New Observation on Camellia

Duo Lei[1], Li Chao[2], and Keqin Feng[3]

[1] Department of Science, National University of Defense Technology,
Changsha, China
Duoduolei@163.com

[2] Department of Science, National University of Defense Technology,
Changsha, China

[3] Department of Math, Tsinghua University,
Beijing, China

Abstract. In this paper, some observations on Camellia are presented, by which the Square attack and the Collision attack are improved. 11-round 256-bit Camellia without FL function is breakable with complexity of 2^{250} encryptions. 9-round 128-bit Camellia without FL function is breakable with the complexity of 2^{90} encryptions. And 10-round 256-bit Camellia with FL function is breakable with the complexity of 2^{210} encryptions and 9-round 128-bit Camellia with FL function is breakable with the complexity of 2^{122} encryptions. These results are better than any other known results. It concludes that the most efficient attack on Camellia is Square attack.

1 Introduction

Camellia [1] is a 128-bit block cipher proposed by NTT and Mitsubishi in 2000. It has the modified Feistel structure with irregular rounds, which is called the FL/FL^{-1} function layers. Camellia had been submitted to the standardization and the evaluation projects such as ISO/IEC JTC 1/SC 27, CRYPTREC, and NESSIE.

The most efficient methods analyzing Camellia are truncated differential cryptanalysis[4][5][6] and higher order differential attack[7][9]. Camellia with more than 11 rounds is secure against truncated differential cryptanalysis. Square attack[11] is a most efficient attack on AES[11][12] . Y. He and S. Qing [2] showed that 6-round Camellia is breakable by it. Y.Yeom, S. Park, and I. Kim [3] improved the result to 8 rounds. Collision attack on Camellia was presented by WL Wu[10].

In this paper, some observations on Camellia are presented, by which the Square attack and the Collision attack are improved. Variant Square Attack can break 11-round 256-bit Camellia without FL function with complexity of 2^{250} encryptions. 9-round 128-bit Camellia without FL function is breakable with the complexity of 2^{90} encryptions. And 10-round 256-bit Camellia with FL function is breakable with the complexity of 2^{210} encryptions and 9-round 128-bit Camellia with FL function is breakable with the complexity of 2^{122} encryptions. These results are better than any other known results.

B. Preneel and S. Tavares (Eds.): SAC 2005, LNCS 3897, pp. 51–64, 2006.

Brief description of Camellia and some new structures equivalent to Camellia are presented in section2. In section 3, active bytes transformations on Camellias are illustrated and some new properties are demonstrated. Our attacking methods are described in section 4. Section 5 is some extension. The paper concludes with our most important results.

2 Equivalent Structures of Camellia

2.1 Description of the Camellia

Camellia has a 128-bit block size and supports 128-, 192- and 256-bit keys. Camellia with a 128-bit key and 256-bit key is written as 128-Camellia, 256-Camellia. The design of Camellia is based on the Feistel structure and its number of rounds is 18(128-bit key) or 24(192-, 256-bit key). The FL/FL^{-1} function layer is inserted in it every 6 rounds in order to thwart future unknown attacks. Before the first round and after the last round, there are pre- and post-whitening layers. F function contains key-addition, S-function and P-function. S-function contains 4 types of S-boxes s_1, s_2, s_3, and s_4. s_2,s_3,s_4 are variations of s_1. The P-function:$\{0,1\}^{64} \mapsto \{0,1\}^{64}$ maps $(z_1, ..., z_8)$ to $(z'_1, ..., z'_8)$, defined as:

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

We refer $X_{(r)}, K_{(r)}$ to the rth round input and subkey, refer $X^L_{(r)}$ and $X^R_{(r)}$ to the left, right half bytes of $X_{(r)}$, which implies $X_{(r)} = (X^L_{(r)}, X^R_{(r)})$. Let $X_{(ri)}$ is the ith byte of $X_{(r)}$, PL and CP be the Plaintext and Ciphertext, and $X_{(L)}$ be the last round output. The round function of Camellia is written as follows (named as Camellia-1) , which is shown in Fig. 1:

$$\begin{gathered}
X^L_{(1)} = PL^L, X^R_{(1)} = PL^R, \\
X^L_{(r+1)} = X^R_{(r)} \oplus P(s(X^L_{(r)} \oplus K_{(r)})), \\
X^R_{(r+1)} = X^L_{(r)}, \\
CP^L = X^L_{(L)}, CP^R = X^R_{(L)}
\end{gathered}$$

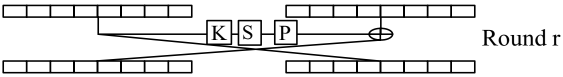

Fig. 1. Round function of Camellia-1

2.2 Three Equivalent Structures of Camellia

We can write Camellia in following form called Camellia-2, where P^{-1} is the inverse transformation of P-function and $\bar{X}_{(r)}$ is the rth round input of Camellia-2. Figure illustration of Camellia-2 is given in Fig. 2:

$$\begin{gathered}\bar{X}^L_{(1)} = P^{-1}(PL^L), \bar{X}^R_{(1)} = P^{-1}(PL^R),\\ \bar{X}^L_{(r+1)} = \bar{X}^R_{(r)} \oplus s(P(\bar{X}^L_{(r)}) \oplus K_{(r)}),\\ \bar{X}^R_{(r+1)} = \bar{X}^L_{(r)},\\ CP^L = P(\bar{X}^L_{(L)}), CP^R = P(\bar{X}^R_{(L)})\end{gathered}$$

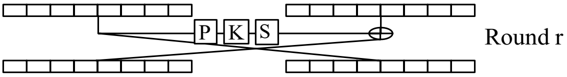

Fig. 2. Round function of Camellia-2

We can also write Camellia in the form of Camellia-3 where $\hat{X}_{(r)}$ is the rth round input. Figure illustration is given in Fig. 3:

$$\begin{gathered}\hat{X}^L_{(1)} = PL^L, \hat{X}^R_{(1)} = P^{-1}(PL^R),\\ \hat{X}^L_{(r+1)} = \hat{X}^R_{(r)} \oplus s(\hat{X}^L_{(r)} \oplus K_{(r)}), \text{ where r is odd}\\ \hat{X}^L_{(r+1)} = P(\hat{X}^R_{(r)} \oplus s(P(\hat{X}^L_{(r)}) \oplus K_{(r)})), \text{ where r is even}\\ \hat{X}^R_{(r+1)} = \hat{X}^L_{(r)},\\ CP^L = \hat{X}^L_{(L)}, CP^R = P(\hat{X}^R_{(L)})\end{gathered}$$

