CA642 - Cryptography and Number Theory
Assignment Report

Li Zeyuan
zeyuan.li8@mail.dcu.ie
15210655

1 Linear Cryptanalysis of the FEAL-4 Block Cipher

1.1 Calculate equations for K0 and $\widetilde{K0}$

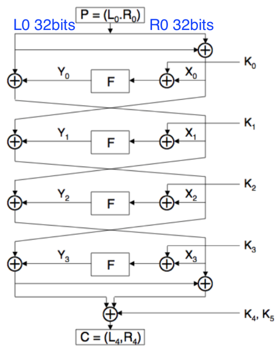


Fig. 1.

From the Fig.1, the FEAL-4 intermediate steps we found the relationship:

$$L0 \bigoplus R0 \bigoplus Y1 \bigoplus Y3 = K4 \bigoplus L4$$

$$S_{13}(Y) = S_{7,15}(X) \oplus S_{23,31}(X) \oplus 1$$
$$S_5(Y) = S_{15}(Y) \oplus S_7(X)$$
$$S_{15}(Y) = S_{21}(Y) \oplus S_{23,31}(X)$$
$$S_{23}(Y) = S_{29}(Y) \oplus S_{31}(X) \oplus 1$$

Fig. 2.

Firs we start with K0, apply the relationship to each formulation in Fig.2 get 4 unknown constant equations for K0 :

$$a = S23,29(L0 \bigoplus R0 \bigoplus L4) \bigoplus S31(L0 \bigoplus L4 \bigoplus R4) \bigoplus S31(F(L0 \bigoplus R0 \bigoplus K0))$$
$$b = S13(L0 \bigoplus R0 \bigoplus L4) \bigoplus S7,15,23,31(L0 \bigoplus L4 \bigoplus R4) \bigoplus S7,15,23,31(F(L0 \bigoplus R0 \bigoplus K0))$$
$$c = S5,15(L0 \bigoplus R0 \bigoplus L4) \bigoplus S7(L0 \bigoplus L4 \bigoplus R4) \bigoplus S7(F(L0 \bigoplus R0 \bigoplus K0))$$
$$d = S15,21(L0 \bigoplus R0 \bigoplus L4) \bigoplus S23,31(L0 \bigoplus L4 \bigoplus R4) \bigoplus S23,31(F(L0 \bigoplus R0 \bigoplus K0))$$

Then combine equation b,c,d we get another constant equation A:

$$S5,13(L0 \bigoplus R0 \bigoplus L4) \bigoplus S21(L0 \bigoplus R0 \bigoplus L4) \bigoplus S15(L0 \bigoplus R4 \bigoplus L4) \bigoplus S15(F(L0 \bigoplus R0 \bigoplus K0))$$

We compress 32-bit K0(b0,b1,b2,b3) to middle 16-bit $\widetilde{K0}$ = (0,a0,a1,0) where $a0 = b0 \bigoplus b1$ and $a1 = b2 \bigoplus b3$. In G1 function of the round function F : $y1 = G1(x0 \bigoplus x1, x2 \bigoplus x3)$. From this we know the part $S15(F(L0 \bigoplus R0 \bigoplus K0))$ of equation A depends only on bits 9 to 15 and 17 to 23. Also, from $S7(a \bigoplus b) = S7(a + b(mod 256))$ we know bit 9 and 17 are XORed in the $\widetilde{K0}$. So the constant equation A depends only on bits 10 to 15 and 18 to 23, 12bits in total, there's $2^{12} = 4096$ possible $\widetilde{K0}$.

