Kasumi Block Cipher

Data Encryptors

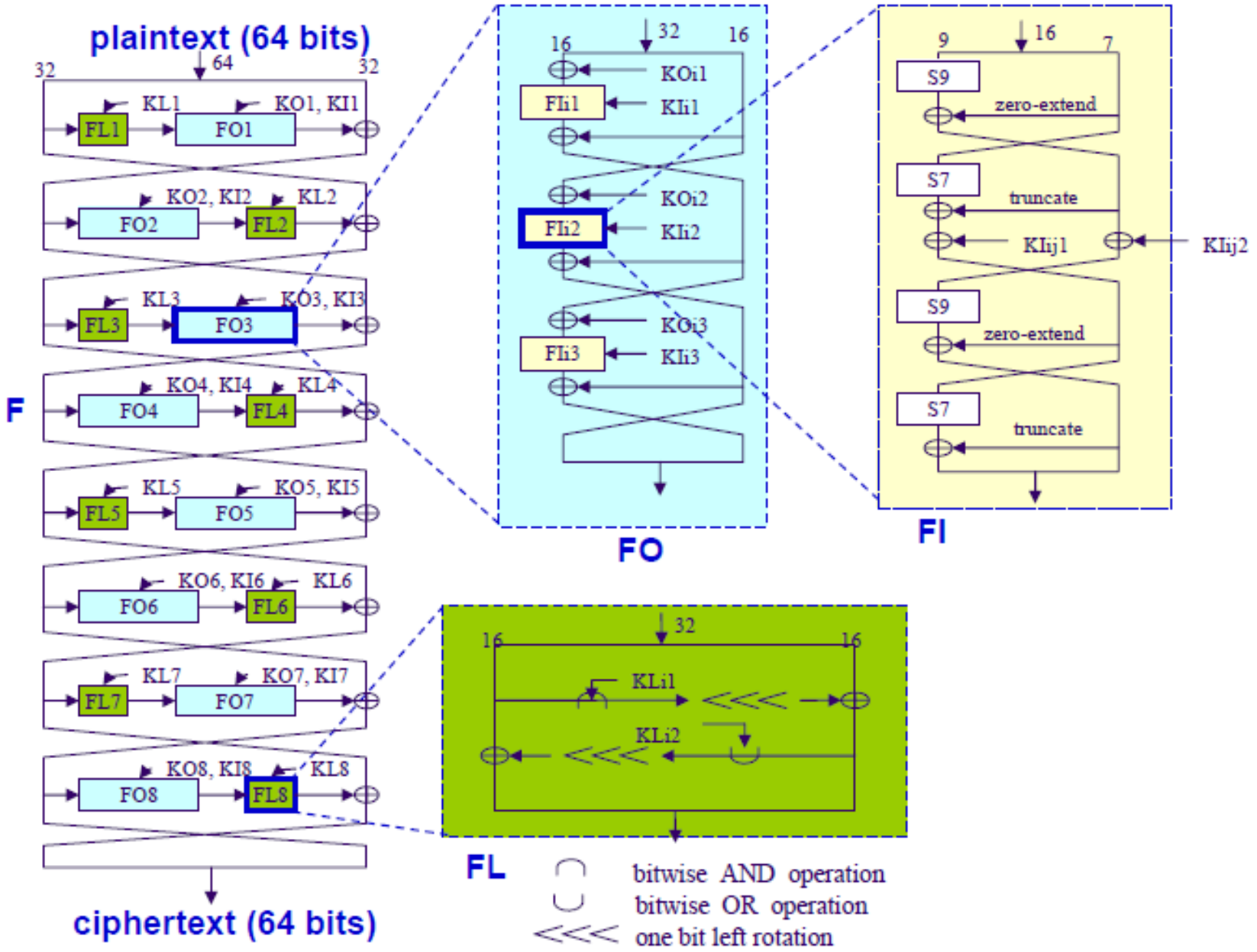
Darshan Gandhi
Rushabh Pasad

Introduction

- Used in 3GPP Confidentiality and Integrity Algorithms.

- Technical Specifications:
  - Key Size: 128 bits
  - Block Size: 64 bits (64 bit output from a 64 bit input).
  - Number of Rounds: 8

Overall Block Diagram



Components of Kasumi

Key Schedule

- The key, K, is 128 bits long.
- Each round of Kasumi uses 128 bit sub-key derived from K.

- Before generating the round keys, two 16-bit arrays, Kj,Kj' are derived as follows:
  - K is split into eight 16 bit values.. K1-K8.
  - Thus, K = K1 || K2 || K3 ||…|| K8.

  - Kj' = Kj $\oplus$ Cj , for each j = 1 to 8 and Cj is a constant value as defined below.

C1	0x0123
C2	0x4567
C3	0x89AB
C4	0xCDEF
C5	0xFEDC
C6	0xBA98
C7	0x7654
C8	0x3210

Components of Kasumi

Key Schedule (contd…)

- The round keys are derived from Kj and Kj' as follows:

	Round 1	Round 2	Round 3	Round 4	Round 5	Round 6	Round 7	Round 8
$KLi_{,1}$	K1<<<1	K2<<<1	K3<<<1	K4<<<1	K5<<<1	K6<<<1	K7<<<1	K8<<<1
$KL_{i,2}$	K3'	K4'	K5'	K6'	K7'	K8'	K1'	K2'
$KO_{i,1}$	K2<<<5	K3<<<5	K4<<<5	K5<<<5	K6<<<5	K7<<<5	K8<<<5	K1<<<5
$KO_{i,2}$	K6<<<8	K7<<<8	K8<<<8	K1<<<8	K2<<<8	K3<<<8	K4<<<8	K5<<<8
$KO_{i,3}$	K7<<<13	K8<<<13	K1<<<13	K2<<<13	K3<<<13	K4<<<13	K5<<<13	K6<<<13
$KI_{i,1}$	K5'	K6'	K7'	K8'	K1'	K2'	K3'	K4'
$KI_{i,2}$	K4'	K5'	K6'	K7'	K8'	K1'	K2'	K3'
$KI_{i,3}$	K8'	K1'	K2'	K3'	K4'	K5'	K6'	K7'

Note: <<<n => Left Circular Rotation of the operand by n bits.

