INSTITUTO FEDERAL
Espírito Santo

Teoria dos Números

congruência e criptografia

Alberson Miranda
2021-08-23

O que é criptografia?

Do grego *kriptós* (oculto, secreto) e *gráphein* (escrita), criptografia é a ciência das técnicas de codificação e decodificação de informações.

Ao longo da história

O documento encriptado mais antigo que se tem registro data do século I AD, utilizado pelos militares no império romano de Júlio César.

fig. 1: Cifra de César

Ao longo da história

A invenção do computador está associada com criptografia. Durante a II Guerra Mundial, o primeiro computador foi desenvolvido para decriptar as comunicações nazistas codificadas através da *Enigma*.

As mensagens geradas pela *Enigma* eram encriptadas com uma chave que mudava diariamente.

fig. 2: Enigma

ALGORITMOS DE CRIPTOGRAFIA

Criptografia de Chave Simétrica

Nos algoritmos de chave simétrica, há apenas uma chave comum usada para trancar e destrancar a "caixa" de encriptação e tanto o remetente quanto o destinatário têm a mesma chave.

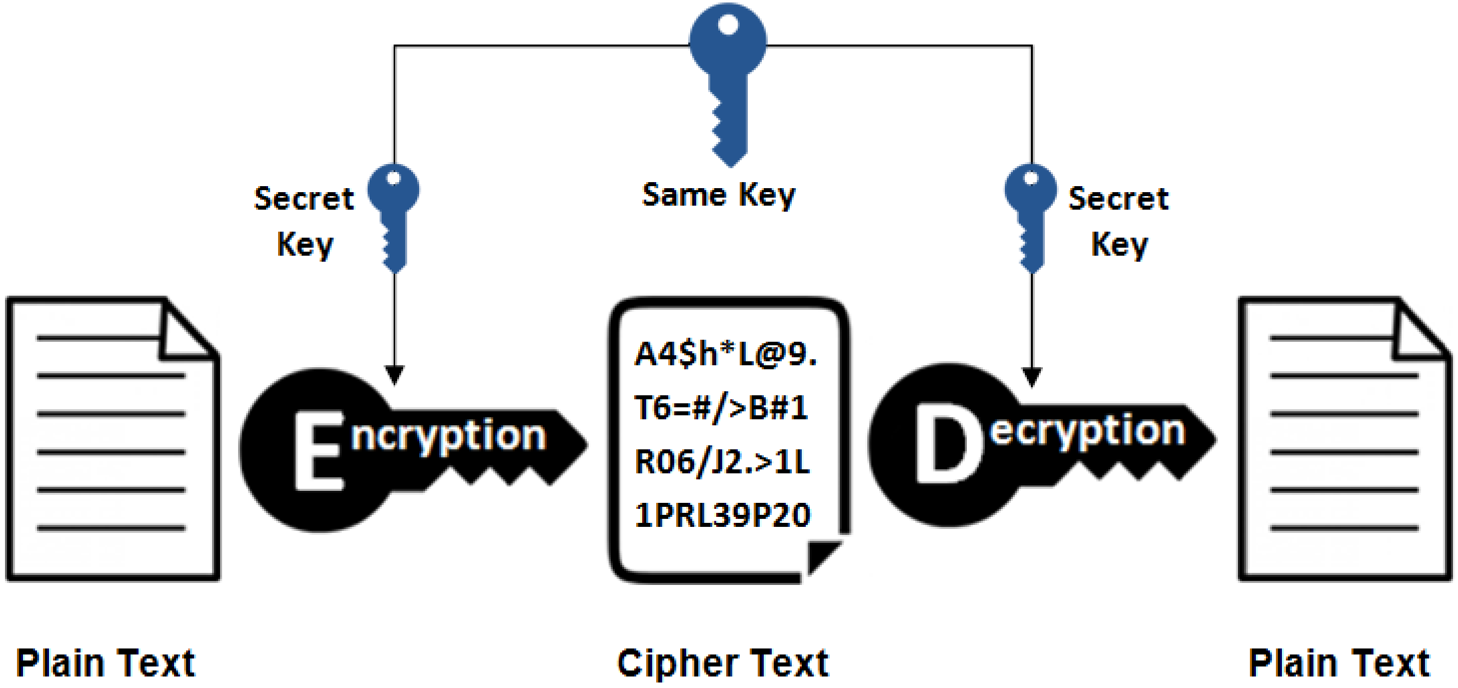

fig. 3: Encriptação simétrica

Onde é usado?

