**Project Report**

|  |  |
| --- | --- |
| Program Title | **Private Messenger**  Secured Chatting Program |
| Description of the program | The program created to facilitate secured communication between two people by implementing RSA Cryptosystem.  The foundation of RSA algorithm is in the formula itself, then developed into a program with practical use.  Detailed program overview can be seen in the **How Program Works.jpg** and **Flow Chart.pdf** |
| The RSA cryptosystem | 1. Generate unique public and private key when user open the program. Unique means the public and private key are different with the last session (when user use the app) 2. Program will generate 2 random prime number for **p** and **q** 3. Share the public key (**e** and **n**) to the txt file, keep the private key inside program variable named **d** and **n.** 4. When encrypting, get the public key of receiver in the txt file, convert every single character into ASCII number, and encrypt the message using that public key. ASCII**e**mod **n** 5. When decrypting, read the encrypted message in txt file (in form of number), get the value of **d** and **n**, then the decryption will be performed. Encryptednumber**d** mod **n** = ASCII number. Convert back ASCII number to char. Show the result in the display.   More Explanation in the **How Program Works.jpg** and **Flow Chart.pdf** |
| Proof of the RSA Cryptosystem | 1. Generate 2 random prime number ( getPrime() in generator.py)      1. The 2 prime numbers stored in an array. Then the program will find **n** and **phi/totient** (defined as z) (messenger.py)      1. Get public and private key      1. Share public key to txt file based on user name      1. Stored public key      1. Original message. 20 characters (symbol, space, etc. included)      1. Encrypted message. 20 lines represent 20 characters      1. Message decrypted in user 2 |