Components of Kasumi

S-boxes

- This algorithm uses 2 S-boxes, S7 and S9.
- The values can be calculated using combinational logic for hardware based implementation or a lookup table for software based implementation. The decimal lookup tables are:

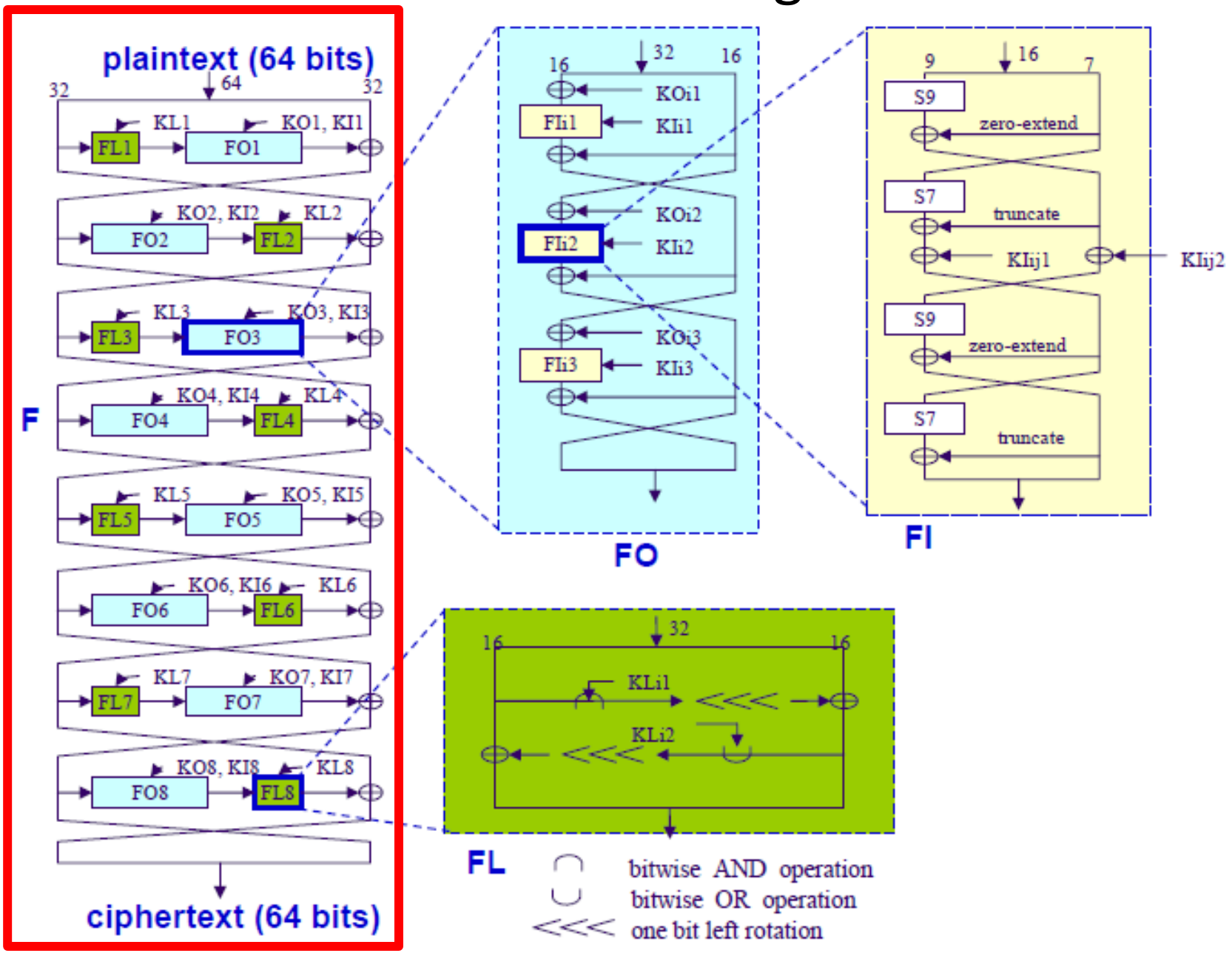
S7:

54,50,62,56,22,34,94,96,38,6,63,93,2,18,123,33,55,113,39,114,21,67,65,12,47,73,46,27,25,111,124,81,53,9,121,79,52,60,58,48,101,127,40,120,104,70,71,43,20,122,72,61,23,109,13,100,77,1,16,7,82,10,105,98,117,116,76,11,89,106,0,125,118,99,86,69,30,57,126,87,112,51,17,5,95,14,90,84,91,8,35,103,32,97,28,66,102,31,26,45,75,4,85,92,37,74,80,49,68,29,115,44,64,107,108,24,110,83,36,78,42,19,15,41,88,119,59,3

S9:

