# Cryptography and Network Security

## Implementing and Breaking RSA

| Name | Section | BN |
|---|---|---|
| Abdullah Adel | 1 | 41 |
| Mohamed Ahmed Fathy | 2 | 10 |

**May 22nd, 2022**

# Implementation

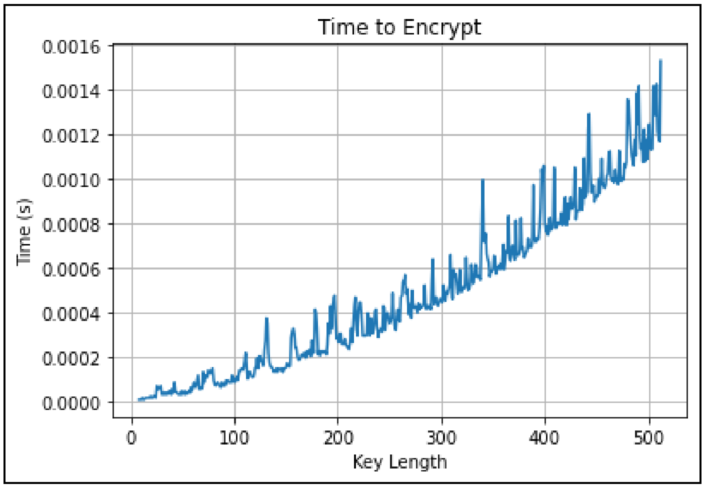
## 1. RSA Implementation

We created some helper functions to convert a string to an integer and vice versa. Then, we implemented necessary intermediate functions like GCD, Extended Euclidean Algorithm, Modulo Inverse, and a function that calculates $a^e \, mod \, n$ which is called PowMod. Finally, the encryption and decryption functions. The encryption function takes a message as a string, converts it to an integer, calculates its PowMod using the passed public key, and then converts it back to a string. Decryption, on the other hand, takes the ciphertext as a string, converts it to an integer, calculates it PowMod using the passed private key, and then converts it back to a string.

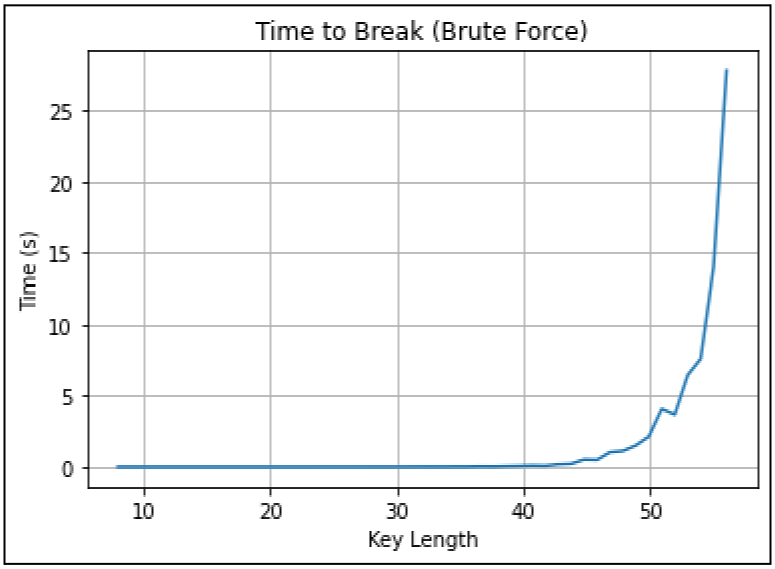
## 2. Sending and receiving program

The program consists of two parts (two files). The sender part is where a socket connection is opened and listens for any connections with it via the socket. The receiver part is where we get connected with the sender. Then, the receiver generates the keys either automatically using random numbers, or by taking the required parameters as inputs from the user. After generating the keys, the receiver passes the public key to the sender through the socket. The sender saves the public key in order to use it in encrypting the messages that are sent from it. The receiver listens for any messages coming from the sender to decrypt them with the private key.

## 3. Plot a graph of RSA encryption time vs. Key length



## 4. Plot a graph of Time to break the private key vs. the value of n



The result of the brute force attack vs. the key length was approximately an exponential function shifted to a key length of 50. When the key length increases by more than 50 bits, a brute force attack takes a much longer time to break.

## 5. Chosen Cipher Text attack for RSA

The attacker will be able to determine the sent message because:

$$X = (C \text{ x } r^{e})^{d} mod\, n$$

$$X = (M^{e} \text{ x } r^{e})^{d} mod\, n$$

$$X = C \text{ x } r\; mod\, n \qquad \because\ e \text{ and } d \text{ are inverses}$$

$X$, $r$, and $n$ are known to the attacker.

$$\therefore M = X \text{ x } r^{-1}\; mod\, n \qquad \text{(Using Extended Euclidean Algorithm to find } r^{-1}\text{)}$$

**Thank you**