Team 1 EN.605.204.81 ARM32 RSA Design Document V2

Rohan Abraham, Tero Suontaka, Sullivan Prellwitz

April 28, 2024

Contents

1 Goals 2

2 Architecture 3

3 Functions 3
- 3.1 libIO.s . . . 3
  - 3.1.1 stringToArray . . . 3
  - 3.1.2 arrayToString . . . 3
  - 3.1.3 writeFile . . . 4
  - 3.1.4 writeArray . . . 4
  - 3.1.5 readArray . . . 4
- 3.2 libMath.s . . . 4
  - 3.2.1 gcd . . . 4
  - 3.2.2 mod . . . 5
  - 3.2.3 isPrime . . . 5
  - 3.2.4 totient . . . 5
- 3.3 libRSA.s . . . 5
  - 3.3.1 cprivexp . . . 5
  - 3.3.2 cpubexp . . . 6
  - 3.3.3 process . . . 6
  - 3.3.4 processArray . . . 6
  - 3.3.5 generateKeys . . . 7
  - 3.3.6 encrypt . . . 7
  - 3.3.7 decrypt . . . 7
- 3.4 main.s . . . 7
  - 3.4.1 main . . . 7

4 Testability 8
- 4.1 Notes on the test project: . . . 8

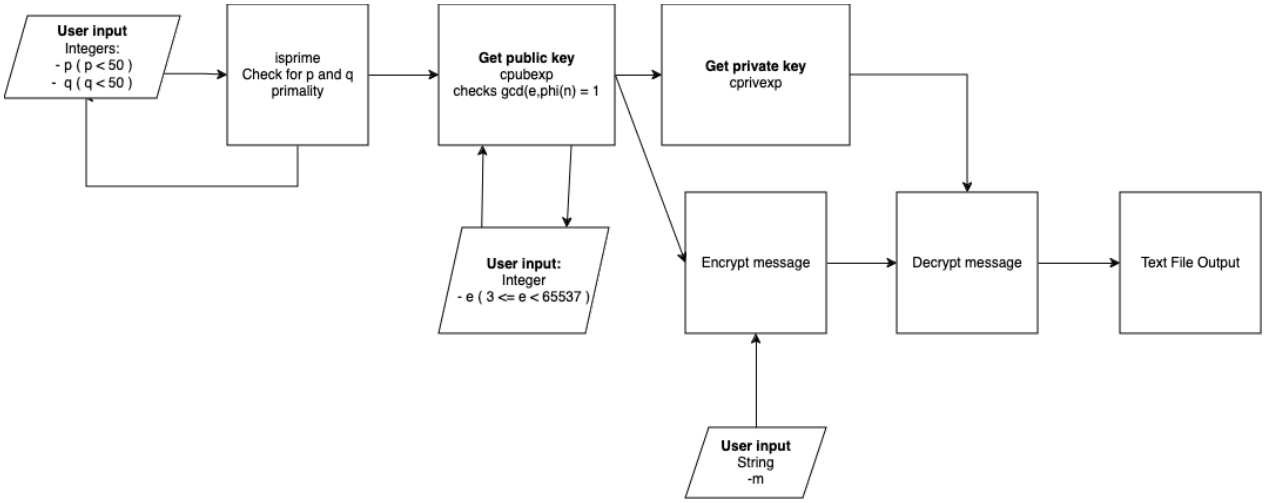
5 Timeline 9
- 5.1 March 11 - 15 . . . 9
- 5.2 March 25 - 29 . . . 9
- 5.3 April 8 - 20 . . . 9
- 5.4 April 21 - 27 . . . 9
- 5.5 April 28 . . . 9

6 Screenshots 10

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 libIO.s

3.1.1 stringToArray

Purpose: Converts a string (byte array) to an array of 32 bit integers
input:

- r0 - pointer to string
- r1 - size of string

Output:

- r0 - pointer to integer
- r1 - size of array

3.1.2 arrayToString

Purpose: Converts an integer array to a null delimited string
input:

- r0 - pointer to integer array
- r1 - size of array

Output:

- r0 - pointer to string
- r1 - size of string

3.1.3 writeFile

Purpose: Write to a file, name provided by user
input:

- r0 - name of file to write
- r1 - pointer to message to write

3.1.4 writeArray

Purpose: Write 32 bit integer array to a file
input:

- r0 - pointer to string
- r1 - pointer to message to write
- r2 - length of string

3.1.5 readArray

Purpose: Read file to 32 bit integer array
input:

- r0 - name of file to read

Output:

- r0 - pointer to array
- r1 - array length

3.2 libMath.s

3.2.1 gcd

Purpose: Computes the greatest common divisor of two integers
input:

- r0 - first integer to compute gcd of
- r1 - second integer to compute gcd of

Output:

- r0 - greatest common divisor of two input integers

3.2.2 mod

Purpose: Modulo calculation: r0 mod r1 = r0
input:

- r0 - first integer to compute modulo
- r1 - second integer to compute modulo

Output:

- r0 - modulo value

3.2.3 isPrime

Purpose: Determines if a number is prime
input:

- r0 - integer to test

Output:

- r0 - binary value indicating primality returns -1 for invalid values

3.2.4 totient

Purpose: Totient calculation $\Phi(n) = (p - 1)(q - 1)$ s.t. p and q are prime
input:

- r0 - p
- r1 - q

Output:

- r0 - return: totient value of (n) or r0 == -1 if p or q are NOT prime (error)

3.3 libRSA.s

3.3.1 cprivexp

Purpose: Calculates the private exponent. Calculates multiplicative inverse of public key over ring of integers mod n
input:

- r0 - public exponent (e)
- r1 - integer such that gcd(r0,r1) = 1 (phi(n))

Output:

- r0 - private exponent returns -1 if gcd(r0,r1) != 1

3.3.2 cpubexp

Purpose: Validates the public exponent s.t. $1 < e < \Phi(n)$ and e is co-prime to $\Phi(n)$ [$\gcd(e, \Phi(n)) = 1$]
input:

- r0 - p
- r1 - q
- r2 - e

Output:

- r0 - pub exponent or -1 if error

3.3.3 process

Purpose: Processes the input for RSA encryption and decryption. For encryption, use private key as exponent. For decryption, use public key as exponent
input:

- r0 - integer base a
- r1 - integer exponent b
- r2 - integer modulus n

Output:

- r0 - a ^ b mod n

3.3.4 processArray

Purpose: Processes an integer array for RSA encryption and decryption. Applies a^b mod n for all a in array.
input:

- r0 - pointer to integer array
- r1 - size of array
- r2 - integer exponent b
- r3 - integer modulus n

Output:

- r0 - pointer to processed integer array
- r1 - size of array

3.3.5 generateKeys

Purpose: Prompt user for primes and public exponent and generate private key

3.3.6 encrypt

Purpose: Encrypts a message given user input public key and modulus and writes to encrypted.txt

3.3.7 decrypt

Purpose: Decrypts a message from encrypted.txt given user input private key and modulus and writes plaintext to plaintext.txt

3.4 main.s

3.4.1 main

Purpose: Drives the generation of keys, encryption, and decryption

4 Testability

To facilitate easy testing the majority of functions are called directly in the Rust project located in the `/tests` directory. This project is made up of three main parts:

1. `lib.rs` - the test library in rust
2. `testHelper.s` - an arm assembly helper file for the test library
3. `Makefile` - the Makefile is responsible for building and linking all related assembly code to a shared library `libRSA.so`

4.1 Notes on the test project:

- Throughout the test project a public key, private key, and modulus value that are referenced. These values are as follows:
  - `pubkey: 557`
  - `privkey: 1493`
  - `mod: 1763`
- Text files created by the test project will live within the `test/` directory
- For tests using a plaintext the string used is `hello plaintext`
- For tests using an array version of the plaintext the string remains the same and the array values are base 10 integers representing character ASCII value
- For tests using a ciphertext the array values provided in the tests are derived from the above plaintext, pub/priv key, and mod values
- To circumvent problems with memory management and lifetime all arrays are dealt with through files to ensure correctness
- For information on compiling and running the tests please see the `README.md` in the `test/` directory