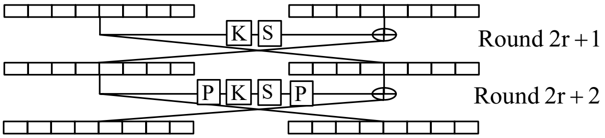

Fig. 3. Round function of Camellia-3

The structure of Camellia-4 is given as follows where $\tilde{X}_{(r)}$ is the rth round input of that. Figure illustration is given in Fig. 4:

$$\begin{gathered}\tilde{X}^L_{(1)} = P^{-1}(PL^L), \tilde{X}^R_{(1)} = PL^R,\\ \tilde{X}^L_{(r+1)} = P(\tilde{X}^R_{(r)} \oplus s(P(\tilde{X}^L_{(r)}) \oplus K_{(r)})), \text{ where r is odd}\\ \tilde{X}^L_{(r+1)} = \tilde{X}^R_{(r)} \oplus s(\tilde{X}^L_{(r)} \oplus K_{(r)}), \text{ where r is even}\\ \tilde{X}^R_{(r+1)} = \tilde{X}^L_{(r)},\\ CP^L = P(\tilde{X}^L_{(L)}), CP^R = \tilde{X}^R_{(L)}\end{gathered}$$

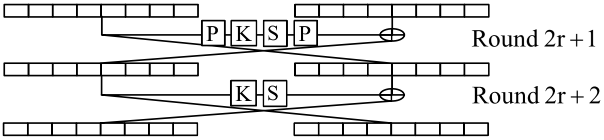

Fig. 4. Round function of Camellia-4

3 New Observations on Camellia

3.1 Preliminaries

Let a $\varLambda$-set be a set of 256 states that are all different in some of the state bytes (the active) and all equal in the other state bytes (the passive). We have:

$$\forall x, y \in \varLambda : \begin{cases} x_i \neq y_i & \mathit{if}\ x_i\ \mathit{is\ active} \\ x_i = y_i & \mathit{else} \end{cases}$$

Let $\varGamma$-set be a set of 256 states that are all equal to zero in summation (the balanced).

$$\forall x \in \varGamma : \sum x_i = 0$$

Applying the S-function or Key-addition on a $\varLambda$-set results in a $\varLambda$-set with the positions of the active bytes unchanged. The result set of applying P-function to a $\varLambda$-set is not always a $\varLambda$-set but always a $\varGamma$-set.

Applying Key-addition or P-function on a $\varGamma$-set results in a $\varGamma$-set. Applying S-function on a $\varGamma$-set results in the active bytes and passive bytes are still balanced. Applying AND operation, OR operation or right shift operation on a $\varGamma$-set results in a $\varGamma$-set.

Here, we give some definitions that are used in following sections.

$\mathcal{F}$: A $\varLambda - set$ has the form of $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, i, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8\}$, in which α_i, β_j are constant, $i \in \{0, .., 255\}$.

$\mathcal{F}_t$: A $\varLambda - set$ has the form of $\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, i, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \gamma_t\}_t$, in which α_i, β_j, γ_k are constant, $i \in \{0, .., 255\}$.

$\widetilde{\mathcal{F}_t}$: A $\varLambda - set$ has the form of $\{i, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \gamma_t, s_1(i \oplus k_1), \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, s_1(\gamma_t \oplus k_2)\}_t$, in which α_i, β_j, γ_k, k_1, k_2 are constant, $i \in \{0, .., 255\}$.

$\widehat{\mathcal{F}_t}$: A $\varLambda - set$ has the form of $\{\beta_1, i \oplus \beta_2, i \oplus \beta_3, i \oplus \beta_4, i \oplus \beta_5, \beta_6, \beta_7, i \oplus \gamma_t, s_1(i \oplus k_1), s_1(i \oplus k_1) \oplus \alpha_2, s_1(i \oplus k_1) \oplus \alpha_3, \alpha_4, s_1(i \oplus k_1) \oplus \alpha_5, \alpha_6, \alpha_7, s_1(i \oplus k_1) \oplus \alpha_8\}_t$, in which α_i, β_j, γ_k, k_1 are constant, $i \in \{0, .., 255\}$.

$\widetilde{\widetilde{\mathcal{F}_t}}$: A $\varLambda - set$ has the form of $\{s_1(i \oplus k_1), \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, s_1(\gamma_t \oplus k_2), \varDelta_1 \oplus i, \varDelta_2, \varDelta_3, \varDelta_4, \varDelta_5, \varDelta_6, \varDelta_7, \varDelta_8 \oplus \gamma_t\}$, in which $\{\varDelta_1, \varDelta_2, \varDelta_3, \varDelta_4, \varDelta_5, \varDelta_6, \varDelta_7, \varDelta_8\}$ satisfy Eq.(1), α_i, β_j, γ_k, η_1, η_2, η_3, η_4, η_5, η_6, η_7, η_8 k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8, k_9 are constant, $i \in \{0, .., 255\}$.