167,239,161,379,391,334,9,338,38,226,48,358,452,385,90,397,183,253,147,331,415,340,51,362,306,500,262,82,216,159,356,177,175,241,489,37,206,17,0,333,44,254,378,58,143,220,81,400,95,3,315,245,54,235,218,405,472,264,172,494,371,290,399,76,165,197,395,121,257,480,423,212,240,28,462,176,406,507,288,223,501,407,249,265,89,186,221,428,164,74,440,196,458,421,350,163,232,158,134,354,13,250,491,142,191,69,193,425,152,227,366,135,344,300,276,242,437,320,113,278,11,243,87,317,36,93,496,27,487,446,482,41,68,156,457,131,326,403,339,20,39,115,442,124,475,384,508,53,112,170,479,151,126,169,73,268,279,321,168,364,363,292,46,499,393,327,324,24,456,267,157,460,488,426,309,229,439,506,208,271,349,401,434,236,16,209,359,52,56,120,199,277,465,416,252,287,246,6,83,305,420,345,153,502,65,61,244,282,173,222,418,67,386,368,261,101,476,291,195,430,49,79,166,330,280,383,373,128,382,408,155,495,367,388,274,107,459,417,62,454,132,225,203,316,234,14,301,91,503,286,424,211,347,307,140,374,35,103,125,427,19,214,453,146,498,314,444,230,256,329,198,285,50,116,78,410,10,205,510,171,231,45,139,467,29,86,505,32,72,26,342,150,313,490,431,238,411,325,149,473,40,119,174,355,185,233,389,71,448,273,372,55,110,178,322,12,469,392,369,190,1,109,375,137,181,88,75,308,260,484,98,272,370,275,412,111,336,318,4,504,492,259,304,77,337,435,21,357,303,332,483,18,47,85,25,497,474,289,100,269,296,478,270,106,31,104,433,84,414,486,394,96,99,154,511,148,413,361,409,255,162,215,302,201,266,351,343,144,441,365,108,298,251,34,182,509,138,210,335,133,311,352,328,141,396,346,123,319,450,281,429,228,443,481,92,404,485,422,248,297,23,213,130,466,22,217,283,70,294,360,419,127,312,377,7,468,194,2,117,295,463,258,224,447,247,187,80,398,284,353,105,390,299,471,470,184,57,200,348,63,204,188,33,451,97,30,310,219,94,160,129,493,64,179,263,102,189,207,114,402,438,477,387,122,192,42,381,5,145,118,180,449,293,323,136,380,43,66,60,455,341,445,202,432,8,237,15,376,436,464,59,461

Overall Block Diagram



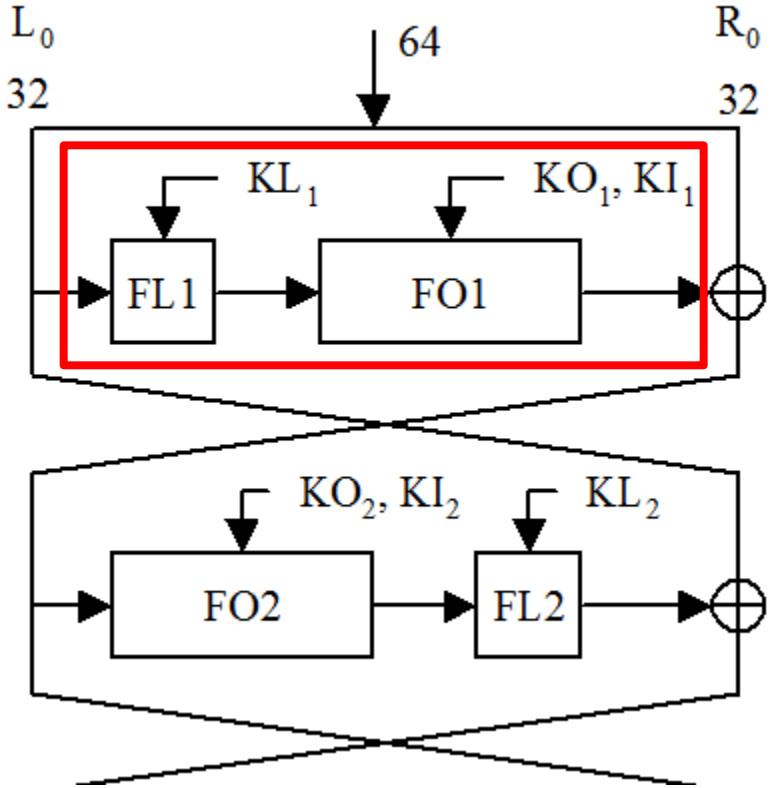
Components of Kasumi

Function f_i

- Input – 32 Bits of Data.
- Output – 32 Bits of Data.

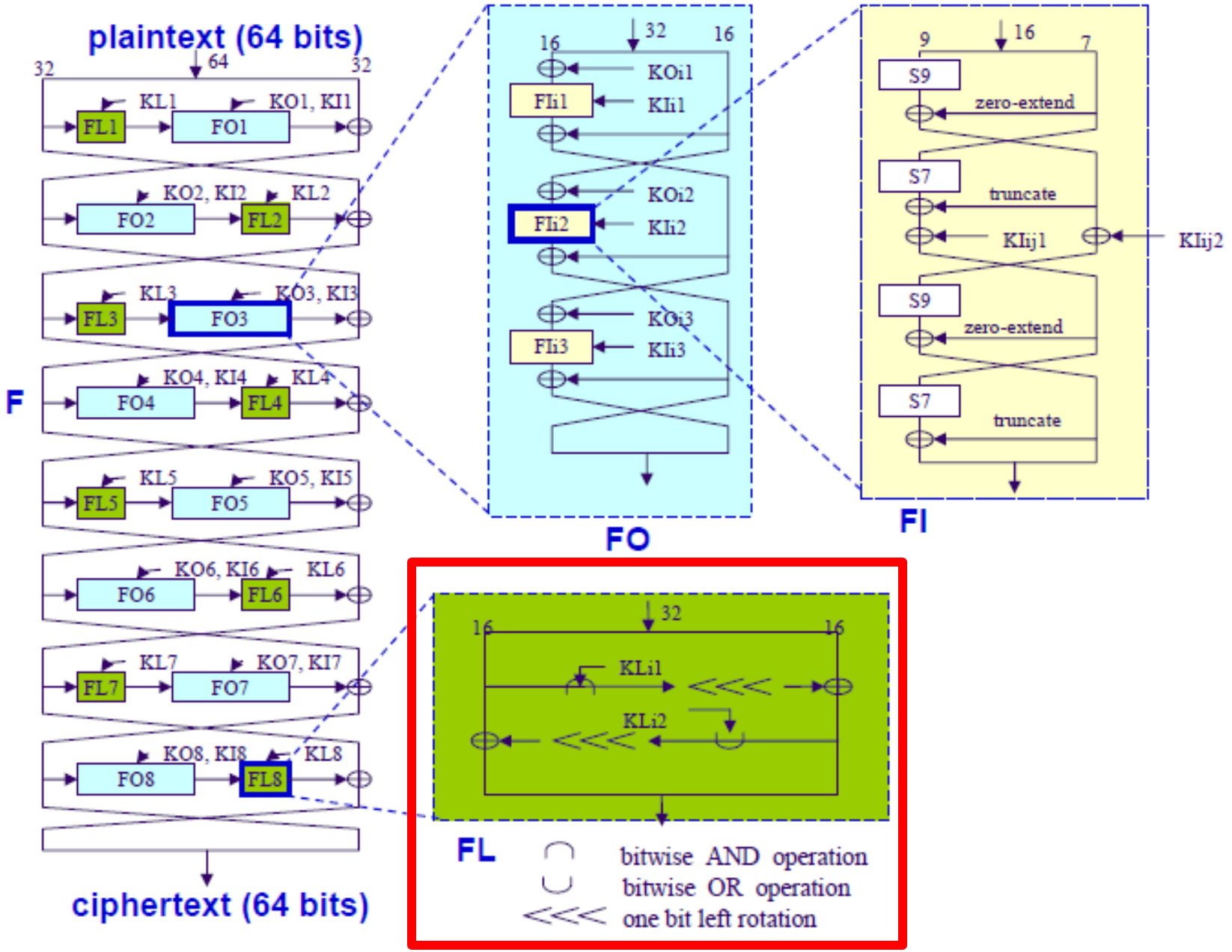
- For odd rounds, data is first passed through FL() and then FO().

