# CS695 Enterprise Cybersecurity

## Lab 3 Encryption with GPG

# Introduction

### Brief description (purpose and overview)

The purpose of this lab is to get students familiar with basic concepts of cryptography and how to use them to protect data using the gpg tool.

### Learning Objectives

After finishing this lab, students shall be able to:

1. Understand the difference between symmetric and asymmetric encryption algorithms
2. Encrypt, sign and decrypt data using gpg tool suite.

### Prerequisite knowledge

- Have very basic knowledge of Linux.
- Have a basic understanding of command line interface (CLI) and be comfortable using it.

### Lab Setup Requirements

- Kali VM

# Part 1: Symmetric Encryption

(Some of the lab instructions are adapted from seed lab:
https://seedsecuritylabs.org/Labs_16.04/PDF/Crypto_Encryption.pdf )

## Detailed instructions

Perform the following actions and take screenshots of your result. Then answer the questions in the Question section.

1. Create a folder named “lab3”, and change your current work directory into lab3.

```
kali@kali:~$ mkdir lab3
kali@kali:~$ cd lab3
```

2. Check the gpg version using the “--version” option. Find all supported algorithms, answer Question 1 in the Question section.

```
kali@kali:~/lab3$ gpg --version
```

3. Find where the gpg is installed using the “whereis” command:

```
kali@kali:~/lab3$ whereis gpg
```

4. Check the manual of the gpg tool using either the “man” command or “--help”option.

```
kali@kali:~/lab3$ man gpg
kali@kali:~/lab3$ gpg --help
```

   a. What is the option used to encrypt using symmetric algorithm?
   b. What is the option used to specify different encryption algorithms?

5. Open Firefox web browser (https://www.fileformat.info/format/bmp/sample/index.htm), download a bmp file to your computer and encrypt the file using both AES256 and TWOFISH. Note you do not want to choose a too small file (You can try it by your own to see what will happen if you use a too small file). The file with the name of `MARBLES.BMP’ is a safe option to choose. The file should be first downloaded to the ~/Download folder. Assume the name of the file downloaded is pic_original.bmp. Change it to the name of the file you downloaded in the following command and move the file to the lab3 folder.

```
kali@kali:~/lab3$ mv ~/Downloads/pic_original.bmp ./
```

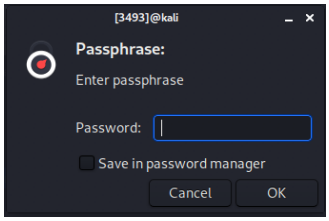
Then encrypt it using AES256 and output an encrypted file named pic_aes_enc

```
kali@kali:~/lab3$ gpg -o pic_aes_enc --symmetric --cipher-algo AES256 pic_original.bmp
```

Also encrypt the original file using TWOFISH and output an encrypted file named pic_twofish_enc.

```
kali@kali:~/lab3$ gpg -o pic_twofish_enc --symmetric --cipher-algo TWOFISH pic_original.bmp
```

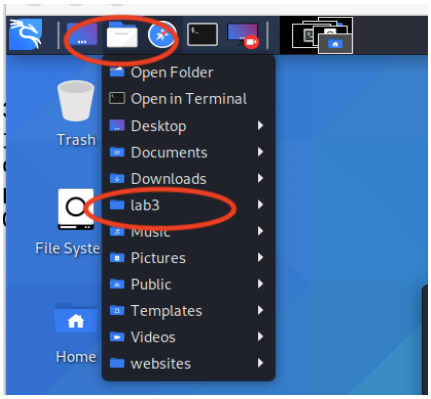
You will be prompted to enter a password phrase twice. Choose the password of your own choice and remember it. Check the generated file size.



