Problem 1

a)

Both MAC and Digital Signatures provide Integrity and Authentication. The difference however is that MAC is generated and verified using the same key (that should be secretly shared between the sender and the receiver). Digital signatures on the other hand are generated using a private key of a key pair (public and private). The sender uses the private key to encrypt but the message can only be decrypted with its public key pair, proving that the message was generated by the sender.

Based on this description I would choose option **number 4** as the best explanation.

Problem 2

For my Encripted File Store first of all I used only C code (not C++).

In general the methodology I used to store files are:

MAC|n. chars for file name (int) | file name (char) | n. chars for message size (int)| IV + message | … append any number of messages

Using this logic I could at any point I could pick the MAC and verify integrity. I could also traverse the file looking for the appropriate message by reading the appropriate integers and comparing the names. Also once the message was extracted I could locate the IV to decrypt the file. The encryption/decryption was done as required by the homework.

I did not have time to implement the getpass() method so I require that a password has to be passed with a -p flag.

Sources I looked for this homework.

I looked at stackoverflow for solving problems I was having with my code. For example this one:

https://stackoverflow.com/questions/2572366/how-to-use-dev-random-or-urandom-in-c

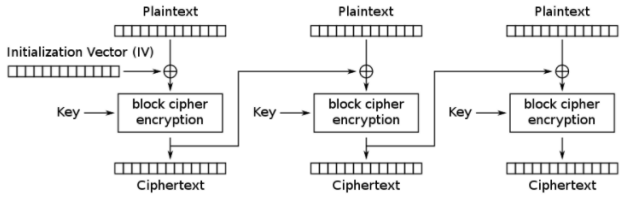
On how to generate random numbers in the required library.

I looked at:

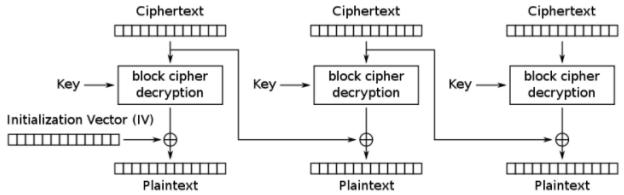
IBM File_encrypt Source Code (File_encrypt Source Code - IBM Documentation) Inspired me on how to read/write files and use a struct to carry the information. Also on how to work with files with different terminations (.pdf, .txt, etc)

I looked at the code in [B-Con/crypto-algorithms: Basic implementations of standard cryptography algorithms, like AES and SHA-1. (github.com)] for understanding how to handle encryption and hashing in general and how to adapt my code to do CBC decryption using aes_encrypt and aes_decrypt.

For cbc encrypt and decrypt I looked at the following plot:



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

And implemented it as follows.