CCA secure encryption and decryption using CBC MAC and OFB encryption

Sarthak Mahajan, 2018111005

March 7, 2022

1 Code structure:

- This folder contains cca.py and run.py code files
- cca.py contains the code to CCA encrypt and decrypt messages.
- run.py is used to take inputs and return outputs of the CCA encryptor/decryptor.

2 Instructions to run the code:

- Run the following command in the folder where this document is contained: python3 run.py
- There will be prompts asking you to input the relevant information serially, and will return the output.

3 Explanation of Functions in run.py:

3.1 run():

- It asks for inputs of the choice of encryption/decryption, key, key length in binary, message or ciphertext(depending on encryption/decryption choice)(all in decimal format)
- Calculates p: the order of the group for DLP and g: the generator of this group.
- prints the encrypted message or decrypted cipher depending on the choice.

4 Explanation of Functions in cca.py:

4.1 CCA_enc(m, k, ,p,g):

- p,g are in decimal format, they are needed to be passed to the dlp function. p is the largest prime possible in n bits, where n= length of the key, g is a random number in the range(1,p-1); note that this is fixed for the program in terms of p(so that same seed gives same value for different iterations of the program), but it could be any random number in the given range.
- All inputs are in decimal format, m is the message to be encrypted, k is the key, key_len is the length of the key in binary format.
- It returns the CCA encrypted message in binary format
- The final output is a concatenation of the OFB-CPA encrypted message and the CBC-MAC tag of the OFB-CPA encrypted message.
- Uses OFB_CPA_enc function for CPA encryption, CBC_MAC function for creating tag.

4.2 CCA_dec(c, k, key_len,p,g):

- p,g are in decimal format, they are needed to be passed to the dlp function. p is the largest prime possible in n bits, where n= length of the key, g is a random number in the range(1,p-1); note that this is fixed for the program in terms of p(so that same seed gives same value for different iterations of the program), but it could be any random number in the given range.
- c is the CCA secure ciphertext to be decrypted in binary format, k is the key in decimal format, key_len represents the length of the key in binary format, the number itself is in decimal format.
- It returns the decryption of the encrypted ciphertext in binary format if the tag is verified, otherwise it returns: ”invalid”
- It uses the OFB_CPA_dec function to decrypt message if tag is verified, CBC_MAC function is used to verify the tag and authenticate the cipher.