Team 1 EN.605.204.81
ARM32 RSA Design Document

Rohan Abraham, Tero Suontaka, Sullivan Prellwitz

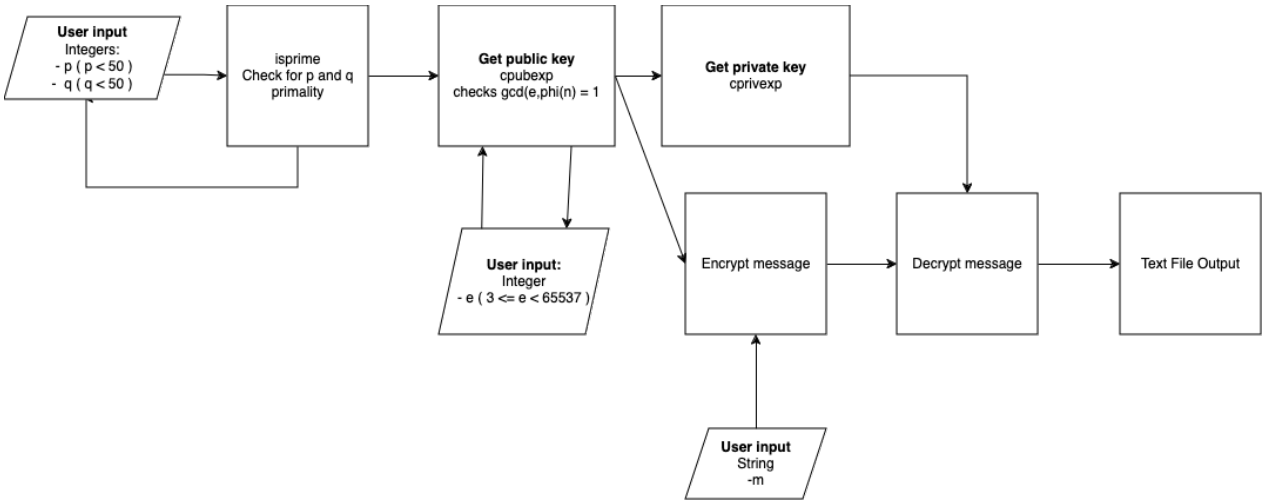
March 03, 2024

Contents

1 Goals

Purpose: Encrypt and decrypt messages using a custom RSA implementation in ARM32 assembly. Implement a modular design for all functions and create a library of assembly code that enables the generation of a public and private RSA keys using user specified values.

2 Architecture



3 Functions

3.1 gcd (Greatest Common Divisor)

Input: 2 integer values.
Output: The greatest common divisor of the input integers.

3.2 pow

Input: 2 integer values, base b and an exponent e.
Output: b^e

3.3 mod (Modulo)

Input: 2 integers a and b.
Output: a mod b

3.4 tot (Totient Calculation)

Input: 2 prime numbers p and q.
Output: $\phi(n) = (p-1)(q-1)$

3.5 cpubexp (Calculation of public key exponent)

Input: 2 prime numbers p and q, and an exponent e such that e is an integer and is not a factor of $\phi(n)$ and $1 < e < \phi(n)$.
Output: Public key exponent.

3.6 cprivexp (Calculation of private key exponent)

Input: e, $\phi(n)$ an integer x. and is not a factor of $\phi(n)$ and $1 < e < \phi(n)$
Output: Private key exponent.

3.7 encrypt

Input: An ASCII string plaintext to be encrypted and associated public key $pubKey$.
Output: ASCII string ciphertext

3.8 decrypt

Input: An ASCII string ciphertext and associated private key $privKey$.
Output: ASCII string plaintext

4 Testability

The individual components of the RSA implementation will be tested using a test script per function. These test scripts will cover normal, edge, and absurd cases in order to ensure proper functionality. The test scripts will be able to be run all at once using a control script.
Test scripts will be help within a `/tests` folder within the project repository and will take the form

```
{function_name}-tests.{file_extension}
```

5 Timeline

5.1 March 11 - 15

- First implementation meeting
- Initialize code repository
- mod function implementation finished, tests written

5.2 March 25 - 29

- Second implementation meeting
- gcd, pow, and tot implementation finished, tests written
- Plan next implementation steps

5.3 April 8 - 20

- Meet as needed
- RSA implementation finished (April 20), tests written
- Creation of testing control script

5.4 April 21 - 27

- Complete testing
- Squash bugs
- Prep repository and extra materials for submission

5.5 April 28

- Submit implementation