6. Since the file header is also encrypted, we will not be able to open the encrypted file as a bmp file. We can replace the header of the encrypted picture with that of the original picture so that we can treat it as a legitimate .bmp file. For the .bmp file, the first 54 bytes contain the header information about the bmp file. We can get this header from the original file and then the rest of the data from the encrypted picture to reconstruct a bmp file. The following screenshot shows how to get the header of the original file using the "head" command and the body of the encrypted file using the "tail" command, then concatenate these two to reconstruct a new encrypted bmp file. You will do the same thing with the twofish-encrypted image.

```
kali@kali:~/lab3$ head -c 54 pic_original.bmp > header
kali@kali:~/lab3$ tail -c +55 pic_aes_enc > encpic
kali@kali:~/lab3$ cat header encpic > pic_ase_enc_withheader.bmp
```

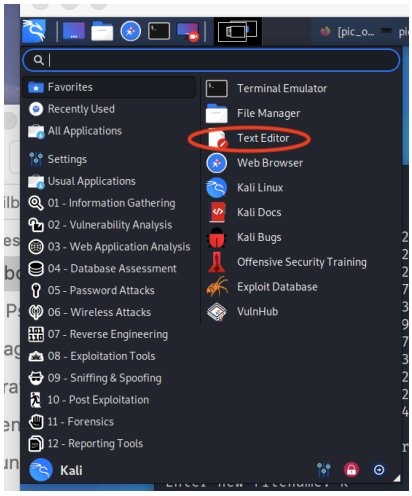
7. Open the lab3 folder using the File Manager tool, and view files as icons. Also try to view the encrypted picture in an image viewer. What are your findings? Answer Question 2 in the Question section.




8. Decrypt your encrypted files (not reconstructed bmp file) and see if it is the same as the original file.

```
kali@kali:~/lab3$ gpg -o decbmp -d pic_aes_enc
```

9. You can use the "cat" command or a text editor to create a short text file with less than 20 bytes and a large text file with more than 1000 bytes. You can fill in any data you want to the files. Save them in the folder of lab3. Encrypt both files using both AES256 and TWOFISH. What are your findings? Answer Question 3 in the Question section.



10. Decrypt your encrypted files and see if it is the same as the original file.
11. Compare AES256 and TWOFISH and answer Question 4 in the Question section.

### Questions

1. Please describe the differences between symmetric encryption, asymmetric encryption and cryptographic hash algorithms? List at least three of each supported by the gpg tool in Kali Linux.
2. When you encrypt the bmp file, what is the size of the encrypted files using AES256 and using TWOFISH? Are they the same? Are they larger or smaller than the original file? Why?
3. When you encrypt the text file, what is the size of the encrypted file using AES256 and using TWOFISH respectively? Are they the same? Are they larger or smaller than the original file?
4. Briefly describe the differences between AES256 and TWOFISH. Which one do you think is more secure based on your experiment results.
5. You will need to use a passphrase to generate a symmetric key for the encryption and decryption. Compare different passphrases (e.g. shorter and weaker vs longer and complexer) and state your findings.

## Part 2: Asymmetric Encryption

### Detailed instructions

Perform the following actions and take screenshots of your result. Then answer the questions in the Question section.

1. To use asymmetric encryption, we need to first create a key pair. You will be prompted to enter your real name and an email address which identifies you. Then you need to enter a passphrase to generate the public and private key pair for you. It will also sign the keys. The public key information will be displayed as the output. You can use "-- gen-key" option which uses RSA and sets the key length as 3072 bits. You can also use " --full-generate-key" option to customize the algorithm and the key length if you want.

```
chival@kali:~/Desktop/lab3$ gpg --gen-key
```

```
kali@kali:~/lab3$ gpg --full-generate-key
```

2. After you generate the key pair, you can also use "-- list-keys" to display the public keys you have.

```
kali@kali:~/lab3$ gpg --list-key
```

3. You can export your public key to a file named xxx_public.key, and provide it to others. (Substitute the "xxx@bu.edu" with your own email address. That is uid of your key. Substitute "xxx" in the "xxx_public.key" with your own name.) Send your public key file to your instructor as early as possible and submit your public key file on blackboard.

```
kali@kali:~/lab3$ gpg --export -a "xxx@bu.edu" > xxx_public.key
```

