RSA – Criptografia Assimétrica

Prof. Ms. Fábio Henrique Cabrini

CyberSecurity (FIAP) e Estudos Avançados em Segurança (FATEC)

Figura 1. Charge

O RSA é um algoritmo de criptografia de dados, que deve o seu nome a três professores do Instituto de Tecnologia de Massachusetts (MIT), Ronald **R**ivest, Adi **S**hamir e Leonard **A**dleman, fundadores da atual empresa RSA Data Security, Inc., que inventaram este algoritmo — até a data (2008) a mais bem-sucedida implementação de sistemas de chaves assimétricas, e fundamenta-se em teorias clássicas dos números. É considerado um dos mais seguros, já que mandou por terra todas as tentativas de quebrá-lo. Foi também o primeiro algoritmo a possibilitar criptografia e assinatura digital e uma das grandes inovações em criptografia de chave pública. *(Wikipedia.org)*

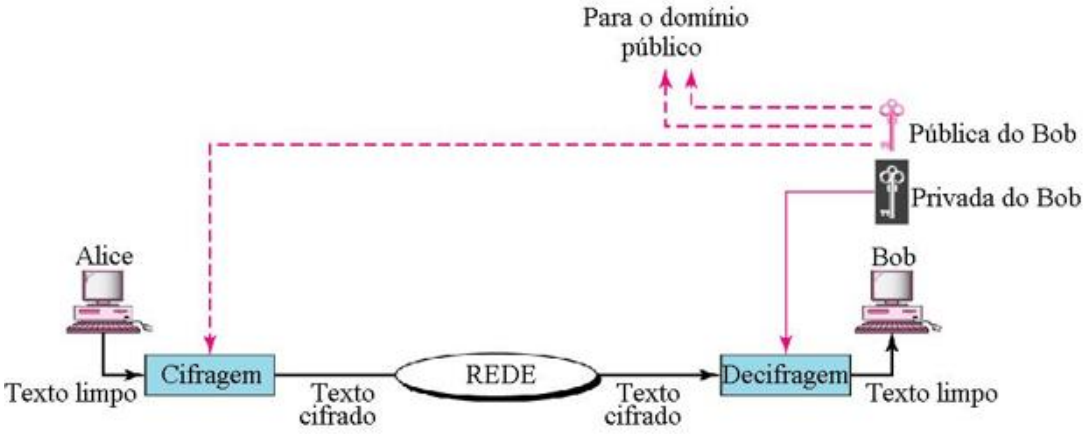

Figura 2. Uso convencional do RSA

Exemplo de código em Java:

```
String msg = "Paz e felicidade a todos";
String msgcifrada = null;
String msgdecifrada = null;
BigInteger n, d, e;
int bitlen = 2048;

//Escolha de forma aleatória dois números primos grandes p e q
SecureRandom r = new SecureRandom();
BigInteger p = new BigInteger(bitlen / 2, 100, r);
BigInteger q = new BigInteger(bitlen / 2, 100, r);

//Compute n = p * q
n = p.multiply(q);

//Compute a função totiente phi(n) = (p -1) (q -1)
BigInteger m = (p.subtract(BigInteger.ONE))
               .multiply(q.subtract(BigInteger.ONE));

//Escolha um inteiro  "e"  , 1 < "e" < phi(n) ,  "e" e phi(n) sejam primos entre si.
e = new BigInteger("3");
while(m.gcd(e).intValue() > 1) e = e.add(new BigInteger("2"));

// d seja inverso multiplicativo de "e"
d = e.modInverse(m);

System.out.println("p:"+p);
System.out.println("q:"+q);
System.out.println("n:"+n);
System.out.println("e:"+e);
System.out.println("d:"+d);

//mensagem cifrada - RSA_encrypt()
msgcifrada = new BigInteger(msg.getBytes()).modPow(e, n).toString();
System.out.println("msg cifrada: "+ msgcifrada);

//mensagem decifrada - RSA_decrypt()
msgdecifrada = new String(new BigInteger(msgcifrada).modPow(d, n).toByteArray());
System.out.println("msg decifrada: " +msgdecifrada);
```

Figura 3. Exemplo do código RSA

Procedimento para geração das chaves

Etapa 1 - Escolher p e q (números primos) para o cálculo de N = p.q

Etapa 2 - Calcular a função totiente $\phi(N) = (p-1).(q-1)$

Etapa 3 - Escolha $1 < e < \phi(N)$, tal que *e* e $\phi(N)$ sejam primos entre si

Etapa 4 - Escolha d tal que $e.d \bmod \phi(N) = 1$

Chaves Assimétricas

Criptografar - Chave Pública (*e*,N) => $C = P^e \bmod N$

Decriptografar - Chave Privada (d,N) => $P = C^d \bmod N$

Exemplo de geração das chaves

Para p=3 e q =5 calcule as chaves pública e privada de acordo com o algoritmo de chave pública RSA.

Etapa 1: N=p.q -> N=3.5 -> N=15

Etapa 2: $\phi(N)=(p-1).(q-1)$ -> $\phi(N)=(3-1).(5-1)$ -> $\phi(N)=8$

Etapa 3: $1 < e < \phi(N)$

Verificar se são primos entre si, quando dois números apresentam um único divisor comum entre eles!

$\phi(N)=8$: 1 ,2, 4, 8

$e=7$: 1, 7

Etapa 4: Para d=15 => e.d mod $\phi(N)=1$ => 7.d mod 8 = 1 -> 1=1 (verdade)

Descrição: No alfabeto de A – Z a letra "C" equivale a 3ª letra, podemos atribuir o valor 3 em decimal. Deste modo, podemos criptografar a letra C usando o algoritmo RSA.

Criptografar: Chave Pública (e,N) => (7,15) => $C = P^e \bmod N$ => $C = 3^7 \bmod 15 = 12$ => "L"

Decriptografar: Chave Privada (d,N) => (15,15) => $P = C^d \bmod N$ => $P = 12^{15} \bmod 15 = 3$ => "C"

Atividade

Construa o algoritmo com a linguagem de programação de sua preferência, ele deve escolher dois valores aleatórios para p e q, gerar as chaves pública e privada de acordo com os quatro passos apresentados. O algoritmo deve ser capaz de criptografar e decriptografar a frase "*The information security is of significant importance to ensure the privacy of communications*".