Volume XXVII 2024
ISSUE no.1
MBNA Publishing House Constanta 2024

Scientific Bulletin of Naval Academy

SBNA PAPER • OPEN ACCESS

Matlab function for automating the affine encryption system

To cite this article: Paul Vasiliu and Tiberiu Pazara, Scientific Bulletin of Naval Academy, Vol. XXVII 2024, pg. 53-60.

Submitted: 15.04.2024
Revised: 20.06.2024
Accepted: 01.08.2024

Available online at www.anmb.ro

ISSN: 2392-8956; ISSN-L: 1454-864X

doi: 10.21279/1454-864X-24-I1-007

Matlab function for automating the affine encryption system

Paul Vasiliu[1], **Tiberiu Pazara**[2]

[1, 2] "Mircea cel Bătrân" Naval Academy, Constanţa, România

E-mail: [1] paul.vasiliu@anmb.ro , [2] tiberiu.pazara@anmb.ro

Abstract. A monoalphabetic encryption system is a system that substitutes each character with another character, always the same, regardless of position. The Caesar encryption system is a monoalphabetic system. The ROT13 encryption system, implemented on UNIX systems, is a Caesar encryption system. The affine encryption system is a generalization of the Caesar system.

Keywords: affine, encryption, system, encoding, decoding

1. A brief introduction to cryptography

The word cryptography was defined in 1658 by the English physicist Thomas Browne.

The term comes from the Greek language and is formed from the Greek words cryptos (hidden) and grafi (writing).

Cryptography is a component of a much broader field called information security. Cryptography is a branch of applied mathematics that is used to secure and maintain the privacy of information.

In practical terms, this involves converting a text (file, string of characters/bits) in clear (plain text) to a cryptic one (called cipher text). The process of converting or encoding plain text is called encryption. The reverse process of converting cipher text into (in) clear text is called decryption. Both processes use (in one form or another) an encryption procedure, called an encryption algorithm.

A message in its original form is called plaintext.

The sender rewrites this message using a method known only to him (possibly also to the recipient); we say that he encrypts (or ciphers) the message, obtaining a ciphertext.

The recipient receives the encrypted text and decrypts it knowing the method used for encryption.

The algorithm that performs the described operations is called an encryption system.

Definition 1 Let V and W be two nonempty sets, usually $V = W = \{0,1\}$, each of them called an alphabet. An encryption system is a quintuple(P, C, K, E, D) where:

$P = \{w \mid w \in V^*\}$ is the set of plain texts, written with elements of the non-empty alphabet V.

$C = \{w \mid w \in W^*\}$ is the set of encrypted texts, written with elements of the non-empty alphabet W.

K the nonempty set of the keys

E the set of the encryption methods

D the set of the deencryption methods

Each key $k \in K$ determines an encryption method $e_k \in E$ and a decryption method $d_k \in D$.

The encryption and decryption methods are the functions $e_k: P \rightarrow C$ (encryption function) and $d_k: C \rightarrow P$ (function of decryption) with the property $d_k(e_k(w)) = w$ for any $w \in P$. In general it is considered $C = \{\alpha \mid (\exists)\, m \in P\, , (\exists)\, k \in K\ s.t. \alpha = e_k(m)\ \}$. The function e_k must be injective. If it is also surjective, so bijective, then $d_k = e_k^{-1}$. In this case the encryption system is called symmetric or block encryption system.

In a symmetric encryption system where $P = C$, the encryption function is a permutation. In other words, if the set of clear texts coincides with that of the encrypted texts, an encryption with a symmetric system does nothing but a rearrangement (permutation) of the texts. This fact results from the bijectivity of the encryption function on finite sets.

An input message x is decomposed into $= x_1 x_2 \cdots x_n$n with $x_i \in P$.

Each x_i is encrypted using the encryption functione_k specified by the encryption key $k \in K$.

The sender calculates $y_i = e_k(x_i) \in C$, $i = 1,2,\cdots,n$ and obtains the encrypted message $y = y_1 y_2 \cdots y_n$ which it sends through the communication channel.

The recipient receives the message $y = y_1 y_2 \cdots y_n$ which he decrypts using the decryption function d_k and obtains $x_i = d_k(y_i)$, $i = 1,2,\cdots,n$, (see paper [9]).

2. Affine encryption system

The affine encryption system is a generalization of the Caesar system. In this paper I worked with the alphabet consisting of all uppercase letters, all lowercase letters, the digits of the decimal numbering system and the space character, numbered from 0 to 62.

Thus, I put this alphabet in correspondence with the ring of modulo 63 residue classes $(Z_{63}, +, \cdot)$.

Table 1. Correspondence with the ring of modulo 63 residue classes $(Z_{63}, +, \cdot)$.

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

char	W	X	Y	Z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
cod	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42

char	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	␣
cod	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62

Since there is no danger of confusion, for the convenience of writing, we will elementary refer to $\hat{x} \in Z_{63}$_by x.

In the affine encryption system, the equalities take place:

$P = C = Z_{63}$,

$K = \{(a,b) \mid a,b \in Z_{63}, gcd(a,63) = 1\}$

Let the key $k = (a,b) \in K$ be fixed.

The encryption function is: $e_k: Z_{63} \to Z_{63}$, $e_k(x) = a \cdot x + b \ (mod\ 63)$.

The decryption function is: $d_k: Z_{63} \to Z_{63}$, $d_k(y) = a^{-1} \cdot y + a^{-1} \cdot (63 - b) \ (mod\ 63)$.

