Final Project: Deliverable 1, Instructions

Implement the block-based symmetric key cryptography system DES with both encryption and decryption components. The specification for DES is laid out in our textbook1, the official DES specification2, and this online webpage that is very helpful3. You may also have read that DES, by itself, has been replaced by AES, but TripleDES is considered as strong as AES and is just a simple extension of DES using 2-3 keys instead of just a single key. Note that in the python programming homework, HW3, you implemented a simple block cipher *(****w****,****P****,****k****)*. DES uses components similar to the ***w*** and ***P*** components of that cipher.

To encrypt or decrypt the plaintext, you should encrypt or decrypt each 64-bit block independently; note that this is known as ECB (Electronic Codebook) mode. Sample secret keys are available in the files keySecret1.txt and keySecret2.txt.

**OneAndZeroes padding**

Note that the message in the input data files may not be a multiple of 64 bits, so we will need to pad these messages. There are many padding schemes out there, but we will use the OneAndZeroes method as described below:

For "OneAndZeroes" padding, given a block size of B bytes (Note B=8 for DES), add a byte of

value 0x80 followed by as many zero bytes as is necessary to fill the input to the next exact

multiple of B. This method always adds padding of length between one and B bytes to the input

before encryption. It is easily removed in an unambiguous manner after decryption.

The "OneAndZeroes" term comes from the fact that this method appends a 'one' bit to the input

followed by as many 'zero' bits as is necessary. The byte 0x80 is 10000000 in binary form.

Examples of OneAndZeroes padding for block length B = 8:

3 bytes: FDFDFD --> FDFDFD8000000000

7 bytes: FDFDFDFDFDFDFD --> FDFDFDFDFDFDFD80

8 bytes: FDFDFDFDFDFDFDFD --> FDFDFDFDFDFDFDFD8000000000000000

We will need to pad the message before encryption, and remove the padding after decryption.

**Note that the DES.py file implements this padding, a large number of DES functions, and the test functions to test your code. You will need to complete the stubbed-out code in the DES.py file. There are also helper functions you can use in the StringConversion.py file.**

Assume the inputs to your code are a plaintext file name (with the plaintext given as binary digits in that file) and a 64-bit key file name (with the key given as 64 binary digits in that file). You must provide the following:

1. **The DES encryption functionality:** The method DES\_encrypt(plaintextFileName, keyFileName, ciphertextFileName) that encrypts the text in the file named by the first parameter with the 64 bit key in the file named in the second parameter and writes the encrypted text to the file named by the third parameter has been implemented for you, along with other methods it calls. To complete this functionality, you will have to complete the following stubbed out methods:
   * + round\_i(array, key\_i)
     + encryption\_round\_key(key, intermediate\_key, i)
     + encrypt(plaintext,key)
2. **The DES decryption functionality:** The method DES\_decrypt(ciphertextFileName, keyFileName, plaintextFileName) that decrypts the text in the file named by the first parameter with the 64 bit key in the file named in the second parameter and writes decrypted text to the file named by the third parameter has been implemented for you, along with other methods it calls. To complete this functionality, you will have to complete the following stubbed out methods:
   * + decryption\_round\_key(key, intermediate\_key, i)
     + decrypt(ciphertext,key)
3. Provide two test cases for your system. One test case has been coded for you in the comments on the testEncryptionAndDecryption() method.
4. Finally, review the attached rubric for this deliverable, and for each performance criterion, please self-evaluate your work by describing whether you completely or partially met expectations, or did not meet expectations, on this criterion.

Format

Please submit your answers:

* + To items 1, 2 and 3 as a zip file containing the DES.py file and the associated test cases, and a readme file with any instructions on how to run your code, and its test cases.
  + To item 4 as a readme file providing the required self-assessment.

Please keep in mind:

* + Your code should run correctly and be implemented efficiently.
  + Proper indentation and variable naming should be followed. Use meaningful variable names.
  + Use helper methods as appropriate.

1. *Understanding Cryptography, Christof Paar and Jan Pelzl, Chapter 3*
2. [*https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-*](https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf)

[*25/documents/fips46-3.pdf*](https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf)

1. [*http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm*](http://page.math.tu-berlin.de/%7Ekant/teaching/hess/krypto-ws2006/des.htm)