AES 256 (CBC)

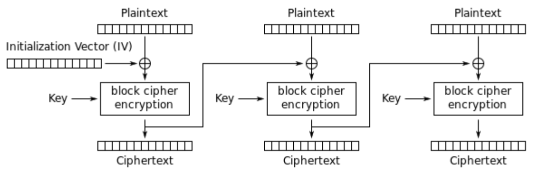
The program is an AES Encryptor/Decrypter with 256bits key. In Cipher Block Chaining mode.
The program aims to encrypt files or an entire folder with a specific key inserted by the user and a random initialization vector the program will encrypt and save the results in a new file or rewrite the main file if user wants, and will make a report file that contain the IV and key
the programs print the time taken to encrypt/decrypt . For decryption process the user is asked to enter the key and the IV as well then same procedures as encryption process.

Advanced Encryption Standard   Name: Omar Masoud   ID: 040190903
the project is completed individually.

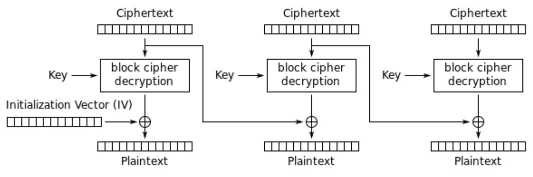
The program starts with asking the user whether to encrypt or decrypt then ask them to enter folder path, the programs will display the contents in the folder and ask user if they want to process a specific file or the entire folder, after that if the process is decryption the user will be asked to enter the initialization vector (16 letters) else the program will produce a random IV , then asks the user for the key (32 letters limit) since the AES works in CBC mode and 256 key bits,the program will start compiling the chosen file(s) and display the processing percentage of each file with a bar, after that a report with the time taken for the process, the used IV and the key entered, the user is also asked if they want to save the data in a new file or in the same file, after that a report file is generated of data of the file(s) as well as the encrypted/decrypted file
note: files shouldn't be bigger than 100kb

for CBC mod
Encryption, each cipher text is added to the next plain-text, for plain-text X0 IV is add, then file is encrypted

Decryption, same procedures but inversed



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

For the 256 bits, 60 arrays are created, W(0) to W(59)

Each W is created by XOR the previous W(i-1) with

With W(i-8),there are two functions g(x) and h(x)

