Jornal Eletrônico de Ensino e Pesquisa de Matemática
(1s.) **v. 2** 2 (2018): 16–24. ©JEEPEMA–ISSN-2594-6323.
www.dma.uem.br/kit/jeepema

Algumas aplicações da aritmética à criptografia

D. Andrade (FEITEP)–Email:doherty200@hotmail.com

RESUMO: Neste trabalho apresentamos uma aplicação da teoria dos números à criptografia. Como veremos os números primos e a relação de congruência módulo inteiro desempenham um papel importante nesta área. Apresentaremos também alguns exemplos.

Sumário

1. Introdução

Atualmente os computadores com algoritmos sofisticados e processadores cada vez mais velozes criam rapidamente tabelas de números primos. Mas em passado recente tabelas eram elaboradas manualmente usando, por exemplo o crivo de Eratóstenes, como foi o caso da tabela publicada em 1914 por Derick Norman Lehemer que continha os números primos menores do que 10 milhões. Veja em [4].

Mesmo com esta facilidade computacional os números primos muito grandes resistem e a busca por eles e por um padrão em que aparecem (se existe) desafia os matemáticos. A utilização dos computadores nos trouxe a possibilidade de procurar números primos cada vez maiores, mas que utilidade tem o conhecimento de números primos cada vez maiores? A resposta é que

o conhecimento dos números primos muito grandes permite criptografar documentos e senhas. E isso não é pouco. Garantir a segurança e a integridade dos dados armazenados em um computador tornou-se atualmente vital para pessoas e companhias. Segurança e integridade de dados são coisas diferentes: segurança significa proteger os dados contra usuários não autorizadas e integridade significa proteger os dados contra usuários autorizados.

Um dos algoritmos atuais mais seguros de encriptação de informações, o algoritmo **RSA**, originou-se dos estudos de Ronald **R**ivest, Adi **S**hamir e Leonard **A**dleman, matemáticos que mudaram a história da Criptografia.

O princípio do algoritmo é construir, utilizando números primos, duas chaves: uma chave chamada de chave pública e outra chamada de chave privada. Uma chave é uma informação restrita que controla toda a operação dos algoritmos de criptografia. Inicialmente devem ser escolhidos dois números primos quaisquer e quanto maior os números escolhidos mais seguro será o algoritmo.

Antes de iniciar com os exemplos, vamos rever a relação de congruência módulo um inteiro $m > 1$. Seja $m > 1$ um inteiro fixado. Se x e y são inteiros, dizemos que x é congruente a y módulo m se $x - y$ é múltiplo de m. Isto é, se existe um inteiro k tal que $x - y = km$. Ou ainda, que m divide $x - y$. Representamos isso por $x \equiv y \mod m$ e lê-se: x é congruente a y módulo m. Observe que se x é congruente a y módulo m, então, quando divididos ambos por m, deixam o mesmo resto. De fato, suponha que $x = k_1m + r_1$ e $y = k_2m + r_2$, onde $0 \leq r_1, r_2 < m$. Calculando $x - y$ temos que:

$$x - y = (k_1 - k_2)m + (r_1 - r_2).$$

Como m divide $x - y$ e m divide $(k_1 - k_2)m$, então, m tem obrigatoriamente que dividir $(r_1 - r_2)$. Como $0 \leq r_1, r_2 < m$, então, $r_1 - r_2 = 0$ e, portanto, $r_1 = r_2$. Essa relação é uma relação de equivalência.

Usamos a notação $x \equiv y (\mod m)$ para representar que x e y são congruentes módulo m. Veja [1].

2. Algumas aplicações simples

2.1. **Código de César.** A aritmética módulo m é muito utilizada na criptografia. O exemplo mais simples (e muito antigo, remonta a Júlio César imperador romano). Veja [2]. Ele usava um método de escrita de mensagens secretas onde cada letra do alfabeto é associada a um número como na tabela abaixo e

depois transladava três casas mais à frente e tomava o módulo $m = 26$ (letras do alfabeto).

