This homework is due on 3/9/2016. This is an individual homework assignment to be done by each student individually. The submission of your homework is your acknowledgement of the honor code statement

I have not copied any solution from anyone and not provided any solution to anyone on this assignment. The solution has been entirely worked out by me and represents my individual effort.

Please

- Use a word processor (MS-Word, LaTex, Framemaker, …) for your solutions. No hand writing submission will be accepted.
- Include your name and G-number at the beginning of your homework submission.
- Pack all your files with zip or tar and gzip and name your packed file as CS468_HW2_<your last name>_<your G#>.zip (or tar.gz)
- Submit your packed solution to the blackboard

This homework requires accessing, using and programming at zeus.vse.gmu.edu. Please create a directory named "CS468_HW2_<your last name>_<your G#> and do everything of this HW there.

1. Use OpenSSL to encrypt/decrypt a given file (20 points)

   A. find appropriate openssl command to use DES ECB mode to encrypt file `CS468-HW2.txt` into file `CS468-HW2.des-ecb` with key `0f1571c947d9e859` and IV `0102030405060708`.

   find appropriate openssl command to use DES ECB mode to decrypt file `CS468-HW2.des-ecb` into file `CS468-HW2-des-ecb.txt` with key `0f1571c947d9e859` and IV `0102030405060708`.

   Take a screenshot of the execution of the two openssl commands, name the screenshot file "`CS468-HW2-1A.*`" (here * depends on the format of your screenshot, e.g., jpg).

   Submit file `CS468-HW2.des-ecb`, `CS468-HW2-des-ecb.txt` and `CS468-HW2-1A.*`.

   B. find appropriate openssl command to use AES-256 CBC mode to encrypt file `CS468-HW2.txt` into file `CS468-HW2.aes256-cbc` with key `000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f20` and IV `d9000a0800ac3b75111d393ad246ff95`.

   find appropriate openssl command to use AES-256 CBC mode to decrypt file `CS468-HW2.aes256-cbc` into file `CS468-HW2-aes256-cbc.txt` with key `000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f20` and IV `d9000a0800ac3b75111d393ad246ff95`.

   Take a screenshot of the execution of the two openssl commands, name the screenshot file "`CS468-HW2-1B.*`" (here * depends on the format of your screenshot, e.g., jpg).

   Submit file `CS468-HW2.aes256-cbc`, `CS468-HW2-aes256-cbc.txt` and `CS468-HW2-1B.*`.

2. Programming with ciphers of OpenSSL (80 points)

A. You need to develop 2 C programs "DES_ECB_Enc.c" and "DES_ECB_Dec.c" that use the ECB mode DES cipher in OpenSSL to encrypt or decrypt a given file (of arbitrary length) into another encrypted or the decrypted file with any given encryption key and IV. Specifically, the programs need to support the following commands line arguments and usage:

```
DES_ECB_Enc –k <key file> -v <IV file> -i <input file> -o <output file>

DES_ECB_Dec –k <key file> -v <IV file> -i <input file> -o <output file>
```

where <key file> contains the 32-byte encryption key in hex format; <IV file> contains the 8-byte IV in hex format; <input file> is the name of the input file (e.g., plaintext or cipertext); <output file> is the name of the output file (e.g., cipher text or plaintext). Note: <key file> and <IV file> are text files, <input file> could be either text or binary file; <output file> is a binary file.

The programs need to do basic sanity check on <key file>, <IV file>, <input file> and gracefully exit when any illegitimate file is found (e.g., the <key file> does not contain 8 bytes in hex).

Take a screen shot of the execution of the following commands:

```
DES_ECB_Enc –k deskey.txt -v desIV.txt -i CS468-HW2.txt -o CS468-HW2.mydesecb

DES_ECB_Dec –k deskey.txt -v desIV.txt -i CS468-HW2.mydesecb -o CS468-HW2-
mydesecb.txt
```

name the screenshot file "`CS468-HW2-3A.*`" (here * depends on the format of your screenshot, e.g., jpg; use provided `deskey.txt` and `desIV.txt` files).

Submit:

- DES_ECB_Enc.c, DES_ECB_Dec.c, corresponding makefiles and/or information about how to generate the executable from the source code. Note, you need to provide everything needed to generate the executable in zeus environment.
- `CS468-HW2-3A.*`, `CS468-HW2.mydesecb` and `CS468-HW2-mydesecb.txt`.

B. You need to develop 2 C programs "AES256_CBC_Enc.c" and "AES256_CBC_Dec.c" that use the CBC mode 256-bit AES cipher in OpenSSL to encrypt or decrypt a given file (of arbitrary length) into another encrypted or the decrypted file with any given encryption key and IV. Specifically, the programs need to support the following commands line arguments and usage:

```
AES256_CBC_Enc –k <key file> -v <IV file> -i <input file> -o <output file>

AES256_CBC_Dec –k <key file> -v <IV file> -i <input file> -o <output file>
```

where <key file> contains the 32-byte encryption key in hex format; <IV file> contains the 16-byte IV in hex format; <input file> is the name of the input file (e.g., plaintext); <output file> is the name of the output file (e.g., cipher text). Note: <key file> and <IV file> are text files, <input file> could be either text or binary file; <output file> is a binary file.

The programs need to do basic sanity check on <key file>, <IV file>, <input file> and gracefully exit when any illegitimate file is found (e.g., the <key file> does not contain 32 bytes in hex).

Take a screen shot of the execution of the following commands

```
AES256_CBC_Enc –k aes256key.txt -v aesIV.txt -i CS468-HW2.txt -o CS468-HW2.myaes256

AES256_CBC_Dec –k aes256key.txt -v aesIV.txt -i CS468-HW2.myaes256 -o CS468-HW2-myaes256.txt
```

name the screenshot file "`CS468-HW2-3B.*`" (here * depends on the format of your screenshot, e.g., jpg; use provided `aes256key.txt` and `aesIV.txt` files).

Submit:

- AES256_CBC_Enc.c, AES256_CBC_Dec.c, corresponding makefiles and/or information about how to generate the executable from the source code. Note, you need to provide everything needed to generate the executable in zeus environment.
- `CS468-HW2-3B.*`, `CS468-HW2.myaes256` and `CS468-HW2-myaes256.txt`.