1.2 Calculate equations for K1 and $\widetilde{K1}$

From the Fig.1, we found the relationship:
$L0 \bigoplus Y0 \bigoplus Y2 \bigoplus L4 \bigoplus K4 = K5 \bigoplus R4$

Again, we apply the relationship to each formulation in Fig.2 get 4 unknown constant equations for K1 :

$$e = S31(L0 \bigoplus R0) \bigoplus S31(F(K1 \bigoplus L0 \bigoplus F(L0 \bigoplus R0K0))) \bigoplus S23,29(F(L0 \bigoplus R0 \bigoplus K0)) \bigoplus 1$$

$$f = S23,31(L0 \bigoplus R0) \bigoplus S23,31(F(K1 \bigoplus L0 \bigoplus F(L0 \bigoplus R0K0))) \bigoplus S15,21(F(L0 \bigoplus R0 \bigoplus K0)) \bigoplus 1$$

$$g = S7(L0 \bigoplus R0) \bigoplus S7(F(K1 \bigoplus L0 \bigoplus F(L0 \bigoplus R0K0))) \bigoplus S5,15(F(L0 \bigoplus R0 \bigoplus K0)) \bigoplus 1$$

$$h = S7,15,23,31(L0 \bigoplus R0) \bigoplus S7,15,23,31(F(K1 \bigoplus L0 \bigoplus F(L0 \bigoplus R0K0))) \bigoplus S13(F(L0 \bigoplus R0 \bigoplus K0)) \bigoplus 1$$

Then combine equation f,g,h we get another constant equation Q:

$$S15(L0 \bigoplus R0) \bigoplus S15(F(K1 \bigoplus L0 \bigoplus F(L0 \bigoplus R0 \bigoplus K0))) \bigoplus S5,13,21(F(L0 \bigoplus R0 \bigoplus K0)) \bigoplus 1$$

Same like $\widetilde{K0}$, we compress K1 to $\widetilde{K1}$.

1.3 Implementation

Programming language : Java.
JDK version : 1.7
Development Platform: Eclipse Luna 4.4.2

Firstly, in the method "readPairs", read all 200 plaintext and ciphertext to line from the "know.txt", then convert them to 64-bit binary string and store in 2 array lists as strings.

Attack $\widetilde{K0}$ In method "attacK0T" of Class LinFealK0, generate all 4096 possible $\widetilde{K0}$ depends on bits 10 to 15 and 18 to 23, and store them in an array list. Then exhaustive search all $\widetilde{K0}$ in 200 plaintext and ciphertext pairs. Split plaintext to left half as L0 and right half as R0, ciphertext to left half as L4 and right half as R4. Calculate the constant equation A $= S5,13(L0 \bigoplus R0 \bigoplus L4) \bigoplus S21(L0 \bigoplus R0 \bigoplus L4) \bigoplus S15(L0 \bigoplus R4 \bigoplus L4) \bigoplus S15(F(L0 \bigoplus R0 \bigoplus K0))$, and count how many time it keeps constant value(0 or 1). When the count equals the number of pairs(200), store the $\widetilde{K0}$ as the candidate $\widetilde{K0}$.

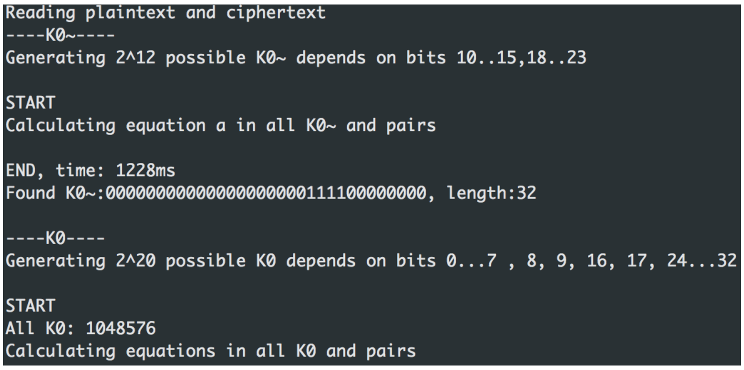


Fig. 3.

As the part of the result in Fig.3, the program took about 1.2 seconds to find the K0 . The bits 10 to 15 are 000000, bits 18-23 are 001111.

Attack K0 In method "attacK0" of Class LinFealK0, generate 2^{20} possible K0 depends on 20 bits 0 to 7, 8, 9, 16, 17, and 24 to 32. This time need to calculate constant equations a, b, c and d.