h(x) substitution process

g(x) substitution, shifting and adding RC(i) to the most left

Byte process

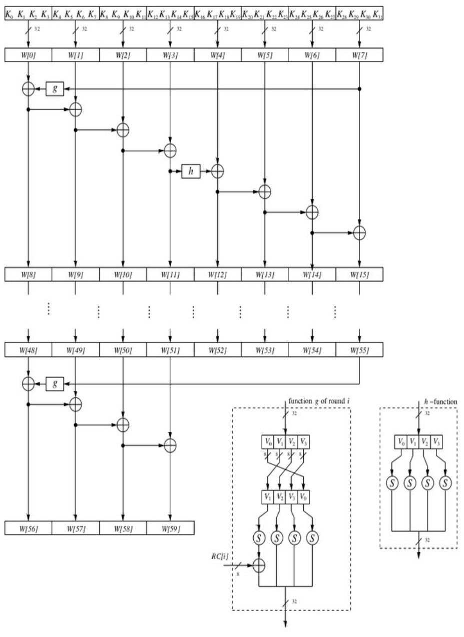
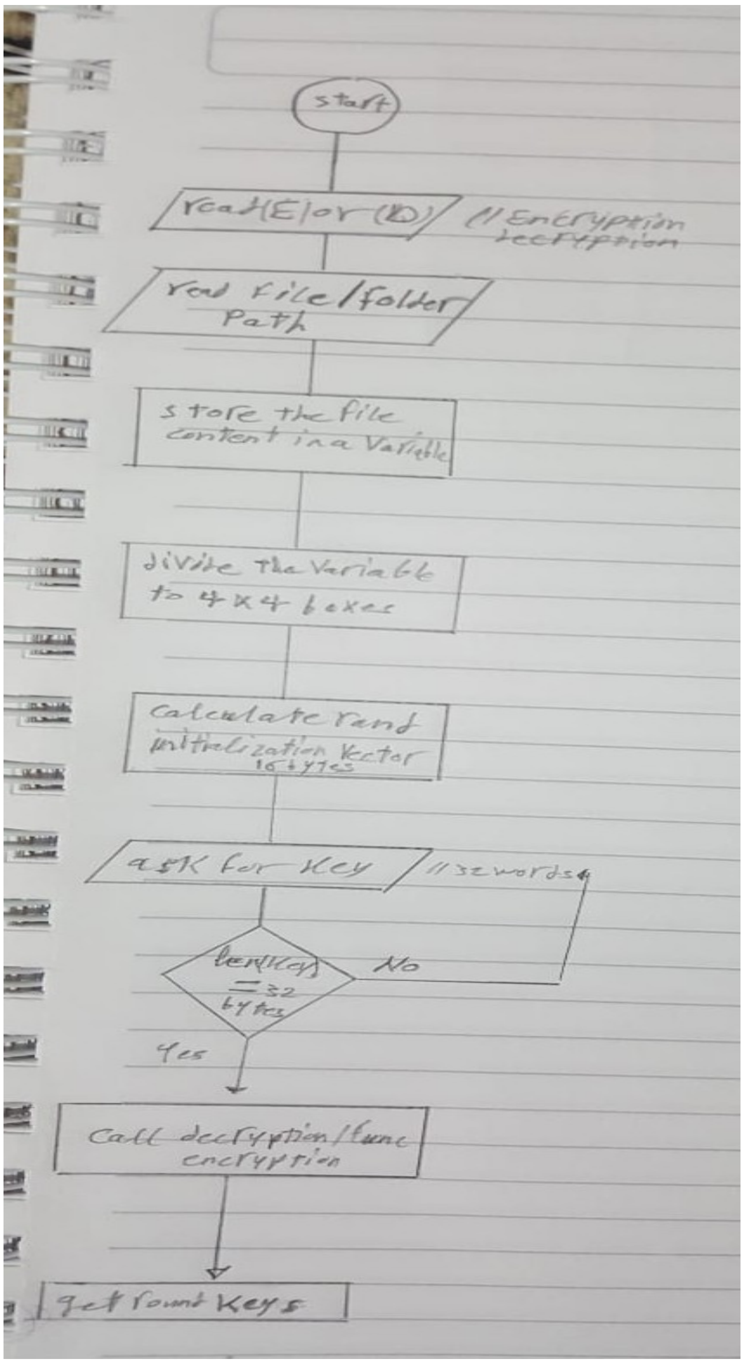
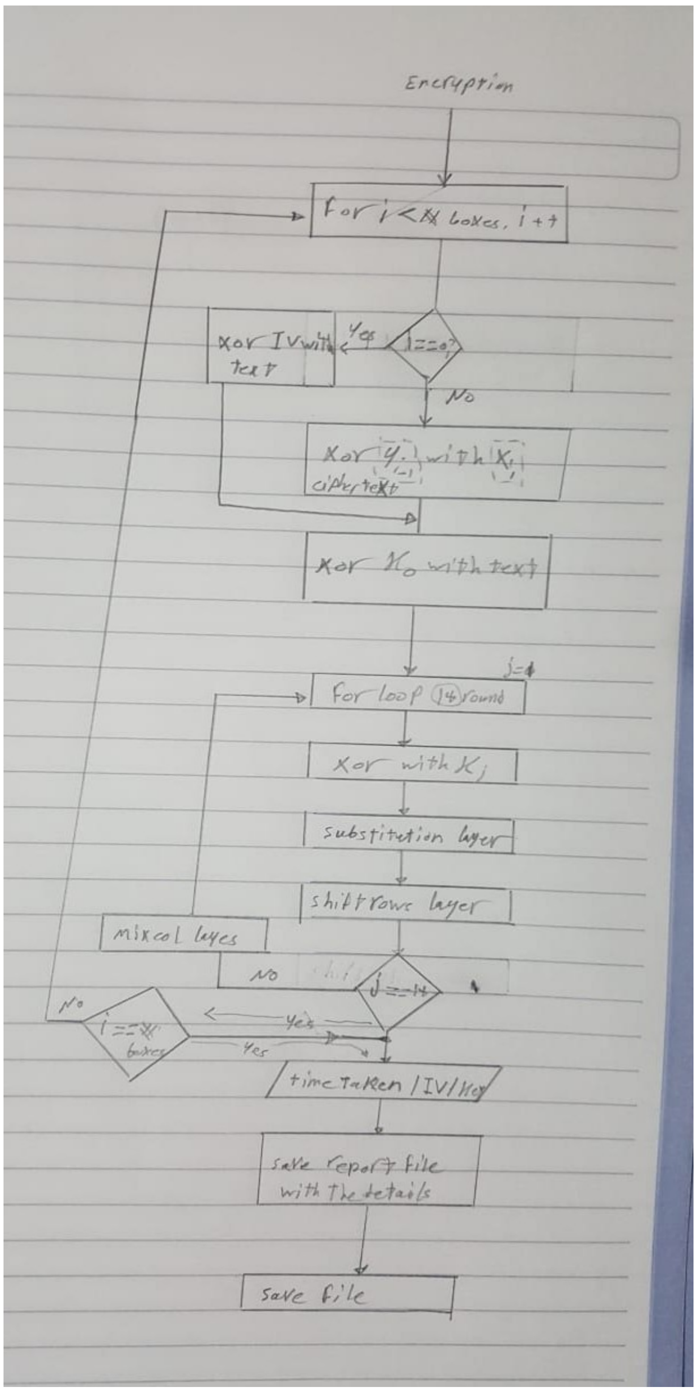
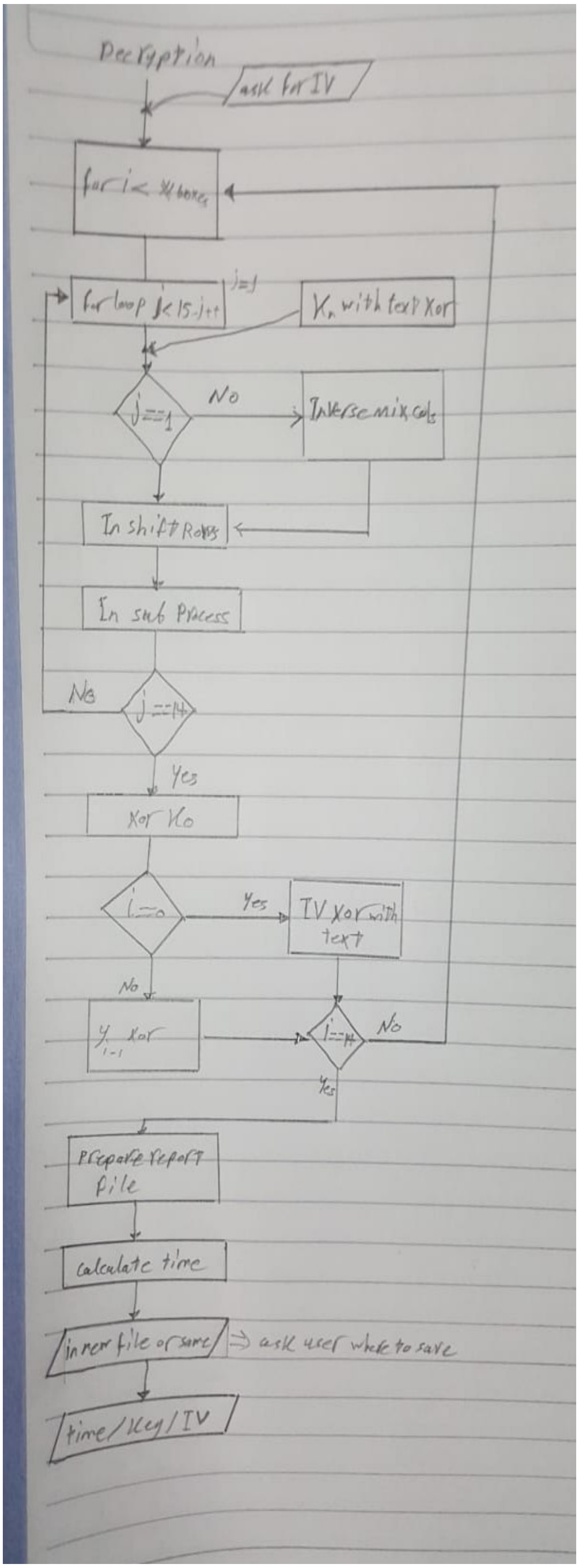


