RSA Public Key Generation Project: Design Document

Sam Demorest and Jason Pearson

CS 6310

Dr. Dedoncker

**Design Overview**

Our program is made up of two separate programs. One program is for the large prime number creation and the second program is for the key creation. The idea is that the first program is to be a standalone prime number creator and the second program is to use the prime number generator. The reason for this is so other programmers can find our prime number generation code and use it in their projects as well.

The prime generator will use a few different methods. The first method we will use is to try and divide the number by common divisors two, three and five. The idea of this one is to cut out all the easy to determine compound numbers out of the mix.

The key generator will call the prime generation method of the first program and that will take care of all the work for prime number generation. After it gets the prime numbers p and q we use those to compute n which is simply p times q. Next we need to determine φ(*n*) which is simply *n* - (*p* + *q* -1). After these are all computed we create a good e value and compute d. The e value is simply a number 1 < e <φ(*n*). Then d = e-1(mod(φ(*n*)), which is the private key part of the encryption.

**AKS References:**

http://www.cse.iitk.ac.in/users/manindra/algebra/primality\_v6.pdf

http://mathworld.wolfram.com/AKSPrimalityTest.html

**Rust References:**

https://doc.rust-lang.org/

http://rustbyexample.com/

https://github.com/rust-lang/rust

**RSA Cryptosystem References:**

http://mathworld.wolfram.com/RSAEncryption.html

http://stackoverflow.com/questions/12749858/rsa-public-key-format

https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture12.pdf