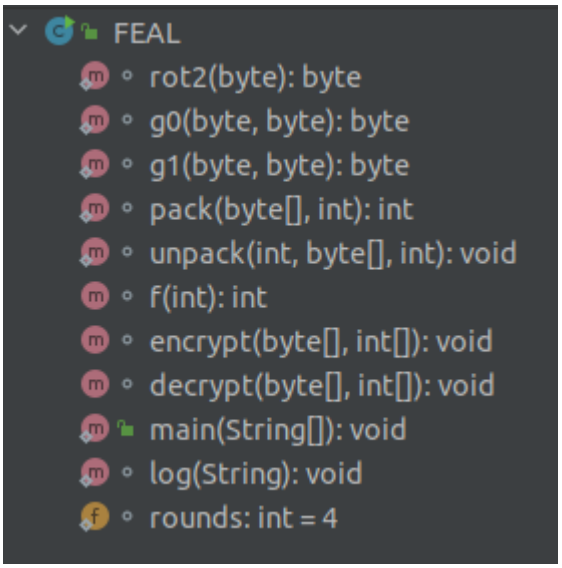
Name	**Vipul Popat**
Student #	**21267549**
Email	**vipul.popat2@mail.dcu.ie**
Program of Study	**Masters in Distributed Ledger Technologies (Blockchain)**
Module #	**CA642I**
Assignment	**Linear Cryptanalysis of FEAL-4 Cipher**
Date of Submission	**28th November 2021**
Word Count	**N/A**

Table of Contents

PROBLEM STATEMENT

FEAL.java - the java implementation of the Feistel FEAL-4 Cipher, which contained the following methods



Problem Statement:
This assignment will involve implementing a linear cryptanalysis attack on the weak block cipher FEAL-4, as described in the course, to find the six secret sub-keys that have been used. This is a VERY hard assignment and will take a considerable amount of time, so please start working on it as soon as possible.

The linear cryptanalysis attack can be implemented in the programming language of your choice. Your program will have to loop through a lot of different possible values, so it should be reasonably efficient. The source code for the FEAL-4 cipher (from which the six secret sub-keys have been removed) in C and Java is provided below in the files FEAL.c and FEAL.java, so you may wish to make use of this code and implement your attack in one of these languages. An executable version of this code which has the secret key built into it was used to generate the 200 random plaintext/ciphertext pairs which can be found below in the file known.txt.

Your task is to discover as many of the bits as possible of the six 32-bit sub-keys K0-K5 used in this cipher. The more bits, the more marks you will get. However, you will get some marks for even finding a few bits of the sub-keys, as this is a very difficult task. You should submit your code along with a written report describing how you went about the cryptanalysis and the results obtained through the Loop page for this module

SOLUTION

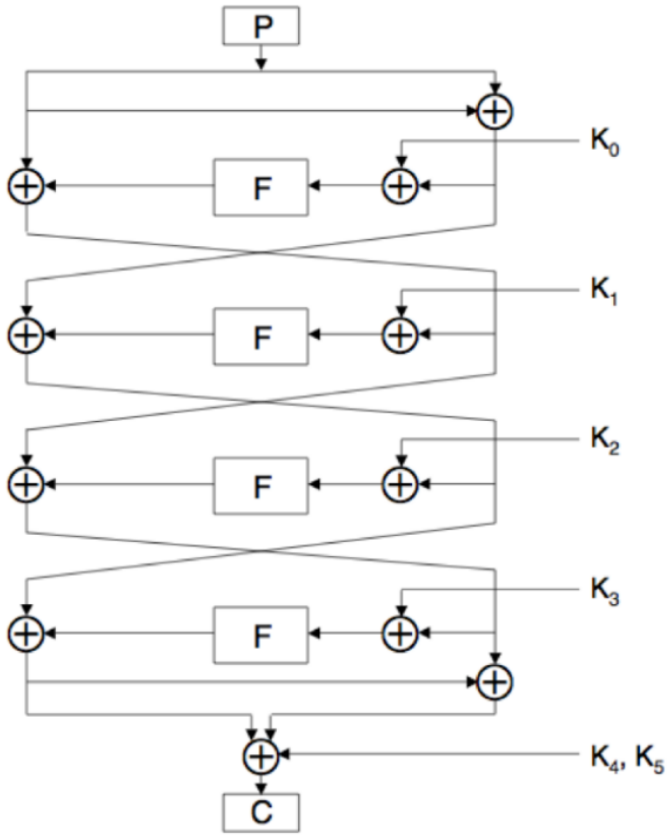


Figure a)

As mentioned in the loop under sections 2.9 - 2.12,

- The real work for this attack is 2^{32} encryptions, plus the work required for the linear attack, which is insignificant in comparison.
- A compression function can be defined.

