MSX Article

Criptografia no MSX

Fulswrjudild qr PVZ

Resumo

O objetivo deste artigo é demonstrar algumas técnicas de criptografia no MSX.

1. Introdução

A criptografia (do grego: kryptós = escondido, graphein = escrita) é o estudo de técnicas para a transformação de uma informação de sua forma original para outra ilegível, de forma que possa ser compreendida apenas por seu destinatário, evitando assim que a informação possa a ser descoberta por pessoas não autorizadas. O receptor possui uma espécie de chave, que permite transformar o texto ilegível de volta a seu estado original.

Adaptado de: http://pt.wikipedia.org/wiki/Criptografia

2- Técnicas de Criptografia

2.1. A Cifra de César (Caesar Cipher Cryptography)

O imperador romano Júlio César (100 a.C. - 44 a.C.) já utilizava textos cifrados (criptografados) para se comunicar com os seus generais. A técnica utilizada na época recebeu seu nome.

Essa técnica é uma das mais simples e consiste na troca de uma letra do alfabeto por outra, distante dela um número fixo de vezes. No caso do Império Romano, a troca era feita por letras distantes três posições à frente.

Original	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Cifrada	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Tabela 1. A cifra de César

Observando a tabela 1, a frase “o coelho come a cenoura” ficaria “R FRHOK FRPH D FHQRXUD”.

Adaptado de: http://pt.wikipedia.org/wiki/Cifra_de_Cesar

2.2. Deslocamento da Tabela ASCII

A técnica da Cifra de César pode ser extendida para a tabela ASCII, utilizada para representar letras, números e símbolos em computação.

Do mesmo modo que é feito na Cifra César, um caractere é substituído por outro da tabela ASCII (não necessáriamente outra letra), distante n posições adiante ou para trás.

Por exemplo, a palavra “casa” possui os seguintes códigos ASCII:

Caractere	c	a	s	a
Código ASCII	99	97	115	97

Ao deslocarmos os caracteres de “casa” 64 posições para trás, o resultado seria:

Texto Original	c	a	s	a
Código ASCII	99	97	115	97
Deslocamento	-64	-64	-64	-64
Novo Código ASCII	35	33	51	33
Texto Cifrado	#	!	3	!

A palavra “casa” cifrada torna-se “#!3!”, no qual é ilegível.

Para se obter de volta a palavra original, deve-se realizar a operação inversa da cifragem, ou seja, somar 64 posições a cada caractere. O ato de “somar 64” é a chave para o receptor decifrar o texto.

Esta técnica pode ser utilizada para esconder textos de jogos e aplicações. Entretanto, não deve ser aplicada ao código de máquina do programa, pois ele se tornaria “ilegível” também para o computador.

Código em Basic para cifrar um texto:

```
10 M$="mensagem a ser cifrada"
20 D=2
30 MC$=""
40 FOR C=1 TO LEN(M$)
50 CC = (ASC(MID$(M$,C,1)) + D) MOD 256
60 MC$ = MC$ + CHR$(CC)
70 NEXT C
80 PRINT "Original: ";M$
90 PRINT "Cifrada: ";MC$
```

Saída:
Original: mensagem a ser cifrada
Cifrada: ogpucigo"c"ugt"ekhtcfc

A variável *D* é a chave e controla o deslocamento.

A instrução MOD é utilizada para realizar o deslocamento circular. Não é necessário, caso o tipo de variável seja inteiro não-sinalizado de 1 byte apenas. No MSX, a variável inteira possui 2 bytes, a de simples precisão 4 e dupla precisão 8 bytes.

Para decodificar a mensagem cifrada anteriormente, basta aplicar o mesmo deslocamento, só que no sentido inverso. Acrescente as seguintes linhas ao programa anterior:

```
100 IF D<0 THEN END
110 M$ = MC$ : D=-D
120 GOTO 30
```

Esse tipo de criptografia é simples e relativamente fácil de ser quebrada. Basta descobrir o valor do deslocamento, que é um número de 0 a 255, e a mensagem é decifrada. Hoje em dia, o uso de computadores para tal fim tornam essa tarefa ainda mais fácil.

Em vista disso, atualmente utilizam-se técnicas mais complexas e chaves bem maiores, no intuito de dificultar ao máximo a decodificação de um texto por uma pessoa não autorizada.

2.3. RSA – Chaves Privada e Pública[2][3]

RSA é um algoritmo de criptografia de dados, que deve o seu nome a três professores do Instituto de Tecnologia de Massachusetts (MIT) que inventaram este algoritmo: Ronald **R**ivest, Adi **S**hamir e Leonard **A**dleman. Até hoje é a mais bem sucedida implementação de sistemas de chaves assimétricas, uma vez que é baseada na dificuldade da fatoração de números primos grandes.

O RSA envolve um par de chaves: uma chave pública que pode ser conhecida por todos e uma chave privada que deve ser mantida em sigilo. Toda mensagem cifrada usando uma chave pública só pode ser decifrada usando a respectiva chave privada.

Geração das chaves

No RSA as chaves são geradas desta maneira:

1. São escolhidos dois números primos extensos, *p* e *q* (geralmente maiores que 10^{100}).
2. Calcula-se $n = p*q$ e $z = (p -1)*(q -1)$.
3. Escolhe-se um número primo *e*, de forma que $1 < e < z$. Essa é a chave pública.
4. Encontramos *d* de forma que $(e*d) \bmod z = 1$. Essa é a chave privada.