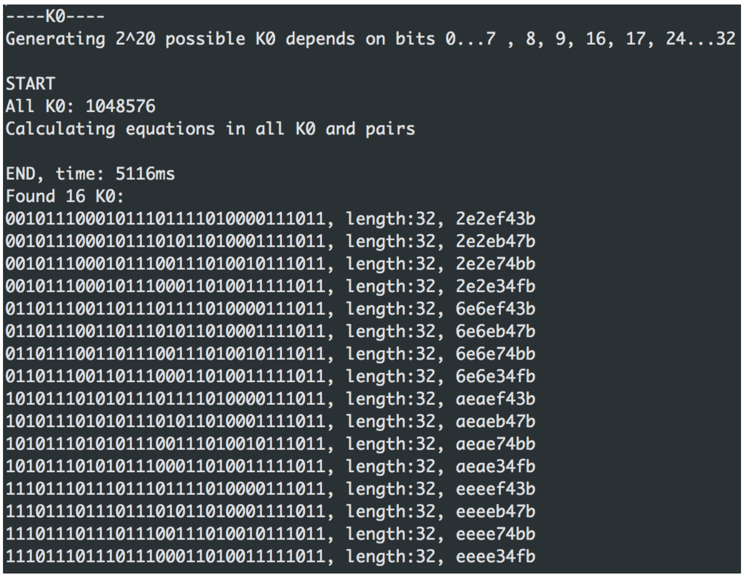


Fig. 4.

As the part of the result in Fig.4, the program took about 5.1 seconds to find 16 possible K0.

00101110001011101111010000111011, 2e2ef43b
00101110001011101011010001111011, 2e2eb47b
00101110001011100111010010111011, 2e2e74bb
00101110001011100011010011111011, 2e2e34fb
01101110011011101111010000111011, 6e6ef43b
01101110011011101011010001111011, 6e6eb47b
01101110011011100111010010111011, 6e6e74bb
01101110011011100011010011111011, 6e6e34fb
10101110101011101111010000111011, aeaef43b
10101110101011101011010001111011, aeaeb47b
10101110101011100111010010111011, aeae74bb
10101110101011100011010011111011, aeae34fb
11101110111011101111010000111011, eeeef43b
11101110111011101011010001111011, eeeeb47b
11101110111011100111010010111011, eeee74bb
11101110111011100011010011111011, eeee34fb

Attack $\widetilde{K1}$ In method "attacK1T" of Class LinFealK1, basically same idea as attack $\widetilde{K0}$, generate all 4096 possible $\widetilde{K1}$ depends on bits 10 to 15 and 18 to 23. Because the constant equation Q contains K0, the different here is exhaustive search all 4096 $\widetilde{K1}$ in 200 plaintext and ciphertext pairs and 16 K0 we found in last round.

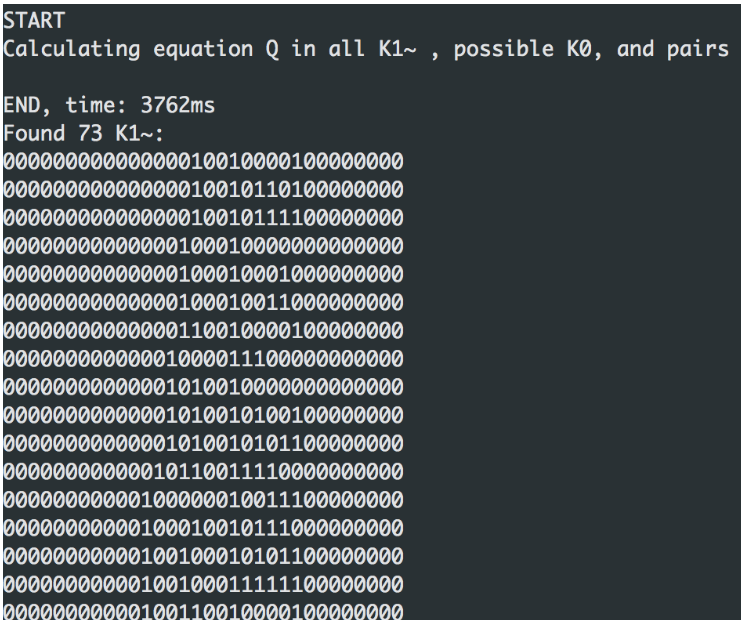


Fig. 5.