  For a 32-bit word $W = (b_0, b_1, b_2, b_3)$, we define
  $M(W) = (0, b_0 \oplus b_1, b_2 \oplus b_3, 0)$

- This is used to squeeze all of the key data into the middle 16 bits.

$$K_0' = M(K_0) = (0, \langle K_0 \rangle_{0..7} \oplus \langle K_0 \rangle_{8..15}, \langle K_0 \rangle_{16..23} \oplus \langle K_0 \rangle_{24..31}, 0)$$

- We can then calculate this, which is going to be a constant,
  - $\texttt{constant} = S_{5,13,21}(L0 \oplus R0 \oplus L4) \oplus S_{15}(L0 \oplus L4 \oplus R4) \oplus S_{15}\ F(L0 \oplus R0 \oplus K0)$
- Due to the properties of the round function, we can replace k0 with k0'
- The remaining bits can also be found by deriving similar expressions and performing an exhaustive search. The overall work factor for this attack is far less than the 2^{32} required for the initial attack.

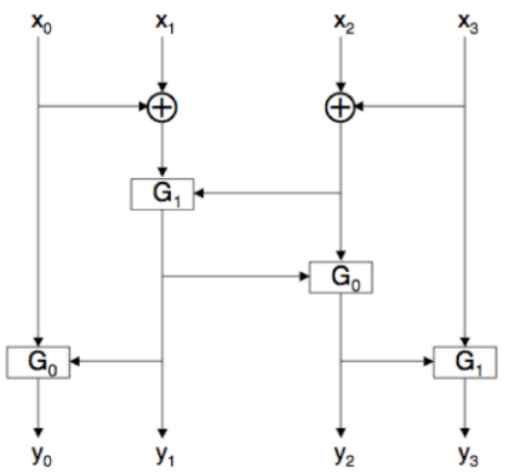


Figure b)

From Figure b) above	Transforming into bits	Move Y terms to the left and group the X's
$y_0 = G_0(x_0, y_1)$	$S_5(Y) = S_{15}(Y) \oplus S_7(X)$	$S_{5,15}(Y) = S_7(X)$
$y_1 = G_1(x_0 \oplus x_1, x_2 \oplus x_3)$	$S_{13}(Y) = S_{7,15,23,31}(X) \oplus 1$	$S_{13}(Y) = S_{7,15,23,31}(X) \oplus 1$
$y_2 = G_0(y_1, x_2 \oplus x_3)$	$S_{15}(Y) = S_{21}(Y) \oplus S_{23,31}(X)$	$S_{15,21}(Y) = S_{23,31}(X)$
$y_3 = G_1(y_2, x_3)$	$S_{23}(Y) = S_{29}(Y) \oplus S_{31}(X) \oplus 1.$	$S_{23,29}(Y) = S_{31}(X) \oplus 1$

Figure c) - the four equations in 3rd column referred to in ascending order as **c1, c2, c3 & c4**

First, we apply all the 4 mentioned formulae to the following relation
$L4 = X0 \oplus Y1 \oplus Y3 \oplus K4$ which is evident from the figure a)

Applying c1

$$S5,15(L4) = S5,15(X0) \oplus S5,15(Y1) \oplus S5,15(Y3) \oplus S5,15(K4)$$

$$S5,15(X0) = S5,15(L0 \oplus R0)$$

$$\begin{aligned} S5,15(Y1) &= S5,15(\mathbf{F}(X1 \oplus K1)) \\ &= S5,15(\mathbf{F}(L0 \oplus Y0 \oplus K1)) \\ &= S7(K1) \oplus \underline{S7(Y0)} \oplus S7(L0) \oplus 1 \end{aligned}$$