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Assim, no método de César define-se uma função $f(n) = (n + 3) \mod 26$ para encriptar uma mensagem, chamada de função encriptadora. Para desencriptar e recuperar a mensagem original utilizava a sua inversa $f^{-1}(n) = (n - 3) \mod 26$, chamada de função desencriptadora.

O método como apresentado acima é pouco seguro, já por volta de 1580 um sistema semelhante foi utilizado pela rainha Maria da Escócia, para conspirar junto com os espanhóis contra sua prima Isabel I. As mensagens de Maria foram facilmente decifradas culminando com a sua decapitação. Desde essa época os sistemas de codificação melhoraram muito. Veja [5].

Como vimos o método de César é pouco seguro e podemos melhorar definindo funções f mais gerais, como por exemplo, $f(n) = (an + b) \mod m$ com inteiros a e b de modo que f tenha inversa. Como por exemplo, a função $f(n) = (7n + 3) \mod 26$ como função encriptadora e, como função desencriptadora, $g(n) = (15n + 7) \mod 26$. Podemos criar um procedimento computacional para verificar o funcionamento desses funções. Veja abaixo um procedimento em Maple.

```
for k from 0 to 26 do
m[k] :=  mod(7*k+3, 26);
n[k] :=  mod (15*m[k]+7, 26);
od;
```

É bom registrar aqui que esta simplificação não leva em conta letras maiúsculas e minúsculas, acentos, espaços e outros caracteres. Há um método de conversão de caracteres ASCII, usado pela maioria dos computadores, está implementado no Maple^{R}. Vejamos um exemplo:

```
convert("Bom dia pessoal", bytes);
```

cujo resultado é

[66, 111, 109, 32, 100, 105, 97, 32, 112, 101, 115, 115, 111, 97, 108]

Para recuperar a expressão original:

```
convert([66, 111, 109, 32, 100, 105, 97, 32,
112, 101, 115, 115, 111, 97, 108], bytes);
```

2.2. **Geração de números aleatórios.** Muitas vezes a codificação de uma mensagem pode ser realizada por meio da geração aleatória de números. A geração de números aleatórios no computador não é de fato aleatório, por isso mesmo, são chamados muitas vezes de números pseudo-aleatórios. A variedade de métodos é grande, por exemplo, um dos métodos utiliza os movimentos do seu mouse armazenados nas últimas horas para gerar esses números. Mas o método mais comumente utilizado é o método das congruências lineares, descrito abaixo, que consiste em escolher 4 números inteiros positivos:

(i) um número m para a aritmética módulo m,

(ii) o multiplicador a,

(iii) o incremento c,

(iv) e a raiz x_0 satisfazendo $2 \leq a < m, 0 \leq c < m$ e $0 \leq x_0 < m$.

Em seguida, iniciando com x_0, gera-se uma sequência de números pseudo-aleatórios x_n com $0 \leq x_n < m$ para qualquer n, por meio da fórmula:

$$x_{n+1} = (ax_n + c) \pmod{m}, n \geq 0. \tag{1}$$

Como exemplo, escolhendo $m = 9, a = 7, c = 5, x_0 = 3$, obtemos de acordo com a equação 1 que:

$$\begin{aligned}
x_1 &= (7x_0 + 5) \mod 9 = 26 \pmod 9 = 8 \\
x_2 &= (7x_1 + 5) \mod 9 = 61 \pmod 9 = 7 \\
x_3 &= (7x_2 + 5) \mod 9 = 54 \pmod 9 = 0 \\
x_4 &= (7 \times 0 + 5) \mod 9 = 5 \pmod 9 = 5 \\
x_5 &= (7x_4 + 5) \mod 9 = 40 \pmod 9 = 4
\end{aligned}$$

Os valores obtidos acima e outros mais estão organizados na seguinte tabela:

n	0	1	2	3	4	5	6	8	8	9
x_n	3	8	7	0	5	4	6	2	1	3

Como $x_9 = x_0$, a sequência tem apenas 9 elementos diferentes, antes de começar a se repetir.

