BIRZEIT UNIVERSITY

INFORMATION SECURITY AND COMPUTER NETWORK LABORATORY – ENCS5121

Assignment 1: Programming using the Crypto Library

In this task, you are given a plaintext and a ciphertext, and your job is to find the key that is used for the encryption. You do know the following facts:

- The `aes-128-cbc` cipher is used for the encryption.
- The key used to encrypt this plaintext is an English word shorter than 16 characters; the word can be found from a typical English dictionary. Since the word has less than 16 characters (i.e. 128 bits), pound signs (`#`: hexadecimal value is `0x23`) are appended to the end of the word to form a key of 128 bits.

Your goal is to write a program to find out the encryption key. Attached you can find the English word list that contains the key (words.txt). You can find the ciphertext in the last page based on your university ID while the plaintext, and IV are the same for all students and listed in the following:

```
Plaintext (total 21 characters): This is a top secret.
IV (in hex format):              0102030405060708090a0b0c0d0e0f
```

After finding the key your program should output the key to "key.txt" in ascii, so the output key should be an English word and not a hexadecimal represntation of the key.

You need to pay attention to the following issues:

- If you choose to store the plaintext message in a file, and feed the file to your program, you need to check whether the file length is 21. If you type the message in a text editor, you need to be aware that some editors may add a special character to the end of the file. The easiest way to store the message in a file is to use the following command (the `-n` flag tells `echo` not to add a trailing newline):

```
$ echo -n "This is a top secret." > file
```

- In this task, you are supposed to write your own program to invoke the crypto library. No credit will be given if you simply use the `openssl` commands to do this task. Sample code can be found from the following URL:

```
https://www.openssl.org/docs/man1.1.1/man3/EVP_CipherInit.html
```

- When you compile your code using `gcc`, do not forget to include the `-lcrypto` flag, because your code needs the `crypto` library. See the following example:

```
$ gcc -o myenc myenc.c -lcrypto
```

- You should use C programming language. No credit will be given if you use any other language.
- Copying from the internet or other students will result in 0 grade.
- Run your code and test it on SEED-Ubuntu20.04 distribution.

- Deadline: Midnight 14-March-2024 (reply before 15-March-2024)
- You should submit the following:

  1- Your C code with the following filename “FIRSTNAME_LASTNAME_ID.c”.
  2- The Makefile used to compile and run your code.

  Note that your code should be well-commented.

```
1180887     2f8e3b6b168e5adda71c19dab24db59d0e34f731fbc9183cd7673adb0f57773f
1190659     7e89de5404a2d589ea7836606230722388822dba85dd21ae8814ed23deef0189
1190886     3549b72ff579a29c92bec11bc620a1144d3ab3cd2960c655cade16625a0fa586
1190994     5fd0b751363f331abcf4e084567d925016a12f0c5039fa3b0947c36f4036070e
1191024     3664cca27b03eda785d19b1b69c5dc55f8aaa1828ce2aa619e1d71043491987c
1191052     d967f96c08002622218dbac9f6b91cd6b366674698c88cb3878cb58be2526b0c
1191072     6f8f1e3936b1bbcff54446121342af538393a9fd895eba07f82d4289119a8625
1191102     9de38d6758c0061a47439b253310e50662ec62e73f4044ea5b50f42716c51958
1191167     12c402ac6a96428b087a568736428d291bba6f370b101a8f25477d02b39d25ed
1191334     ef0c1599f6d9a9c3e1983003b7939113b715de339f8bb463fc7cda5a6aefd07f
1191408     97fd3f8526495fcd1287ec37962f22b998d3ad42195a86f2fdbe5b885e188109
1191448     ecbcab2ee96e76579848013e928e72f866a1dd94eadd6f184b8d38093b822cab
1191522     8a539c6c4aca3cefc5caac11f9cd0b29bf71eb8dbfce195821e5521ecef25844
1191648     f08bf3e27b49b578d5be7435c4266fc7f13c2eb96fddd15f5087d17aa45ca7a7
1191708     5c89d1ae6760cc0d2eef573e3b68b619570b7296bb75531d8508b10a0e24d8a8
1191868     9a6e582c4a0a2859f91d84e9949da87c6051c1b5fccde584b7b9d36f1f633c66
1192364     6c1ee159c84e67ebc12a0aba28c866917984341bf26f0f5c23afd4d322c82670
1192454     d1db6e0c4cef61a10dd7c910f83326c67cd87317eb381dcadcc937523e179bbb
1200006     aefb7a8d8dc47c7110d0e69036f56763e1a9d0b1c6825c25f2b996c8c1e06472
1200284     6029c22a7b6c95852d2053104b65433fd6c39c1fda0d6564e59414fd82dc01a7
1200488     9ae7d9ba2efbb297683353154ead164fca6957f90cfec17f87501c3c40202bc9
1201225     08a4ad025c5b0264169e76ec68ec0bed73adbde92f2715b492990f636832da80
1201766     4fb4ecfc6298352cb7e071a68f5cfde6f754d56692f764ff870160571fd1b682
1190119     83aa36ed93d717e674b9f0cba68ce1dd463ff3c436bf1f28d16bea58803502a7
1190186     606ce8038916fad8459702dfbae5fcc6e53f1c8552f1c605de2f3715a570eb76
1190324     cca522f76b3994b3f1601227bf039b06c38e52b71d6ef68368f02a6ad46fb210
1190515     74410584df66996d6d8c07a7911493c82d7dc0d04d6aba21778598a6de853522
1190585     3537633c9fa0decf53468fc7b077f7bfdd09d8e75518a946c3f99d51e651f8e1
1190715     961f8a44cc6bdd37f9358267cff36e1b1f56d7ed3a0cf280b1098d29ac8448a6
1191514     0a6de6b3982a4b93570a50577e2c2d7e151700e3abac5bdbffa07ddef1809e0b
1191590     4528cf62d16b9780144144226be31a6b306bc60bf8baf57fe321cc38229c0248
1191740     0b0670be60042ce89e6ec6f1928a4d9bc23e5dacb49e5ab84b72b320ece54190
1191749     5549bc1ca64a3ebf0ecf77eb80050ed034a8e9428cd1b37d81b36458cb947ec2
1192054     469f52bd6a0dae032a2bbeef55a545b8611f92d2ce9a15592e3a5f4a1f216600
1192401     a75dc5638c27a658025968a062af61ab0315a76d8c036858db5a34b5d1b44734
1192439     1fffeee0df3953c177c9494738389b52f3e9b922351ccd769809bd64889a5415
1192567     d696c003b9ab79482d5d37379feec32967a34f32db8697436e543fd4f75852bb
1193121     3ebc53dd7be4aa17d269099043dc8b920e7f88e95e4f72a6bc87eaf7a4af8622
1200757     03c4aa11469713a4c2ca58e171f8bd73a418a6d6042456590bbe7bab2946f35d
1200905     cf5fb851481e9f59dc711b1e1641a5f7c8fec5500a4a481671c8f4ef3bdb2d6e
1201134     ee075fdad8069b11d3e2798c8aed19e4476f1b8bbc211d44a80de83d7968a5e6
1201959     f08e1b341e41f9dd9172e1b25d16c134d9da4e1c43e409968578d3bb413719e6
1202093     ed88eccbfae1455c8ea4d64424dc7dc604cdb83418a03f33a764872fefa3a1dd
1202384     77b6d07a228adca115bf6ecbc462c895d6f1a3853e9741750547ba20d8a75eee
```