Elaborating S7(Y0)

$$\begin{aligned} S7(Y0) &= S7(\mathbf{F}(X0 \oplus K0)) \\ &= S7(\mathbf{F}(L0 \oplus R0 \oplus K0)) \end{aligned}$$

$$S5,15(Y1) = S7(K1) \oplus S7(\mathbf{F}(L0 \oplus R0 \oplus K0)) \oplus S7(L0) \oplus 1$$

$$\begin{aligned} S5,15(Y3) &= S7(X) \oplus 1 \\ &= S7(L4 \oplus K4 \oplus R4 \oplus K5 \oplus K3) \oplus 1 \\ &= S7(L4 \oplus R4) \oplus S7(K4 \oplus K5 \oplus K3) \oplus 1 \end{aligned}$$

Hence

$$\begin{aligned} S5,15(L4) = {} & S5,15(L0 \oplus R0) \oplus \\ & S7(K1) \oplus S7(\mathbf{F}(L0 \oplus R0 \oplus K0)) \oplus S7(L0) \oplus 1 \oplus \\ & S7(L4 \oplus R4) \oplus S7(K4 \oplus K5 \oplus K3) \oplus 1 \oplus \\ & S5,15(K4) \end{aligned}$$

$$\mathbf{constant\_c1} = S5,15(L0 \oplus R0 \oplus L4) \oplus S7(L0 \oplus L4 \oplus R4) \oplus S7(\mathbf{F}(L0 \oplus R0 \oplus K0))$$

Where $\mathbf{constant\_c1} = S7(K1 \oplus K4 \oplus K5 \oplus K3) \oplus S5,15(K4)$

Applying c2)

$$S13(L4) = S13(X0) \oplus S13(Y1) \oplus S13(Y3) \oplus S13(K4)$$

$$S13(X0) = S13(L0 \oplus R0)$$

$$\begin{aligned} S13(Y1) &= S13\ F(X1 \oplus K1) \\ &= S13\ F(L0 \oplus Y0 \oplus K1) \\ &= S7,15,23,31(K1) \oplus \underline{S7,15,23,31(Y0)} \oplus \\ &\quad S7,15,23,31(L0) \oplus 1 \end{aligned}$$

$$\begin{aligned} S7,15,23,31(Y0) &= S7,15,23,31(\mathbf{F}(X0 \oplus K0)) \\ &= S7,15,23,31(\mathbf{F}(L0 \oplus R0 \oplus K0)) \end{aligned}$$

Hence

$S13(Y1) = S7,15,23,31(K1) \oplus S7,15,23,31(\mathbf{F}(L0 \oplus R0 \oplus K0)) \oplus S7,15,23,31(L0) \oplus 1$

$S13(Y3) = S7,15,23,31(X) \oplus 1$
$= S7,15,23,31(L4 \oplus K4 \oplus R4 \oplus K5 \oplus K3) \oplus 1$
$= S7,15,23,31(L4 \oplus R4) \oplus S7,15,23,31(K4 \oplus K5 \oplus K3) \oplus 1$

Hence

$S13(L4) = S13(L0 \oplus R0) \oplus S7,15,23,31(K1) \oplus S7,15,23,31(\mathbf{F}(L0 \oplus R0 \oplus K0)) \oplus S7,15,23,31(L0) \oplus 1 \oplus S7,15,23,31(L4 \oplus R4) \oplus S7,15,23,31(K4 \oplus K5 \oplus K3) \oplus 1 \oplus S13(K4)$

$\mathbf{constant\_c2} = S13(L0 \oplus R0 \oplus L4) \oplus S7,15,23,31(L0 \oplus L4 \oplus R4) \oplus S7,15,23,31\ F(L0 \oplus R0 \oplus K0)$

Where $\mathbf{constant\_c2} = S7,15,23,31(K1 \oplus K4 \oplus K5 \oplus K3) \oplus S13(K4)$

Applying c3)

$S15,21(L4) = S15,21(X0) \oplus S15,21(Y1) \oplus S15,21(Y3) \oplus S15,21(K4)$

$S15,21(X0) = S15,21(L0 \oplus R0)$

$S15,21(Y1) = S15,21(\mathbf{F}(X1 \oplus K1))$
$= S15,21(\mathbf{F}(L0 \oplus Y0 \oplus K1))$
$= S23,31(K1) \oplus \underline{S23,31(Y0)} \oplus S23,31(L0) \oplus 1$

$\underline{S23,31(Y0)} = S23,31(\mathbf{F}(X0 \oplus K0))$
$= S23,31(\mathbf{F}(L0 \oplus R0 \oplus K0))$

