Write a program(s) (in any language) to help you answer these questions:

**What IS the Modulus Tree Encryption/Decryption algorithm?**

See the code.

**What should dictate the choice of moduli, both how many and which values?**

**Choice of moduli, Traditional:**

**Choice of moduli, Reduced:**

**How many:** As many as we need.

**Which:** More the better. Choose smaller ones to get more option. Can they be guessed?

**How difficult is it to crack the encryption?**

**Traditional Tree:** You can craft a special file and pass it to the encrypt function to build the tree. For example, with 128 bit, you only need to create a file with the first 103 number to find out the first prime. Repeat until you have found all the primes.

**Option 1:** Multiply by some other relatively-prime number. How many levels deeper – TBD.

**Does knowledge of any plaintext/ciphertext pairs allow recovering the key?**

Yes, partially.

**If so, which ones, how many, and how are they related to each other?**

If you know enough values that are adjacent on the tree you can tell the size of the group which is the current prime number. **Note:** Could be on small only.

**If not, provide a proof that the algorithm is “secure enough” for all practical purposes.**

Let n= # of co-prime numbers, a=number of bits per chunk,

Example: If you have a tree with n=10, and a=128, the possible combinations and permutations of co-primes that can be used to encrypt numbers between , or