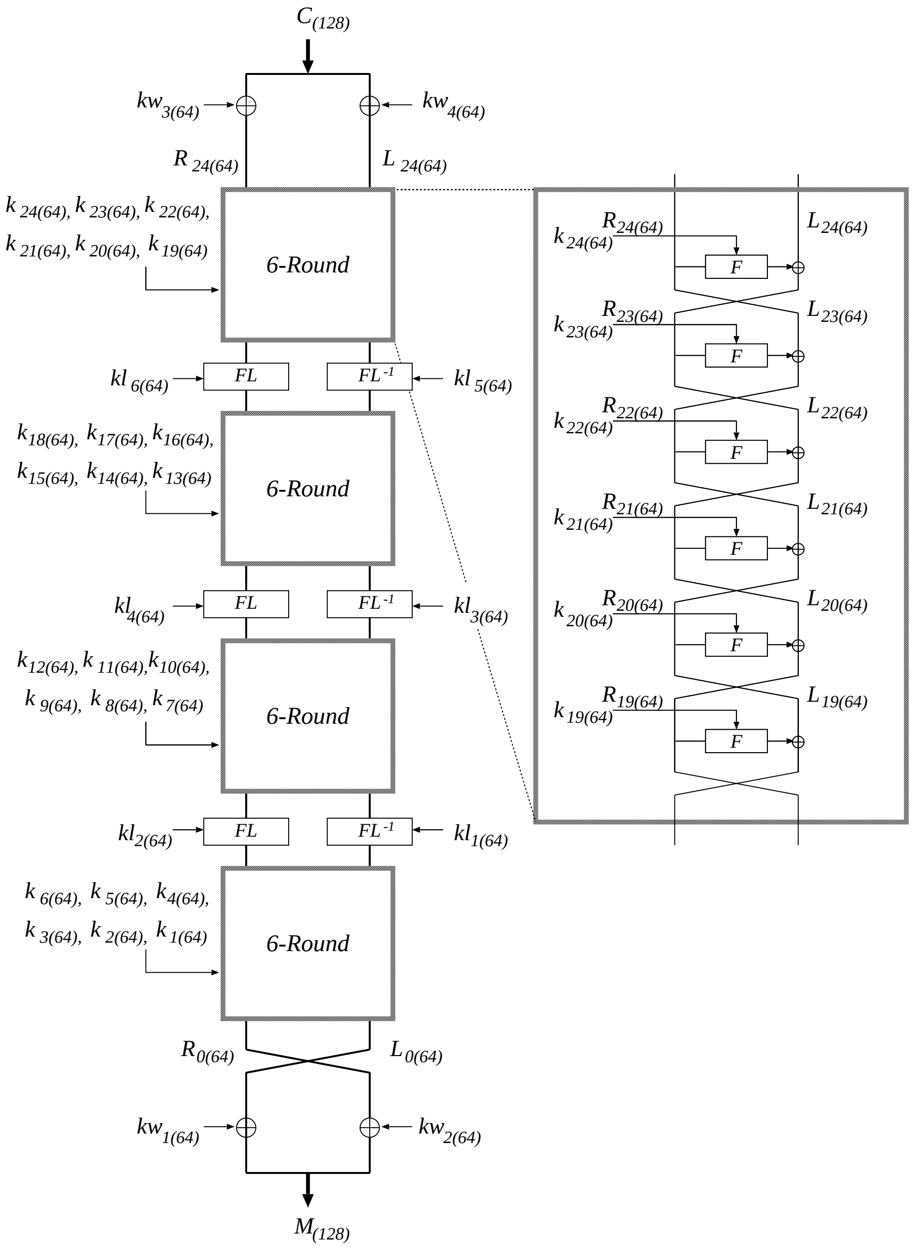

Figure 4: Decryption Procedure of *Camellia* for 192-bit and 256-bit key

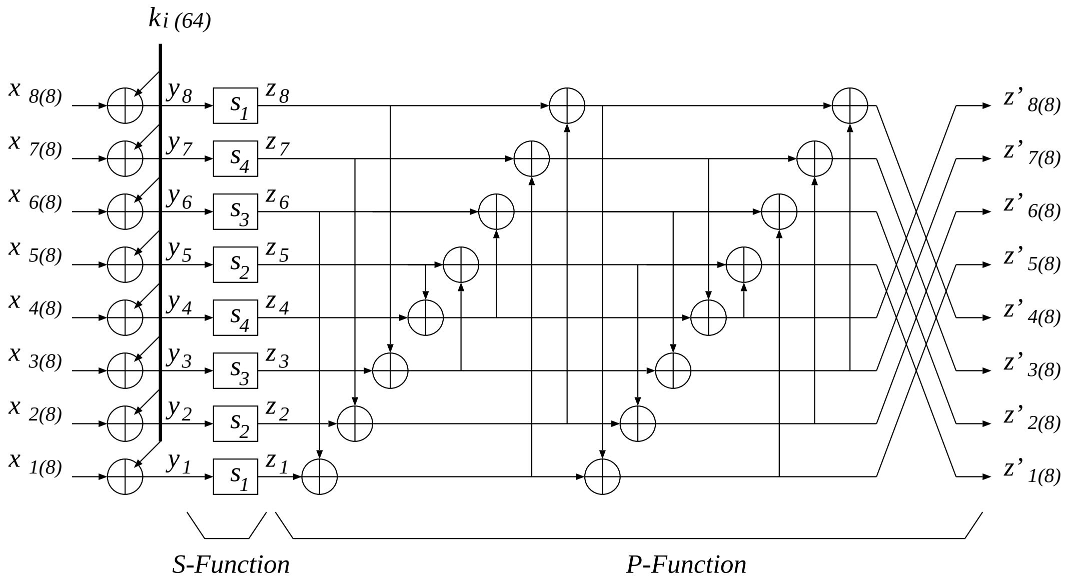

Figure 5: F-function

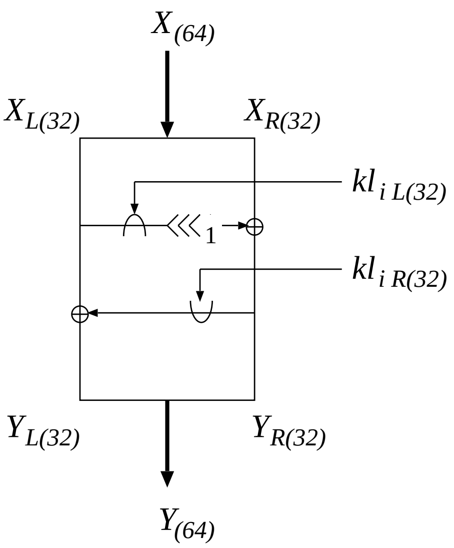

Figure 6: FL-function

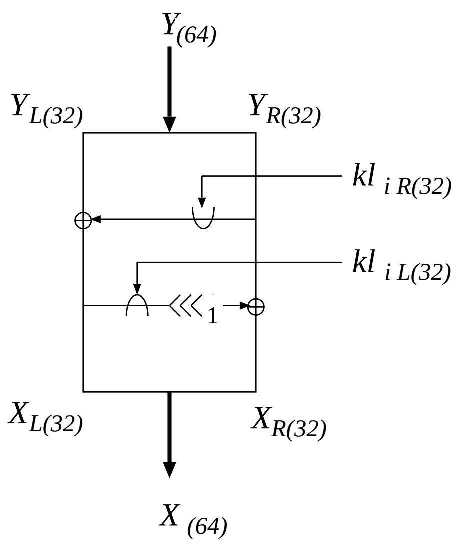

Figure 7: FL^{-1}-function

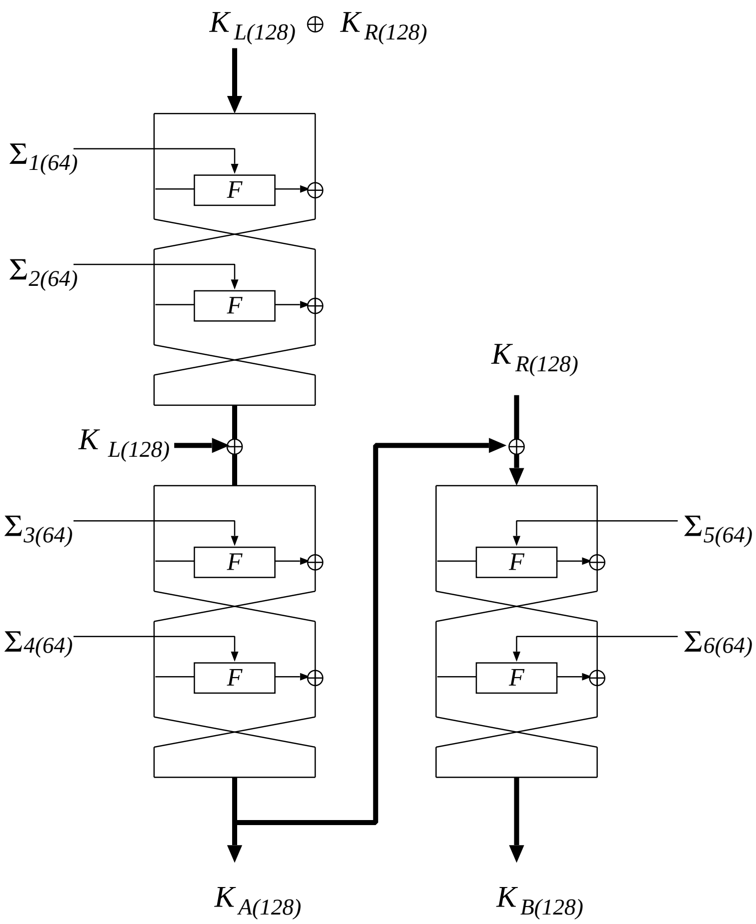

Figure 8: Key Schedule

B Test Data

The following is test data for *Camellia* in hexadecimal form:

128-bit key

key	`01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10`
plaintext	`01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10`
ciphertext	`67 67 31 38 54 96 69 73 08 57 06 56 48 ea be 43`

192-bit key

key	`01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10`
	`00 11 22 33 44 55 66 77`
plaintext	`01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10`
ciphertext	`b4 99 34 01 b3 e9 96 f8 4e e5 ce e7 d7 9b 09 b9`

256-bit key

key	`01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10`
	`00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff`
plaintext	`01 23 45 67 89 ab cd ef fe dc ba 98 76 54 32 10`
ciphertext	`9a cc 23 7d ff 16 d7 6c 20 ef 7c 91 9e 3a 75 09`