- For even rounds, data is first passed through FO() and then FL().



Do for 8 rounds.

Overall Block Diagram



Components of Kasumi

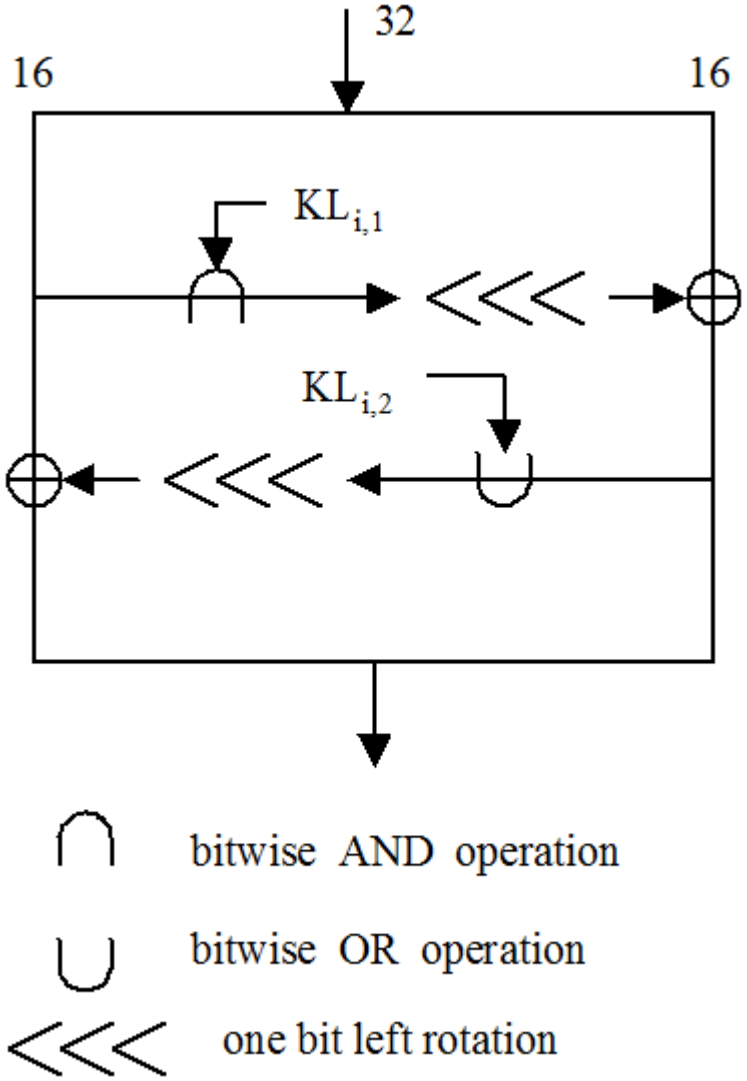
Function FL()

- Input – 32 Bits of Data, I, and 32 Bits of sub-key, KL.
- Output – 32 Bits of Data.

