Cryptographic Algorithms for RSA, Shift Cipher, and Vigenere Cipher

Fin Goscha

March 19, 2025

1 Introduction

This document describes the implementation of several cryptographic algorithms: modular exponentiation, GCD, modular inverse, RSA encryption, Shift Cipher, and Vigenere Cipher.

2 Modular Exponentiation

The `fast_mod_exp` This function is essential for calculating. The algorithm iteratively squares the base and multiplies the result when the exponent is odd, reducing the number of computations.

3 GCD and Modular Inverse

The `gcd` function computes the greatest common divisor of two numbers using the Euclidean algorithm. The `modInverse` function finds the modular inverse of an integer modulo M using a brute-force approach if the GCD is 1, otherwise, it returns -1 if the inverse doesn't exist.

4 RSA Functions

The `totient` function calculates Euler's Totient, $(p-1)(q-1)$, where p and q are primes. The `RSA` function allows encryption and decryption using modular exponentiation, where the user can input the public or private key.

5 Shift Cipher

The `shift` function applies a simple shift cipher by shifting each ASCII value of a message by a given key.

6 Vigenere Cipher

The `vigenereEncrypt` and `vigenereDecrypt` functions perform encryption and decryption using the Vigenere cipher. The `generate_key` function ensures that the key is extended to match the length of the message.

7 Main Function

The `main` function provides an interactive interface where the user can choose to use the Shift Cipher, Vigenere Cipher, or RSA encryption, with options for encryption or decryption.