$S15,21(Y3) = S23,31(X3) \oplus 1$
$= S23,31(L4 \oplus K4 \oplus R4 \oplus K5 \oplus K3) \oplus 1$
$= S23,31(L4 \oplus R4) \oplus S23,31(K4 \oplus K5 \oplus K3) \oplus 1$

On the similar lines of above

$\mathbf{constant\_c3} = S15,21(L0 \oplus R0 \oplus L4) \oplus S23,31(L0 \oplus L4 \oplus R4) \oplus S23,31(\mathbf{F}(L0 \oplus R0 \oplus K0))$

Applying c4)

$$S_{23,29}(L4) = S_{23,29}(X0) \oplus S_{23,29}(Y1) \oplus S_{23,29}(Y3) \oplus S_{23,29}(K4)$$

$$S_{23,29}(X0) = S_{23,29}(L0 \oplus R0)$$

$$\begin{aligned} S_{23,29}(Y1) &= S_{23,29}(\mathbf{F}(X1 \oplus K1)) \\ &= S_{23,29}(\mathbf{F}(L0 \oplus Y0 \oplus K1)) \\ &= S_{31}(K1) \oplus \underline{S_{31}(Y0)} \oplus S_{31}(L0) \oplus 1 \end{aligned}$$

$$\begin{aligned} \underline{S_{31}(Y0)} &= S_{31}(\mathbf{F}(X0 \oplus K0)) \\ &= S_{31}(\mathbf{F}(L0 \oplus R0 \oplus K0)) \end{aligned}$$

$$\begin{aligned} S_{23,29}(Y3) &= S_{31}(X3 \oplus K3) \oplus 1 \\ &= S_{31}(L4 \oplus K4 \oplus R4 \oplus K5 \oplus K3) \oplus 1 \\ &= S_{31}(L4 \oplus R4) \oplus S_{31}(K4 \oplus K5 \oplus K3) \oplus 1 \end{aligned}$$

On the similar lines of above

$$\mathbf{constant\_c4} = S_{23,29}(L0 \oplus R0 \oplus L4) \oplus S_{31}(L4 \oplus R4 \oplus L0) \oplus S_{31}(\mathbf{F}(L0 \oplus R0 \oplus K0))$$

Combining the constants **constant_c1, constant_c2 & constant_c3** which pertain to the bits of k0' would result in the following

$$S_{5,13,21}(L0 \oplus R0 \oplus L4) \oplus S_{15}(L0 \oplus L4 \oplus R4) \oplus S_{15}\, F(L0 \oplus R0 \oplus K0)$$

We will use the above expression to determine the bits of k0'. This has been implemented in code using

```
public int evaluateConstant_k0_prime(PlainAndCipherText pair, int k0_prime)
```

We can use all the above constants for evaluating k0'. We need to generate all the possible values of k0' for the bits 10..15 && 18...23, $2^{12} = 4096$

This is calculated for all the plain text/ciphertext pairs, and the key for which the constant remains the same is saved as candidate keys for k0'

It would be logical to use **constant_c2** for the evaluation of the keys as it contains all the required bits of the outer two bytes we need to evaluate for k0 evaluation. This has been implemented in code using

```
public int evaluateConstant_k0(PlainAndCipherText pair, int k0)
```

K1 evaluation

From figure a), we can very well derive the following equation

$$L0 \oplus Y0 \oplus Y2 \oplus L4 \oplus K4 = K5 \oplus R4$$

Applying c4

$$S23,29(R4) = S23,29(L0) \oplus S23,29(Y0) \oplus S23,29(Y2) \oplus S23,29(L4) \oplus S23,29(K4) \oplus S23,29(K5)$$

Elaborating each of the terms above

$$S23,29(Y0) = S23,29(\mathbf{F}(X0 \oplus K0)) = S23,29(\mathbf{F}(L0 \oplus R0 \oplus K0)) = S31(L0) \oplus S31(R0) \oplus S31(K0) \oplus 1$$

$$S23,29(Y2) = S23,29(\mathbf{F}(X2 \oplus K2)) = S23,29(\mathbf{F}(X0 \oplus Y1 \oplus K2)) = S31(X0) \oplus S31(Y1) \oplus S31(K2) \oplus 1 = S31(L0 \oplus R0) \oplus \underline{S31(Y1)} \oplus S31(K2) \oplus 1$$

$$\underline{S31(Y1)} = S31(\mathbf{F}(X1 \oplus K1)) = S31(\mathbf{F}(L0 \oplus Y0 \oplus K1)) = S31(\mathbf{F}(L0 \oplus F(L0 \oplus R0 \oplus K0) \oplus K1))$$

