The University of Newcastle
School of Electrical Engineering and Computer Science

COMP3260 Data Security
Assignment 2

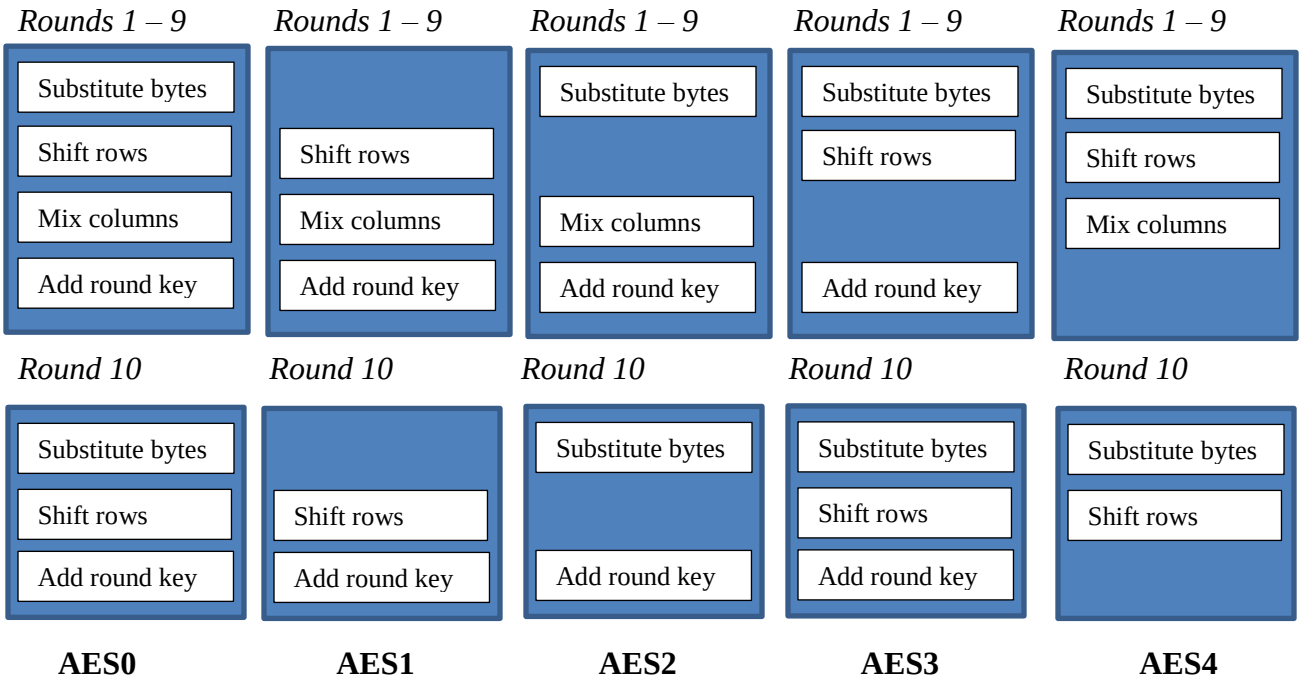
Due on Wednesday, 15th May 2019, end of day, in the Assignment2 in Blackboard.
Total mark: 100

Note:
Before you start working on the Assignment please read the information on academic integrity, which can be found at http://www.newcastle.edu.au/service/academic-integrity/. All available strategies will be used for detecting possible plagiarism and all suspicious cases will be referred to the SACO (Student Academic Conduct Officer).

In this assignment you will implement 10 round AES encryption and decryption of a single plaintext block. Your program will take as input a 128 bit plaintext block and a 128 bit key and produce as output a 128 bit ciphertext block, or a 128 bit ciphertext block and a 128 bit key and produce as output a 128 bit plaintext block. You will use your implementation to explore the Avalanche effect of the original Advanced Encryption Standard denoted as AES0, as well as AES1, AES2, AES3 and AES4, where in each version an operation is missing in each round as follows:

0. AES0 - the original version of AES
1. AES1 – SubstituteBytes is missing from all rounds
2. AES2 – ShiftRows is missing from all rounds
3. AES3 – MixColumns is missing from all rounds
4. AES4 - AddRoundKey is missing from all rounds

For additional clarity, the encryption algorithm for the five versions of AES is given in the picture bellow.



In addition to the original plaintext block P and the key K, your program should use another plaintext block P_i and key block K_i that differ only in bit i from P and K respectively, and use them to explore the Avalanche effect in AES as follows.

The program will encrypt plaintext P under key K. Then it will encrypt plaintext P_1 under key K and plaintext P under key K_1 and it will find the number of different bits after each of the 10 rounds between
a) P under K, and P_1 under K;

b) P under K, and P under K_1.
Then the above will be repeated for each of the remaining 127 plaintexts P_i that differ from P in a single bit, and 127 key blocks K_i that differ from K in a single bit, and the average results for all 128 data/key blocks will be presented in the output.

Your program MUST be well commented, include a header stating the authors and purpose of the program, and be easy to understand. You MUST NOT use any available AES code or a portion of it (including AES libraries).

ENCRYPTION: INPUT FILE

The following is an example of an input file, where the first row is the plaintext P and the second row is key K.

```
000…0
111…0
```

OUTPUT FILE

The following is a format of an output file (note that the numbers provided are sample values and not necessarily what you will obtain for different inputs):

```
ENCRYPTION
Plaintext P: 000…0
Key K: 111…0
Ciphertext C: 010…0
Running time: XXX
Avalanche:
P and Pi under K
Round        AES0    AES1    AES2    AES3    AES4
    0          1       1       1       1       1
    1         20     etc
    2         58
    3         59
    4         61
    5         68
    6         64
    7         67
    8         65
    9         61
   10         58

P under K and Ki
Round        AES0    AES1    AES2    AES3    AES4
    0          0     etc
```

1	22
2	58
3	67
4	63
5	81
6	70
7	74
8	67
9	59
10	53

In the above, 'Round 0' refers to the plain text before the beginning of the encryption. The column AESi contains the number of bits that differ between the original plaintext P, and the intermediate result in each round of the encryption performed by AESi defined above.

DECRYPTION: For decryption, the INPUT FILE should contain the ciphertext and the key, and the OUTPUT FILE should contain the ciphertext, the key and the plaintext.

The following is an example of an input file:

```
0000…0
111…0
```

The following is a format of an output file (note that the numbers provided are sample values and not necessarily what you will obtain for different inputs):

```
DECRYPTION
Ciphertext C: 000…0
Key K: 111…0
Plaintext P: 010…0
```

PROGRAM REQUIREMENTS

Assignment must be completed in either Java or C++, unless you first obtain permission to use another language – please contact your marker Joel Wong to discuss other languages. If implementing in C++ the program must include a *make* file, and must compile and run on the University machines. If implemented in Java, it must also compile and run on the University machines. For all implementations please include a Readme.txt file outlining what each class handles. Please name the main runnable class Application. If possible, please also provide a Windows executable file.

Assessment criteria:

1	AES encryption and decryption – working and correct	55
2	Avalanche analysis, correct	35
3	Comments throughout the program	10
	TOTAL	100

If your AES encryption and decryption are not working correctly you can score at most 40 marks in total.