RSA Assignment Problem Statement

Cryptographic & Security Implementations

Graded Assignment

Submission Deadline: **12 August 2025**

August 7, 2025

Objective

The objective of this graded assignment is to implement the RSA public-key cryptographic algorithm using the C programming language on a Linux-based operating system. The focus will be on measuring the performance of RSA key generation, encryption, and decryption in terms of clock cycles. The GNU MP (GMP) library's `mpz_t` data type must be used for large integer arithmetic. The implementation must be compiled using the `gcc` compiler.

Assignment Tasks

The assignment is divided into several clearly defined steps:

Step 1: Prime Number Generation

- Generate two large prime numbers p and q, each of 512 bits.
- Use GMP's `mpz_t` functions to generate these primes uniformly at random.
- Repeat the prime generation process one million (1,000,000) times.
- For each iteration, record the number of clock cycles taken to generate the primes.
- At the end, compute and print:
  - Minimum number of clock cycles
  - Maximum number of clock cycles
  - Average number of clock cycles

Step 2: Compute RSA Modulus and Euler's Totient

- Compute $N = p \times q$
- Compute $\phi(N) = (p - 1) \times (q - 1)$
- Record the number of clock cycles taken to compute these values

Step 3: Public and Private Key Generation

- Choose a fixed public exponent $e = 2^{16} + 1$
- Compute the private key d such that $e \cdot d \equiv 1 \mod \phi(N)$
- Record the number of clock cycles required to compute d

Step 4: Message Encryption and Decryption

- Prepare a message m of 1023 bits.
- Encrypt the message: $c = m^e \mod N$
- Decrypt the ciphertext: $m' = c^d \mod N$
- Verify that the decrypted message m' matches the original message m

Step 5: Repeat with Larger Key Sizes

Repeat the entire process (Steps 1–4) using:

- 768-bit primes for p and q
- 1024-bit primes for p and q

This will result in three datasets corresponding to key sizes:

- 512-bit primes
- 768-bit primes
- 1024-bit primes

Implementation Requirements

- Programming Language: C
- Operating System: Any Linux-based OS
- Compiler: `gcc`
- Library: GMP (GNU Multiple Precision Arithmetic Library)
- Data Type: Use `mpz_t` for all big integer computations
- Performance Measurement: Record the number of clock cycles for all computational steps

System Information

Include the following system specifications in your final report:

- CPU model and specifications
- RAM size and type
- Operating System version
- Compiler version (output of `gcc --version`)

Submission Guidelines

- Submit a complete and working source code with appropriate comments.
- Submit a report (in PDF) prepared using LaTeX (Overleaf or offline) that includes:
  - Problem statement (this document)
  - Output datasets and analysis
  - System specifications
  - Observations
- Mention any external sources, libraries, or documentation you took help from.
- **Submission Deadline: 12 August 2025**
- **Submit the report and code in the class group before the deadline.**