$$\begin{bmatrix} \varDelta_1 \\ \varDelta_2 \\ \varDelta_3 \\ \varDelta_4 \\ \varDelta_5 \\ \varDelta_6 \\ \varDelta_7 \\ \varDelta_8 \end{bmatrix} = \begin{bmatrix} 1\,0\,1\,1\,0\,1\,1\,1 \\ 1\,1\,0\,1\,1\,0\,1\,1 \\ 1\,1\,1\,0\,1\,1\,0\,1 \\ 0\,1\,1\,1\,1\,1\,1\,0 \\ 1\,1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,1\,0\,1\,0\,1\,1 \\ 0\,0\,1\,1\,1\,1\,0\,1 \\ 1\,0\,0\,1\,1\,1\,1\,0 \end{bmatrix} \begin{bmatrix} s_1(s_1(i \oplus k_1) \oplus \eta_1 \oplus s_1(\gamma_t \oplus k_2)) \\ s_2(s_1(i \oplus k_1) \oplus \eta_2 \oplus s_1(\gamma_t \oplus k_2)) \\ s_3(s_1(i \oplus k_1) \oplus \eta_3 \oplus s_1(\gamma_t \oplus k_2)) \\ s_4(\eta_4) \\ s_2(s_1(i \oplus k_1) \oplus \eta_5 \oplus s_1(\gamma_t \oplus k_2)) \\ s_3(\eta_6 \oplus s_1(\gamma_t \oplus k_2)) \\ s_4(\eta_7 \oplus s_1(\gamma_t \oplus k_2)) \\ s_1(s_1(i \oplus k_1) \oplus \eta_8) \end{bmatrix} \tag{1}$$

$\hat{\hat{\mathcal{F}}}_t$: A $\Lambda - set$ has the form of $\{s_1(i \oplus k_1),\ s_1(i \oplus k_1) \oplus \alpha_2,\ s_1(i \oplus k_1) \oplus \alpha_3,\ \alpha_4,\ s_1(i \oplus k_1) \oplus \alpha_5,\ \alpha_6,\ \alpha_7,\ s_1(i \oplus k_1),\ s_1(s_1(i \oplus k_1) \oplus k_2),\ s_2(s_1(i \oplus k_1) \oplus k_3) \oplus i,\ s_3(s_1(i \oplus k_1) \oplus k_4) \oplus i,\ \alpha_4 \oplus i,\ s_2(s_1(i \oplus k_1) \oplus k_5) \oplus i,\ \alpha_6,\ \alpha_7,\ s_1(s_1(i \oplus k_1) \oplus k_6) \oplus i \oplus \gamma_t)\}_t$, in which α_i, β_j, γ_k, k_1, k_2, k_3, k_4, k_5, k_6 are constant, $i \in \{0, .., 255\}$.

In next section, we trace the position changes of the active bytes through 5 rounds transformations of Camellia-1∼4 and the plaintext set $\{PL\}$ is a $\Lambda - set\ \mathcal{F}$. We just describe the evolution of left half bytes of round outputs, for the left half bytes of previous round pass to right half bytes of next round unchanged.

3.2 Active Bytes Changing Properties

In Camellia-1, $X^R_{(1,1)}$ is active byte, 1st round transformations convert the active byte to $X^L_{(2,1)}$, other bytes are passive. In 2nd round transformations, P-function converts the active byte to 5 active bytes which are $X^L_{(31)}$, $X^L_{(32)}$, $X^L_{(33)}$, $X^L_{(35)}$, $X^L_{(38)}$ and three passive bytes. In 3rd round transformation, P-function converts the active bytes to 8 balanced bytes. In 4th round transformation, S-function converts balanced bytes to unbalanced bytes. After 5th round transformation all bytes are unbalanced. Evolutions of active bytes are illustrated in Fig. 5.

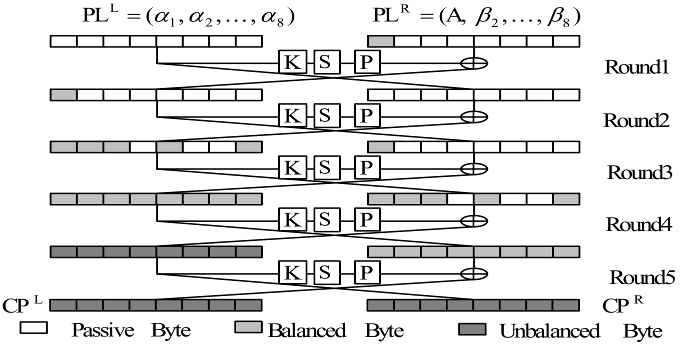
$PL^L = (\alpha_1, \alpha_2, \ldots, \alpha_8)$
$PL^R = (A, \beta_2, \ldots, \beta_8)$
K
S
P
Round1
Round2
Round3
Round4
Round5
CP^L
CP^R
Passive Byte
Balanced Byte
Unbalanced Byte

Fig. 5. Round function of Camellia-1

In Camellia-2, 1st byte of $\{PL\}$ is active, the pre-P^{-1}-function converts the active byte to 5 active bytes. 1st round transformations convert the 5 active bytes and 3 passive bytes to 5 active bytes and 3 passive bytes. 2nd round transformations convert the 5 active bytes and 3 passive bytes to 1 active byte and 7 passive bytes. 3rd round transformation convert the 5 active bytes and 3 passive bytes to 2 active bytes, 4 balanced bytes and 2 passive bytes, where $\bar{X}^L_{(41)}, \bar{X}^L_{(42)}$ are active, $\bar{X}^L_{(42)}, \bar{X}^L_{(43)}, \bar{X}^L_{(45)}, \bar{X}^L_{(48)}$ are balanced and $\bar{X}^L_{(46)}, \bar{X}^L_{(47)}$ are passive. 4th round transformations convert those bytes to unbalanced bytes. Evolutions of active bytes are illustrated in Fig. 6.

Details of 2nd and 3rd transformation are given in Eq.(2)and Eq.(3).

$$\begin{aligned}\bar{X}^L_{(3)} &= \bar{X}^R_{(2)} \oplus s(P(\bar{X}^L_{(2)} \oplus K_{(2)}) \\ &= \bar{X}^L_{(1)} \oplus s(P(\bar{X}^R_{(1)} \oplus s(P(\bar{X}^L_{(1)}) \oplus K_{(1)})) \oplus K_{(2)}) \\ &= \bar{X}^L_{(1)} \oplus s(P(P^{-1}(PL^R) \oplus s(P(\bar{X}^L_{(1)}) \oplus K_{(1)})) \oplus K_{(2)}) \\ &= \bar{X}^L_{(1)} \oplus s((PL^R \oplus s(P(\bar{X}^L_{(1)}) \oplus K_{(1)})) \oplus K_{(2)})\end{aligned} \tag{2}$$

