MSX Article

MARMSX

Cryptography on MSX

Fubswrjudskb rq PVZ

Summary

This article presents some cryptography techniques applied on MSX.

1. Introduction

Cryptography (from Greek: kryptós = hide, secret, graphein = writing) is the practice and study of techniques for converting some information from its original form to a nonsense form, in order to be only discovered by the proper receiver, avoiding the third parties or the public from reading private messages. The receiver owns a key that allows to convert the nonsense text to the original text.

Adapted from: http://en.wikipedia.org/wiki/Cryptografy

2- *Cryptography Techniques*

2.1. Caesar Cipher Cryptography

The roman imperor Julius Ceasar (100 b.C. - 44 b.C.) already used encrypted messages on communications with his generals. This technique received his name.

This is a simple cryptography technique and consists in replacing each letter in the text by a letter some fixed number of positions forward or backwards the alphabet. In the Romans case, the shift was performed 3 letters forward.

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
Encrypted	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

Table 1. Caesar Cipher

According to the table 1, the phrase “the rabbit likes carrot” would be encrypted as “WKH UDEELW OLNHV FDUURW”.

Adapted from: http://en.wikipedia.org/wiki/Caesar_cipher

2.2. ASCII table shift

The Caesar Cipher technique may be extended to the ASCII table, used to represent letters, numbers and symbols on computers.

As the same way performed on Caesar Cipher, one character is replaced by another character distant n positions forward or backwards. For example, the word “goal” has the following ASCII codes:

Character	g	o	a	l
ASCII code	103	111	97	108

If we shift the word “goal” 64 characters backwards, the result would be:

Original Text	g	o	a	l
ASCII Code	103	111	97	108
Shift	-64	-64	-64	-64
New ASCII Code	39	47	33	44
Encrypted text	'	/	!	,

The word “goal” encrypted turned to “'/!,”, which is a nonsense word.

In order to recover the original word, we must apply the reverse way from the encryption, or, shift 64 characters forward. In that case, the act of adding 64 to each letter is the receiver's key to decode the text.

Cryptography is useful to hide some information on games or applications. Nevertheless, it should never be applied to machine code, once it turns the code unreadable for the MSX.

The following program in Basic encrypts a phrase using ASCII shift.

```
10 M$="message to be encrypted"
20 D=2
30 MC$=""
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 "Encrypted: ";MC$
```

Exit:
```
Original: message to be encrypted
Encrypted: oguucig"vq"dg"gpet{rvgf
```

The variable *D* is the key and controls the shift.

The Basic instruction MOD is used to make a circular shift. It would not be necessary if the numeric variable were a 1 byte type. But, MSX numeric variables uses 2 bytes for integers, 4 bytes for simple precision and 8 bytes for double precision.

In order to decode the encrypted message resulted on the previous program, just apply the same shift multiplied by -1. So, introduce the following lines to that program:

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

This cryptografy technique is quite simple to break. If you find the key value, a number ranging from 0 to 255, the message is discovered. Today, the use of computers for that is even a easier task.

Nowadays, more complex techniques and larger keys are used to make almost impossible to avoid text reading from third parties.

2.3. RSA – Private and Public keys[2][3]

RSA is one of the first public-key cryptosystems and widely used for secure data transmission. The acronym is related to three researches from MIT who created the algorithm: Ronald **R**ivest, Adi **S**hamir and Leonard **A**dleman. The RSA is based on the difficulty of the factorization of the product of two large prime numbers, the "factoring problem".

The RSA uses a public key, which is known by anyone, and a private key, which must be kept secret. Every message encrypted using a public-key can only be decrypted using the corresponding private-key.

Key generation

The keys on RSA are generated as follows:

1. Choose two distinct prime numbers, p and q (generally greater than 10^{100}).
2. Compute $n = p*q$ and $z = (p -1)*(q -1)$.
3. Choose a prime number e, where $1 < e < z$. This is the public-key.
4. Choose an integer d such that $(e*d) \bmod z = 1$. This is the private-key.

Characteristics:

- The text is divided into blocks, in such way that the message M lies on the interval $0 <= p < n$.
- To encrypt M, compute $C = M^e \bmod n$.
- To decrypt C, compute $M = C^d \bmod n$.
- The public-key is the pair (n,e), whereas the private-key is d.

Example:

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

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

Let $e=7$, once 7 is a prime number and not a divisor for 20.

For d, the value 3 is valid on the equation $(e*d) \bmod z = 1$.

For each text letter, get the corresponding numeric code which must range from 0 to n. For example, the letter B code is 2.

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

To get the original letter back, compute:

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

The following Basic program encrypt and decrypt a text using the RSA algorithm.

```
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$="PETROPOLIS"
90 PRINT"P:";P
100 PRINT"Q:";Q
110 PRINT"N:";N
120 PRINT"D:";D
130 PRINT"E:";E
140 PRINT"Original text: ";M$ : F$=M$ : GOSUB 400
150 '
160 ' Encrypt
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$(C)
250 NEXT I
260 PRINT"Encrypted text: " : F$=C$ : GOSUB 400
270 '
280 ' Decrypt
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$(M+65)
370 NEXT I
380 PRINT"Decrypted text: ";MM$ : F$=MM$ : GOSUB 400
390 END
400 '
420 ' Print 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. In order to respect the limit of *n=33*, the original text must have capital letters ranging from A-Z. According to that, the ASCII code was shifted 65 positions backwards.

2. The MSX Basic MOD operator overflows with big numbers. To avoid that limitation, the modulus was calculated manually.
3. The maximum *n* value indicates the size of each encrypted text “letter”. The example above has the maximum *n* value equal 32, which fits on one byte.

2.4. Logical Operator XOR[4]

This technique is based on the following XOR (exclusive OR) property:

A XOR B = C
C XOR B = A

According to that, we may use *B* as a key to encrypt a message *A*, resulting on an encrypted message *C*. Using the same key, we can decrypt *C* and get the message *A* back.

```
10 K=73:C$="":D$=""
20 M$="I love donuts"
30 '
31 ' Encrypt
32 '
40 FOR I=1 TO LEN(M$)
50 A = ASC(MID$(M$,I,1))
60 C = A XOR K
70 C$ = C$ + CHR$(C)
80 NEXT I
90 '
91 ' Decrypt
92 '
100 FOR I=1 TO LEN(C$)
110 C = ASC(MID$(C$,I,1))
120 A = C XOR K
130 D$ = D$+CHR$(A)
140 NEXT I
150 '
151 ' Print
152 '
160 PRINT"Original: ";M$
170 PRINT"Encrypted: ";C$
180 PRINT"Decrypted: ";D$
```

2.5. Correspondence table[4]

Another interesting technique is a conversion table between the characters. We create an ordered character array and then another array with a random distribution of characters. For example:

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

Original message: “FADE”
Encrypted message: “AKBY”

3- Credits and References

This article was originally written in Portuguese and translated into English by Marcelo Silveira on March 2004, reviewed on July 2017, April 2018 and October 2018.

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

[1] – Cryptografy – at: http://en.wikipedia.org/wiki/Cryptografy
[2] – RSA – at: http://en.wikipedia.org/wiki/RSA_(cryptosystem)
[3] – Class notes from Professor Orlando Bernardo, UERJ, Redes de Computadores, at: http://www.eng.uerj.br/~orlando
[4] – CPU MSX magazine, Brazil, #4.