Ciência da Computação

Marco Túlio Alves de Barros RA: 202100560105

Vinícius Ferreira Schiavon RA: 202100560434

Criptografia RSA

Relatório do trabalho de Matemática Discreta e Finita II

Londrina

05/2022

Descrição do método: dada uma frase, desejamos criptografá-lá usando um método de chave pública, neste caso, o RSA.

É necessário transformar tal frase em um número, para isso usa-se o padrão determinado a seguir: 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 escolhidos dois números primos muito grandes p e q (essa informação é mantida em segredo). Tendo esses, podemos calcular n = p * q (sendo n a chave privada) e phi(n) = (p – 1) * (q – 1).

Também deve ser determinado um e, tal que e < phi(n) e mdc(e, phi(n)) = 1, e também um d, sendo d < phi(n) tal que 1 = k * phi(n) + d * e (equação diofantina).

Ao final dos cálculos teremos: p, q, n, phi(n), e, k, d.

Então temos que a chave de codificação será (e, n) e a chave de decodificação será (d, n).

Explicação do código: inciamos o código importando as funções que serão usadas ao longo do programa, nesse caso não usamos nenhum import relacionado à bibliotecas de RSA, usamos dois imports para resolução da equação diofantina, um para ter acesso às letras minúsculas da tabela ASCII e outro para usar a função gcd (mdc), respectivamente.

Agora é necessário calcular os valores que serão usados na codificação. Iniciamos definindo p e q. Logo após essa definição, já é possível obter os valores n, phi(n) e e. Por último, a equação diofantina phi(n) = (p – 1) * (q – 1) é resolvida, o resultado é filtrado e obtêm-se os valores de k e d.

Terminando os cálculos matemáticos é possível partir para leitura da frase, que nesse caso será feita através de um arquivo de texto definido como user-input.txt.

Para transformar a frase em um número é necessário ter o padrão A = 10, B = 11, etc., e para isso foi feito um dicionário obedecendo tais regras. Com o dicionário já pronto é possível pré-codificar a frase. Após a frase ter sido transformada em um número, é possível começar a divisão em blocos, que nesse caso obedece à regra, sendo: primeiro tenta-se agrupar um bloco de 3 dígitos menor que n, caso não seja possível, o mesmo teste é realizado para um bloco de 2 dígitos e por fim para um bloco de 1 dígito; todos os casos respeitam a regra do próximo dígito após o bloco atual não ser 0. Os blocos vão sendo armazenados em uma lista.

Por fim, podemos, usando as chaves de codificação (e, n) e decodificação (d, n), codificar e decodificar cada bloco.

Para o processo de codificação é criada uma lista que receberá na respectiva posição de cada bloco o valor resultante da operação mod(bloco^e, n), sendo “bloco” o valor numérico de cada bloco após a divisão.

Para o processo de decodificação é criada uma lista que receberá na respectiva posição de cada bloco o valor resultante da operação mod(bloco^D, n), sendo D = phi(n) + d e “bloco” o valor numérico de cada bloco encriptado.

Aplicação do exemplo: como exemplo usaremos o que foi passado no slide, sendo p = 11 e q = 13, por consequência: n = 143, phi(n) = 120, e = 7, k = 1, d = -17. A frase escolhida foi “paraty e linda”, que transformada em números fica:

25102710293499149921182313 10

dividindo em blocos:

25, 102, 7, 102, 93, 49, 91, 49, 92, 118, 23, 13, 10

codificando:

64, 119, 6, 119, 102, 36, 130, 36, 27, 79, 23, 117, 10

decodificando:

25, 102, 7, 102, 93, 49, 91, 49, 92, 118, 23, 13, 10