Configuration Manual

MSc Research Project
MSc in Cybersecurity

Lee Kearns
Student ID: x15728099

School of Computing
National College of Ireland

Supervisor: Mr. Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Lee Kearns		
Student ID:	x15728099		
Programme:	Master of Science in Cybersecurity	**Year:**	1
Module:	MSc Research Project / Internship		
Supervisor:	Mr. Vikas Sahni		
Submission Due Date:	6th September 2021		
Project Title:	Enhancing Data Security through Comparative Analysis & Implementation of a Hybrid Cryptosystem with Emphasis on Levels of Entropy.		
Word Count: 1028	**Page Count:** 3		

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Lee Kearns

Date: 01/09/21

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	

Penalty Applied (if applicable):	

Configuration Manual

Lee Kearns
Student ID: x15728099

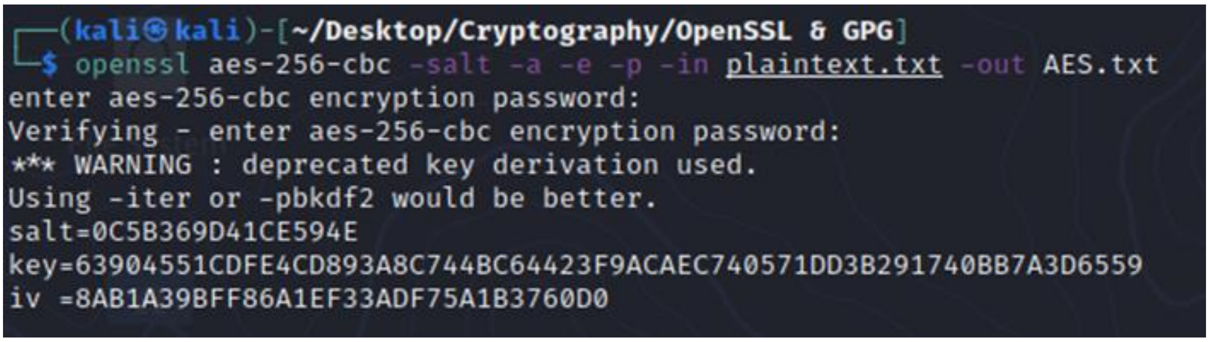
1 Introduction

The purpose of this configuration manual is to briefly outline how a user implements the encryption processes portrayed in the research paper. It provides the OpenSSL and GPG encryption scripts used and describes what each part means. In addition, the configuration manual also portrays the Shannon Index Formula used for calculating the entropy levels and describes what each symbol of the formula means.

2 Project Implementation

2.1 AES Encryption using OpenSSL Script

OpenSSL software [1] was chosen for symmetric encryption process for efficiency. AES-256 in CBC mode is the algorithm selected for symmetric encryption. The user will choose the file they wish to encrypt and will add the name of this file inside the symmetric encryption script. They will also provide the name of the outputted encrypted file. When the user hits enter, they will be prompted to insert a password. The password corresponds to the encryption key used and will need to be entered in the future if they wish to decrypt the data. Figure 1 displays the implementation of the encryption script.



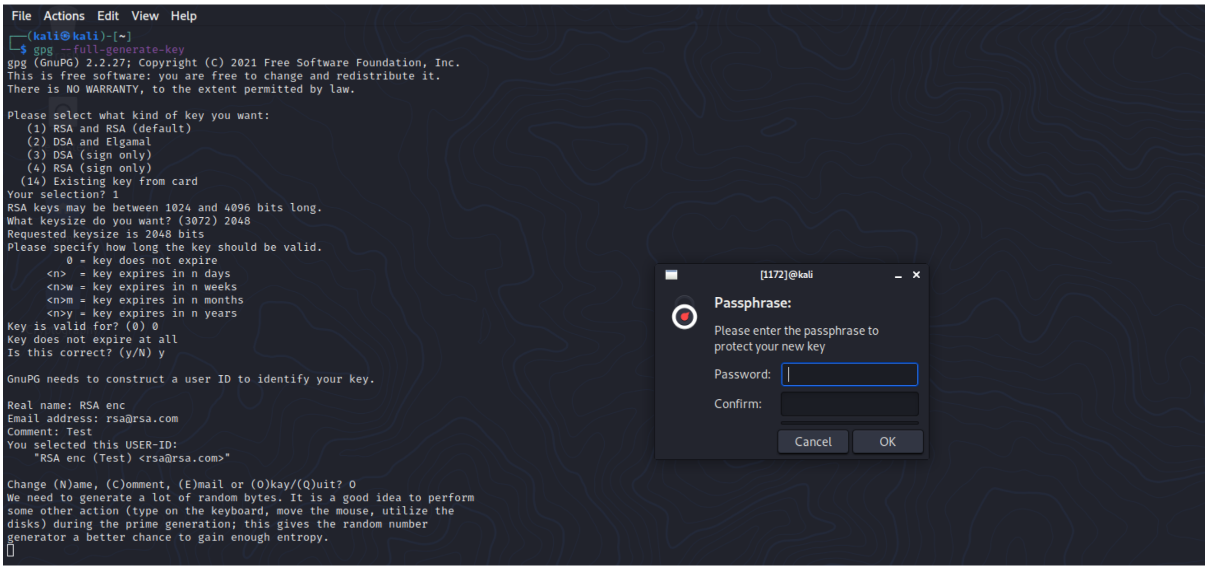
```
┌──(kali㉿kali)-[~/Desktop/Cryptography/OpenSSL & GPG]
└─$ openssl aes-256-cbc -salt -a -e -p -in plaintext.txt -out AES.txt
enter aes-256-cbc encryption password:
Verifying - enter aes-256-cbc encryption password:
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
salt=0C5B369D41CE594E
key=63904551CDFE4CD893A8C744BC64423F9ACAEC740571DD3B291740BB7A3D6559
iv =8AB1A39BFF86A1EF33ADF75A1B3760D0
```