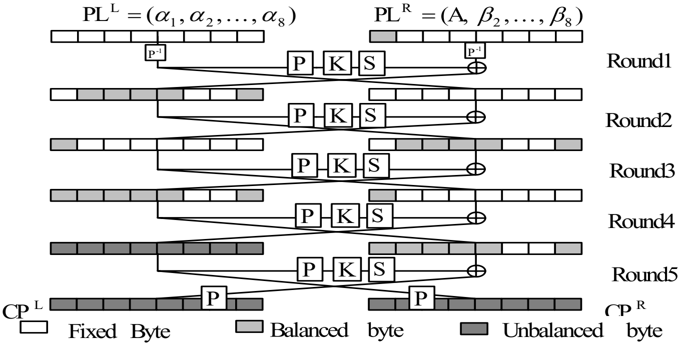

Fig. 6. Round function of Camellia-2

$\bar{X}^L_{(11)}$ is the only active byte in $\{\bar{X}^L_{(1)}\}$, since applying addition and S-function on it results in $\{\bar{X}^L_{(3)}\}$ with position of active byte unchanged, demonstrated in Eq.(2). Each byte of $\bar{X}^L_{(4)}$ can be written in the form of Eq.(3), in which $\bar{X}^L_{(41)}, \bar{X}^L_{(44)}$ are influenced by an active byte $\bar{X}^L_{(31)}$ so active, $\bar{X}^L_{(42)}$, $\bar{X}^L_{(43)}$, $\bar{X}^L_{(45)}$, $\bar{X}^L_{(48)}$ are influenced by 2 active bytes thus balanced and $\bar{X}^L_{(46)}, \bar{X}^L_{(47)}$ are derived from passive bytes, still passive.

$$\begin{aligned}\bar{X}^L_{(41)} &= s(\bar{X}^L_{(31)} \oplus \bar{X}^L_{(33)} \oplus \bar{X}^L_{(34)} \oplus \bar{X}^L_{(36)} \oplus \bar{X}^L_{(37)} \oplus \bar{X}^L_{(38)} \oplus K_{(31)}) \oplus \bar{X}^R_{(31)} \\ \bar{X}^L_{(42)} &= s(\bar{X}^L_{(31)} \oplus \bar{X}^L_{(32)} \oplus \bar{X}^L_{(34)} \oplus \bar{X}^L_{(35)} \oplus \bar{X}^L_{(37)} \oplus \bar{X}^L_{(38)} \oplus K_{(32)}) \oplus \bar{X}^R_{(32)} \\ \bar{X}^L_{(43)} &= s(\bar{X}^L_{(31)} \oplus \bar{X}^L_{(32)} \oplus \bar{X}^L_{(33)} \oplus \bar{X}^L_{(35)} \oplus \bar{X}^L_{(36)} \oplus \bar{X}^L_{(38)} \oplus K_{(33)}) \oplus \bar{X}^R_{(33)} \\ \bar{X}^L_{(44)} &= s(\bar{X}^L_{(32)} \oplus \bar{X}^L_{(33)} \oplus \bar{X}^L_{(34)} \oplus \bar{X}^L_{(35)} \oplus \bar{X}^L_{(36)} \oplus \bar{X}^L_{(37)} \oplus K_{(34)}) \oplus \bar{X}^R_{(34)} \\ \bar{X}^L_{(45)} &= s(\bar{X}^L_{(31)} \oplus \bar{X}^L_{(32)} \oplus \bar{X}^L_{(36)} \oplus \bar{X}^L_{(37)} \oplus \bar{X}^L_{(38)} \oplus K_{(35)}) \oplus \bar{X}^R_{(35)} \\ \bar{X}^L_{(46)} &= s(\bar{X}^L_{(32)} \oplus \bar{X}^L_{(33)} \oplus \bar{X}^L_{(35)} \oplus \bar{X}^L_{(37)} \oplus \bar{X}^L_{(38)} \oplus K_{(36)}) \oplus \bar{X}^R_{(36)} \\ \bar{X}^L_{(47)} &= s(\bar{X}^L_{(33)} \oplus \bar{X}^L_{(34)} \oplus \bar{X}^L_{(35)} \oplus \bar{X}^L_{(36)} \oplus \bar{X}^L_{(38)} \oplus K_{(37)}) \oplus \bar{X}^R_{(37)} \\ \bar{X}^L_{(48)} &= s(\bar{X}^L_{(31)} \oplus \bar{X}^L_{(34)} \oplus \bar{X}^L_{(35)} \oplus \bar{X}^L_{(36)} \oplus \bar{X}^L_{(37)} \oplus K_{(38)}) \oplus \bar{X}^R_{(38)}\end{aligned} \tag{3}$$

In Camellia-3, 1st byte of $\{PL\}$ is active, the pre-P-function converts the active byte to 5 active bytes. 1st round transformations convert the 5 active bytes, 3 passive bytes to 5 active bytes,3 passive bytes. 2nd round transformations convert the 5 active bytes, 3 passive bytes to 5 active bytes, 3 active bytes. 3rd round transformation convert 5 active bytes, 3 passive bytes to 2 active bytes, 4 balanced bytes and 2 passive bytes, where $\hat{X}^L_{(41)}, \hat{X}^L_{(42)}$ are active, $\hat{X}^L_{(42)}, \hat{X}^L_{(43)}, \hat{X}^L_{(45)}, \hat{X}^L_{(48)}$ are balanced and $\hat{X}^L_{(46)}, \hat{X}^L_{(47)}$ are passive.And 4th round transformations convert those bytes to unbalanced bytes. The deducing procedure is similar to that of Camellia-2. Figure illustration is given in Fig.7.

