**Universidade Estadual de Londrina** 30/05/2022

Relatório Trabalho Matémática Discreta e Finita II – Criaptografia RSA

Aluno: Marco Túlio Alves de Barros RA: 202100560105

Aluno: Vinícus Ferreira Schiavon RA: 202100560434

Descrição do Metódo: Dada um frase desejamos criptografá-lá em números.

Para tal temos que cada letra do alfabeto e o espaço correspondem a um número: A=10, B=11, C=12, D=13, E=14, F=15, G=16, H=17, I=18, J=19, K=20, L=21, M=22, N=23, O=24, P=25, Q=26, R=27, S=28, T=29, U=30, V=31, W=32, X=33, Y=34, Z=35, ESPAÇO=99.

Serão fornecidos dois números primos muito grandes P e Q (essa informação é mantida em segredo). Para fins de testes, usaremos números primos de 1 a 100.

Deve ser calculado N = P \* Q, tal que N é a chave pública.

Deve ser calculado PHI(N) = (P - 1) \* (Q – 1).

Também deve ser calculado um E, tal que:

* E < PHI(N);
* Mdc (E, PHI(N)) = 1.

Portanto existe um D < PHI(N) e D > 0, que originam a equação diofantina:

* 1 = K \* PHI(N) + D \* E

Por fim temos que a chave de codificação será (E, N) e a chave de decodificação será (D, N).

Explicação do código: O primeiro passo é realizar a leitura da frase a ser codificada, a frase deve ser apresentada em um arquivo txt de input.

Os valores de P e Q são inicializados diretamente no começo do programa. Assim podemos calcular N e PHI(N). O valor de E é obtido logo em seguida em um laço de repetição que ja verifica a validade de E.

Agora calculamos a equação diofantina para encontrar os valores de K e D, tal que 1 = K \* PHI(N) + D \* E. Esse trecho foi realizado via função importada da biblioteca SYMPY (o foco não é resolver uma diofantina) e assim, após filtras o resultado da função, temos os valores para K e D.

A seguir, é necessário montar uma variável do tipo dicionário e carregá-la com o valores das letras e a quais números elas se referem. Após carregar todas de A a Z, define-se que ESPAÇO(“ “) tem valor 99.

O próximo passo é ler toda frase a ser codificada e converter todos seus caracteres ao devidos números. Assim temos toda a string em números.

Agora basta percorrer essa string numérica e analisar em grupos de 3 ou 2 números, também precisamos criar uma variável do tipo lista (blocks). O critério para tal se dá comparando se os 3 números, lidos como um único valor, são estritamente menores que N, se sim, carregamos o índice atual de blocks com esse números. Caso seja maior ou igual a N, devemos carregar blocks com o número referente aos 2 números, lidos como um único número. Repetimos esse processo até que toda a string seja percorrida. Devemos ter atenção para que nenhum bloco formado comece com 0 (zero).

Por fim, temos nossa chave de decodificação e a chave de codificação prontas, a frase passada e o conjunto de blocos resultante.

Com um simple laço de repetição (for), percorremos todos os elementos de blocks, usando a chave de codificação (E, N) e realizando as operações de potência e módulo para que ao fim dessa etapa a variável de lista (coded) esteja complemente carregada com a frase codificada.

Por fim, realizamos uma abordagem muito semelhante, com outro laço for, utilizando agora a chave de decodificação (D, N) e ao final desse laço temos a frase decodificada novamente.

Todas as etapas foram percorridas, isto é, após a passagem dos dois primos e a frase todas as outras variáveis têm seus determinados procedimentos para atribuição de valor e a frase pode ser corretamente codificada ou decodificada.