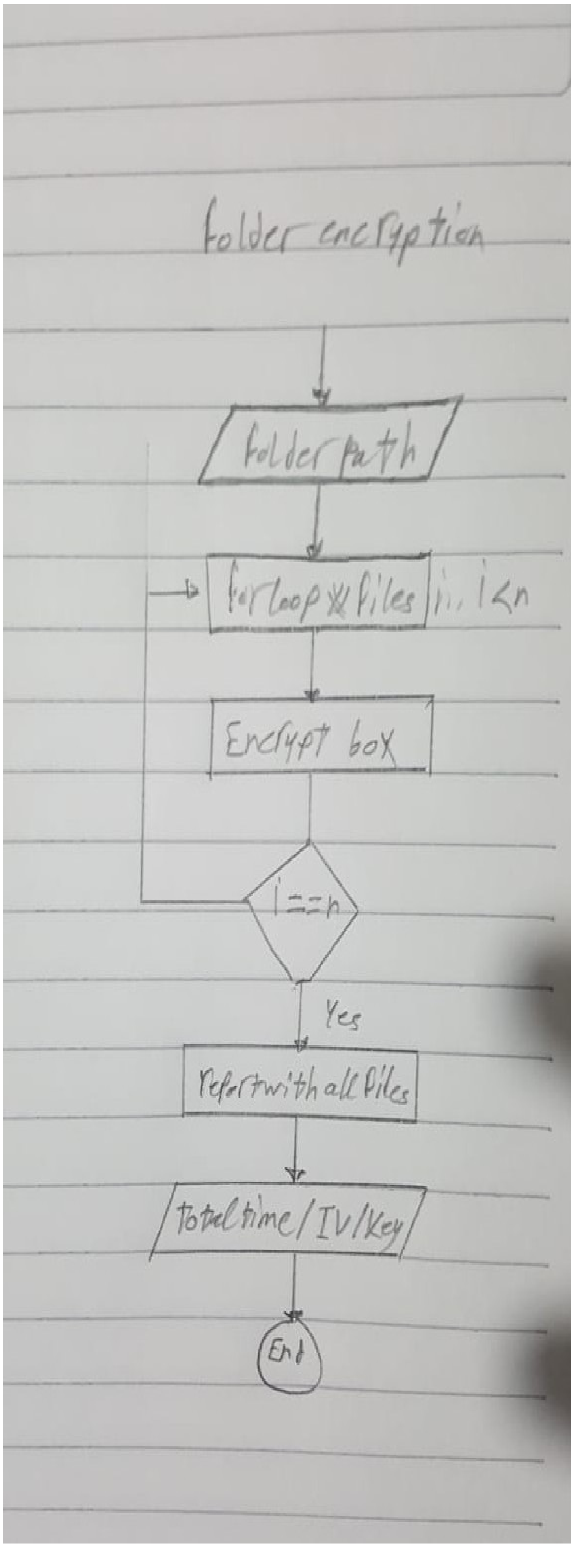
Fig. 4.7 AES key schedule for 256-bit key size