4. Import the instructor (or the facilitator's) public key (the other file of Assignment 3) into your key store. Then use "-- list-key" option to check if the imported key is there.

```
kali@kali:~/lab3$ gpg --import instructor_public.key
```

5. You can verify if the key is authentic by comparing the fingerprint generated with the provided fingerprint. (substitute "instructoremail@bu.edu with real email of your instructor or facilitator)

```
kali@kali:~/lab3$ gpg --fingerprint instructoremail@bu.edu
```

6. Decrypt the encrypted message that the instructor (or the facilitator) sent to you. In the example below, message.txt.asc is the encrypted message you got from the instructor.

```
kali@kali:~/lab3$ gpg --output message_decrypted.txt --decrypt message.txt.asc
```

7. Compose a new message to reply to the received message using any text editor and save it into a file, for example messagetoinstructor.txt, and then encrypt it and sign it. It will prompt you to input your passphrase that you use to generate your keys. Submit the encrypted message to the blackboard.

```
kali@kali:~/lab3$ gpg --encrypt --sign --armor  -r "instructor@bu.edu" messagetoinstructor.txt
```

8. Use the "--encrypt" option to encrypt the bmp file in the first part and specify the receiver as yourself, and then decrypt it to see if you can get the same bmp file.

```
kali@kali:~/lab3$ gpg --encrypt --sign -a -r "yourusername@bu.edu" pic_original.bmp
```

9. Now use "--compress-algo=none" to disable compression in the encryption.

```
kali@kali:~/lab3$ gpg --output pic_enc_nocompress.asc --encrypt --compress-algo=none --sign --armor
-r "yourusername@bu.edu" pic_original.bmp
```

If you did something wrong in the middle and want to restart, you need to delete all the public keys and private keys you have created.
(1) First, delete your private keys.
Run below commands to list all the public keys.

```
chival@kali:~/lab3$ gpg --list-secret-keys
```

For each private key you see from the output, run below command by replacing 'email' with the email address associated with each private key.

```
chival@kali:~/lab3$ gpg --list-secret-key email
```

(2) Then delete all public keys.
Run below commands to list all the public keys.

```
chival@kali:~/lab3$ gpg --list-keys
```

For each public key you see from the output, run below command by replacing 'email' with the email address associated with each public key.

```
chival@kali:~/lab3$ gpg --delete-key email
```

## Questions:

1. Check the manual of the gpg command using either the man command or "--help" option and explain the following options:
    a. --encrypt
    b. --decrypt
    c. --sign

    d. --armor
    e. --output
    f. -r
2. Gpg uses a hybrid approach to encrypt the message, which uses the public key to encrypt a session key and use the session key to encrypt the message. Explain this in more detail about how the message you want to send to the instructor is encrypted and signed as well as the decryption process. You can read the opengpg standard https://tools.ietf.org/html/rfc4880, particularly Section 2.
3. Compare the size of the original text message and encrypted & signed message, and state your findings and reasoning.
4. Compare the file size of the original bmp file and encrypted bmp file with and without compression respectively, state your findings and reasoning.

# Deliverables

Please submit the lab report in a single document named *CS695_yourusername_Lab3*. Please submit a DOC document and/or a PDF file in case the format is not compatible across the platform. The lab report should include:

1. Title, author(s)
2. Table of Content
3. The detailed steps and results using text description and/or screenshots that answer the above questions and demonstrate your lab progress.
4. A summary of your own reflection of the lab exercise, such as:
    a. What is the purpose of the lab in your own words?
    b. What do you learn? Do you achieve the objectives?
    c. Is this lab hard or easy? Are the lab instructions clear?
    d. What do you think about the tools used? What worked? What didn't? Are there other better alternatives?
    e. Any other feedback?