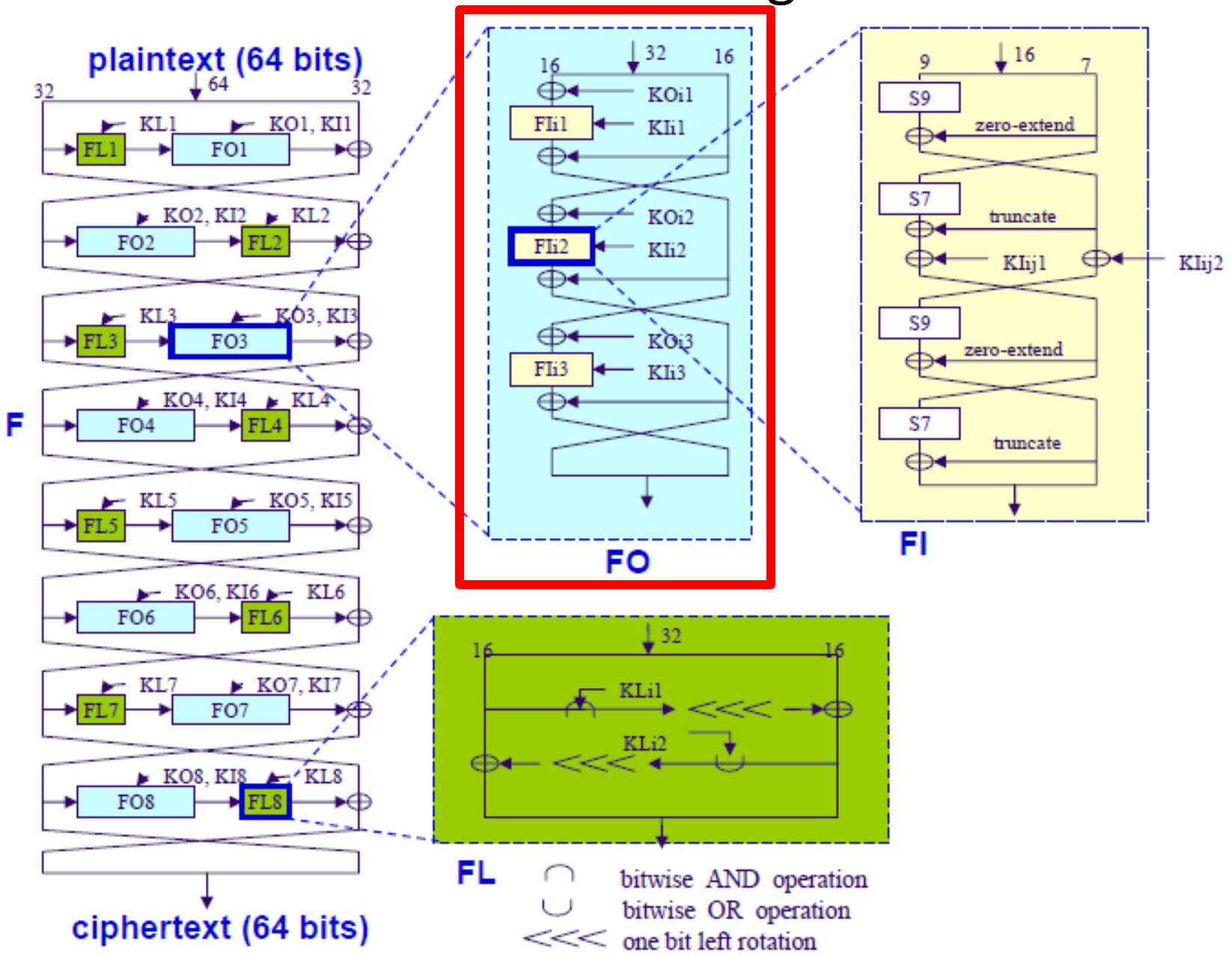
- Data is split into 2 halves of 16 bits, L and R.
  Thus, I = L || R.
- Key is split into 2 halves of 16 bits, $KL_{i,1}$ and $KL_{i,2}$.
  Thus, KL = $KL_{i,1}$ || $KL_{i,2}$.

- The operations are defined as:
  - R' = R $\oplus$ ROL(L $\cap$ $KL_{i,1}$).
  - L' = L $\oplus$ ROL(R $\cup$ $KL_{i,2}$).

- Output will be (L' || R').



Overall Block Diagram



Components of Kasumi

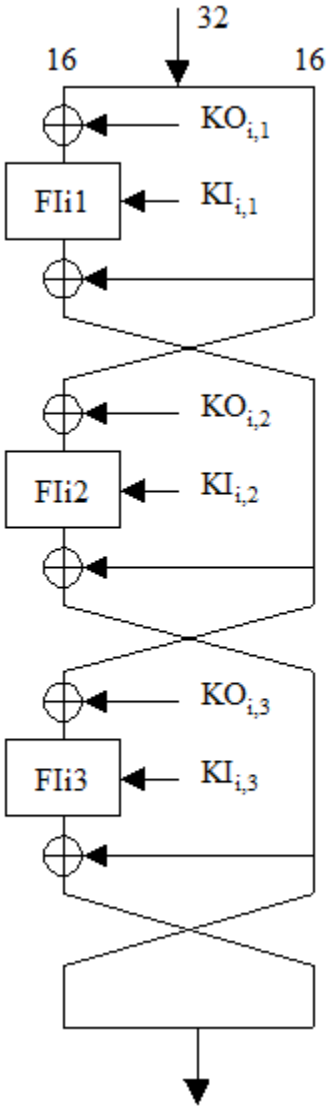
Function FO()

- Input – 32 Bits of Data, I, and two 48 Bit sub-keys, KO_i and KI_i.
- Output – 32 Bits of Data.

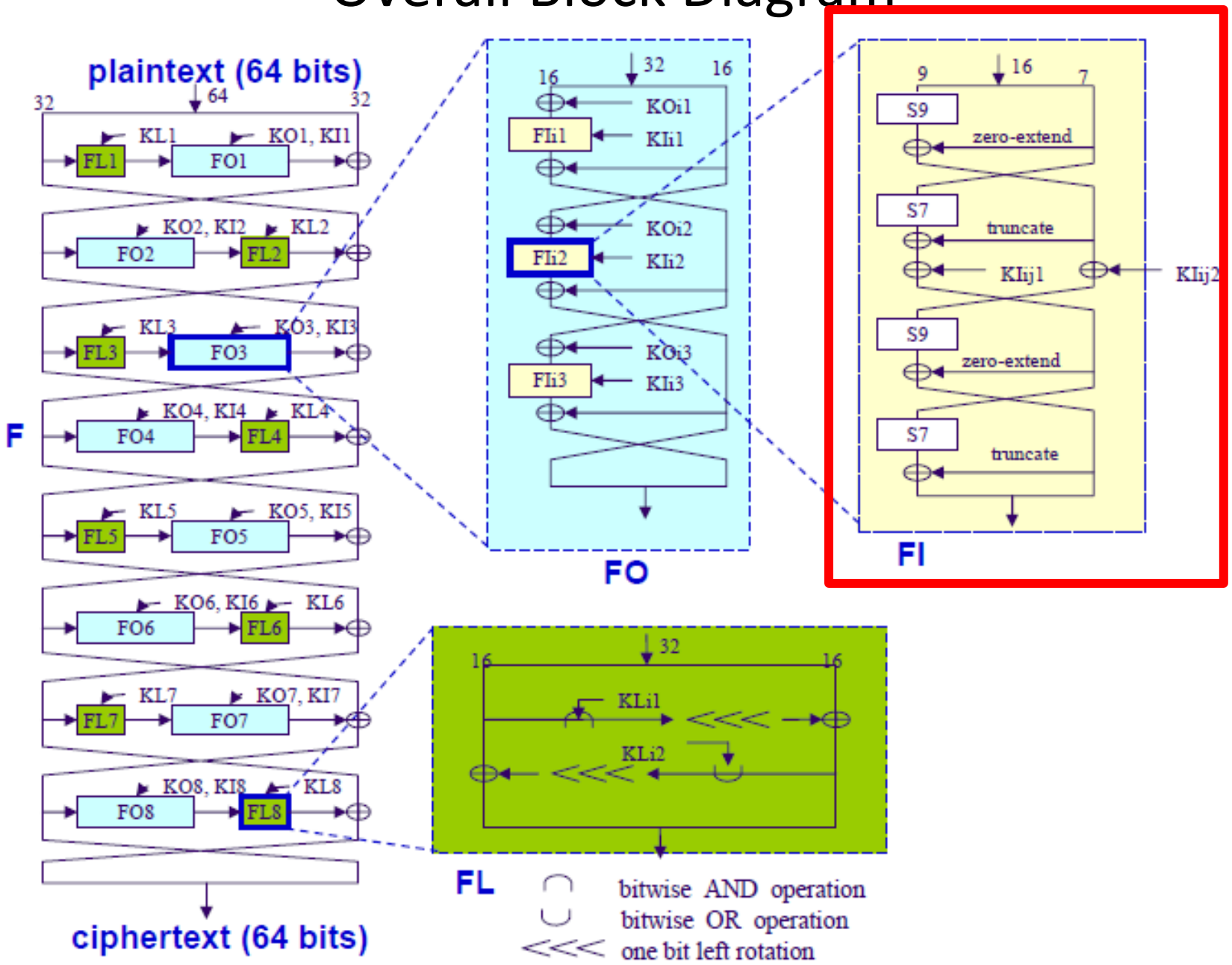
- Data is split into 2 halves of 16 bits, L and R.
  Thus, I = L || R.
- Both keys are split into three 16-bit keys.
  Thus, $KO_i = KO_{i,1} || KO_{i,2} || KO_{i,3}$ and
  $KI_i = KI_{i,1} || KI_{i,2} || KI_{i,3}$.

