

**RSA Implentation from Scratch**

==========================

**Overview**

This project implements the RSA algorithm, a widely used method for secure data transmission. RSA is based on the principle of asymmetric cryptography, where a pair of keys is used: one for encryption (public key) and another for decryption (private key).

**Features**

* **Key Generation**: The project generates two large prime numbers p and q, each greater than 2048 bits, using the Rabin-Miller primality test.
* **Public and Private Key Calculation**: Calculates the public key e and private key d based on the generated primes.
* **Encryption and Decryption**: Encrypts plaintext using the public key and decrypts ciphertext using the private key.

**How It Works**

**1. Key Generation**

1. **Prime Numbers**: Generate two large prime numbers p and q.
2. **Modulus Calculation**: Calculate n = p \* q.
3. **Totient Function**: Calculate Ø(n) = (p-1) \* (q-1).

**2. Public Key Calculation**

* Choose a public key e such that e and Ø(n) are coprime (i.e., their greatest common divisor is 1).

ie **gcd(e,Ø(n)) =1**

**3. Private Key Calculation**

* Calculate the private key **d** as the **modular inverse** of **e** modulo **Ø(n).**

**4. Encryption and Decryption**

* **Encryption**: Use the **public key e** to encrypt **plaintext into ciphertext.**

**ciphertext = (plaintext^e) % n**

* **Decryption**: Use the **private key** **d** to decrypt **ciphertext back into plaintext.**

**plaintext= (ciphertext^d) % n**