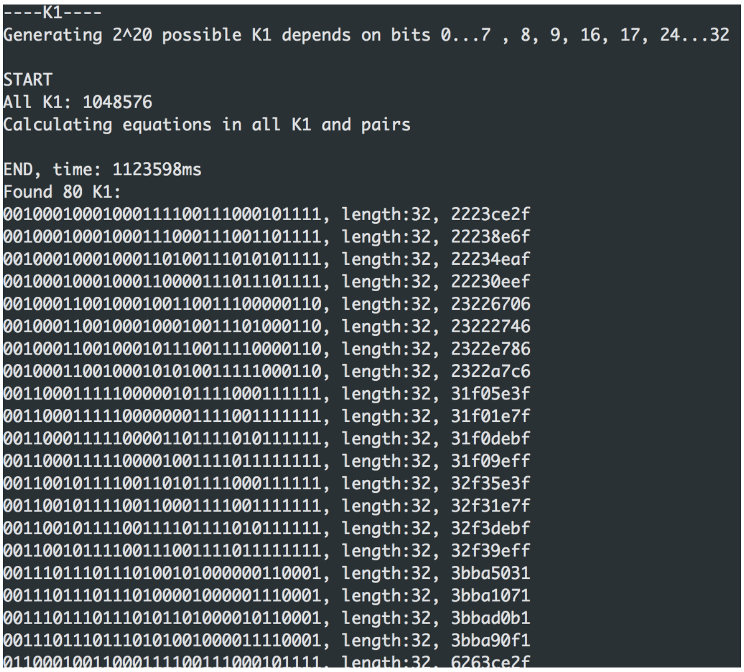
As the part of the result in Fig.4, the program took about 3.7 seconds to find 73 possible $\widetilde{K1}$.

1. 00000000000000010010000100000000
2. 00000000000000010010110100000000
3. 00000000000000010010111100000000
4. 00000000000000100010000000000000
5. 00000000000000100010001000000000
6. 00000000000000100010011000000000
7. 00000000000000110010000100000000
8. 00000000000001000011100000000000
9. 00000000000001010010000000000000
10. 00000000000001010010100100000000
11. 00000000000001010010101100000000
12. 00000000000010110011100000000000
13. 00000000000100000010011100000000
14. 00000000000100010010111000000000
15. 00000000000100100010101100000000
16. 00000000000100100011111100000000

17. 00000000000100110010000100000000
18. 00000000000100110010101100000000
19. 00000000000100110011111000000000
20. 00000000000101000010011000000000
21. 00000000000101000011001100000000
22. 00000000000101100010011100000000
23. 00000000000101100011000100000000
24. 00000000000101100011101100000000
25. 00000000000101110010010000000000
26. 00000000000101110011011100000000
27. 00000000000101110011101000000000
28. 00000000000110000010010000000000
29. 00000000000110010011110000000000
30. 00000000000110110010101000000000
31. 00000000000111010010001000000000
32. 00000000000111110010111100000000
33. 00000000001000000001010100000000
34. 00000000001000010001010000000000
35. 00000000001000010001011100000000
36. 00000000001000010001110000000000
37. 00000000001000100001110000000000
38. 00000000001000100001110100000000
39. 00000000001000100001111100000000
40. 00000000001000110000110000000000
41. 00000000001000110001011100000000
42. 00000000001001100000001000000000
43. 00000000001001100000101000000000
44. 00000000001001110000100000000000
45. 00000000001010000001101000000000
46. 00000000001010000001110000000000
47. 00000000001010010000010100000000
48. 00000000001010010000011100000000
49. 00000000001010100001101100000000
50. 00000000001010110001100000000000
51. 00000000001011000001010100000000
52. 00000000001011000001011100000000
53. 00000000001011000001110000000000
54. 00000000001011010000000100000000
55. 00000000001011010000001100000000
56. 00000000001011010001000000000000
57. 00000000001011010001010100000000
58. 00000000001011010001011000000000
59. 00000000001011100001000100000000
60. 00000000001011100001001100000000
61. 00000000001011100001110100000000
62. 00000000001101000000001000000000