- The operations are defined as:
  - For each j = 0 to 3
    - $R_j = FI(L_{j-1} \oplus KO_{i,j}, KI_{i,j}) \oplus R_{j-1}$
    - $L_j = R_{j-1}$

- Output will be $(L_3 || R_3)$.



Overall Block Diagram



Components of Kasumi

Function FI()

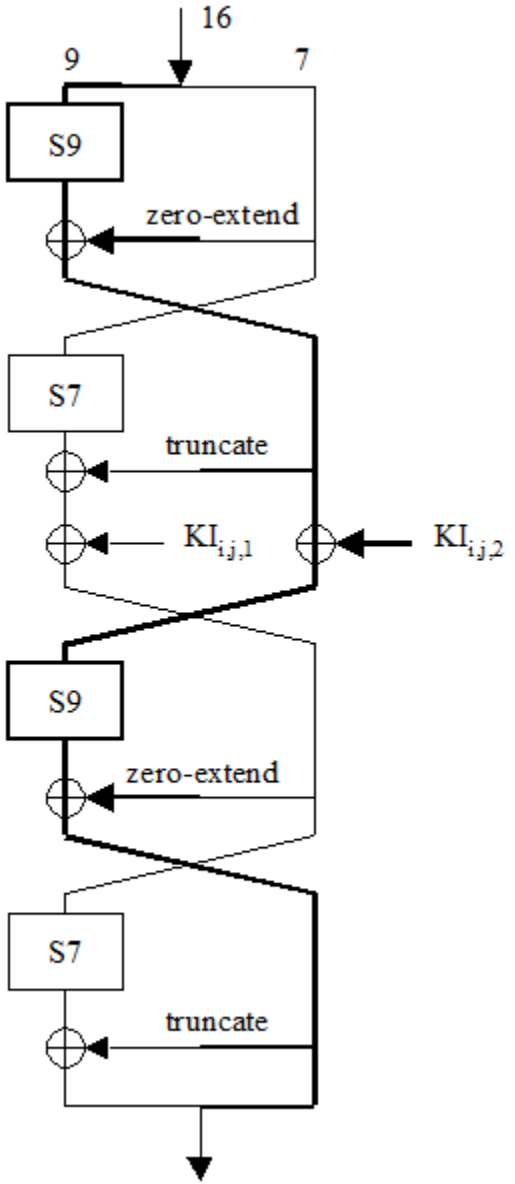
- Input – 16 Bits of Data, I, and 16 Bit sub-key $KI_{i,j}$.
- Output – 16 Bits of Data.

- Data is split into 2 unequal halves of 9 bits, L_0, and 7 bits, R_0.

  Thus, $I = L_0 \,||\, R_0$.
- Sub key is also split into 2 unequal halves of 7 bits, $KI_{i,j,1}$, and 9 bits, $KI_{i,j,2}$.

  Thus, $KI_{i,j} = KI_{i,j,1} \,||\, KI_{i,j,2} \,||\, KO_{i,3}$.

- The function uses two S-boxes:
  - S7: maps 7 bit input to 7 bit output.
  - S9: maps 9 bit input to 9 bit output.



