1 SMA0180 - Matemática Discreta I: Entrega de Trabalho 7

Nome: Rafael Zimmer; nUsp: 12542612 Data: 24/10/2021

2 Algoritmo de Exponenciação Modular

2.1 Explicação do Algoritmo

O algoritmo do RSA usando n de 16 bits requer o uso de dois valores, p e q, tais que $pq = n$, e ambos p e q são primos. Para achar os valores de p e q, foi usado uma tabela de referência com os números primos menores que 1 bilhão, e escolhido dois valores tais que seu produto coubesse em apenas 16 bits de memória. Após isso, será obtido o produto $\phi = (p-1)(q-1)$, assim como um outro número $e \in Z_\phi$, que é também invertível em Z_ϕ. Esse número e será usado como chave pública de codificação do Algoritmo RSA16, e para a decodificação da mensagem, o inverso de $f = e^{-1} em Z_\phi$ será obtido e usado como chave privada, usando o algoritmo de Euclides extendido, já apresentado e desenvolvido no trabalho 5.

Após obtermos os valores de ambas as chaves, para a codificação basta executar a exponenciação modular com a mensagem como base, a chave como expoente e $n = pq$ como módulo, usando novamente uma função desenvolvida previamente: Ex:

$$c(15) = 15^e \mod n \tag{1}$$

$$15 = [c(15)]^f \mod n \tag{2}$$

3 Casos Teste

```
gcc -o main main.o
./main c 15

Chave publica: 24329
Resultado da codificacao =  24207


./main d 24207

Chave privada: 15629
Resultado da decodificacao =  15
```

```
gcc -o main main.o
./main c 827

Chave publica: 24329
Resultado da codificacao =  11229


./main d 11229

Chave privada: 15629
Resultado da decodificacao =  827
```

3.1 Caso Extra

```
gcc -o main main.o
./main c 34343

Chave publica: 24329
Resultado da codificacao =  42591
```

```
6
7
8 ./main d 42591
9
10 Chave privada: 15629
11 Resultado da decodificacao =  34343
```