To test that the key generated with our Diffie-Hellman Key Exchange implementation, we decided to utilize a CBC block encryption cipher. However, we did not wish to use an existing CBC implementation but work in out own. So, it was decided that in order to easily make use of the CBC implementation, it would be incorporated as an object.

The object itself does not have a construction or destructor and only has three public functions, encrypt, decrypt, and get\_IV. These functions take in two file names, the key and in the case of the decryption function, it also takes in the previously generated IV. Since the CBC uses a random IV for encrypting the first block and that IV is needed to decrypt that same block, the IV is randomly generated within the public encryption function and returned as an int. Without this the public decryption function would not be able to decrypt the first block. CBC uses 64-bit blocks, so to read blocks of this size the program uses fixed integer size blocks of uint64\_t and uint32\_t. With these blocks the date read in by the fread function is already formatted into the right 64-bit block size. The only problem with this is that it reads past the end of the file padding it with whatever is next in the memory. Because of this is a lot harder to unpad this junk data as it may NULLs or may be some random data. The public decryption and encryption functions each have their own helper functions Dec and Enc respectively. These functions act as the block encryption and decryption functions that the key is used to encrypt and decrypt each block of data. These are both implemented as Feistel structures that uses bitwise XOR operations to encrypt and decrypt the data. Since the size of the key needs to match the block size for CBC, the helper Feistel functions mod the inputted key to the correct size.

The user has access to the three public functions. They are encrypt, decrypt, and get\_IV. The user simply needs to call on these object functions with the file names of what they wish to encrypt, the output file as well as the key they wish to use. The functions open and read from the input file and either encrypt of decrypt to the output file named. If the user wishes to not save the random IV value to some variable, they can simply utilize the get\_IV function which will return the value that was generated. There are two functions that the user does not have access to, these are the Dec and Enc functions, which serve no purpose to the user and are only there to help with the encryption and decryption of the files.