RSA Encryption Implementation Writeup

Ariel Young, Nashir Janmohamed

June 14th, 2020

Summary

1 Introduction

Talk about motivation and application and overall process.

2 Theory

2.1 RSA Algorithm

Give overview of algorithm here, just copy wikipedia

2.1.1 Random number generation

The *linear congruential method* [1] . . .

2.1.2 Generating large primes

Various approaches were investigated, including using the *Sieve of Eratosthenes* [2], the *Sieve of Atkin* [3], ... After more research on best practices for generating large primes we learned that an alternative and more scalable approach is to generate random large numbers, and then performing primality tests to determine their viability. The *Miller Rabin test* [4] . . .

2.1.3 Modular Multiplicative Inverse

To compute the exponent used in the private key, we implemented the Extended Euclidean Algorithm for computing the *modular multiplicative inverse* [5] . . .

3 Implementation

3.1 one subsection for each portion of program

Maybe put pseudocode?

4 Analysis

4.1 Chi Squared Test

4.2 Bitmap

4.3 Runtime Analysis

Talk about big O to make him excited :o

5 Usage

5.1 CLI

Show that using a new key will produce garbage

6 References

examples pls remove and replace

References

[1] Add source here

[2] Add source

[3] Add source

[4] Add source

[5] Add source

[6] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley, Reading, Massachusetts, 1993.

[7] Albert Einstein. *Zur Elektrodynamik bewegter Körper*. (German) [*On the electrodynamics of moving bodies*]. Annalen der Physik, 322(10):891–921, 1905.

[8] Knuth: Computers and Typesetting,
`http://www-cs-faculty.stanford.edu/~uno/abcde.html`