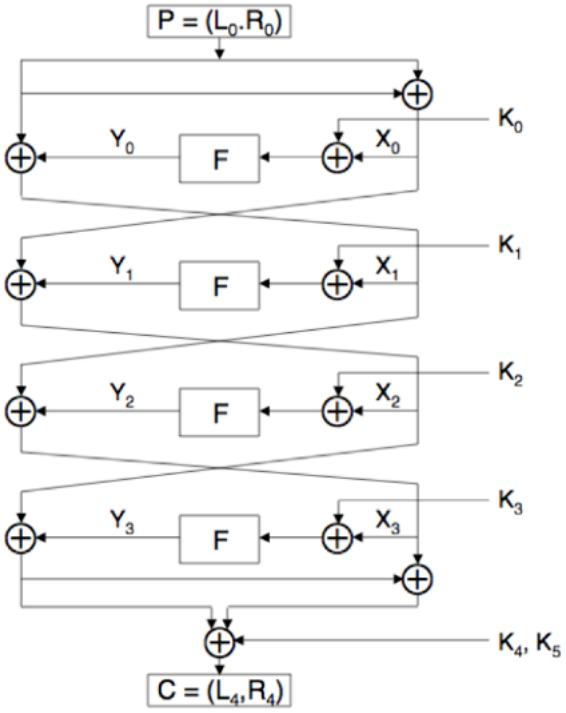
Linear Cryptanalysis of Feal 4

Name	Harrish Dhaithya Kutte Ajan Babu
Student ID	A00049171
Course Code	CSC1132

Introduction to Feal 4:

FEAL stands for Fast data Encryption Algorithm. It is a symmetric block cipher that is developed as a simple and efficient alternative of DES. It operates on 64-bit data blocks and uses a 64-bit key, applying four rounds of a Feistel-style structure to perform encryption and decryption.



The above block diagram represents the structure of FEAL-4. From this we can derive the below 2 equations.

L4 = X0^Y1^Y3^K4 ---------- (1)

R4 = L0^Y0^Y2^L4^K4^K5 -------(2)

Problem:

Given 200 plain texts and ciphertext pairs, we need to find the 6 keys used for encryption i.e. K0,K2,...,K5 using Linear Cryptanalysis.

Let F(X) = Y, where X is the input, Y is the output and F is FEAL-4 function.

We can say that the below equation always stands true for FEAL-4

S13(Y) = S7,15,23,31(X) ^ 1 ------(3)

S5,15(Y) = S7(X) ----------(4)

S15,21(Y)=S23,31(X)----------(5)

S23,29(Y)=S31(X)+1----------(6)

Attack K0:

Using (1) we can expand S13(L4) as,

S13(L4) = S13(X0)^S13(Y1)^S13(Y3)^S13(K4) ----- (7)

Now, we can expand each term in (7)

S13(X0) = S13(L0^R0)

S13(Y1) = S13 F(X1^K1), Expanded with FEAL structure

= S13 F(L0^Y0^K1)

= S7,15,23,31(L0) ^ S7,15,23,31(Y0) ^ S7,15,23,31(K1) ^ 1

= S7,15,23,31(L0) ^ S7,15,23,31 F(L0 ^ R0 ^ K0) ^ S7,15,23,31(K1) ^ 1

S13(Y3) = S7,15,23,31(X3^k3) ^ 1 , From (3)

= S7,15,23,31(L4^K4^R4^K5^K3) ^ 1

= S7,15,23,31(L4^R4)^S7,15,23,31(K3^K4^K5)^1

By substituting all these in (7), we get

S13(L4)= S13(L0^R0) ^ S7,15,23,31(L0) ^ S7,15,23,31 F(L0 ^ R0 ^ K0) ^ S7,15,23,31(K1)
^ S7,15,23,31(L4^R4)^S7,15,23,31(K3^K4^K5)^S13(K4)

By Taking all the Keys except K0 to the LHS, we get,

S13(K4)^S7,15,23,31(K1^K3^K4^K5) = S13(L0^R0^L4) ^S7,15,23,31(L0^L4^R4)

^S7,15,23,31 F(L0^R0^K0)

The above equation will always be constant for all 200 pair for candidates of k0

So, We can say as,

a = S13(L0^R0^L4) ^S7,15,23,31(L0^L4^R4) ^S7,15,23,31 F(L0^R0^K0) ----- (8)

similarly, we can derive other constants using (4) , (5) , (6)

b = S5,15(L0 ^R0 ^L4) ^ S7(L0 ^L4 ^R4) ^ S7 F(L0 ^R0 ^K0) ----------(9)

c=S15,21(L0^R0^L4)^S23,31(L0^L4^R4)^S23,31 F(L0^R0^K0)----------(10)

d=S23,29(L0^R0^L4)^S31(L4^R4^L0)^S31 F(L0^R0^K0)----------(11)