Substituting the above into the main equation above and combining the terms, we get the following constant

$$\mathbf{constant\_c4} = S5,13,21(L0 \oplus L4 \oplus R4) \oplus S15(\mathbf{F}(L0 \oplus \mathbf{F}(L0 \oplus R0 \oplus K0) \oplus K1))$$

Following the same process after applying equations c1, c2 and c3, We would arrive at the following constants

$$constant\_c1 = S5,15(L0 \oplus L4 \oplus R4) \oplus S7\ F(L0 \oplus F(L0 \oplus R0 \oplus K0) \oplus K1)$$
$$constant\_c2 = S13(L0 \oplus L4 \oplus R4) \oplus S7,15,23,31\ F(L0 \oplus F(L0 \oplus R0 \oplus K0) \oplus K1)$$
$$constant\_c3 = S15,21(L0 \oplus L4 \oplus R4) \oplus S23,31\ F(L0 \oplus F(L0 \oplus R0 \oplus K0) \oplus K1)$$
$$constant\_c4 = S23,29(L0 \oplus L4 \oplus R4) \oplus S31\ F(L0 \oplus F(L0 \oplus R0 \oplus K0) \oplus K1)$$

We combine constant_c1, constant_c2 and constant_c3 to get

$$S5,13,21(L0 \oplus L4 \oplus R4) \oplus S15\ F(L0 \oplus F(L0 \oplus R0 \oplus K0) \oplus K1)$$

This constant has been implemented using this method in code.

```
public int evaluateConstant_k1_prime(PlainAndCipherText pair, int k0, int key)
```

We follow the same process to get all the possible values of k1' using the above constant. And then iterate on each of the saved values of k1' to get the k1 by evaluating the Constant_c2 below, which is a representation of the outer two bytes.

```
constant_c2 = S13(L0 ⊕ L4 ⊕ R4) ⊕
          S7,15,23,31 F(L0 ⊕ F(L0 ⊕ R0 ⊕ K0) ⊕ K1)
```

This constant has been implemented using this method in code.

```
public int evaluateConstant_k1(PlainAndCipherText pair, int k0, int key)
```

K2 evaluation

From figure a), we can very well derive the following equation

```
L4 = X0 ⊕ Y1 ⊕ Y3 ⊕ K4
```

Applying equations c1, c2, c3 and c4 results into the following four constants

```
constant_c1 = S23,29(L0 ⊕ R0 ⊕ L4) ⊕ S31(F(L0 ⊕ R0 ⊕
                (F(L0 ⊕ (F(L0 ⊕ R0 ⊕ K0)) ⊕ K1)) ⊕ K2))
constant_c2 = S13(L0 ⊕ R0 ⊕ L4) ⊕ S7,15,23,31(F(L0 ⊕ R0 ⊕
                (F(L0 ⊕ (F(L0 ⊕ R0 ⊕ K0) ⊕ K1))) ⊕ K2))
constant_c3 = S5,15(L0 ⊕ R0 ⊕ L4) ⊕ S7(F(L0 ⊕ R0 ⊕ (F(L0 ⊕
                (F(L0 ⊕ R0 ⊕ K0)) ⊕ K1)) ⊕ K2))
constant_c4 = S15,21(L0 ⊕ R0 ⊕ L4) ⊕ S23,31(F(L0 ⊕ R0 ⊕ (F(L0 ⊕
                (F(L0 ⊕ R0 ⊕ K0)) ⊕ K1)) ⊕ K2))
```

We combine constant_c1, constant_c2 and constant_c3 to get

```
constant = S5,13,21(L0 ⊕ R0 ⊕ L4) ⊕ S15(F(L0 ⊕ R0 ⊕ (F(L0 ⊕
           (F(L0 ⊕ R0 ⊕ K0) ⊕ K1))) ⊕ K2))
```

This constant has been implemented using this method in code.

```
public int evaluateConstant_k2_prime(PlainAndCipherText pair, int k0, int k1, int key)
```

We can use the equation for constant_c2 for evaluating the constant to evaluate k1

```
constant = S13(L0 ⊕ R0 ⊕ L4) ⊕ S7,15,23,31(F(L0 ⊕ R0 ⊕
                (F(L0 ⊕ (F(L0 ⊕ R0 ⊕ K0) ⊕ K1))) ⊕ K2))
```

This constant has been implemented using this method in code.

```
public int evaluateConstant_k2(PlainAndCipherText pair, int k0, int k1, int key)
```

K3 evaluation

From figure a), we can very well derive the following equation

```
L0 ⊕ Y0 ⊕ Y2 ⊕ L4 ⊕ K4 = K5 ⊕ R4
```

We will substitute Y2 here with `Y2 = F(L4 ⊕ K4 ⊕ Y3 ⊕ K2)`

Following the same process in all of the earlier steps, the constant to determine k3' would be

```
Constant = 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)
```

This constant has been implemented using this method in code

```
public int evaluateConstant_k3_prime(PlainAndCipherText pair, int k0, int k1, int k2, int key)
```

```
And the constant_c2 for evaluation of k3 would be
constant_c2 = 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)
```

This constant has been implemented using this method in code

```
public int evaluateConstant_k3(PlainAndCipherText pair, int k0, int k1, int k2, int key)
```

K4 evaluation

We can derive the following equation from figure a)

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

K5 evaluation

We can derive the following equation from figure a)

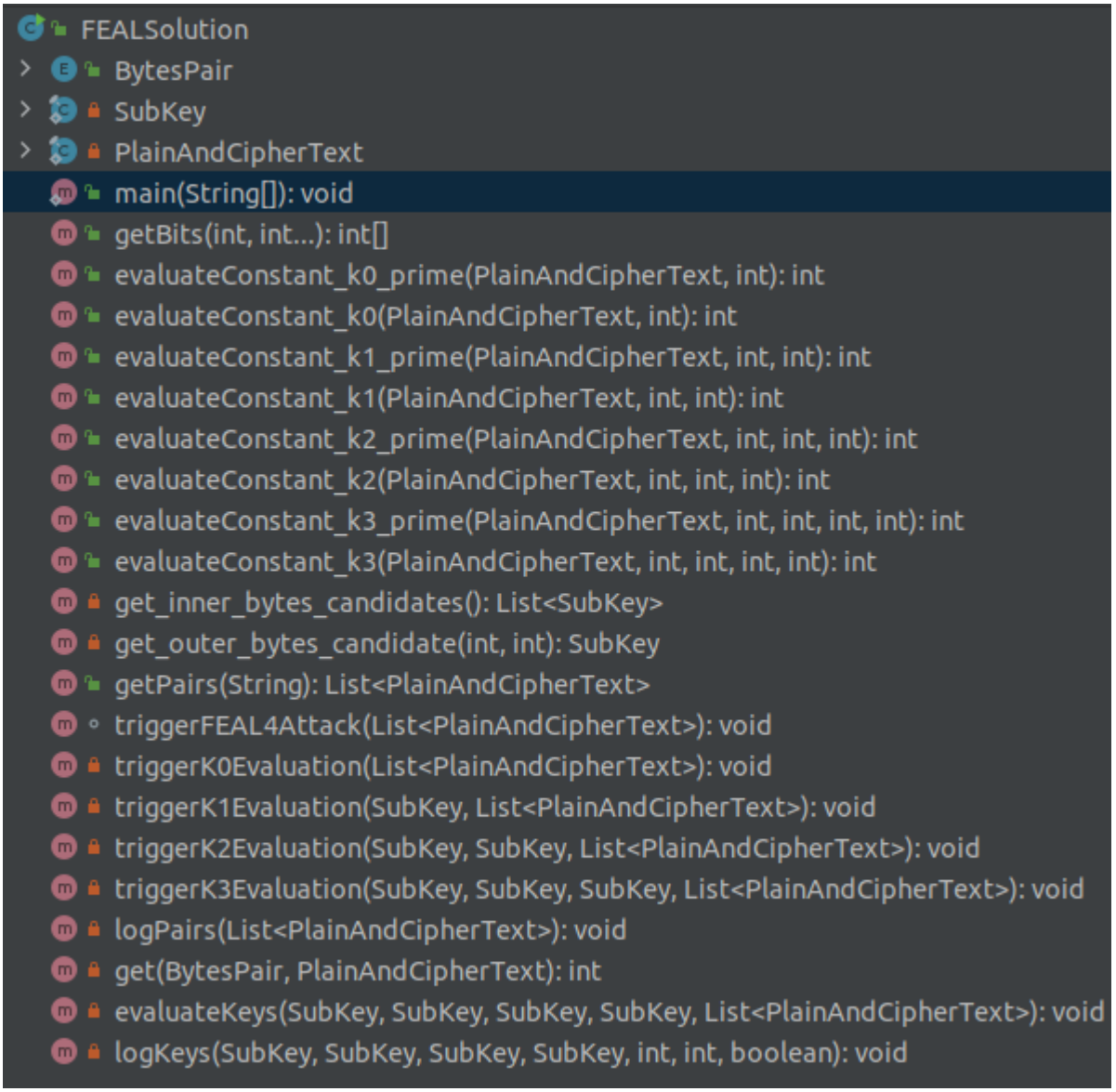
$$\begin{aligned} K5 &= L0 \oplus R0 \oplus Y1 \oplus Y3 \oplus L0 \oplus Y0 \oplus Y2 \oplus R4 \\ &= R0 \oplus Y1 \oplus Y3 \oplus Y0 \oplus Y2 \oplus R4 \end{aligned}$$

KEYS VALIDATION

Once we calculate all the keys, i.e. k0 .... k5, We would need only to use those keys which decrypt the ciphertext correctly.

INSTRUCTIONS TO EXECUTE

The solution has been implemented in **FEALSolution.java,** which **extends** the base functionality given in **FEAL.java**
The methods and inner classes to model keys and the Plain text ciphertext pairs are implemented in FEALSolution are as below.



To run the program

> java FEALSolution <full path to the known.txt file>

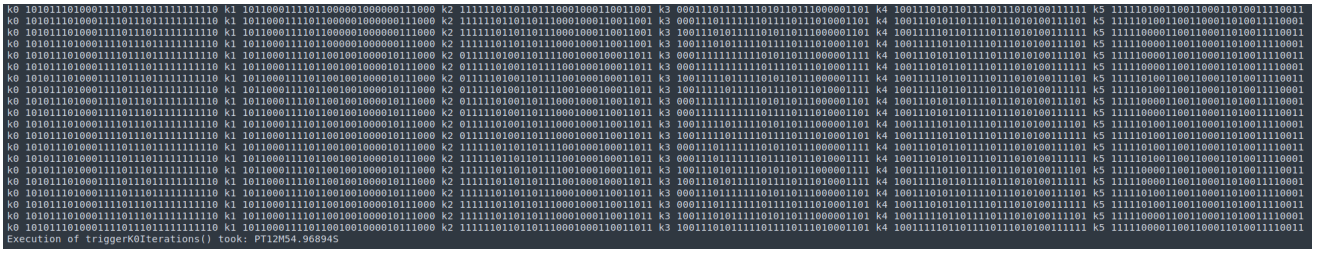
If you need to keys in hex/binary - it is just available with the toggle of a boolean flag in this line of code

```
logKeys(k0, k1, k2, k3, key4, key4, true);
```

RESULTS

Got 256 valid key combinations which were able to decrypt the ciphertext to plain text

Execution of triggerFEAL4Attack() took: PT10M59.344715S ~ **11 minutes**



I had gone through all the relevant keys and observed a pattern in the bits. The pattern is available in the table below

Key	32 bits Pattern	Unknown Bits Index
k0	**X**0101111**X**0100111**X**1001100**X**1000000	0,8,16,24
k1	**X**11111**X**1**X**1100011**X**1100010**X**10010**X**1	0,6,8,16,24,30
k2	**X**00100**X**1**X**0111110**X**1100011**X**10010**X**0	0,6,8,16,24,30
k3	**X**10010**X**1**X**1101111**X**0110101**X**11000**X**1	0,6,8,16,24,30
k4	100010**X**10101010011011001000101**X**0	6,30
k5	011001**X**01000001010110000011000**X**0	6,30
	X could be 0 or 1 dependent on the combination.	**From Figure c), it can be noticed that all the equations of constants were dependent on these bits.**

IMPORTANT CONSIDERATIONS

1. It was noticed that the iterations over plain text and cipher text for constant evaluations could be shortened if the previous constant evaluation does not equate to the current one as we need to check whether the constant is 1 or 0 for all the 200 pairs. This reduced a lot of computation time.
2. For the compressed keys, i.e. the prime k's, we just need to evaluate all the possible combinations of 12 bits and not the 16 bits, which reduce the computation time by a significant margin.