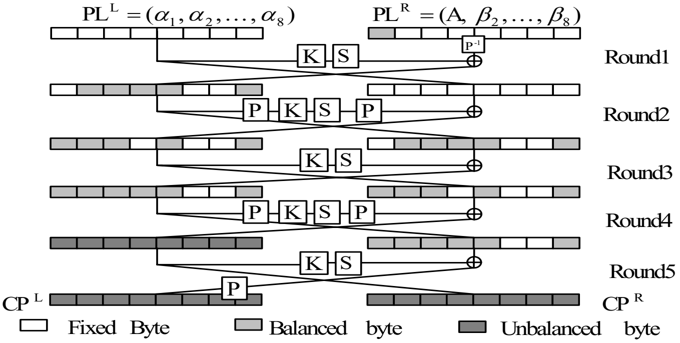

Fig. 7. Round function of Camellia-3

The outstanding properties of Camellia-3 are Eq.(4) and Eq.(5), that are used in Improved Square attack and Improved Collision attack in section 4.2 and section 4.3.

$$\sum_{PL\in\mathcal{F}} \hat{X}^R_{(5i)} = 0 \Rightarrow \sum_{PL\in\mathcal{F}} (s(\hat{X}^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{X}^L_{(6i)} = 0, 6 \leq i \leq 7. \tag{4}$$

$$\hat{X}^R_{(5i)} \equiv C \Rightarrow s(\hat{X}^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{X}^L_{(6i)} \equiv C, PL \in \mathcal{F}, 1 \leq i \leq 8. \tag{5}$$

Camellia-3 also have the property of Eq.(6) that is used in section 4.4.

$$\hat{X}^R_{(5i)} = B \Rightarrow s(\hat{X}^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{X}^L_{(6i)} = B,\ PL \in \mathcal{F},\ B\ is\ active, i \in \{1, 4\}. \tag{6}$$

In Camellia-4, $X^R_{(11)}$ is active, 1st round transformations convert the active byte to an active byte, 2nd round transformations convert the active byte to an active byte and 3rd round transformations convert the active byte to 8 active bytes. Figure illustration is shown in Fig.8. The outstanding property of Camellia-4 is that seven 3rd round output bytes are passive, which will be used in variant Square attack in section 4.1.

$$\textstyle\sum_{PL\in\mathcal{F}} \tilde{X}^L_{(4i)} \oplus K_{(4i)} = 0, i \neq 1, \tilde{C}_i\ is\ a\ constant.$$

$$\Rightarrow \sum_{PL\in\mathcal{F}} (s^{-1}(\tilde{X}^L_{(5i)}) \oplus \tilde{C}_i) = 0, i \neq 1, \tilde{C}_i\ is\ a\ constant. \tag{7}$$

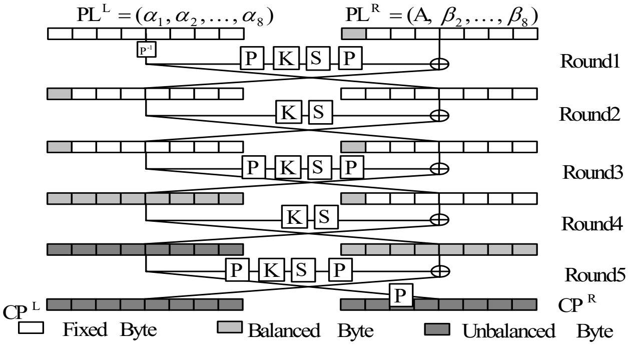

Fig. 8. Round function of Camellia-4

4 Some Attacks

In this section, we construct the attacks on Camellia without pre-, post- whitening and FL/FL^{-1} function. The influences of FL/FL^{-1} function are discussed in section 5.

4.1 Variant Square Attack

The 6-round variant Square attack use the property of Eq.(8), which is derives from Eq.(7), and use the structure of Camellia-4. This attack can be described by the following steps.

$$\sum_{PL \in \mathcal{F}_t} (s^{-1}(s(\tilde{X}^{(R)}_{(7i)} \oplus K_{(6i)}) \oplus \tilde{X}^{L}_{(7i)} \oplus \tilde{C}_i) = 0, \tilde{C}_i \ is \ a \ constant, i \neq 1. \qquad (8)$$

- Step 1: Select the Plaintext sets $\{PL\}_t$ as $\{PL\}_t = \mathcal{F}_t, 1 \leq t \leq 3$, calculate the values of $(X^L_{(7)}, X^R_{(7)})$ and record them, which will be used in following steps.
- Step 2: Guess $k_{(65)}$ and $\tilde{C}_5$, then check whether Eq.(8) is satisfied or not, where the Plaintext set is $\mathcal{F}_1$. If Eq.(8) is satisfied, record the values of $k_{(65)}$ and $\tilde{C}_5$ that may be the correct pair. Step 2 is ended until all possible values are checked.
- Step 3: For all 'correct' pairs from step 2, checks whether Eq.(8) is satisfied or not, where the Plaintext set are $\mathcal{F}_t, 2 \leq t \leq 3$. ($\gamma_t$ does not influence the value of $\tilde{C}_5$).

In this 6-round attack, the time that step 1 takes is 3×2^8 6-round encryptions takes. Since Eq.(8) has $256 + 256 \times 3$ additions, 256×2 substitutions, and 6-round Camellia has 44×6 additions, 8×6 substitutions, the time Eq.(8) takes is nearly 6 times of that 6-round encryptions of Camellia take. In step 2, Eq.(8) repeats $2^8 \times 2^8$ times. The probability of wrong key passing the checking is 2^8, so there is 2^8 pairs passing the checking in step 2, that implies the time of step

3 and 4 take $2^8 \times 3$ and 3 times of that 6-round encryptions take, respectively. In this 6-round attack, we only do the step 1, step 2 and step 3 one time. So the 6-round attack's complexity is $2^8 \times 3 + 2^{16} \times 6 + 2^8 \times 6 \times 2 \approx 2^{18.6}$. The counts of selected Plaintexts are $2^8 \times 3$.

In 7-round attack, we add one round at the beginning, use the structure of Camellia-3 and select the input sets $\{\tilde{X}_{(1)}\}_t$ as $\widetilde{\mathcal{F}}_t$, where k_1 and k_2 are guessing key, for that: if $k_1 = k_{(11)}, k_2 = k_{(18)}$ *then* $\forall \tilde{X}_{(1)} \in \widetilde{\mathcal{F}}_t \Rightarrow \tilde{X}_{(2)} \in \mathcal{F}_t$.