By performing XOR operation on (8),(9),(10), we get another constant

e=S5,15,21(L0^R0^L4)^S15(L0^L4^R4) ^S15 F(L0^R0^K0) ----------(12)

Instead of performing exhaustive search across 32 bits, we can first calculate the candidates of middle 12 bits i.e. 10…15 and 18…23 using (12) and extend the solution of outer 20 bits for selected candidates using (8). By doing this way, we get 16 possible candidates for k0.

Attack K1:

For identifying K1, we are expanding the S13(R4) in the similar way using (2).

S13(R4) = S13(L0)^S13(Y0)^S13(Y2)^S13(L4)^S13(K4)^S13(K5) ----------(13)

S13(Y0) = S13 F(X0^K0)

= S13 F(L0^R0^K0)

= S7,15,23,31(L0)^ S7,15,23,31(R0)^ S7,15,23,31(K0)^1

S13(Y2) = S13 F(X2^K2)

= S13 F(X0^Y1^K2)

= S7,15,23,31(L0^R0)^ S7,15,23,31(Y1)^ S7,15,23,31(K2)^1

= S7,15,23,31(L0^R0)^ S7,15,23,31 F(L0^F(L0^R0^K0)^K1)^ S7,15,23,31(K2)^1

By substituting in (13), we get

S13(R4) = S13(L0)^ S7,15,23,31(L0)^ S7,15,23,31(R0)^ S7,15,23,31(K0)^1

^ S7,15,23,31(L0^R0)^ S7,15,23,31 F(L0^F(L0^R0^K0)^K1)^ S7,15,23,31(K2)^1

^K13(K4^K5)

By Moving keys to LHS, we get,

S13(K4^K5)^S7,15,23,31(K0^K2)=S13(L0^L4^R4)^ S7,15,23,31 F(L0^F(L0^R0^K0)^K1)

Using this we can derive a constant,

a = S13(L0^L4^R4)^ S7,15,23,31 F(L0^F(L0^R0^K0)^K1) ----------(14)

On similar expansion of (3),(4),(5),

We can find other constants, as below

b = S5,15(L0^L4^R4)^ S7 F(L0^F(L0^R0^K0)^K1) ----------(15)

c = S15,21(L0^L4^R4)^ S23,31 F(L0^F(L0^R0^K0)^K1) ----------(16)

d = S23,39(L0^L4^R4)^ S31 F(L0^F(L0^R0^K0)^K1) ----------(17)

By performing XOR operation across (14),(15),(16), we can arrive with a new constant,

e = S5,13,21(L0^L4^R4)^ S15 F(L0^F(L0^R0^K0)^K1)----------(18)

Similar to K0, we can use (18) to calculate the inner 12 bit and (14) to calculate the other bits, by doing so, we will be getting 64 candidates for K1.

Attack K2:

For k2, we use the same equation used for cracking K0 (1) with a slight modification in the expansion of Y3.

S13(Y3) = S13 F(X3^K3)

= S13 F(Y2^X1^K3)

= S7,15,23,31(Y2) ^ S7,15,23,31(X1) ^ S7,15,23,31(K3)^1

= S7,15,23,31 F(X2^K2) ^ S7,15,23,31(L0^Y0) ^ S7,15,23,31(K3)^1

= S7,15,23,31 F(X0^F(X1^K1)^K2) ^ S7,15,23,31(L0^F(X0^K0)) ^ S7,15,23,31(K3)^1

= S7,15,23,31 F(L0^R0^F(L0^Y0^K1)^K2) ^ S7,15,23,31(L0^F(L0^R0^K0)) ^ S7,15,23,31(K3)^1

= S7,15,23,31 F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2) ^ S7,15,23,31(L0^F(L0^R0^K0)) ^ S7,15,23,31(K3)^1 ----------(19)

By applying this to (7), we get the following constant,

a = S13(L0^R0^L4) ^ S7,15,23,31 F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2) ----------(20)

on similar expansion of (3),(4),(5), we get

b = S5,15(L0^R0^L4) ^ S7 F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2) ----------(21)

c = S15,21(L0^R0^L4) ^ S23,31 F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2) ----------(22)

d = S23,39(L0^R0^L4) ^ S31 F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2) ----------(23)

By performing XOR across (19),(20) and (21), we can derive

e = S5,13,21(L0^R0^L4) ^ S15 F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2) ----------(24)

with equation (24), we can find candidates for inner 12 bits and with (19), we can expand for outer 20 bits with selected candidates. By doing this we will be getting 256 candidates for K2.