63. 00000000001110010001100100000000
64. 00000000001110100001011100000000
65. 00000000001110100001100000000000
66. 00000000001110100001101000000000
67. 00000000001110100001111000000000
68. 00000000001110110001011000000000
69. 00000000001110110001100100000000
70. 00000000001111000000101100000000
71. 00000000001111010001100000000000
72. 00000000001111100001001100000000
73. 00000000001111110001001000000000

Attack K1 In method "attacK1" of Class LinFealK1, same as attack K0, generate 2^{20} possible K0 depends on 20 bits 0 to 7, 8, 9, 16, 17, and 24 to 32. This time need to calculate constant equations e, f, g and h. Again, because all constant equations contains K0, exhaustive search all K1 in 200 plaintext and ciphertext pairs and 16 K0 we found in last round.



```
----K1----
Generating 2^20 possible K1 depends on bits 0...7 , 8, 9, 16, 17, 24...32

START
All K1: 1048576
Calculating equations in all K1 and pairs

END, time: 1123598ms
Found 80 K1:
00100010001000111100111000101111, length:32, 2223ce2f
00100010001000111000111001101111, length:32, 22238e6f
00100010001000110100111010101111, length:32, 22234eaf
00100010001000110000111011101111, length:32, 22230eef
00100011001000100110011100000110, length:32, 23226706
00100011001000100010011101000110, length:32, 23222746
00100011001000101110011110000110, length:32, 2322e786
00100011001000101010011111000110, length:32, 2322a7c6
00110001111100000101111000111111, length:32, 31f05e3f
00110001111100000001111001111111, length:32, 31f01e7f
00110001111100001101111010111111, length:32, 31f0debf
00110001111100001001111011111111, length:32, 31f09eff
00110010111100110101111000111111, length:32, 32f35e3f
00110010111100110001111001111111, length:32, 32f31e7f
00110010111100111101111010111111, length:32, 32f3debf
00110010111100111001111011111111, length:32, 32f39eff
00111011101110100101000000110001, length:32, 3bba5031
00111011101110100001000001110001, length:32, 3bba1071
00111011101110101101000010110001, length:32, 3bbad0b1
00111011101110101001000011110001, length:32, 3bba90f1
01100010011000111100111000101111, length:32, 6263ce2f
```

Fig. 6.

As the part of the result in Fig.4, the program took about 1123 seconds to find 80 possible K1.