Figure 1: AES OpenSSL Encryption Script

Each part of the encryption script is described below:
-aes-256-cbc: This is the symmetric cipher chosen for encryption (**aes**) using 256 bits (-**256**) in Cipher Block Chaining mode (-**cbc**)
-salt: This is telling the system to add a random salt to the encryption.
-a: This is the command the user enters that generates the key used for encryption.
-e: This is the command the user enters that assigns the IV to the encryption.
-p: This command prints out the salt, key, and IV used in the encryption process.
-in: This is the command the user enters to inform the system of the file they wish to encrypt.
-out: This is the command the user enters to assign a name to the encrypted file.

2.2 Generating Asymmetric Keys

GPG software [2] was chosen for asymmetric process for efficiency. RSA-2048 is the algorithm selected for asymmetric encryption. Before the asymmetric encryption process begins, the user must first generate the public and private keys. To do so, they must create an account with the GPG software, entering a username and password, choosing the algorithm they want to use for encryption (RSA) and choosing the size (2048 bits). This will then generate two mathematically connected keys; one is public, and one is private. Figure 2 highlights the key generation process.



```
┌──(kali㉿kali)-[~]
└─$ gpg --full-generate-key
gpg (GnuPG) 2.2.27; Copyright (C) 2021 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
  (14) Existing key from card
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (3072) 2048
Requested keysize is 2048 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct? (y/N) y

GnuPG needs to construct a user ID to identify your key.

Real name: RSA enc
Email address: rsa@rsa.com
Comment: Test
You selected this USER-ID:
    "RSA enc (Test) <rsa@rsa.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
```

[1172]@kali

Passphrase:

Please enter the passphrase to protect your new key

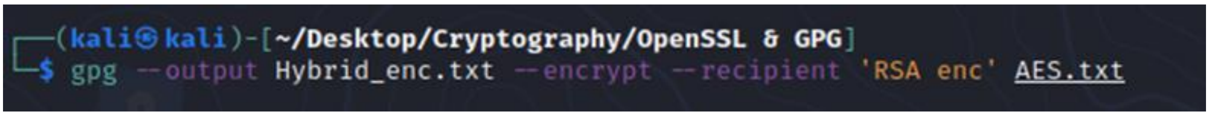
Password:

Confirm:

Figure 2: Asymmetric Key Generation

2.3 RSA Encryption using GPG Script

GPG software [2] was chosen for the asymmetric encryption process for efficiency. RSA-2048 is the algorithm selected for asymmetric encryption. Now that the keys have been fully generated in the previous step, the asymmetric encryption process can begin. The user will choose the file that has already been encrypted with AES. They will insert the name of this file inside the asymmetric encryption script, alongside the username they entered when generating the keys and the name of the file that will be outputted. The reason for entering the username is that the system corresponds the generated keys to the username provided and will encrypt the file using their generated public key. Figure 3 conveys this encryption process.



```
┌──(kali㉿kali)-[~/Desktop/Cryptography/OpenSSL & GPG]
└─$ gpg --output Hybrid_enc.txt --encrypt --recipient 'RSA enc' AES.txt
```

Figure 3: RSA GPG Encryption Script

Each part of the encryption script is described below:
--output: This is the name assigned to the encrypted file that is outputted.
--encrypt: This is the command the user gives to tell the system they wish to encrypt a file.
--recipient: This is the username inputted when generating the keys so the system knows which public key to use for encryption.

3 Evaluation

3.1 Calculating Entropy Levels

The formula used to calculate levels of entropy is the Shannon Index Formula. [3] The formula is as follows: $H = -n\sum i=1\ p_i\ \mathrm{log2}\ p_i$.

H: This indicates the entropy level.
P: This indicates the proportion (n/N) of a number/letter found in the ciphertext (n) divided by the total number of number/letters found in the ciphertext (N).
Log2: This represents the natural log used.
∑: This is the scientific symbol for the sum of the calculations.
N: This represents the total number of numbers/letters found in the ciphertext.

References

[1] Kekayan, ‘Encrypt files using AES with OPENSSL’, *Medium*, Jul. 07, 2018. https://kekayan.medium.com/encrypt-files-using-aes-with-openssl-dabb86d5b748 (accessed Sep. 01, 2021).

[2] ‘Getting started with commandline encryption tools on Linux’, *HowtoForge*. https://www.howtoforge.com/tutorial/linux-commandline-encryption-tools/ (accessed Sep. 01, 2021).

[3] ‘Shannon Entropy Index Calculator - Online Information Entropy Finder’. https://www.dcode.fr/shannon-index (accessed Sep. 01, 2021).