This 7-round attack use Eq.(9) for checking and select the T as 7, for the probability of all key bytes pass the checking is $2^{-8\times T}$. The guessing step is as follows:

$$\sum_{\tilde{X}_{(1)} \in \widetilde{\mathcal{F}}_t} (s^{-1}(s(\tilde{X}^{(R)}_{(8i)} \oplus K_{(7i)}) \oplus \tilde{X}^L_{8i} \oplus \tilde{C}_i) = 0, i \neq 1, 1 \leq t \leq T\}. \qquad (9)$$

- Step 1: Guess the values of $k_{(11)}$ and $k_{(18)}$.
- Step 2: For each $X_{(1)} \in \widetilde{\mathcal{F}}_t$ calculate the result of $\{X_{(8)}\}_t$.
- Step 3: Guess $k_{(75)}$ and $\widetilde{C}_5$, and record the values that pass the checking Eq.(9),where the Plaintext set is $\widetilde{\mathcal{F}}_1$.
- Step 4: For all 'correct' pairs from step 3, checks whether Eq.(9) is satisfied or not, where $2 \leq t \leq 7$, if all the pairs can't pass the check, go to step 1.

The attacking complexity is $2^{16} \times (2^8 \times 7 + 2^{16} \times 6 + 2^8 \times 6 \times 6) \approx 2^{33.6}$. The complexity of 256-Camellia is $2^{25.6}$, since its 1st round key bits are the same as 7th round key bits.

The 8-round attack is similar to 7-round attack, the only difference is that $\tilde{X}^{(R)}_{(8i)}$ is unknown. Getting $\tilde{X}^{(R)}_{(8i)}$ from $\tilde{X}_{(9)}$ needs five 8th round key bytes, then complexity of this attack is $2^{16} \times (2^8 \times 12 + 2^{56} \times 6 + 2^{48} \times 6 \times 11) \simeq 2^{74.6}$, the complexity of 256-Camellia becomes $2^{66.6}$.

In 9-round attack, we add one round at the beginning and use the structure of Camellia-4, where the selected special plaintexts should satisfy the properties of that $\{\tilde{X}^{(2)}\}_t$ is a $\Gamma - set$ with the from of $\widetilde{\mathcal{F}}_t$. So we select $\{\tilde{X}_{(1)}\}_t$ as $\widetilde{\widetilde{\mathcal{F}}}_t$, where k_1,...,k_9 are guessing key. . The complexity of this attack is $2^{72} \times (2^8 \times 19 + 2^{56} \times 4 + 2^{48} \times 4 \times 18) \approx 2^{130}$. In 256-Camellia, the 2nd round key bytes are the same as 8th round key bytes, so the complexity of the attack is 2^{122}. In 128-Camellia 1st round key bytes are the same as 9th round key, so the complexity is 2^{90}. In 10-round attack, we add 1 round at the end and use the structure of Camellia-4, it will need to guess another 8 bytes key. The complexity is 2^{186}. And attacking on 11-round Camellia, the complexity is 2^{250}.

4.2 Improved Square Attack

The best result of Square attack against Camellia was given by Y.Yeom, S. Park, and I. Kim [3]. In this paper we improved the attacking result, since Camellia-3 satisfying Eq.(4), so called Improved Square attack.

The basic attack on 5-round camellia use the property of $s(\hat{X}^R_{(65)} \oplus K_{(55)}) \oplus \hat{X}^L_{(65)}$ is balanced byte if the plaintext set is $\mathcal{F}_t$, which is illustrated in Eq.(4), and the probability of wrong key pass the checking is 2^{-8}. The attacking details are as follows.

- Step 1: Choose $\mathcal{F}_t, 1 \le t \le 2$ as plaintext sets, calculate the values of $\hat{X}^L_{(65)}$ and record them.
- Step 2: For each possible values of $K_{(55)}$, check whether Eq.(4) is satisfied or not, where the Plaintext set is $\mathcal{F}_1$, and record the passed key bytes, which may be the correct key byte. Go to next step until all possible values are checked.
- Step 3: For all 'correct' key bytes in step 2, checks whether Eq.(4) is satisfied or not, where the Plaintext set is $\mathcal{F}_2$.

For this 5-round attack, the time that step 1 takes equals the time that 2×2^8 5-round encryptions takes. Since Eq.(4) has $256 \times 2 + 256$ additions and 256 substitutions, and 5-round Camellia has 44×5 additions and 8×5 substitutions, so the time Eq.(4) takes is nearly 5 times of that 5-round Camellia encryptions take. In step 2, Eq.(4) repeats 2^8 times. The probability of wrong key passing the checking is 2^{-8}, so there will be few passing the checking, that means the time of step 3 takes 5 times of that 5-round encryptions take. So the complexity of 5-round attack is $2^8 \times 2 + 2^8 \times 5 + 5 \approx 2^{10.6}$.

6-round attack add one round at the begin and select the $\widehat{\mathcal{F}}_t, 1 \le t \le 5$ as plaintext sets and use the structure of Camellia-4. If k_1 in $\widehat{\mathcal{F}}_t, 1 \le t \le 5$ equals $k_{(11)}$, then $s(\hat{X}^R_{(75)} \oplus K_{(65)}) \oplus \hat{X}^L_{(75)}$ is balanced byte. So in this attack for each guessing k_1 we check wether the byte $s(\hat{X}^R_{(75)} \oplus K_{(65)}) \oplus \hat{X}^L_{(75)}$ is balanced or not, similar as 5-round attack. The complexity of the attack is $2^8 \times (2^8 \times 5 + 2^8 \times 4 + 4 \times 4) \approx 2^{18}$. We extend 6 round attacks to 7-round by adding one round at the end and select the plaintexs as 6-round attack. Then to get to know $\hat{X}^R_{(75)}$ we have to guess 5 7th round key bytes, so the complexity is $2^8 \times (2^8 \times 10 + 2^{48} \times 4 + 2^{40} \times 4) \approx 2^{58}$.

