Fábrica de Noobs

Criptografia – Criptografia RSA

A Criptografia RSA é um dos algoritmos de encriptação mais seguros e populares existentes na atualidade. Seu funcionamento baseia-se em princípios matemáticos, e se trabalhado com chaves grandes o suficiente, é sem dúvida um excelente algoritmo para se implementar.

Aqui vou tentar explicar o processo de encriptação e encriptação manual, o que vai envolver alguns conceitos matemáticos, que considero melhor explicar antes.

Número primo é qualquer número que seja divisível somente por 1 e por ele mesmo. Por exemplo, 17,11,37,41.

Podemos indicar restos de divisões utilizando **congruências**. Por exemplo:

$$8 \equiv 3 \mod(5)$$

(lê-se 8 congruente a 3 módulo 5)

A notação acima mostra que 8 apresenta o mesmo resto que 3 ao ser dividido por 5. Veja outros exemplos:

$$101 \equiv 1 \mod(100)$$

$$478 \equiv 3 \mod(5)$$

$$256 \equiv 4 \equiv 1 \mod(3)$$

$$2 \equiv 0 \mod(2)$$

$$2^3 \equiv 0 \mod(2)$$

O ramo da Matemática que trabalha com tais propriedades chama-se Aritmética Modular, e é onde toda a criptografia RSA está baseada.

Para codificar uma mensagem, o primeiro passo é transformar todo o texto em números. Para isso, usamos a conhecida tabela ASCII:

Tabela ASCII (códigos de caracteres 0 - 127)

000		016	►	032		048	0	064	@	080	P	096	`	112	p
001	☺	017	◄	033	!	049	1	065	A	081	Q	097	a	113	q
002	☻	018	↕	034	"	050	2	066	B	082	R	098	b	114	r
003	♥	019	‼	035	#	051	3	067	C	083	S	099	c	115	s
004	♦	020	¶	036	$	052	4	068	D	084	T	100	d	116	t
005	♣	021	§	037	%	053	5	069	E	085	U	101	e	117	u
006	♠	022	■	038	&	054	6	070	F	086	V	102	f	118	v
007		023	↨	039	'	055	7	071	G	087	W	103	g	119	w
008		024	↑	040	(	056	8	072	H	088	X	104	h	120	x
009		025	↓	041	)	057	9	073	I	089	Y	105	i	121	y
010		026	→	042	*	058	:	074	J	090	Z	106	j	122	z
011	♂	027	←	043	+	059	;	075	K	091	[	107	k	123	{
012	♀	028	∟	044	,	060	<	076	L	092	\	108	l	124	|
013		029	↔	045	-	061	=	077	M	093	]	109	m	125	}
014	♫	030	▲	046	.	062	>	078	N	094	^	110	n	126	~
015	☼	031	▼	047	/	063	?	079	O	095	_	111	o	127	⌂

Por exemplo, codificaremos “morte ao miojo” da seguinte forma:

109 111 114 116 101 032 097 111 032 109 105 111 106 111

Aqui, terminamos o processo de pré-codificação e passamos para a codificação propriamente dita.

Importante: é conveniente, para evitar uma possível (apesar de improvável) quebra por frequência linguística, que o texto a ser codificado esteja em alguma cifra que não haja repetição de letras. Por exemplo, cifra de Vigenère.

Agora precisamos encontrar dois parâmetros RSA. Vamos chamá-los de p e q. Esses números precisam ser necessariamente primos. Em situações reais, esses números são astronômicos (algo no mínimo da ordem de 10^{100}), o que impossibilita qualquer tentativa de decodificação sem a chave. Aqui tem alguns http://compoasso.free.fr/primelistweb/page/prime/liste_online_en.php. Mas no nosso caso, usaremos esses:

$$p = 17$$
$$q = 23$$

Depois, determinamos o produto desses dois números, e chamamos tal produto de n.

$$n = pq$$

$$n = 17 * 23$$
$$n = 391$$

Vamos chamar n de chave pública do código. Ela pode ser divulgada para qualquer pessoa que queira criptografar e lhe enviar uma mensagem.

Já p e q são chaves privadas, e não devem ser reveladas. Se n for um número pequeno, é fácil deduzir os outros valores, mas isso é impossível para valores grandes.

Agora, vamos determinar a chamada função totiente em n, que não é nada mais que:

$$\varphi(n) = (p - 1) * (q - 1)$$

No caso:

$$\varphi(n) = (16) * (22)$$
$$\varphi(n) = 352$$

Daí, basta escolhermos um número que satisfaça $1 > e > \varphi(n)$ e $d(e) \notin d(\varphi n)$ ou seja, que esteja entre 1 e o valor que encontramos anteriormente e não possua divisores comuns (além do 1, claro) com ele. Para facilitar a codificação, vou escolher 3.

$$e = 3$$

Depois, precisamos encontrar um número d que satisfaça:

$$d * e \ \equiv 1 \ \mathrm{mod}(\varphi n)$$

No nosso exemplo:

$$d * 3 \ \equiv 1 \ \mathrm{mod}(352)$$

Em aritmética modular, d é chamado de inverso multiplicativo de e: $\varphi(n)$. Felizmente, temos calculadoras online que já fazem esse serviço.

Modular inversion

Use the [extended Euclidean algorithm] to compute a [modular multiplicative inverse]

Computes m for $n^{-1} = m \pmod{p}$, where n and p are coprime.

Displays the steps of the extended Euclidean algorithm.

