EE 337: Microprocessors Laboratory (Spring 2025)
Electrical Engineering
Indian Institute of Technology Bombay
MIDSEM: 25 points (Duration: 2.5 hours) Date: February 8, 2025

Instructions:

- You are allowed to use **only your** codes from previous labs. Sharing material or old codes is strictly not permitted.
- You must put your laptops in airplane mode.
- Any discussion during exam is not permitted.
- After the completion of **two questions**, **show your solution to your evaluating TA** for evaluation.
- TAs have one **additional test case** to evaluate. Performance on this test case will be **evaluated only once** and the **resulting score will be the sole determinant on this particular evaluation**.

Introduction

RSA (Rivest-Shamir-Adleman) is one of the most widely used public-key cryptosystems for secure data transmission. It is based on the mathematical difficulty of factoring large prime numbers, ensuring strong cryptographic security.

The RSA algorithm consists of three main steps: key generation, encryption, and decryption. In key generation, a pair of public and private keys is created using large prime numbers. Encryption involves converting plaintext into ciphertext using the recipient's public key, making it unreadable to unauthorized parties. Decryption is performed using the private key to retrieve the original plaintext.

RSA is commonly used in digital signatures, secure communications, and cryptographic protocols, providing confidentiality and authentication in modern security systems. For example, RSA signatures are used in SSL/TLS certificates in HTTPS. The objective of this exercise is to understand how RSA works by doing the Message Encryption and Message Decryption.

Example for Encryption Using RSA Algorithm:

- Suppose if you want to send a encrypted message over a channel.
- First choose two large distinct numbers p and q.
- If p or q is not prime, increment it until it becomes prime number.
- Example, given $p = 4$, $q = 8$,

$$p = 5 \text{ (incremented from 4 to prime)}, \quad q = 11 \text{ (incremented from 8 to prime)}$$

- Calculate the **modulus** n :

$$n = p \times q$$

Example:

$$n = 5 \times 11 = 55$$

The modulus will be made public but the factors p and q will be kept secret. If the primes are large, it will be very difficult to factorize n.

- Calculate the **totient** x :

$$x = (p - 1) \times (q - 1)$$

Example:

$$x = (5 - 1) \times (11 - 1) = 4 \times 10 = 40$$

- The **encryption exponent** e is some integer satisfying below conditions:

$$1 < e < x \quad \text{and} \quad \gcd(x, e) = 1.$$

  Here, $\gcd(x, e)$ is the greatest common divisor of x and e.
  Example: $e = 3$ that is satisfying the conditions.

- **Public Key:** The public key is the pair (n,e) required for the encryption and decryption. Where n is **modulus** and e is **encryption exponent**.

- **Message Encryption:** The **message m** is an integer satisfying

$$1 < m < n \quad .$$

  Anyone wishing to send an encrypted version of m to someone caculate c as

$$c = m^e \mod n$$

  This c is called the **ciphertext**. The sender can send this ciphertext over a insecure channel.

- Example: $m = 4$, $e = 3$, and $n = 55$:

$$c = 4^3 \mod 55 = 9.$$

  The given example is for understanding purpose only.

Q1. Encryption Using RSA Algorithm:

Given the numbers $p = 1$, $q = 4$ and $m = 6$. The task is to calculate the public key(n,e) and ciphertext c.

Steps to be followed:

- First do all the hand calculations for RSA Encryption for the given p, q and m values(follow the steps in given example).
- After doing the hand calculations, start with the code and check whether p and q are prime or not.
- To check both p and q are prime or not, write a prime number detection function.
- If p or q is not prime, increment by 1 each time in a loop until it becomes prime number and check it using prime number detection function.
- Store the values of p and q (after they became prime numbers) in **memory locations 70H and 71H**.
- Calculate the **totient** x and store that in **memory location 72H**.
- Calculate the **modulus** n and store that in **memory location 73H**.
- The **encryption exponent** e:
  Reduced the range of e to $e \in \{2, 3\}$ to avoid complex computations. Choose a number e satisfying the conditions from set {2,3}.
  No need to write code for gcd function, consider the value of e from hand calculation.
- Store the **encryption exponent** e in **memory location 74H**.
- Store the **message** m in **memory location 75H**.
- Calculate the **ciphertext** c and store that in **memory location 76H**.

Example for Decryption of RSA Ciphertext:

- Suppose you have a RSA encrypted message and you want to decrypt it.
- **Public Key:** The public key is the pair (n,e) after the encryption. Where n is **modulus** and e is **encryption exponent**.
- Example $n = 55$ and $e = 3$(from encryption example).
- **Decryption exponent d:** Use the equation, $(d \times e) \mod x = 1$ to calculate **decryption exponent** d, where x is **totient**. This can be rewritten as:

$$d = \frac{(x \times i) + 1}{e}$$

  where i is an integer $i \in \{1, 2, 3, \ldots, 255\}$.
  Increment i starting from 1 until the remainder of $((x \times i) + 1) \mod e = 0$, where the quotient becomes the private key d.
- Example, $x = 40$, $e = 3$, then from $(d \times e) \mod x = 1$, the value of $d = 27$.
- **Message Decryption**: The ciphertext c is decrypted as

$$m' = c^d \mod n$$

  This m' is the **decrypted message** of cipher text c.
- Example: If $d = 27$, $c = 9$, and $n = 55$:

$$m' = 9^{27} \mod 55 = 4.$$

  The given example is for understanding purpose only.
- Observe that message m and decrypted message m' are same.

Q2. Decryption of RSA Ciphertext:

Given the public key (n, e) and totient x, the task is to calculate the private key d and decrypted message m'.

Steps to be followed:

- First do all the hand calculations for RSA Decryption with the values n, e, c and x obtained from question1.
- Consider the **public key** (n, e) and **totient** x values obtained from question1.
- Write a function to calculate the **decryption exponent** d and store that in **memory location 77H**.
- Calculate the decrypted message m' and store that in **memory location 78H**.
- Show that the message m and decrypted message m' are same.

BONUS:

After successfully demonstrating the above two questions to your respective TA, then only you can try the bonus part.

- For Q1, write a gcd function to calculate the encryption exponent e.
- Given first number, $a = 12$ and second number, $b = 26$. Find the GCD for these numbers.

RUBRICS

- Hand calculations:
  - **2 marks** for encryption hand calculations.
  - **2 marks** for decryption hand calculations.
- Question 1:
  - **5 marks** for prime number detection function.
  - **6 marks** for message encryption.
- Question 2:
  - **10 marks** for message decryption.