In 8-round attack, adding one round at the beginning, use the structure of Camellia-4. The input sets $\{\hat{X}_{(1)}\}_t$ is selected as $\hat{\hat{\mathcal{F}}}_t, 1 \le t \le 15$. The key bytes are used in attack. The complexity of the attack is $2^{48} \times (2^8 \times 15 + 2^{48} \times 4 + 2^{40} \times 14) \approx 2^{98}$. The complexity of the attack on 256-Camellia is 2^{82}, since the 1st and 2nd round key bytes are the same as 7th and 8th round key bytes, respectively. The complexity of the attacks on 9 and 10 rounds are 2^{146} and 2^{212}, respectively.

4.3 Improved Collision Attack

Collision attack on Camellia is given by WL wu[10]. We improve the attacking results, called Improved Collision attack. In 5-round attack, Eq.(10) is used for checking, which is derived from Eq.(5).

$$s(\hat{X}^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{X}^L_{(6i)} \equiv s(\hat{X}'^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{X}'^L_{(6i)}, \hat{X}_{(1)}, \hat{X}'_{(1)} \in \mathcal{F}, i \in \{6, 7\}. \quad (10)$$

The procedure of this attack is similar to that of 5 round Improved Square attack. The time Eq.(10) takes is nearly 1/4 times that of 1-round Camellia encryptions take, then the complexity of 5 round attack is $4 + 4 \times 2^8/(4 \times 5) \approx 2^{5.8}$. Similarly as section 4.2, The complexities of the attack on 6,7,8,9 rounds are $2^8 \times (5 + 5 \times 2^8/(6 \times 4)) \approx 2^{13.7}$, $2^8 \times (10 + 10 \times 2^{48}/(7 \times 4)) \approx 2^{54.5}$, $2^{48} \times (15 + 15 \times 2^{48}/(8 \times 4)) \approx 2^{94.9}$ and $2^{48} \times (23 + 23 \times 2^{112}/(9 \times 4)) \approx 2^{159.4}$. The complexity of 9-round attack on 128-Camellia is $2^{119.4}$. The complexities of the attack on 256-Camellia with 7,8,9 and 10 are $2^{46.5}$, $2^{78.9}$, $2^{143.4}$ and $2^{205.6}$.

4.4 Other Observations

We can build a new attack on Camellia based on Eq.(6), which implies the result of $s(\hat{X}^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{(}X)^L_{(6i)}$ is a active byte. We select the $\Lambda - set\ \mathcal{F}$ as plaintext sets,then check whether $s(\hat{X}^R_{(6i)} \oplus K_{(5i)}) \oplus \hat{(}X)^L_{(6i)}$ is active byte or not for guessing key byte $K_{(5i)}$.

There is also an interesting property in Camellia-4. If we select the Plaintext $(\tilde{X}^L_1, \tilde{X}^R_1) \in \mathcal{F}$,then $\tilde{X}^R_{(58)}$ with the form of Eq.(11).

$$\tilde{X}^R_{(58)} = s_1(s_1(B \oplus \gamma) \oplus s_2(B \oplus \gamma) \oplus \delta) \oplus \epsilon,\ \gamma, \delta, \epsilon\ are\ passive,\ B\ is\ active \quad (11)$$

5 The Influence of FL/FL^{-1}

In this section, we construct the attacks on Camellia with FL/FL^{-1} function and without pre- and post-whitening.

5.1 Variant Square Attack

In 7-round variant Square attack, we use the structure of Camellia-4 and select the plaintexts $\{\tilde{X}^L_{(1)}, \tilde{X}^R_{(1)}\}_t$ as a series of $\Lambda - sets\ \bar{\bar{\mathcal{F}}}_t$. The equation used in this attack is Eq.(12).

$$\sum_{\tilde{X}^L_{(1)}, \tilde{X}^R_{(1)} \in \bar{\bar{\mathcal{F}}}_t} (s^{-1}(\tilde{X}^L_{(7i)} \oplus \tilde{C}_i) = 0, i \in \{1, 2, ..., 8\}, t \in \{1, 2, ..., T\}. \quad (12)$$

If there is a FL/FL^{-1} function in Camellia-4, we have $\tilde{X}^L_{(7i)} \neq \tilde{X}^R_{(8i)}$. Getting $\tilde{X}^L_{(7i)}$ from $\tilde{X}^R_{(8i)}$ needs to guess eight key bytes, which are used in $FL^{-}1$ function. Hence the attacking complexity is $2^{72} \times (2^8 \times 21 + 2^{72} \times 6 + 2^{64} \times 6 \times 20) \approx 2^{146.6}$, where the value of T is 21. In 128-Camellia,the complexity becomes $2^{90.6}$, since the key bytes used in FL^{-1} function are the same as 1st round key bytes.

In this 8-round attack, we add one round at the end. Then the attacking complexity is $2^{146.6+64}$. It becomes $2^{194.6}$ in 256-Camellia, since in 256-Camellia 2nd round key bytes are the same as 8th round key.

5.2 Improved Square Attack

In 7-round Improved Square attack, we use the structure of Camellia-3 and select the $\{\hat{X}^L_{(1)}, \hat{X}^R_{(1)}\}$ as $\bar{\bar{\mathcal{F}}}_t$, and use Eq.(13) for checking .

$$\sum_{X_{(1)} \in \bar{\bar{\mathcal{F}}}} s(\hat{X}^R_{(8i)} \oplus K_{(7i)}) \oplus \hat{X}^L_{(8i)} = 0, i \in \{1, 2, ..., 8\}. \quad (13)$$

If there is a FL/FL^{-1} function in Camellia-3, we have to consider the property of $FL(\hat{X}^{L}_{(7)})$. Since the FL function results a $\Gamma-set$ in a $\Gamma-set$, we have a conclusion that whether there is a FL/FL^{-1} or not Eq.(13) always holds. Hence the complexity of that attack is $2^{48} \times (2^8 \times 10 + 2^8 \times 6 + 3 \times 9) \approx 2^{58.6}$, the key bytes required in this attack are $\{k_{(11)}, k_{(12)}, k_{(13)}, k_{(15)}, k_{(18)}, k_{(21)}, k_{(75)}\}$.