Set **n** = 3

Set **p** = 352

Submit

---

step	quotient	t_1	t_2	t_3	u_1	u_2	u_3	v_1	v_2	v_3
0					1	0	352	0	1	3
1	117	1	-117	1	0	1	3	1	-117	1
2	3	-3	352	0	1	-117	1	-3	352	0

The modular inverse is:

235

Basta acessarmos http://www.cs.princeton.edu/~dsri/modular-inversion-answer.php, informarmos o valor de e na primeira caixa e de $\varphi(n)$ na segunda:

Set **n** = 3

Set **p** = 352

Como nos mostrou, o resultado é:

$$d = 235$$

Ao final do processo, terminamos com os seguintes valores:

Chaves públicas: $(n; e)$

Chaves privadas: $(p; q; d)$

Ou, no nosso caso:

Chaves públicas: $(391; 3)$

Chaves privadas: $(17{,}23{,}235)$

É só depois de ter esses 5 valores que realizamos a codificação propriamente dita.

Voltamos a mensagem pré-codificada originalmente:

109 111 114 116 101 032 097 111 032 109 105 111 106 111

Vamos denotar cada conjunto de números por bloco, e chama-los, um por um, de M. A formula para codificar cada bloco M é:

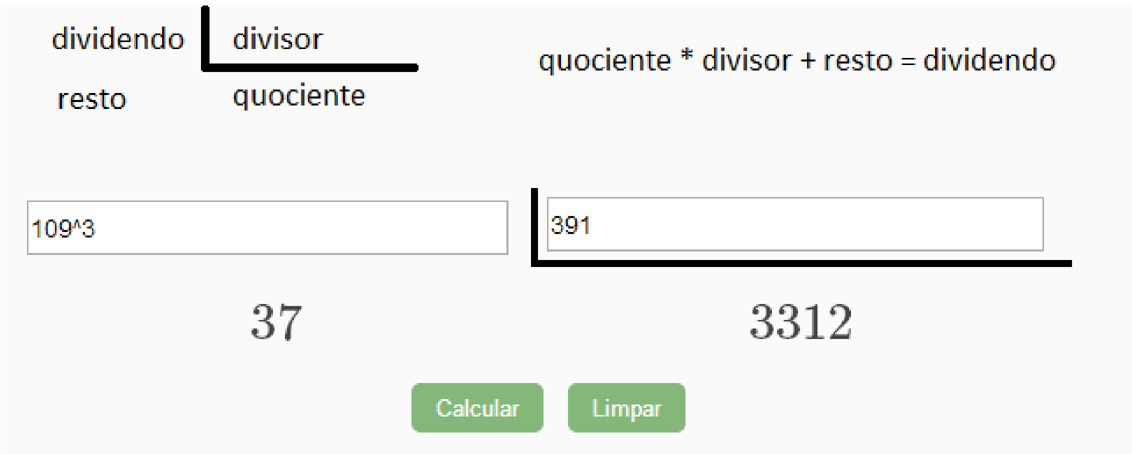
$$C = M^e \bmod (n)$$

Sendo C o bloco codificado que obteremos após aplicar a fórmula.

Veja o exemplo:

$$C = 109^3 \bmod (391)$$

Também felizmente, podemos fazer isso com uma calculadora de restos (http://www.calculadoraonline.com.br/divisao-polinomios) , tornando o trabalho infinitamente mais fácil:



Logo:

$$109^3 \bmod (391) = 37$$

E daí temos nosso primeiro bloco codificado. Devemos repetir o processo para todos os outros blocos. Realizando-o, temos a seguinte mensagem codificada já em RSA:

37 304 45 24 16 315 79 304 315 37 265 304 30 304

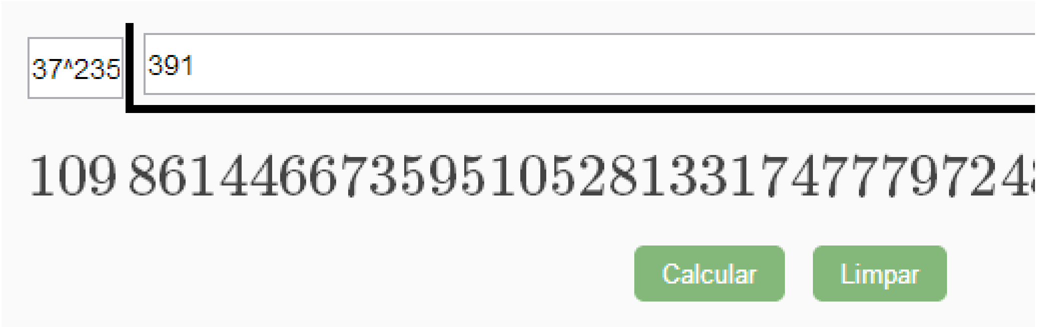
Se quisermos decodificar, podemos fazê-lo utilizando outra fórmula:

$$M = C^d \bmod (n)$$

No caso do primeiro bloco:

$$M = 37^{235} \bmod (391)$$

O que também pode ser feito usando a calculadora de restos, e dá:



Logo:

$$109 = 37^{235} \bmod (391)$$

Que é o valor do primeiro bloco. Realizando o processo completo, temos novamente:

109 111 114 116 101 32 97 111 32 109 105 111 106 111

Comparando com a tabela ASCII, chegamos no resultado original e deciframos a mensagem.

Se quisermos automatizar o processo podemos usar esse site, https://www.cs.drexel.edu/~introcs/Fa11/notes/10.1_Cryptography/RSA_Express_EncryptDecrypt.html que oferece um codificador e decodificador automático de RSA com base na tabela ASCII.

Basta informarmos os valores de n e e para codificar, e de d para decifrar: