**RSA Implementation**

Karim Taha

**Supervised By:**

Dr. Samir Shaheen

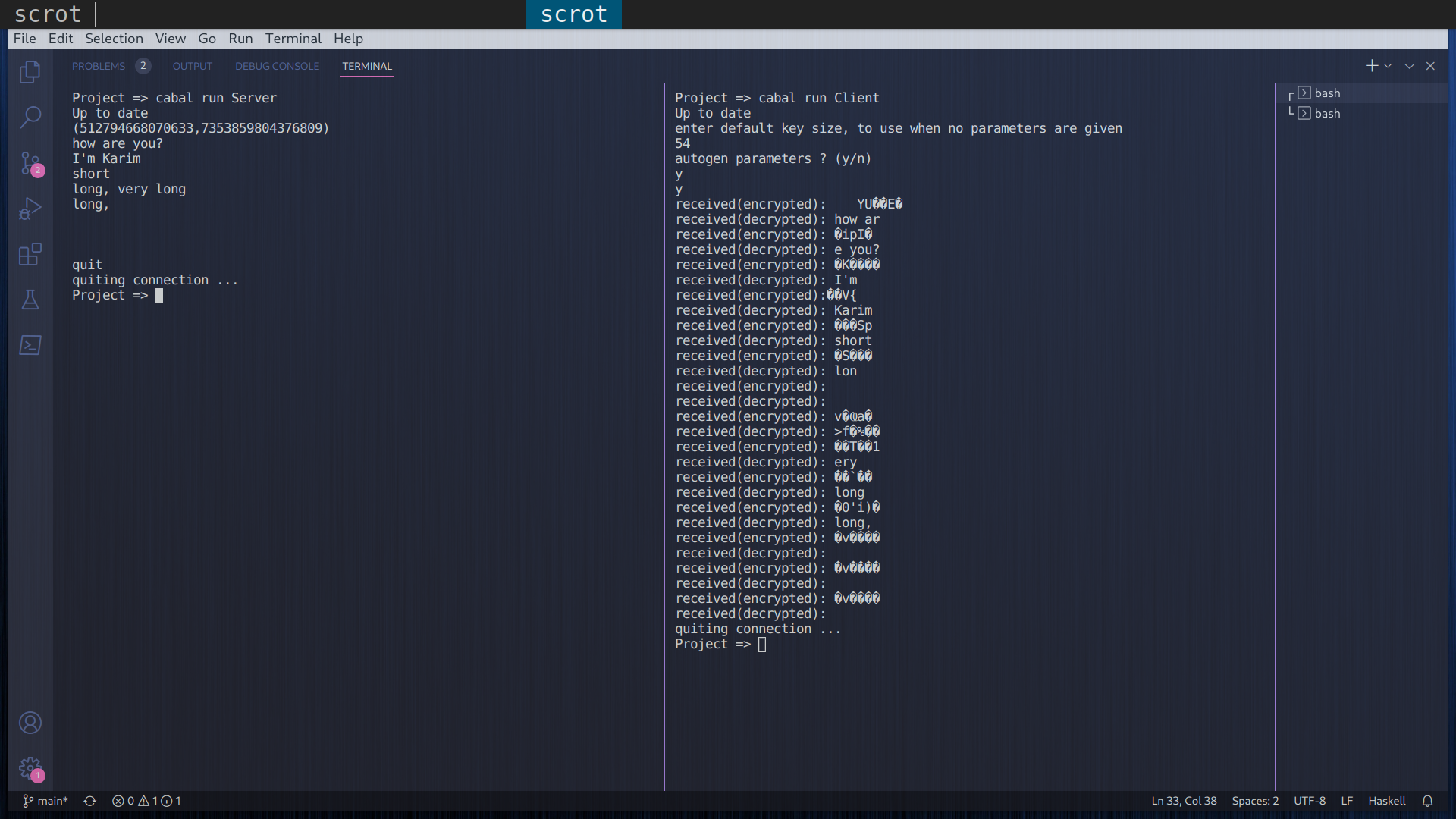
Eng. Sandra Wahid

Implementation:

Language : Haskell

Chat:

