# EHN P3 INSTRUCTIONS

***rev 1.0: initial release***

## 1. Key generation “rsakeygen”

**usage:**
./rsakeygen -bitLen <bit_string> -fopub <public_key_file> -fopriv <private_key_file> -init <rc4_init>
**example:**
./rsakeygen -bitLen 1024 -fopub pubkey.txt -fopriv privkey.txt -init “rc4initkey”

**** NB: I left several spaces in between the arguments for clarity purposes ONLY. ****

| Parameter | Description |
|---|---|
| <bits_string> | Specifies the number of bits that need to be generated for the given key. |
| <public_key_file> | The name of the file to store the generated output public key to. As a “txt” file. |
| <private_key_file> | The name of the file to store the generated output private key to. As a “txt” file. |
| <rc4_init> | String used to initialise the RC4 RNG **(ASCII only)** |

## 2. RC4 encryption using RC4 “rc4”

Your code must be able to encrypt **3 different** file types (pdf, jpg and png). The key will be provided as a text file **only**.

**Usage:**
./rc4 [-e/ -d] -fi <input file> -fo <output file> -key <key-file>
**Example:**
./rc4 -e -fi test.txt -fo rc4_testfile.rc4 -key rc4key.txt
./rc4 -d -fi doe.rc4 -fo rc4_testfile.png -key rc4key.txt

| Parameter | Description |
|---|---|
| -e / -d | Indicates encryption or decrryption. |
| <input file> | The name of the file to be encrypted or decrypted. |
| <Outputfile> | The name of the file to output the result to. |
| <keyfile> | The name of the file which contains the key used in the RNG generation. |

## 3. Encrypt RC4 key using RSA “rsaencrypt”

**usage:**
./rsaencrypt -key <key string> -fo <output file> -fopub <public key file>
**example:**
./rsaencrypt -key “QkZ9;xzv0Ja” -fo rsa_enc.rsa -fopub pubkey.txt

| Parameter | Description |
|---|---|
| <key string> | The name of the file which contains the plaintext. **(ASCII only)** |
| <output file> | The name of the file to output the ciphertext to. |
| <public key file> | The name of the file which contains the public key. |

## 4. Decrypt RC4 key using RSA “rsadecrypt”

**Usage:**
./rsadecrypt -fi <input file> -fopriv <private key file> -fo <output file>
**Example:**
./rsadecrypt -fi rsa_encrypted.rsa -fopriv priv.txt -fo decrypted_key.txt

| Parameter | Description |
|---|---|
| <inputfile> | The name of the file which contains the ciphertext. |
| <private key file> | The name of the file which contains the private key. |
| <output file> | The name of the file to output the plaintext to. |

- You **MUST** make sure that your program can write file names that correspond to the ones given via the arguments.
- In order for you to know the correct extension of a decrypted file, you have to read its magic number. Then add the extension before saving the file.