INTRODUCTION TO CRYPTOGRAPHY – LAB 5

B.Tech. Computer Science and Engineering (Cybersecurity)

|  |  |
| --- | --- |
| Name: Anish Sudhan Nair | Roll No.: K041 |
| Batch: K2/A2 | Date of performance: 09/02/2022 |

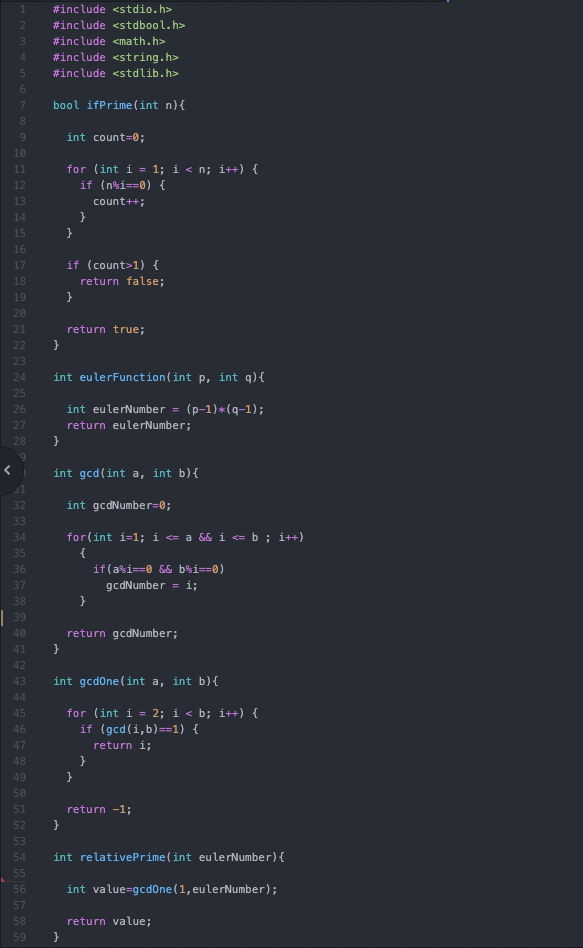
Aim: To code the RSA algorithm

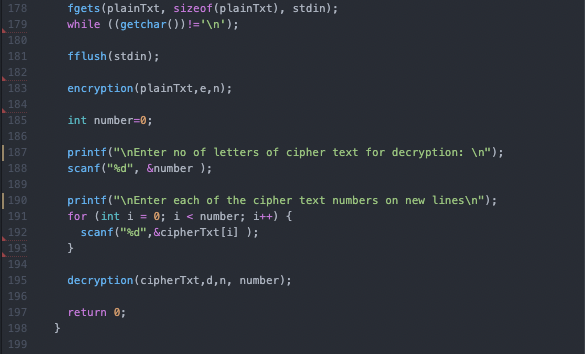
**Code:**

Language: C

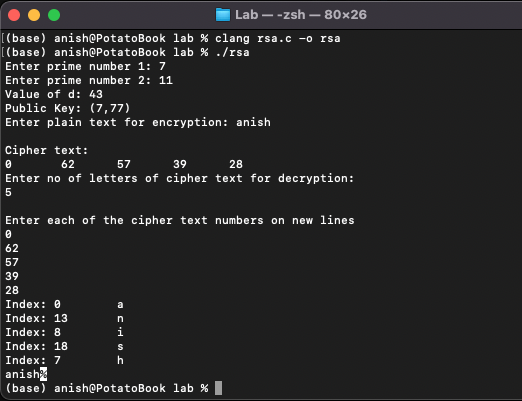
Editor: Atom

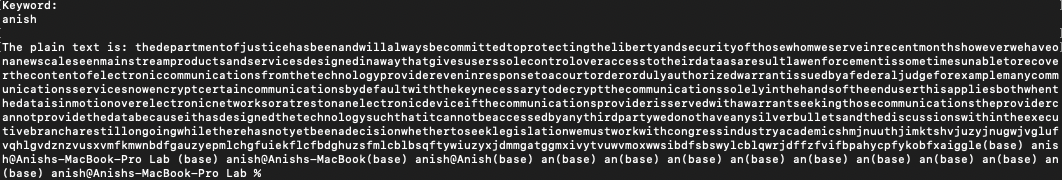
Compiler: clang/ZSH





**Complete Output:**





Questions:

1. Compare and contrast symmetric key encryption and asymmetric key encryption.

The Diffie-Hellman key exchange algorithm enables encrypted communication over an unsafe channel by using a public key.

Consider two user Anish and Hasin, who wish to communicate over a channel. In DH algorithm, each user would have a private key generated for them that is not disclosed to anyone. Using the private key and publicly informed prime numbers they each generate a key which they then share with each other. Using this shared key, their private key and the prime numbers, they generate the symmetric key which is used for encrypting and decrypting their messages.

2. Explain few of the applications of RSA.

- Secure Shell (SSH) [More secure than Telnet]

- Internet Protocol Security (IPSec)

- Transport Layer Security (TLS)

3. List advantages and limitations of RSA.

Advantages:

* It provides a symmetric key which enables communication in insecure channels
* Due to the symmetric key exchange the users need not necessarily know each other
* The keys are generated using large numbers and so manually cracking them is quite exhaustive

Disadvantages:

* It can only be used for symmetric key exchange
* It is very resource exhaustive (computational power)
* The algorithm itself does not perform the encryption
* There is no authentication process involved opening vulnerabilities for Man in the Middle Attacks

4. What are the most popular values of e in practice? Why?

65537

Fermat prime

Squaring

Only two 1s in the binary

Efficiency

5. Why does decryption using RSA take more time as compared to encryption?

In practice the public exponent is small, therefore encryption is faster in practice. On the other hand the private exponent is much larger (needs to be for proper security), therefore decryption is much slower.

RSA is considerably slow due to the calculation with large numbers. In particular the decryption where d is used in the exponent is slow. There are ways to speed it up by remembering p and q, but it is still slow in comparison to symmetric encryption algorithms.

The CRT replace one modular exponentiation with two, but these two exponentiations use half-size modulus and exponents, so each of them is about eight times faster than the non-CRT exponentiation. Thus, CRT speeds up RSA decryption by a factor of about 4. CRT requires knowledge of modulus factorization, so it cannot be applied to encryption, only to decryption.

On the other hand, RSA encryption uses the public exponent, which can be extremely small. A traditional RSA public exponent is 65537, thus 17 bits long. Exponentiation to the power 65537, a 17-bit integer, should be about 60 times faster than exponentiation to a 1024-bit power 𝑑 (the private exponent). Even with the CTR speed-up, RSA encryption should still be about 15 times faster than RSA decryption.