Attack K3:

For K3, we can expand (2) as K1, with a slight modification i.e. Y2 = F(L4^K4^Y3^K2)

S13(Y2) = S13 F(L4^K4^Y3^K2)

=S7,15,23,31(L4) ^ S5,7,23,31(K4) ^ S5,7,23,31(Y3) ^ S5,7,23,31(K2)

S5,7,23,31(Y3) = S5,7,23,31 F(X3^K3)

= S5,7,23,31 F(F(X0^Y1^K2)^L0^Y0^K3)

= S31 F(L0^ F(L0^R0^K0)^F(L0^R0^F(L0^F(L0^R0^K0)^K1)^K2)^K3)

By substituting this in (2), we get,

a = S13(L0 ^ L4 ^ R4) ^ S7,15,23,31(L0 ^ R0^L4) ^ S7,15,23,31
F(L0^F(L0^R0^K0)^F(L0^R0^F(L0^F(L0 ^ R0 ^ K0) ^ K1) ^ K2) ^ K3) ----------(25)

On similar expansion of (3),(4),(5), we get

b = S5,15(L0 ^ L4 ^ R4) ^ S7 (L0 ^ R0^L4) ^ S7 F(L0^F(L0^R0^K0)^F(L0^R0^F(L0^F(L0 ^ R0 ^ K0) ^ K1) ^ K2) ^ K3) ----------(26)

c = S15,21(L0 ^ L4 ^ R4) ^ S23,31 (L0 ^ R0^L4) ^ S23,31
F(L0^F(L0^R0^K0)^F(L0^R0^F(L0^F(L0 ^ R0 ^ K0) ^ K1) ^ K2) ^ K3) ----------(27)

d = S23,29(L0 ^ L4 ^ R4) ^ S31 (L0 ^ R0^L4) ^ S31
F(L0^F(L0^R0^K0)^F(L0^R0^F(L0^F(L0 ^ R0 ^ K0) ^ K1) ^ K2) ^ K3) ----------(28)

By performing (25),(26),(27), we can derive another constant,

e = S5,13,21(L0 ^ L4 ^ R4) ^ S15 (L0 ^ R0^L4) ^ S15
F(L0^F(L0^R0^K0)^F(L0^R0^F(L0^F(L0 ^ R0 ^ K0) ^ K1) ^ K2) ^ K3) ----------(29)

Same as k0 and k1 and k2, we can use (29) for inner 12 bit and (25) for outer 20 bit. By doing this, we will get 1024 candidates for k3.

K4 and K5:

Now, using k0, k1, and k2, we can derive k4 and k5 from (1) and (2) as below,

K4 = L4^ X0^Y1^Y3 = L4^L0^R0^Y1^Y3, From (1) ----------(30)

K5 = L0^Y0^Y2^L4^K4^R4, From (2) ----------(31)

After computing K4, K5 and validating, we will get 256 combination of Keys k1…k5

I have pasted few of them below.

0x63cab942 0xa0c541 0x4674095a 0x64204c03 0x4b37d10a 0xd0a24877

0x63cab942	0xa0c541	0x4674095a	0x6420cc83	0x4b37d108	0xd0a24875
0x63cab942	0xa0c541	0x4674095a	0xe4a04c03	0x4937d10a	0xd2a24877
0x63cab942	0xa0c541	0x4674095a	0xe4a0cc83	0x4937d108	0xd2a24875
0x63cab942	0xa0c541	0x467489da	0x64204c01	0x4b37d10a	0xd0a24875
0x63cab942	0xa0c541	0x467489da	0x6420cc81	0x4b37d108	0xd0a24877
0x63cab942	0xa0c541	0x467489da	0xe4a04c01	0x4937d10a	0xd2a24875

Other keys can be found in result.txt

By Examining the keys, we can observe the following pattern.

```
Key 0:
x1100011x1001010x0111001x1000010
Key 1:
x00000x0x0100000x1000101x10000x1
Key 2:
x10001x0x1110100x0001001x10110x0
Key 3:
x11001x0x0100000x1001100x00000x1
Key 4:
010010x10011011111010001000010x0
Key 5:
110100x0101000100100100011101x1
```

Where, x is 0/1.