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

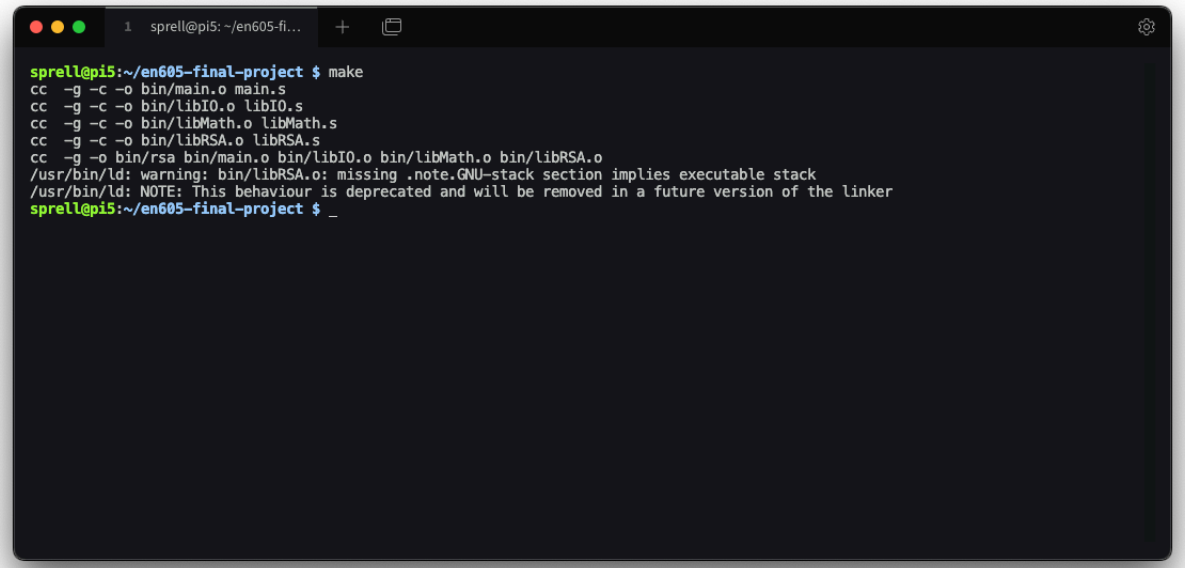
5.4 April 21 - 27

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

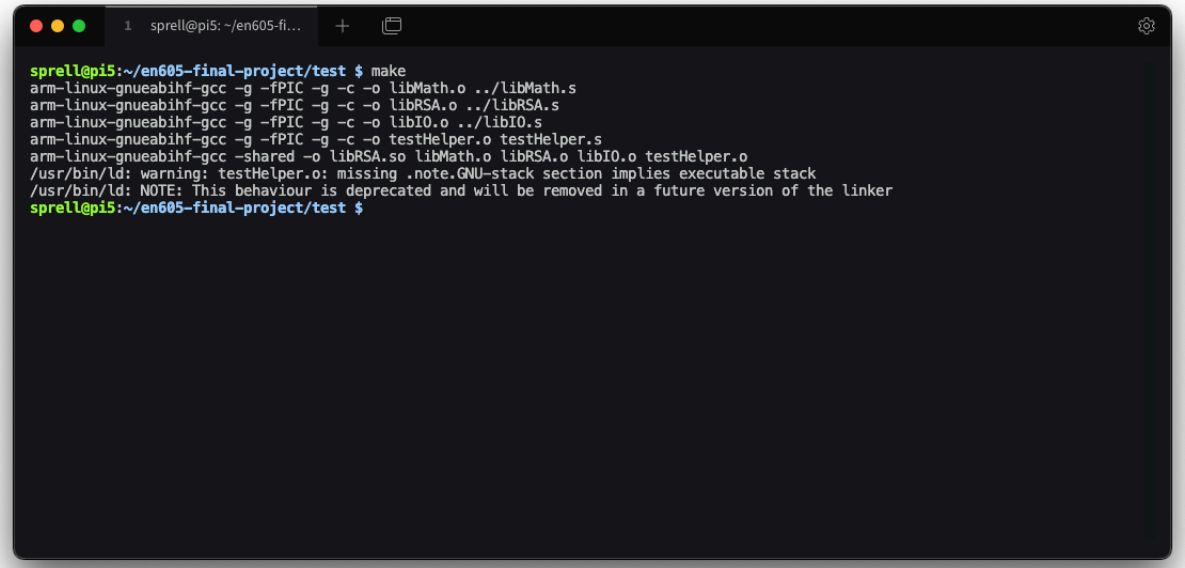
5.5 April 28

- Submit implementation

6 Screenshots



```
sprell@pi5:~/en605-final-project $ make
cc  -g -c -o bin/main.o main.s
cc  -g -c -o bin/libIO.o libIO.s
cc  -g -c -o bin/libMath.o libMath.s
cc  -g -c -o bin/libRSA.o libRSA.s
cc  -g -o bin/rsa bin/main.o bin/libIO.o bin/libMath.o bin/libRSA.o
/usr/bin/ld: warning: bin/libRSA.o: missing .note.GNU-stack section implies executable stack
/usr/bin/ld: NOTE: This behaviour is deprecated and will be removed in a future version of the linker
sprell@pi5:~/en605-final-project $ _
```



```
sprell@pi5:~/en605-final-project/test $ make
arm-linux-gnueabihf-gcc -g -fPIC -g -c -o libMath.o ../libMath.s
arm-linux-gnueabihf-gcc -g -fPIC -g -c -o libRSA.o ../libRSA.s
arm-linux-gnueabihf-gcc -g -fPIC -g -c -o libIO.o ../libIO.s
arm-linux-gnueabihf-gcc -g -fPIC -g -c -o testHelper.o testHelper.s
arm-linux-gnueabihf-gcc -shared -o libRSA.so libMath.o libRSA.o libIO.o testHelper.o
/usr/bin/ld: warning: testHelper.o: missing .note.GNU-stack section implies executable stack
/usr/bin/ld: NOTE: This behaviour is deprecated and will be removed in a future version of the linker
sprell@pi5:~/en605-final-project/test $
```

```
sprell@pi5:~/en605-final-project $ ./bin/rsa
Please enter:
1 - to generate keys
2 - to encrypt plaintext
3 - to decrypt ciphertext
Selection: 1

=== Small key size RSA generation ===
Please prepare the following information:
- Positive integers P and Q such that P & Q are both prime
- Public key value e s.t. 1 < e < Φ(n) and e is co-prime to Φ(n) [ gcd(e, Φ(n)) = 1 ]
Enter first prime: 41
Enter second prime: 43
Enter desired public key: 557
Modulus: 1763
Public Key: 557
Private Key: 1493
sprell@pi5:~/en605-final-project $
```

```
sprell@pi5:~/en605-final-project $ ./bin/rsa
Please enter:
1 - to generate keys
2 - to encrypt plaintext
3 - to decrypt ciphertext
Selection: 1

=== Small key size RSA generation ===
Please prepare the following information:
- Positive integers P and Q such that P & Q are both prime