Novamente, podemos escrever facilmente um procedimento computacional para gerar números aleatórios como descrito acima. Apresentamos a seguir um procedimento em Maple que gera os números pseudo-aleatórios do exemplo acima.

```
m := 9; a := 7; c := 5; x[0] := 3;
for n from 0 to (m-1) do x[n+1] := modp(a*x[n]+c, m);od;
```

A maioria dos computadores utiliza este procedimento para gerar números aleatórios com o número de Mersenne $m = 2^{31} - 1$, com incremento $c = 0$ e multiplicador $a = 7^5$, o que permite gerar $2^{31} - 2$ números diferentes antes de iniciar a repetição. Veja [2].

3. Aplicação ao sistema de criptografia RSA

O sistema de criptografia **RSA**, iniciais de seus desenvolvedores **R**ivest, **S**hamir e **A**dleman surgiu em 1976. Atualmente o governo dos Estados Unidos da América detem a sua propriedade. Esse sistema é na verdade uma família de sistemas criptográficos diferentes se escolhermos os parâmetros diferentes, essa família é definida da seguinte forma:

(i) o espaço da mensagens: $\mathbb{Z}_n$, onde $n = p \times q$, com p, q números primos quaisquer.

(ii) $u(x) = x^a \pmod{n}$, para qualquer $x \in \mathbb{Z}_n$.

(iii) $v(y) = y^b \pmod{n}$, para qualquer $y \in \mathbb{Z}_n$,

onde a e b são tais que $a \times b \pmod{((p-1) \times (q-1))} = 1$.

As chaves u (pública) e a chave v (privada) devem satisfazer a duas propriedades para que o sistema seja seguro:

(i) $v(u(x)) = x$, para qualquer mensagem x.

(ii) Não deve ser possível obter x conhecendo-se $u(x)$ e não conhecendo $v(x)$. Note que resolver esta questão é um problema em aberto, com consequências importantes para a utilização da criptografia.

Consideremos ainda que $\text{MDC}(a, n) = 1$ e seja b a solução da congruência linear $ab \equiv 1 \pmod{n}$.

No sistema RSA, podemos começar por traduzir as mensagens (sequência de letras) em sequências de números inteiros. Por simplicidade e para ilustrar consideremos apenas as letras do alfabeto português:

A	B	C	D	E	F	G	H	I	J	L	M
00	01	02	03	04	05	06	07	08	09	10	11
N	O	P	Q	R	S	T	U	V	X	Z	
12	13	14	15	16	17	18	19	20	21	22	

O inteiro x resultante é então transformado, com a ajuda da chave pública, em um inteiro

$$u(x) \equiv x^a \pmod{n}.$$

Como exemplo tomemos $p = 43, q = 59$ e $a = 13$ e vamos codifique a palavra **GOLD**. Para as contas utilizamos o software Maple com o pacote numtheory.

Neste caso $n = 43 \times 59 = 2537$. Como $a = 13$ é primo, $\text{MDC}(3, 42 \times 58) = 1$. Codificando, obtemos

G	O	L	D
06	13	10	03

Logo, a mensagem corresponde é $x = 06131003$. Aplicando a chave pública u à mensagem x, obteremos uma mensagem codificada:

$$u(x) = (06131003)^{13} \pmod{2537} = 1868,$$

com apenas 4 dígitos. Para manter o mesmo número de dígitos, agrupamos em x os dígitos em blocos de 4 e depois aplicamos u a cada um dos blocos[1].

$$\begin{aligned} 0613 &\longrightarrow 0613^{13} \pmod{2537} = 1767 \\ 1003 &\longrightarrow 1003^{13} \pmod{2537} = 0885. \end{aligned}$$

[1] deve-se ter $x < n$.

A mensagem criptografada é 1767 0885.

Quando esta mensagem for recebida, o receptor a decodifica com a chave privada v que só ele conhece por meio de b:

$$x = v(u(x)) = u(x)^b \pmod{n}.$$

Assim, para determinar a chave privada b temos que resolver a congruência

$$13b \equiv 1 \pmod{2436}.$$

Resolvendo, obtemos $b = 937$.

