Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 11, n. 1, 2025.

Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Teoria de Código: O Algoritmo RSA

Lucas O. A. Mauricio[1]
UNICAMP, Campinas, SP

A criptografia RSA utiliza as propriedades singulares dos números inteiros para assegurar a transmissão segura de dados, aproveitando a complexidade associada ao desafio conhecido como problema de fatoração.

No sistema assimétrico onde o sistema RSA é implementado, cada usuário possui um par de chaves: uma chave pública (n, e) para quem deseja enviar uma mensagem e uma chave privada (n, d) que é mantida em posse exclusiva do destinatário da mensagem.

Os termos e e d no contexto do algoritmo, assim como o termo n, são expressões que envolvem números inteiros. Essas expressões não apenas representam relações entre os números, mas também influenciam diretamente o funcionamento do algoritmo. Vamos explorar essas relações e entender como elas são fundamentais para o desempenho do algoritmo.

O primeiro passo para utilizar o método RSA envolve a conversão da mensagem em uma sequência de números. Suponhamos que a mensagem original seja um texto que contenha apenas palavras, sem números. Logo, utilizando a Tabela 1, que pode ser encontrada em [1], obtemos o número correspondente de cada letra:

Tabela 1: Tabela de Conversão.

A	B	C	D	E	F	G	H	I	J	K	L	M
10	11	12	13	14	15	16	17	18	19	20	21	22
N	O	P	Q	R	S	T	U	V	X	W	Y	Z
23	24	25	26	27	28	29	30	31	32	33	34	35

Os espaços entre as palavras serão denotado pelo número 99 e o resultado dessa conversão será chamada de *pré-codificação*. Também é necessário definir os parâmetros do sistema RSA que serão utilizados, esses parâmetros consistem em dois números primos distintos denotados por p e q [3], nos quais $p \cdot q = n$. Depois, a próxima etapa envolve a divisão em blocos do número resultante, onde cada bloco é menor que o número n.

Para codificarmos a mensagem precisamos de n e de um número $e \in \mathbb{N}$ tal que $mdc(e, \varphi(n)) = 1$ e, portanto, inversível módulo $\varphi(n)$ [2]. Procederemos codificando cada bloco separadamente, onde a mensagem codificada será então formada pela sequência dos blocos previamente codificados. Seja b um inteiro positivo menor que n tal que b é um bloco pré codificado, vamos denotar o bloco codificado por $C(b)$, na qual $C(b)$ é o resto da divisão de b^e por n, isto é, $b^e \equiv C(b)\ (mod\ n)$.

Agora, para o processo de decodificação dos blocos da mensagem codificada, será necessária duas informações essenciais: o valor de n e, o inverso de e módulo $\varphi(n)$, que iremos representar como d, em que $d \in \mathbb{N}$. Suponha que $a = C(b)$ seja os blocos da mensagem codificada; então, a^d será o resultado obtido após o processo de decodificação, isto é, $a^d \equiv D(C(b))\ (mod\ n)$, na qual $D(C(b))$ será o resto da divisão de a^d por n. Para encontrar d, basta aplicar o Algoritmo de Euclides Estendido entre $\varphi(n)$ e e [2].

[1] l188523@dac.unicamp.br

Exemplificando, considere a seguinte frase: Congresso Nacional de Matemática Aplicada e Computacional. Logo, utilizando a Tabela 1 (em que teremos que ocultar algumas situações), temos:

$$1224231627142828249923101218242310219913149922102914221029181210991025211812101310991491912242225302910121824231021$$

Tomando $p = 104729$ e $q = 100669$, temos que $n = 10542963701$. Agora dividindo o número resultante da conversão em blocos onde cada um dos blocos seja menor do que n e que nenhum bloco comece por zero, obtemos:

$$1224231627 - 1428282499 - 2310121824 - 2310219913 - 1499221029 - 1422102918$$
$$1210991025 - 2118121013 - 1099149912 - 2422253029 - 1012182423 - 1021$$

Calculando, temos que $\varphi(n) = 10542758304$. Tomando $e = 5 \in \mathbb{N}$ o menor inteiro positivo tal que $mdc(e, \varphi(n)) = 1$ e fazendo os devidos cálculos, temos os seguintes blocos codificados:

$$2911738655 - 3993497202 - 9506984187 - 6483552492 - 6331154245 - 7316306265$$
$$5572506387 - 3901110192 - 5588335852 - 210856203 - 3135340864 - 4258450665$$

Para decodificação, aplicando o Algoritmo de Euclides Estendido entre $\varphi(n)$ e e, temos que $d = 2108551661$. Assim, ao decodificar cada um dos blocos codificados, obtemos:

$$1224231627 - 1428282499 - 2310121824 - 2310219913 - 1499221029 - 1422102918$$
$$1210991025 - 2118121013 - 1099149912 - 2422253029 - 1012182423 - 1021$$

que correspondem aos blocos originais. Daí, seguindo a Tabela 1, temos: CONGRESSO NACIONAL DE MATEMATICA APLICADA E COMPUTACIONAL (Feito no ambiente Jupyter).

Portanto, podemos concluir que com apenas alguns parâmetros, conseguimos colocar em prática o algoritmo RSA. Devido à sua implementação fácil, ele é muito utilizado e apresenta segurança considerável, especialmente devido ao desafio de fatoração, o que torna difícil a quebra de seu sigilo.

Referências

[1] S. C. Coutinho. **Números Inteiros e Criptografia RSA**. 3a. ed. Rio de Janeiro: IMPA, 2023. ISBN: 9788524405273.

[2] A. Hefez. **Aritmética**. 3a. ed. Rio de Janeiro: SBM, 2022. ISBN: 9788583371816.

[3] J. P. O. Santos. **Introdução à Teoria dos Números**. 3a. ed. Rio de Janeiro: IMPA, 2010. ISBN: 9788524404962.