Características:

- O texto simples (uma string de bits) é dividido em blocos, de modo que cada mensagem de texto simples *M* fique no intervalo $0 <= p < n$.
- Para criptografar a mensagem *M*, é calculado $C = M^e \bmod n$.
- Para decriptografar *C*, é calculado $M = C^d \bmod n$.
- A chave pública é o par (n,e), enquanto que a chave privada é o *d*.

Exemplo:

Escolhemos $p=3$ e $q=11$.

$n = 3*11 = 33$
$z = (3-1)*(11-1) = 2*10 = 20$

Escolhe-se $e=7$, uma vez que 7 e 20 não possuem fatores em comum.

Para *d*, o valor 3 satisfaz a equação $(e*d) \bmod z = 1$.

Para cada letra da mensagem, obter o código da letra que deverá estar entre 0 e *n*. Por exemplo, o código da letra *B* é 2.

$C = 2^7 \bmod 33 = 29$

Para obter a letra original de volta, calcula-se:

$M = 29^3 \bmod 33 = 2$

O programa em Basic a seguir criptografa e descriptografa um texto, usando o RSA.

```
10 P=3 : Q=11
20 N=P*Q
30 Z=(P-1)*(Q-1)
40 E=7
50 FOR I=1 TO Z
60 D=I
70 IF (E*D) MOD Z <> 1 THEN NEXT
80 M$="PETROPOLIS"
90 PRINT"P:";P
100 PRINT"Q:";Q
110 PRINT"N:";N
120 PRINT"D:";D
130 PRINT"E:";E
140 PRINT"Texto original: ";M$ : F$=M$ : GOSUB 400
150 '
160 ' Criptografa
170 '
180 C$=""
190 FOR I=1 TO LEN(M$)
200 M=ASC(MID$(M$,I,1))-65
210 C1=(M^E)
220 C2=INT(C1/N)
230 C=C1-C2*N
240 C$=C$+CHR$(C)
250 NEXT I
260 PRINT"Texto cifrado: " : F$=C$ : GOSUB 400
270 '
280 ' Descriptografa
290 '
300 MM$=""
310 FOR I=1 TO LEN(C$)
320 C=ASC(MID$(C$,I,1))
330 M1=(C^D)
340 M2=INT(M1/N)
350 M=M1-M2*N
360 MM$=MM$+CHR$(M+65)
370 NEXT I
380 PRINT"Texto decifrado: ";MM$ : F$=MM$ : GOSUB 400
390 END
400 '
420 ' Imprime ASCII
430 '
440 FOR I=1 TO LEN(F$)
450 PRINT USING"## ";ASC(MID$(F$,I,1));
460 NEXT I : PRINT : PRINT
470 RETURN
```

Obs:

1. Para caber no limite de *n=33*, o texto somente poderá ter letras maiúsculas de A-Z. Por isso, o código ASCII de cada letra é deslocado de 65 posições para trás.
2. Para números grandes no Basic, o cálculo do módulo através da função MOD dá *overflow*. Dessa forma, fez-se o cálculo do módulo manualmente.
3. O máximo valor de *n* irá indicar o tamanho da palavra cifrada. No caso, o valor é 32 e ocupa apenas um byte.

2.4. Operador Lógico XOR[4]

Esta criptografia se baseia na propriedade do operador lógico XOR (ou exclusivo), onde:

A XOR B = C
C XOR B = A

Assim, podemos utilizar *B* como uma chave para criptografar a mensagem *A*, obtendo uma mensagem criptografada *C*. Aplicando a mesma chave *B* em *C*, podemos recuperar a mensagem original *A*.

```
10 K=73:C$="":D$=""
20 M$="O rato roeu a roupa do rei de Roma"
30 '
31 ' Criptografa
32 '
40 FOR I=1 TO LEN(M$)
50 A = ASC(MID$(M$,I,1))
60 C = A XOR K
70 C$ = C$ + CHR$(C)
80 NEXT I
90 '
91 ' Descriptografa
92 '
100 FOR I=1 TO LEN(C$)
110 C = ASC(MID$(C$,I,1))
120 A = C XOR K
130 D$ = D$+CHR$(A)
140 NEXT I
150 '
151 ' Imprime
152 '
160 PRINT"Original: ";M$
170 PRINT"Criptografada: ";C$
180 PRINT"Descriptografada: ";D$
```

2.5. Criptografia por tabela de correspondência[4]

Outra técnica interessante é criar uma tabela de correspondência entre os caracteres, gerando um vetor ordenado e outro com uma disposição aleatória dos caracteres. Exemplo:

Original	A	B	C	D	E	F	G	H	...
Cifrada	K	Z	T	B	Y	A	O	E	...

Mensagem original: “CADE”
Mensagem criptografada: “TKBY”

3- Créditos e Referências

Este artigo foi escrito por Marcelo Silveira em Março de 2004, revisado em Julho de 2017, Abril de 2018 e Outubro de 2018.

E-mail: flamar98@hotmail.com
Homepage: http://marmsx.msxall.com

[1] – Criptografia – em: http://pt.wikipedia.org/wiki/Criptografia
[2] – RSA – em: http://pt.wikipedia.org/wiki/RSA
[3] – Notas de Aula do Professor Orlando Bernardo, UERJ, Redes de Computadores, em: http://www.eng.uerj.br/~orlando
[4] – Revista CPU MSX, número 4.