**CENG418 Assignment 1**

**1. RSA Encryption:**

For smaller bit sizes such as 2 and 4 bit N is too small to hold the numeric value of the message 24930. Even though such a small message size used with just 2 characters it could not encrypt the message. Later on when 8 bit key and larger is used, message is encrypted succesfully.

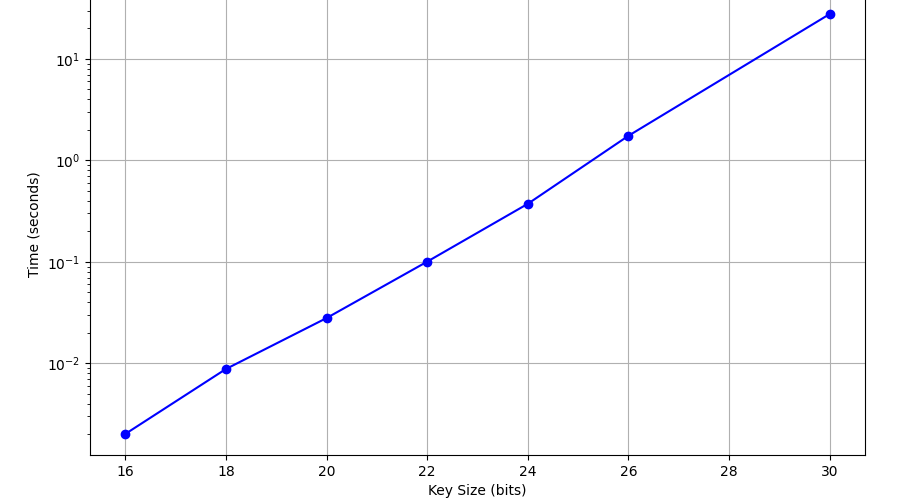
**2. Brute Force Cracking:**

**3. 256-bit RSA Breaking Time Calculation:**

For an n-bit RSA key, the number of possible keys is approximately 2ⁿ. Thus, a 256-bit RSA has:

Key Space= 2^256

Since testing a 256-bit key directly is infeasible, I measured brute-force times for smaller key sizes (8-bit to 30-bit) on my machine(CPU: AMD Ryzen 7,RAM: 16GB)



The recorded times were used to model the exponential growth of computation time with key size.

The time (T) required to brute-force an n-bit:

T(n)=a⋅2^(b.n) where: a = constant (hardware efficiency), b = growth rate (algorithmic efficiency)

I performed a log-linear regression on the measured data to estimate a and b (since the relationship is exponential)

Using the fitted model, I calculated:

T(256) = a⋅2^(b⋅256)

It was ***4.13e+67 seconds***, Approximately 1.31e+60 years for my computer.

This is far beyond practical feasibility, demonstrating why RSA-256 remains secure against brute-force attacks.

**4. Supercomputer Comparison:**

**5. Protocol Coding & Testing:**

.