- *Enigma* (II GM)
- armazenamento de dados
- onde velocidade é importante (e.g., transações de cartão de crédito)

Riscos

Como há uma única chave, ela deve ser compartilhada. Se o canal onde a chave é transmitida for comprometido e a chave for interceptada, a terceira parte poderá decifrar o texto.

Criptografia de Chave Assimétrica

Nos algoritmos de chave assimétrica, apenas o receptor detém a chave. O receptor envia publicamente uma fechadura — o método de cifragem para trancar essa caixa hipotética —, para a qual apenas o receptor possui a chave. A fechadura é chamada de **chave pública** e a chave é chamada de **chave privada**, existindo apenas uma chave pública para cada chave privada.

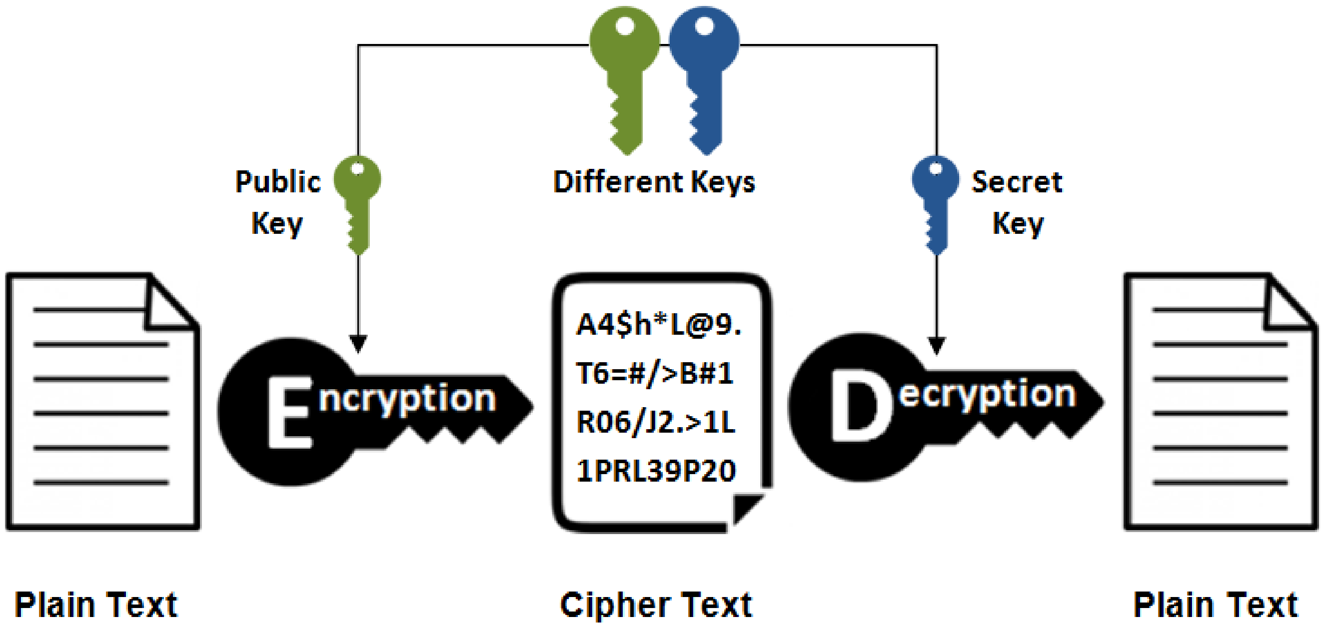

fig. 4: Encriptação assimétrica

Onde é usado?

- verificação de identidade (e.g., assinatura digital, blockchain, apps de chat)
- onde segurança é importante

Riscos

A criptografia assimétrica oferece melhor segurança porque usa duas chaves diferentes - uma chave pública que só é usada para criptografar mensagens, tornando-a segura para qualquer pessoa, e uma chave privada para descriptografar mensagens que nunca precisam ser compartilhadas.

- a chave privada nunca precisa ser compartilhada
- garante que apenas o destinatário pretendido possa descriptografar as mensagens codificadas e criar uma assinatura digital à prova de violação.