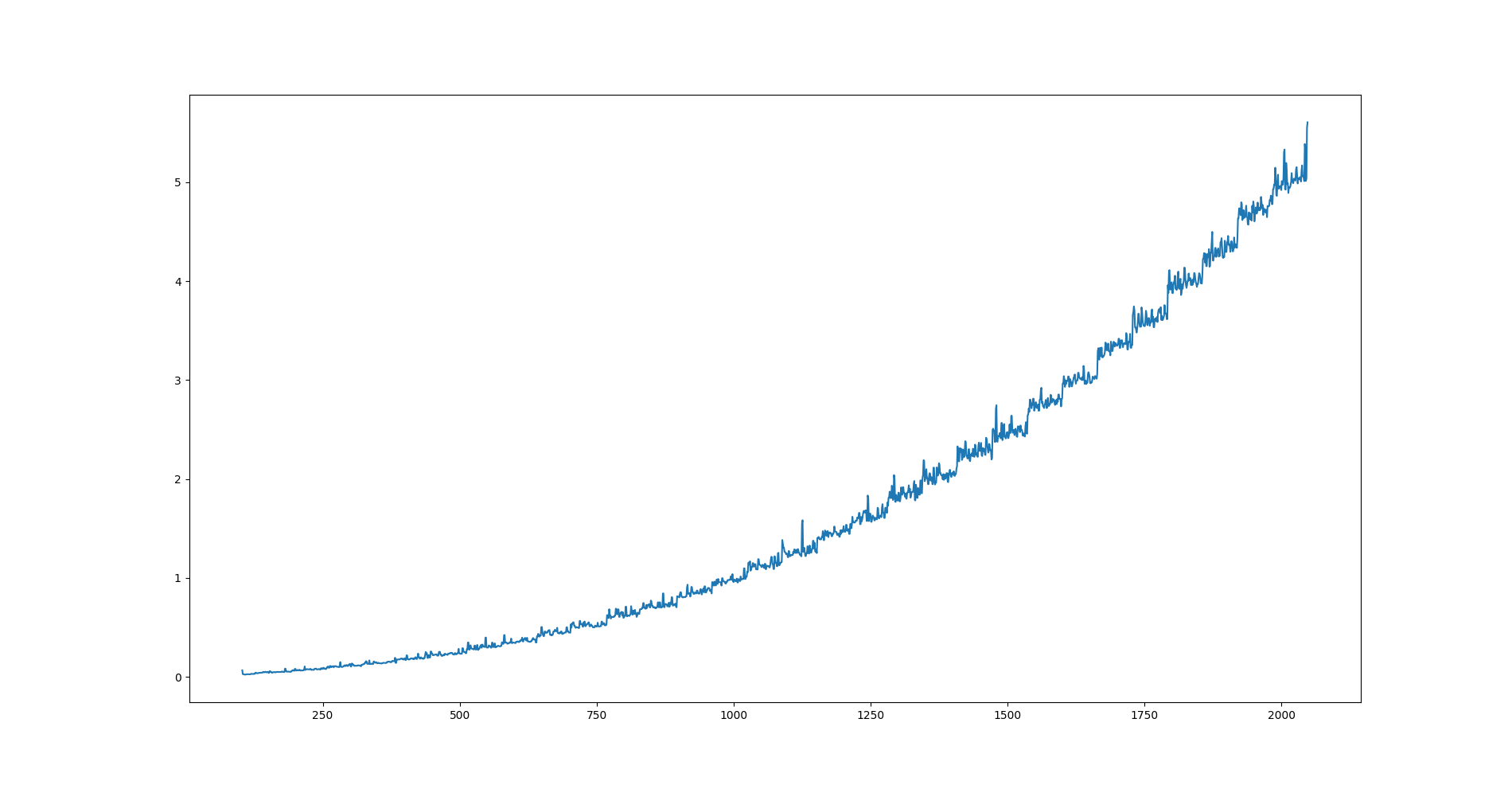
* TCP/IP socket based.
* The sender creates the socket, and the receiver connects to it.
* The receiver starts the communication by sending his public key to the sender.
* The sender then uses the public key to send encrypted messages to the receiver.
* If the message cannot fit within the keysize, it is chunked and sent one by one.



RSA :

* String message is converted to the corresponding 256 base number.
* A suitable keysize is chosen to fit the message, unless the user intereferes, in which the ciphering scheme cuts the string into chunks.
* Two random prime numbers p,q are generated, both having sizes equal to half the chosen keysize, so that n=p\*q <2^keysize.
* The public key is {e,n}, where e is random number with gcd(e,phi(n))=1, phi is the Euler totient function.
* The private key is {d,n}, where d is the inverse of e mod phi(n).
* Let the message be M, then C = Me mod n.
* Let the cipher be C, then M = Cd mod n.
* Finally, a function converts from the integer representation to the corresponding string.

