Documentação algoritmo de criptografia RSA em Python

Naelson Douglas

Abril de 2019

1 O algoritmo

A implementação do algoritmo RSA neste código compreende os seguintes passos:

1. São gerados dois números aleatórios p e q
2. É gerado um número n a partir da multiplicação de p por q
   - n = p x q.
3. É calculada a função totiente de n, que por sua vez é um número gerado em função de p e q
   $$\phi(n) = (p-1)(q-1)$$
4. Então o algoritmo escolhe um número ex coprimo deste número tal que:
   - $$1 < ex < \phi(n)$$
5. Então o algoritmo encontra um número d, tal que:
   - $$d * ex \equiv 1 \, mod \, \phi(n)$$

   Agora tendo em mãos todas as variáveis, o algoritmo pode prosseguir e gerar as chaves. As quais são geradas da seguinte maneira:
6. A chave pública é a tupla (n, ex)
7. A chave privada é a tupla (n, d)

Além da geração de chaves públicas e privadas, dois outros passos cruciais no algoritmo são o de criptografar e descriptografar mensagens. Ambas funções ocorrem ao se ler a mensagem caractere por caractere, convertendo eles para seus respectivos números na tabela UTF-8 e aplicando a devida regra matemática neste número a fim de transforma-lo em um outro número que representa o caractere criptografado (quando fazendo a criptografia da mensagem) ou retorna ao caractere original (quando descriptografando). Tendo a chave pública em mãos (n, ex), a criptografia de uma mensagem M é feita ao aplicar a cada um de seus caracteres C a seguinte fórmula, a qual os converte no caractere criptografado X e ao fim convertendo a mensagem M para a sua versão criptografada N.

$$x = C^e x \; mod \, n$$

O caminho inverso, para trazer X de volta a C, e consequentemente descriptografando a mensagem N de volta para M, é feito usando a chave privada (n, d)

$$C = X^d mod \, n$$

2 Uso

Para usar a implementação do algoritmo, basta utilizar as funções

- cryptography.setup((n, d), (n, ex)) ou setup() Onde na primeira assinatura ela simplesmente retornará as chaves já fornecidas pelo usuário, ou na segunda irá gerar aleatoriamente um par de chaves a ser usado ao assinar e desassinar as mensagens
- cryptography.encrypt((n, ex), m) Função que criptografa a string m utilizando a chave pública (n, ex) e retorna uma string x criptografada
- cryptography.decrypt((n, d), x) Recebe a chave privada (n, d) junto à string criptografada x e retorna a string descriptografada m.