Segue que $x_1 = 1767^{937} \pmod{2537} = 613 = 0613$ e $x_2 = 0885^{937} \pmod{2537} = 1003$, e, portanto, $x = x_1x_2 = 0613\ 1003$ que é a mensagem original **GOLD**.

Agora vamos ilustrar como resolver a congurência linear $13b \equiv 1 \pmod{2436}$ que utilizamos acima. É um exercício simples de máximo divisor comum entre dois inteiros positivos que recai nas equações Diofantinas.

Note que $13b \equiv 1 \pmod{2436} \Leftrightarrow 13b + 2436y = 1$, que é uma equação Diofantina. Como MDC(2436,13)=1, a congruência tem solução. Vamos determinar uma solução particular utilizando MDC(2436,13). Veja [1].

	187	2	1	1	2
2436	13	5	3	2	1
5	3	2	1	0	

Agora, reescrevendo o MDC, usando que

$$\begin{aligned} 2436 &= 13 \times 187 + 5 \\ 13 &= 5 \times 2 + 3 \\ 5 &= 3 \times 1 + 2 \\ 3 &= 2 \times 1 + 1. \end{aligned}$$

$$\begin{aligned} 1 &= 3 - 2 \times 1 \\ &= 3 - (5 - 3 \times 1) = 3 \times 2 - 5 \\ &= (13 - 5 \times 2) \times 2 - 5 = 13 \times 2 - 5 \times 5 \\ &= 13 \times 2 - (2436 - 13 \times 187) \times 5 = 13 \times 937 - 2436 \times 5. \end{aligned}$$

Segue que $b = 937$.

Resumindo, uma vez escolhidos os números primos P e Q, defina os números:

$$\begin{aligned} N &= P \times Q \\ Z &= (P-1) \times (Q-1). \end{aligned}$$

Agora escolha um número D que seja primo com relação ao número Z. De posse desses números começa o processo das chaves públicas e privadas. Para isto é necessário encontrar um número E que satisfaça à seguinte propriedade:

$$(E \times D) \pmod{Z} = 1.$$

Isto é, ED ao ser dividido por Z deixa resto igual a 1. Com esse processo definem-se as chaves de encriptação e desencriptação.

Para encriptar utilizamos E e N – esse par de números será utilizado como chave pública. Para desencriptar utilizamos D e N – esse par de números é utilizado como chave privada. As formas gerais são:

Texto Criptografado = ((Texto Original)^E) (mod N)
Texto Original = ((Texto Criptografado)^D) (mod N)

Mais um exemplo, tomemos $P = 17$e $Q = 13$ dois números primos. Assim, $N = PQ = 221$ e $Z = (P-1) \times (Q-1) = 192$. Em seguida, escolhemos o número $D = 7$ que é primo com relação a Z e o número $E = 55$ que é congruente a 1 módulo Z. Temos os seguintes resultados:

Encriptando	Desencriptando
Original =3	Criptografado = 198
Criptografado = (3^{55}) (mod 221)	Original = $((198)^7)$ (mod 221)
= (17444921100912017907117050 7) (mod 221)	= (11930436453209472) (mod 221)
Criptografado = 198	Original= 3

Como podemos ver, quanto maior os números escolhidos mais protegidos estarão os dados criptografados. A corrida por número primo muito grande já começou há algum tempo e não deve parar logo. Outros detalhes em [5].

Referências

1. MONTEIRO, L. H. Jacy, **Elementos de Álgebra**. LTC, 1978. 17, 22
2. ROSEN, Kenneth H., **Discrete Mathematics and its applications**. McGrill, 1995. 17, 20
3. ANDRADE, D. **Matemática Discreta: Notas de aula**. 2005.
4. Texto da internet. http://locomat.loria.fr/lehmer1914/lehmer1914doc.pdf. Acessado em julho/2016. 16
5. Texto da internet. http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica _modular/criptografia.html#1. Acessado em julho/2016. 18, 23