start
read(E)or(D) // Encryption Decryption
read file/folder Path
store the file content in a Variable
divide the Variable to 4x4 boxes
Calculate rand initialization Vector 16 bytes
ask for key // 32 words
len(key) == 32 bytes
No
Yes
Call decryption/func encryption
get round keys




Decryption
ask for IV
for i < # boxes
for loop j < 15 j++
j=1
Kn with text Xor
j==1
No
Inverse mix cols
In shift Rows
In sub Process
No
j==14
Yes
Xor K0
i==0
Yes
IV Xor with text
No
Y i-1 Xor
i==14
No
Yes
Prepare report file
calculate time
in new file or same
⇒ ask user where to save
time/Key/IV

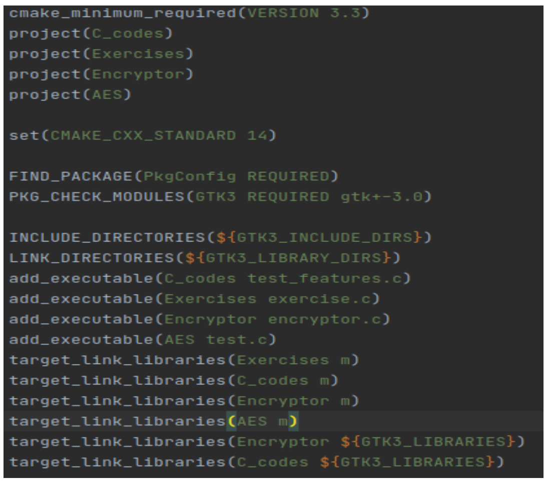

folder encryption
folder path
for loop # files
i, i<n
Encrypt box
i==n
Yes
Report with all files
Total time/IV/Key
End

Multiplications and additions are done in Galois Field GF(2^8) mod 2 in addition, and

mod P(X) in multiplication P(X) = X^8 + X^ 4 + X^3 + X +1 == 0b 100011011

the substitution process is inverse in GF(2^8)

OS: linux (Ubuntu)
IDE : Clion (jet brains) setting Cmakelists to read cmath and other libraries
the output files are texts



```
cmake_minimum_required(VERSION 3.3)
project(C_codes)
project(Exercises)
project(Encryptor)
project(AES)

set(CMAKE_CXX_STANDARD 14)

FIND_PACKAGE(PkgConfig REQUIRED)
PKG_CHECK_MODULES(GTK3 REQUIRED gtk+-3.0)

INCLUDE_DIRECTORIES(${GTK3_INCLUDE_DIRS})
LINK_DIRECTORIES(${GTK3_LIBRARY_DIRS})
add_executable(C_codes test_features.c)
add_executable(Exercises exercise.c)
add_executable(Encryptor encryptor.c)
add_executable(AES test.c)
target_link_libraries(Exercises m)
target_link_libraries(C_codes m)
target_link_libraries(Encryptor m)
target_link_libraries(AES m)
target_link_libraries(Encryptor ${GTK3_LIBRARIES})
target_link_libraries(C_codes ${GTK3_LIBRARIES})
```

Since the AES algorithm are complex especially in CBC mode, each 4 by 4 box is affected by the previous boxes and the IV, so any small change leads to completely different output. Besides, my IDE some times take the input as two input , text and enter key, so I had to add some getchar() to avoid such a problem, I was planing to add some graphics using gtk but the time was limited I can add them in the presentation day if possible.