CRIPTOGRAFIA E TEORIA DOS NÚMEROS
CIFRA DE CÉSAR

Implementação da Cifra de César

Cada letra é associada a um número:

letras	A	B	C	D	E	F	G	H	I	J	K	L	M
numeros	1	2	3	4	5	6	7	8	9	10	11	12	13

letras	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
numeros	14	15	16	17	18	19	20	21	22	23	24	25	26

A palavra IFES, por exemplo, ficaria:

$$\text{IFES} = 9\ 6\ 5\ 19$$

Implementação da Cifra de César

Escolhemos então um número b qualquer para deslocar as letras. Tomando $b = 3$, por exemplo:

letras	A	B	C	D	E	F	G	H	I	J	K	L	M
numeros	1	2	3	4	5	6	7	8	9	10	11	12	13

letras	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
numeros	14	15	16	17	18	19	20	21	22	23	24	25	26

E a palavra IFES codificada fica:

$$12\ 9\ 8\ 22 = \mathrm{LIHV}$$

Implementação da Cifra de César

Esse é um caso de chave simétrica. b deve ser compartilhado com o receptador de forma que ele pudesse realizar a operação inversa para retornar à mensagem original:

$$\begin{aligned} \text{LIHV} = 12\ 9\ 8\ 22 \\ (-3) \\ \text{IFES} = 9\ 6\ 5\ 19 \end{aligned}$$

Descrevendo matematicamente,

$$C \equiv P + b \pmod{26}$$

Sendo C o código numérico do texto cifrado, P o código numérico do texto original e b a chave.

CRIPTOGRAFIA E TEORIA DOS NÚMEROS
CRIPTOGRAFIA RSA

Implementação de Algoritmo de Chave Assimétrica

Assim como na Cifra de César, começamos por transformar a mensagem em números:

letras	A	B	C	D	E	F	G	H	I	J	K	L	M
numeros	10	11	12	13	14	15	16	17	18	19	20	21	22

letras	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
numeros	23	24	25	26	27	28	29	30	31	32	33	34	35	99

$$\text{TEORIA DOS NUMEROS} = 29\ 14\ 24\ 27\ 18\ 10\ 99\ 13\ 24\ 28\ 99\ 23\ 30\ 22\ 14\ 27\ 24\ 28$$

Implementação de Algoritmo de Chave Assimétrica

- Para encurtar as mensagens, utiliza-se conversão binário-texto[1]

```
mensagem = bignum("291424271810991324289923302214272428")
mensagem_base64 = base64_encode(mensagem)
mensagem_base64
```

```
## [1] "OCBTcBVvkyuloL66hwWs"
```

[1] não faz parte do processo matemático de encriptação

Implementação de Algoritmo de Chave Assimétrica

Com a mensagem pronta, o remetente gera duas chaves públicas n e e e uma chave privada d da seguinte forma:

1. escolhe dois números primos p e q muito grandes, tal que $n = p * q$, pois

$$\phi(n) = \phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$$

2. escolhe outro número primo e, tal que e seja inversível módulo $\phi(n)$ pois este será usado no processo de decodificação

1º problema: achar números primos

Solução: **Crivo de Eratóstenes** (Hefez, Abramo [Hef93], p. 88)

```
crivo = function(n) {

  # garantir que n é inteiro
  n = as.integer(n)

  # criando lista de primos de 1 até n supondo todos verdadeiros
  primos = rep(TRUE, n)

  # definindo 1 como não primo
  primos[1] = FALSE

  # definindo 2 como último primo
```

```
# obtendo os primos até 1000
crivo(1000)
```

```
##   [1]   2   3   5   7  11  13  17  19  23  29  31  37  41  43  47  53  59  61  67  71  73  79
##  [54] 251 257 263 269 271 277 281 283 293 307 311 313 317 331 337 347 349 353 359 367 373 379
## [107] 587 593 599 601 607 613 617 619 631 641 643 647 653 659 661 673 677 683 691 701 709 719
## [160] 941 947 953 967 971 977 983 991 997
```

1º problema: achar números primos

Tempo de executação para encontrar 10^6 primos:

```
inicio = Sys.time()
n1 = crivo(1e+06)
fim = Sys.time()

fim - inicio
```

```
## Time difference of 0.02000093 secs
```

1º problema: achar números primos

Tempo de executação para encontrar 10^9 primos:

```
inicio = Sys.time()
n2 = crivo(1e+09)
fim = Sys.time()

fim - inicio
```

```
## Time difference of 3.497244 mins
```

1º problema: achar números primos

- Tempo de processamento para encontrar os primos até 10^9 é 2,1 mil vezes maior do que para encontrar os primos até 10^6
- Seguindo a mesma proporção, levaria 47,3 horas para encontrar os primos até 10^{12}
- Normalmente são utilizados números maiores que 10^{100} e a busca por primos maiores é contínua

2º problema: garantir a possibilidade de decodificação

- Para que seja possível a decodificação, e deve ser inversível módulo $\phi(n)$, ou seja, $mdc(e, \phi(n)) = 1$.