Table 1. The Summary of known attacks on Camellia

Rounds	FL/FL^{-1}	Methods	$Time$	$Time$	Notes
			$128-bit$	$256-bit$	
5	No	SA	2^{48}		[2]
5	No	SA	2^{16}		[3]
5	No	Improved SA	$2^{10.6}$		This Paper
5	No	Improved CA	$2^{5.8}$		This Paper
6	No	SA	2^{56}		[3]
6	No	Higher Order DC	2^{18}		[9]
6	No	Improved SA	2^{18}		This Paper
6	No	Improved CA	$2^{13.7}$		This Paper
6	No	Variant SA	$2^{18.6}$		This Paper
7	No	Truncated DC	192		[5]
7	No	Higher Order DC	2^{57}		[9]
7	Yes	SA	—	$2^{57.2}$	[3]
7	No	Improved SA	2^{58}	2^{50}	This Paper
7	No	Improved CA	$2^{54.7}$	$2^{46.7}$	This Paper
7	No	Variant SA	$2^{33.5}$	$2^{25.6}$	This Paper
7	Yes	Improved SA	$2^{58.6}$		This Paper
7	Yes	Variant SA	$2^{90.6}$	$2^{146.6}$	This Paper
8	No	Truncated DC	$2^{55.6}$		[5]
8	No	Higher Order DC	2^{120}		[9]
8	Yes	SA	—	2^{116}	[3]
8	Yes	Improved SA	2^{98}	2^{82}	This Paper
8	No	Improved CA	$2^{94.9}$	$2^{78.9}$	This Paper
8	No	Variant SA	$2^{74.6}$	$2^{66.6}$	This Paper
8	Yes	Improved CA	$2^{74.6}$	$2^{66,6}$	This Paper
8	Yes	Variant SA	—	$2^{194.6}$	This Paper
9	No	Higher Order DC	—	2^{188}	[9]
9	Yes	SA	—	$2^{181.4}$	[3]
9	Yes	Improved SA	2^{122}	2^{146}	This Paper
9	No	Improved CA	$2^{119.4}$	$2^{143.4}$	This Paper
9	No	Variant SA	2^{90}	2^{122}	This Paper
10	No	Higher Order DC	—	2^{252}	[9]
10	Yes	Improved SA	—	2^{210}	This Paper
10	No	Improved CA	—	$2^{207.4}$	This Paper
10	No	Variant SA	—	2^{186}	This Paper
11	No	Higher Order DC	—	$2^{259.6}$	[9]
11	No	Variant SA	—	2^{250}	This Paper

In 8-round attack, we add one round at the end and still use Eq.(13) for checking, than the attacking procedure becomes the same as that of section 4.2. The attacks on 9-and 10-round Camellia with FL function are also no difference from that of without FL function, which have been described in section 4.2.

6 Conclusions

Variant Square attack can break 9-round 128bit Camellia, 11-round 256 bit Camellia without FL function, further more, it is faster than exhaustive key search. The conclusions can be made that key schedule and P-function influence the security of Camellia and Square attack is still the best attack on Camellia. Table(1) give a summary of known attacks on Camellia.

Acknowledgement. We would like to thank Vincent Rijmen for his help for important comments and correctness of mistakes that improved the technical quality of the paper. The reviewer has kindly pointed out many technical problems and give suggestions in the submitted version of this draft. We would also like to thank the referees of the paper, for their constructive comments and suggestions.

References

1. K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima and T. Tokita.: Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis. In: Proceedings of Selected Areas in Cryptography. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (2000) 39-56.
2. Y. He and S. Qing: Square Attack on Reduced Camellia Cipher. In: ICICS 2001, Lecture Notes in Computer Science, Vol. 2229. Springer-Verlag, Berlin Heidelberg New York (2001) 238-245.
3. Y.Yeom, S. Park, and I. Kim.: On the security of Camellia against the Square attack.: In: Proceed-ings of Fast Software Encryption. Lecture Notes in Computer Science, Vol. 2365. Springer-Verlag, Berlin Heidelberg New York (2002) 89-99.
4. M. Kanda and T. Matsumoto.: Security of Camellia against Truncated Differential Cryptanalysis. In: Proceedings of Fast Software Encryption Lecture Notes in Computer Science, Vol. 2355. Springer-Verlag, Berlin Heidelberg New York (2001) 286-299.
5. S. Lee, S. Hong, S. Lee, J. Lim and S. Yoon.: Truncated Differential Cryptanalysis of Camellia. In: ICISC 2001, Lecture Notes in Computer Science, Vol. 2288. Springer-Verlag, Berlin Heidel-berg New York (2001).32-38.
6. M. Sugita, K. Kobara and H. Imai.: Security of Reduced Version of the Block Cipher Camellia against Truncated and Impossible Differential Cryptanalysis. In: ASIACRYPT 2001, Lecture Notes in Computer Science, Vol. 2248. Springer-Verlag, Berlin Heidelberg New York (2001) 193-207.
7. T. Kawabata and T. Kaneko.: A Study on Higher Order Differential Attack of Camellia.: The 2nd open NESSIE workshop (2001).

8. J. Daemen, L. R. Knudsen and V. Rijmen.: The Block Cipher SQUARE. In Fast Software En-cryption, Lecture Notes in Computer Science, Vol. 1267. Springer-Verlag, Berlin Heidelberg New York (1997) 149-165.
9. Y.Hatano,H.Sekine, and T.Kaneko.: Higher order differential attack of Camellia(II). In: Proceed-ings of Selected Areas in Cryptography-SAC'02, Lecture Notes in Computer Science, Vol. 2595. Springer-Verlag, Berlin Heidelberg New York (2002) 39-56.
10. W.L.Wu,F. D.G. Feng: Collision attack on reduced-round Camellia, Science in China Series F-Information Sciences, 2005, Vol.48, No.1pp.78-90.
11. Joan Daemen and Vincent Rijmen, The Design of Rijndael, AES - The Advanced Encryption Standard, Springer-Verlag 2002, (238 pp.).
12. Niels Ferguson,John Kelsey,Stefan Lucks,Bruce Schneier,Mike Stay,David Wagner and Doug : Improved Cryptanalysis of Rijndael Whiting. In Proceedings of Fast Software Encryption-FSE'00, Vol 1978, Springer-Verlag, Berlin Heidelberg New York (2000) 213-230.