1. 00100010001000111100111000101111, length:32, 2223ce2f
2. 00100010001000111000111001101111, length:32, 22238e6f
3. 00100010001000110100111010101111, length:32, 22234eaf
4. 00100010001000110000111011101111, length:32, 22230eef
5. 00100011001000100110011100000110, length:32, 23226706
6. 00100011001000100010011101000110, length:32, 23222746
7. 00100011001000101110011110000110, length:32, 2322e786
8. 00100011001000101010011111000110, length:32, 2322a7c6
9. 00110001111100000101111000111111, length:32, 31f05e3f
10. 00110001111100000001111001111111, length:32, 31f01e7f
11. 00110001111100001101111010111111, length:32, 31f0debf
12. 00110001111100001001111011111111, length:32, 31f09eff
13. 00110010111100110101111000111111, length:32, 32f35e3f
14. 00110010111100110001111001111111, length:32, 32f31e7f
15. 00110010111100111101111010111111, length:32, 32f3debf
16. 00110010111100111001111011111111, length:32, 32f39eff
17. 00111011101110100101000000110001, length:32, 3bba5031
18. 00111011101110100001000001110001, length:32, 3bba1071
19. 00111011101110101101000010110001, length:32, 3bbad0b1
20. 00111011101110101001000011110001, length:32, 3bba90f1
21. 01100010011000111100111000101111, length:32, 6263ce2f
22. 01100010011000111000111001101111, length:32, 62638e6f
23. 01100010011000110100111010101111, length:32, 62634eaf
24. 01100010011000110000111011101111, length:32, 62630eef
25. 01100011011000100110011100000110, length:32, 63626706
26. 01100011011000100010011101000110, length:32, 63622746
27. 01100011011000101110011110000110, length:32, 6362e786
28. 01100011011000101010011111000110, length:32, 6362a7c6
29. 01110001101100000101111000111111, length:32, 71b05e3f
30. 01110001101100000001111001111111, length:32, 71b01e7f
31. 01110001101100001101111010111111, length:32, 71b0debf
32. 01110001101100001001111011111111, length:32, 71b09eff
33. 01110010101100110101111000111111, length:32, 72b35e3f
34. 01110010101100110001111001111111, length:32, 72b31e7f
35. 01110010101100111101111010111111, length:32, 72b3debf
36. 01110010101100111001111011111111, length:32, 72b39eff
37. 01111011111110100101000000110001, length:32, 7bfa5031
38. 01111011111110100001000001110001, length:32, 7bfa1071
39. 01111011111110101101000010110001, length:32, 7bfad0b1
40. 01111011111110101001000011110001, length:32, 7bfa90f1
41. 10100010101000111100111000101111, length:32, a2a3ce2f
42. 10100010101000111000111001101111, length:32, a2a38e6f
43. 10100010101000110100111010101111, length:32, a2a34eaf
44. 10100010101000110000111011101111, length:32, a2a30eef
45. 10100011101000100110011100000110, length:32, a3a26706
46. 10100011101000100010011101000110, length:32, a3a22746

47. 10100011101000101110011110000110, length:32, a3a2e786
48. 10100011101000101010011111000110, length:32, a3a2a7c6
49. 10110001011100000101111000111111, length:32, b1705e3f
50. 10110001011100000001111001111111, length:32, b1701e7f
51. 10110001011100001101111010111111, length:32, b170debf
52. 10110001011100001001111011111111, length:32, b1709eff
53. 10110010011100110101111000111111, length:32, b2735e3f
54. 10110010011100110001111001111111, length:32, b2731e7f
55. 10110010011100111101111010111111, length:32, b273debf
56. 10110010011100111001111011111111, length:32, b2739eff
57. 10111011001110100101000000110001, length:32, bb3a5031
58. 10111011001110100001000001110001, length:32, bb3a1071
59. 10111011001110101101000010110001, length:32, bb3ad0b1
60. 10111011001110101001000011110001, length:32, bb3a90f1
61. 11100010111000111100111000101111, length:32, e2e3ce2f
62. 11100010111000111000111001101111, length:32, e2e38e6f
63. 11100010111000110100111010101111, length:32, e2e34eaf
64. 11100010111000110000111011101111, length:32, e2e30eef
65. 11100011111000100110011100000110, length:32, e3e26706
66. 11100011111000100010011101000110, length:32, e3e22746
67. 11100011111000101110011110000110, length:32, e3e2e786
68. 11100011111000101010011111000110, length:32, e3e2a7c6
69. 11110001001100000101111000111111, length:32, f1305e3f
70. 11110001001100000001111001111111, length:32, f1301e7f
71. 11110001001100001101111010111111, length:32, f130debf
72. 11110001001100001001111011111111, length:32, f1309eff
73. 11110010001100110101111000111111, length:32, f2335e3f
74. 11110010001100110001111001111111, length:32, f2331e7f
75. 11110010001100111101111010111111, length:32, f233debf
76. 11110010001100111001111011111111, length:32, f2339eff
77. 11111011011110100101000000110001, length:32, fb7a5031
78. 11111011011110100001000001110001, length:32, fb7a1071
79. 11111011011110101101000010110001, length:32, fb7ad0b1
80. 11111011011110101001000011110001, length:32, fb7a90f1