**Lab 6: Programming Symmetric & Asymmetric Crypto (Report)**

Submitted by:

**Md. Mehadi Hasan(2014331049)**

**Md. Yusuf Ali(2014331046)**

Submitted to:

**Dr. Md Sadek Ferdous,**

Assistant Professor, CSE dept., SUST

We have used five integer values as **input command** for different type of tasks in a loop to process like:

‘***0’*** for “Exit”,

‘***1’*** for “AES encryption and decryption”,

***‘2’*** for “RSA encryption and decryption”,

***‘3’*** for “RSA Signature”,

‘***4’*** for “SHA-256 hashing”

And for the others invalid command we have used try-except block to handle that.

If the input command value is **‘1’** then there are two options to use ‘ECB’ and ‘CBC’ mode and then options for using key length 128 and 256 and an user input option for the previous each option.

After that we will get the cipher text and the decrypted text with respect to the input values. And we have used the two *fixed key* for 128 and 256 bits which are “'abcdefghijklmnop” and “abcdefghijklmnop1234569874123658” respectively. Here we have used the tutorial that is “<https://techtutorialsx.com/2018/04/09/python-pycrypto-using-aes-128-in-ecb-mode/>”.

If the input command value is **‘2’** then there is an user option of giving the message(plain text). After this we have stored the private and public key to respectively the ‘private\_key.txt‘ and ‘public\_key.txt’ files. And we will get the RSA encrypted and decrypted message.

Here we have used 1024 bit encryption keys. And we followed the stackoverflow link that is “<https://stackoverflow.com/questions/30056762/rsa-encryption-and-decryption-in-python>”.

If the input command value is **‘3’** then there is an user option of giving the message(plain text). After that we have used the previous 2 tasks and then read the the message and encrypted message files then we have decrypt the encrypted message then we saw that whether the hash value of this decrypted message is equal or not to the message. Here, we have followed the tutorial that is “<https://www.laurentluce.com/posts/python-and-cryptography-with-pycrypto/>”.

If the input command value is **‘4’** then there is an user option of giving the message(plain text). After that we will get the hashed value of the message using SHA-256 machanishm. Here, we have followed the tutorial that is “<https://www.laurentluce.com/posts/python-and-cryptography-with-pycrypto/>”.

We have kept the functionality to show the **elapsed time** of each performed operation.