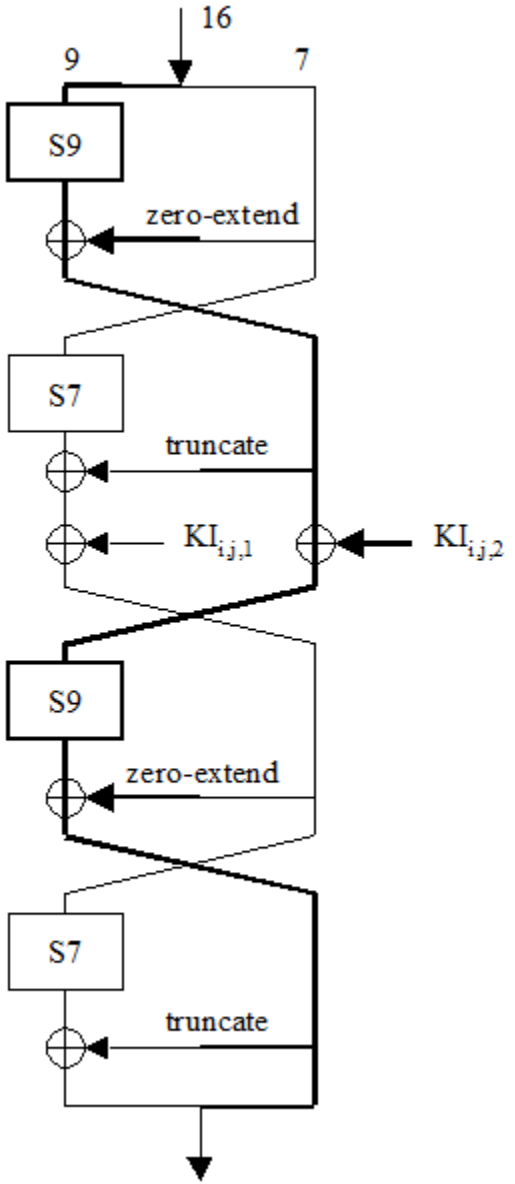
Components of Kasumi

Function FI() (contd…)

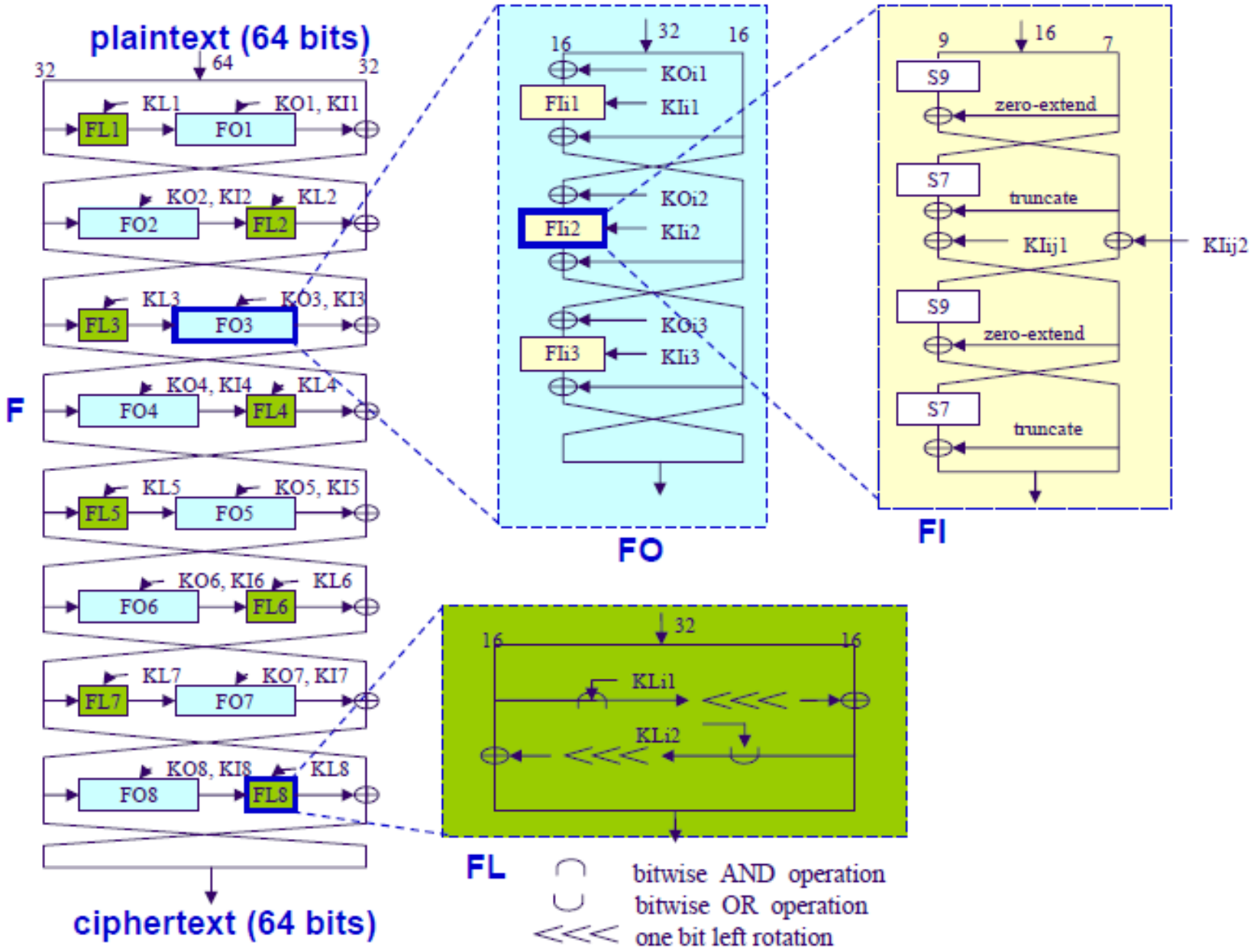
- This function also defines two more operations, zero extend(ZE) and truncate(TR).
  - ZE() – Adds two zero bits to the most-significant end.
  - TR() – Discards two most-significant bits.

- The operations are defined as:
  - $L_1 = R_0$; $R_1 = S9[L_0] \oplus ZE(R_0)$
  - $L_2 = R_1 \oplus KI_{i,j,2}$; $R_2 = S7[L_1] \oplus TR(R_1) \oplus KI_{i,j,1}$
  - $L_3 = R_2$; $R_3 = S9[L_2] \oplus ZE(R_2)$
  - $L_4 = S7[L_3] \oplus TR(R_3)$; $R_4 = R_3$

- Output will be $(L_4 \,||\, R_4)$.



Overall Block Diagram



Statistical Test Suite

- NIST
  - Developed to test randomness of binary sequences produced by hardware or software based cryptographic random or pseudorandom number generators.
  - Focuses on variety of different types of non-randomness that could exist in a sequence.
  - Consists of 15 sub-tests.

Statistical Test Suite(contd.)

