rsa.py: This program implements an RSA class for the encryption and decryption of data. It works as follows, it initializes with a public key and reads two prime numbers from files to create a private key. It uses the Extended Euclidean Algorithm to find the modular inverse. The encrypt method encrypts text in blocks and writes the ciphertext to a file, while decrypt uses the Chinese Remainder Theorem to efficiently decrypt and write the plaintext to a file. The class is designed to be run from the command line with options for encryption or decryption and file inputs.

breakRSA.py: This Python program defines a class breakRSA that aims to break RSA encryption using low public exponent attacks. It generates three sets of RSA keys with a public exponent of 3, calculates the corresponding private decryption exponents, and saves key information. The encrypt method encrypts a plaintext file into three separate files using the generated keys. The crack method attempts to decrypt ciphertext using the Chinese Remainder Theorem (CRT) and a cube root attack, indicating it's designed to exploit the vulnerability when the same message is encrypted with small public exponent across different public keys. It's intended to be executed from the command line with options to either encrypt or crack the encryption.