Relembrando (Hefez, Abramo [Hef93], p. 118-121):

- o conjunto de todas as classes residuais módulo m é representado por $\mathbb{Z}_m$
- a vantagem das classes residuais é que transformam a congruência $a \equiv b \ (mod\ m)$ na igualdade $\bar{a} = \bar{b}$ e podemos realizar operações com essas classes
- um elemento $\bar{a} \in \mathbb{Z}_m$ é invertível se e somente se $(a, m) = 1$
- se $\bar{a}$ é invertível, então existe $\bar{b} \in \mathbb{Z}$ tal que $\bar{1} = \bar{a}.\,\bar{b}$, logo $a.\,b \equiv 1 \ (mod\ m)$

2º problema: garantir a possibilidade de decodificação

- conseguimos ver pela tabela de multiplicação que isso ocorre $\forall\ \bar{a} \in \mathbb{Z}_m$ apenas se m for primo
- se m não for primo, há pelo menos uma classe $\bar{a}$ não congruente a 1 módulo m

Implementação de Algoritmo de Chave Assimétrica

Garantida a possibilidade de decodificação, o receptor envia a chave pública (i.e. o par (n, e)) ao emissor da mensagem para o processo de encriptação.

$$\text{TEORIA DOS NUMEROS} = 29\ 14\ 24\ 27\ 18\ 10\ 99\ 13\ 24\ 28\ 99\ 23\ 30\ 22\ 14\ 27\ 24\ 28$$

Para encriptar um bloco b_i, toma-se o resto da divisão de b_i^e por n, ou seja, $b_i^e \pmod n$. Supondo $p = 13$, $q = 17$, $n = p.q = 221$ e $e = 5$, então:

letras	T	E	O	R	I	A
numeros	139	131	215	40	18	108

Implementação de Algoritmo de Chave Assimétrica

O emissor então envia a mensagem encriptada ao receptor que, de posse da chave privada, (i.e. o par (n, d)), sendo $d = (e \ mod \ \phi(n))^{-1}$ é o único capaz de decriptar os blocos a_i.

$$a_i = 139 \ 131 \ 215 \ 40 \ 18 \ 108$$

Para decriptar um bloco a_i, toma-se o resto da divisão de a_i^d por n, ou seja, $a_i^d \ (mod \ n)$. Calculando $\phi(n)$:

$$\phi(n) = (p - 1)(q - 1) = (13 - 1).(17 - 1) = 192$$

E podemos calcular a inversa de $5 \ (mod \ 192)$ aplicando o algoritmo euclidiano estendido, chegando a:

$$192.(-2) + 5.77 = 1$$

concluindo que $5.77 \equiv 1 \ (mod \ 192)$ e obtendo $d = (5 \ mod \ 192)^{-1} = 77$.

Por fim, podemos calcular $a_1 = 139^{77} \ (mod \ 221) = 29 = T$

Implementação de Algoritmo de Chave Assimétrica

De forma programática:

```
encriptar = function(mensagem, p, q, e) {

  if (!is.character(c(mensagem, p, q, e))) {
    stop("os argumentos devem ser caracteres")
  }

  p = openssl::bignum(p)
  q = openssl::bignum(q)

  # chave pública
  n = p * q
  e = openssl::bignum(e)
```

Implementação de Algoritmo de Chave Assimétrica

Exemplo de encriptação de mensagem

```
encriptar(
  mensagem = "demorei maior tempão pra entender essa parada",
  p = "112481050639317229656723018120659623829736571015511322021617837187076258724819",
  q = "89185111938335771293328323333111422985697062149139368049232365065924632677343",
  e = "65537"
)
```

```
## [1] "chave pública: n = 1003163509220912149867498786164902216377106156513044137355558453704745
## [1] "chave privada: 6886694454027199678759881881737814611139109676622117091683160716943132564‍8
## [1] "mensagem: P26Dr9AhfkSkweOaW/8GAGsJaKlEzCAIUwPs3/X1kolRrzqWdop/wNKHM1g3sZzmQXboH7hfE02ZfAG
```

Implementação de Algoritmo de Chave Assimétrica

Exemplo de decriptação de mensagem

```
decriptar(
  mensagem = "P26Dr9AhfkSkweOaW/8GAGsJaKlEzCAIUwPs3/X1kolRrzqWdop/wNKHM1g3sZzmQXboH7hfE02ZfAGG6W
  d = "68866944540271996787598818817378146111391096766221170916831607169431325648630077153452457
  n = "10031635092209121498674987861649022163771061565130441373555584537047455688991931937563110
)
```

```
## [1] "demorei maior tempão pra entender essa parada"
```

Referências

Hefez, Abramo (1993). *Curso de Algebra*. Vol. 1. Instituto de Matematica Pura e Aplicada, CNPq, p. 226.