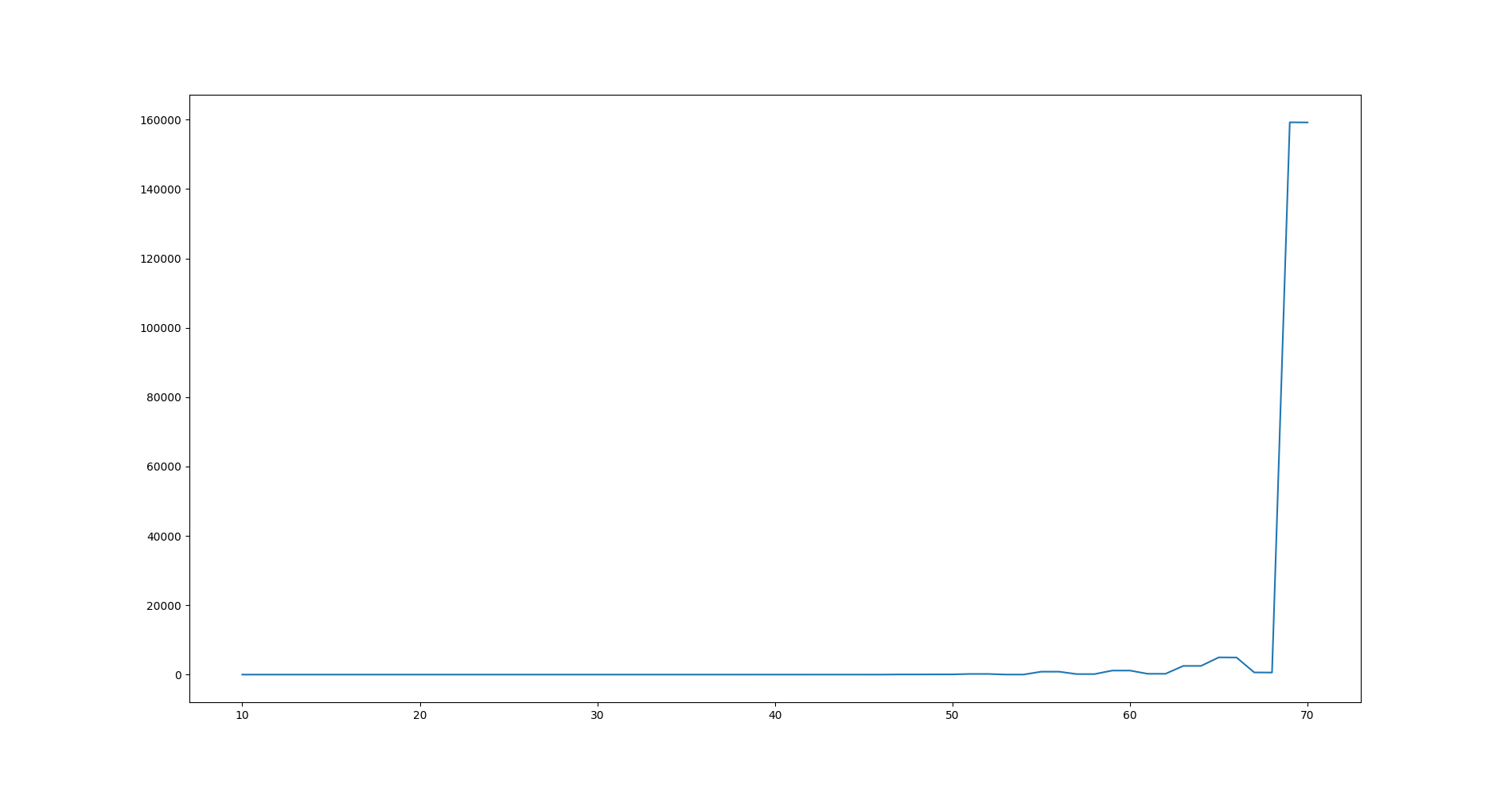
Encryption Time vs Key size:



The vertical axis is measured in milliseconds.

Observation: The encryption is quite fast, grows polynomially (lineary?) with the keysize.

Bruteforce attack on Private Key:

 The vertical axis is in milliseconds.

Observation: The bruteforce is so slow, time grows exponentially with the keysize.

It is implemented as a linear search between sqrt of n and 1.

Chosen Ciphertext Attack

* CCA is an attack where the cryptanalyst can gather information by obtaining the decryptions of chosen ciphertexts.
* From these pieces of information the adversary can attempt to recover the hidden secret key used for decryption.