1. The Frequency Test

   - Also known as Monobit test.
   - Focuses on the proportion of zeros and ones for the entire sequence.
   - Determines whether the number of ones and zeros in a sequence are approximately same(as expected for a truly random sequence).
   - All subsequent tests depend on the passing of this test.

Statistical Test Suite(contd.)

2. Frequency Test within a Block

   – Focuses on the proportion of ones within M-bit blocks.

   – Determines whether the frequency of ones in a M-bit block is approximately M/2(as would be expected under an assumption of randomness).

   – For block size M=1, it is same as Frequency Test.

Statistical Test Suite(contd.)

3. The Runs Test

   – Focuses on the total number of runs in the sequence, where a run is an uninterrupted sequence of identical bits.

   – Run of length *k* consists of *k* identical bits bounded by a bit of opposite value.

   – Determines whether the number of runs of ones and zeros of various lengths is as expected for a random sequence.

Statistical Test Suite(contd.)

4. Test for the Longest Run of Ones in a Block

   – Focuses on the longest run of ones within M-bit blocks.

   – Determines whether the length of the longest run of ones within the tested sequence is as expected for a random sequence.

   – Irregularity in the expected length of the longest run of ones implies that there is also an irregularity in the expected length of the longest run of zeros.

Statistical Test Suite(contd.)

5. The Binary Matrix Rank Test

   – Focuses on the rank of disjoint sub-matrices of the entire sequence.

   – Checks for linear dependency among fixed length substrings of the original sequence.

   – Appears in DIEHARD battery of tests.

Statistical Test Suite(contd.)

6. The Discrete Fourier Transform Test

   - Also Known as Spectral Test.
   - Focuses on the peak heights in the Discrete Fourier Transform of the sequence.
   - Detects periodic features in the tested sequence that would indicate a deviation from the assumption of randomness.
   - Intended to detect whether the number of peaks exceeding the 95% threshold is significantly different than 5%.

Statistical Test Suite(contd.)

7. The Non-overlapping Template Matching Test

   - Focuses on the number of occurrences of pre-specified target strings.
   - Detects generators that produce too many occurrences of a given non-periodic pattern.
   - Uses m-bit window to search for a specific m-bit pattern:
     - If pattern not found, window slides one bit position.
     - If pattern is found, window is reset to the bit after the found pattern and search resumes.

Statistical Test Suite(contd.)

7. The Non-overlapping Template Matching Test(contd.)

For example, if $\varepsilon = 10100100101110010110$, then $n = 20$. If $N = 2$ and $M = 10$, then the two blocks would be 1010010010 and 1110010110.

For the above example, if $m = 3$ and the template $B = 001$, then the examination proceeds as follows:

Bit Positions	Block 1		Block 2	
	Bits	W_1	Bits	W_2
1-3	101	0	111	0
2-4	010	0	110	0
3-5	100	0	100	0
4-6	001 (hit)	Increment to 1	001 (hit)	Increment to 1
5-7	Not examined		Not examined	
6-8	Not examined		Not examined	
7-9	001	Increment to 2	011	1
8-10	010 (hit)	2	110	1

Statistical Test Suite(contd.)

8. The Overlapping Template Matching Test

   – Focuses on the number of occurrences of pre-specified target strings.

   – Uses m-bit window to search for a specific m-bit pattern:
      - If pattern not found, window slides one bit position.
      - If pattern is found, window slides only one bit before resuming the search.

Statistical Test Suite(contd.)

8. The Overlapping Template Matching Test(contd.)

For the above example, if $m = 2$ and $B = 11$, then the examination of the first block (*1011101111*) proceeds as follows:

Bit Positions	Bits	No. of occurrences of $B = 11$
1-2	10	0
2-3	01	0
3-4	11 (hit)	Increment to 1
4-5	11 (hit)	Increment to 2
5-6	10	2
6-7	01	2
7-8	11 (hit)	Increment to 3
8-9	11 (hit)	Increment to 4
9-10	11 (hit)	Increment to 5

Statistical Test Suite(contd.)

9. Maurer's "Universal Statistical" Test

   - Focuses on the number of bits between matching patterns.
   - Detects whether or not the sequence can be significantly compressed without loss of information.
   - Significantly compressible sequence is considered to be non-random.

Statistical Test Suite(contd.)

10. The Linear Complexity Test

- Focuses on the length of a Linear Feedback Shift Register(LFSR).
- Determines whether or not the sequence is complex enough to be considered as random.
- Short LFSR implies non-randomness.

Statistical Test Suite(contd.)

11. The Serial Test

- Focuses on the frequency of all possible overlapping m-bit patterns across the entire sequence.
- Determines whether the number of occurrences of the 2^m m-bit overlapping patterns is approximately the same as would be expected for a random sequence.
- For m=1, equivalent to Frequency Test.

Statistical Test Suite(contd.)

12. The Approximate Entropy Test

- Focuses on the frequency of all possible overlapping m-bit patterns across the entire sequence.

- Compares the frequency of overlapping blocks of two adjacent lengths(against the expected result for a random sequence).

Statistical Test Suite(contd.)

13. The Cumulative Sums Test

- Also Known as Cusum.
- Focuses on the maximal excursion(from zero) of the random walk defined by the cumulative sum of adjusted (-1,+1) digits in the sequence.
- Determines whether the cumulative sum of the partial sequences is too large or too small(relative to the expected behavior for random sequences).

Statistical Test Suite(contd.)

14. The Random Excursions Test

- Focuses on the number of cycles having exactly K visits in a cumulative sum random walk.
- Determines if the number of visits to a particular state within the cycle deviates from what one would expect for a random sequence.
- Series of eight tests and conclusions. One test and conclusion for each of the states: -4, -3, -2, -1, +1, +2, +3, +4.

Statistical Test Suite(contd.)

15. The Random Excursions Variant Test

- Focuses on the total number of times that a particular state is visited in a cumulative sum random walk.
- Detects deviations from the expected number of visits to various states in the random walk.
- Series of eighteen tests and conclusions. One test and conclusion for each of the states: -9, -8,..., -1, +1, +2,…, +9.

Questions/Comments/Suggestions?