- Public key value e s.t. 1 < e < Φ(n) and e is co-prime to Φ(n) [ gcd(e, Φ(n)) = 1 ]
Enter first prime: 62
Enter second prime: 42
ERROR: One of the given integers was not prime
Enter first prime: 55
Enter second prime: 22
ERROR: One of the given integers was not prime
Enter first prime: 41
Enter second prime: 43
Enter desired public key: 4
ERROR: Invalid public key
Must be between 1 and 1680 and coprime to 1680
Enter desired public key: 557
Modulus: 1763
Public Key: 557
Private Key: 1493
sprell@pi5:~/en605-final-project $ _
```

```
sprell@pi5:~/en605-final-project $ ./bin/rsa
Please enter:
1 - to generate keys
2 - to encrypt plaintext
3 - to decrypt ciphertext
Selection: 2
Enter public key: 557
Enter modulus: 1763
Enter text to encrypt: hello plaintext
Encrypted text is in encrypted.txt
sprell@pi5:~/en605-final-project $ cat encrypted.txt
263 762 309 309 1715 237 1094 309 1741 373 1218 235 762 3 235 sprell@pi5:~/en605-final-project $
```

```
sprell@pi5:~/en605-final-project $ ./bin/rsa
Please enter:
1 - to generate keys
2 - to encrypt plaintext
3 - to decrypt ciphertext
Selection: 3
Enter private key: 1493
Enter modulus: 1763
Decrypted text is in plaintext.txt
sprell@pi5:~/en605-final-project $ cat plaintext.txt
hello plaintextsprell@pi5:~/en605-final-project $
```

```
    Finished test [unoptimized + debuginfo] target(s) in 0.58s
     Running unittests src/lib.rs (target/debug/deps/lib-7f0f868d4213ced7)

running 14 tests
test tests::array_to_string_expect_true ... ok
test tests::cpubexp_expect_valid ... ok
test tests::cprivexp_expect_valid ... ok
test tests::decrypt_expect_true ... ok
test tests::gcd_expect_true ... ok
test tests::is_prime_expect_correct ... ok
test tests::is_prime_expect_false ... ok
test tests::string_to_array_expect_true ... ok
test tests::read_array_expect_true ... ok
test tests::totient_pq_not_prime ... ok
test tests::totient_pq_prime ... ok
test tests::write_array_expect_true ... ok
test tests::write_to_file_expect_true ... ok
test tests::encrypt_expect_true ... ok

test result: ok. 14 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

   Doc-tests lib

running 0 tests

test result: ok. 0 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s

sprell@pi5:~/en605-final-project/test $ _
```