The condition that a is prime to 63 ensures the existence of a^{-1} in the ring $(Z_{63}, +, \cdot)$) and the injectivity of the encryption function e_k.

In other words, there are no different characters that have the same encryption.

For example, we consider the coding function $e_k(x) = 10 \cdot x + 1 \ (mod\ 63)$, defined for any $x \in Z_{63}$. Obviously, $e_k(0) = 1$ which shows that letter A turns into letter B.

Obviously, $e_k(13) = 1$ which shows that letter N turns into letter B.

From these examples it follows that the coding function defined by $e_k(x) = 10 \cdot x + 1 \ (mod\ 63)$ is not good.

It can be shown that $14 \notin Im(e_k)$ and thus the letter O is not the image of a letter in the substitution alphabet.

Using the function function invzn(a,n) from paper [8] we obtained the invertible elements in the ring $(Z_{63}, +, \cdot)$, the set $U(Z_{63})$, $card\big(U(Z_{63})\big)$ and their inverses:

```
List of elements in U(Z63)
The element 1 is invertible and its inverse is 1
The element 2 is invertible and its inverse is 32
The element 4 is invertible and its inverse is 16
The element 5 is invertible and its inverse is 38
The element 8 is invertible and its inverse is 8
The element 10 is invertible and its inverse is 19
The element 11 is invertible and its inverse is 23
The element 13 is invertible and its inverse is 34
The element 16 is invertible and its inverse is 4
The element 17 is invertible and its inverse is 26
The element 19 is invertible and its inverse is 10
The element 20 is invertible and its inverse is 41
The element 22 is invertible and its inverse is 43
The element 23 is invertible and its inverse is 11
The element 25 is invertible and its inverse is 58
The element 26 is invertible and its inverse is 17
The element 29 is invertible and its inverse is 50
The element 31 is invertible and its inverse is 61
The element 32 is invertible and its inverse is 2
The element 34 is invertible and its inverse is 13
The element 37 is invertible and its inverse is 46
The element 38 is invertible and its inverse is 5
The element 40 is invertible and its inverse is 52
The element 41 is invertible and its inverse is 20
The element 43 is invertible and its inverse is 22
The element 44 is invertible and its inverse is 53
The element 46 is invertible and its inverse is 37
The element 47 is invertible and its inverse is 59
The element 50 is invertible and its inverse is 29
The element 52 is invertible and its inverse is 40
The element 53 is invertible and its inverse is 44
The element 55 is invertible and its inverse is 55
The element 58 is invertible and its inverse is 25
The element 59 is invertible and its inverse is 47
The element 61 is invertible and its inverse is 31
The element 62 is invertible and its inverse is 62
card(U(Z63)) = 36
```

For example $5^{-1} = 38$ and $19^{-1} = 10$.

The set of values of a is the set $U(Z_{63})$.

The set of values of b is the set Z_{63}.

The parameter b can have 63 values, which are taken independently of the values of a, with the only exception where $a = 1$ and $b = 0$.

The case where $a = 1$ and $b = 0$ leads to the identical encryption function, $e_k(x) = x\ (mod\ 63)$, which does no encryption.

Any key $k \in K$ is completely determined by the values $a \in U(Z_{63})$ and $b \in Z_{63}$. The total number of keys is equal to $card(K) = card\big(U(Z_{63})\big) \cdot card(Z_{63}) - 1 = 36 \cdot 63 - 1 = 2267$, a small enough number for a brute force attack to succeed.

3. Example 1

Let $a = 5$ and $b = 11$. Obviously $gcd(5,63) = 1$. The inverse of $a = 5$ in the ring Z_{63} is $a^{-1} = 38$. The encryption key is the pair $k = (5,11)$.

The encryption function is: $e_k(x) = 5 \cdot x + 11\ (mod\ 63)$.

The decryption function is: $d_k(y) = 38 \cdot y + 38 \cdot (63 - 11)\ (mod\ 63) = 38 \cdot y + 23\ (mod\ 63)$.

The encryption function values are calculated:

Table 2. Encryption function values.

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$e_k(x)$	11	16	21	26	31	36	41	46	51	56	61	3	8	13	18	23	28	33	38	43	48

x	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
$e_k(x)$	53	58	0	5	10	15	20	25	30	35	40	45	50	55	60	2	7	12	17	22	27

x	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
$e_k(x)$	32	37	42	47	52	57	62	4	9	14	19	24	29	34	39	44	49	54	59	1	6

The result is the encoding of the characters of the alphabet proposed in the work:

Table 3. Encoding of the characters.

$char$	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
$e_k(char)$	L	Q	V	a	f	k	p	u	z	4	9	D	I	N	S	X	c	h	m	r	w	1	6	A

$char$	Y	Z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
$e_k(char)$	F	K	P	U	Z	e	j	o	t	y	3	8	C	H	M	R	W	b	g	l	q	v	0	5	␣

$char$	x	y	z	0	1	2	3	4	5	6	7	8	9	␣
$e_k(char)$	E	J	O	T	Y	d	i	n	s	x	2	7	B	G

The values of the decryption function are calculated:

Table 4. Values of the decryption function.

y	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
$d_k(y)$	23	61	36	11	49	24	62	37	12	50	25	0	38	13	51	26	1	39	14	52	27

y	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
$d_k(y)$	2	40	15	53	28	3	41	16	54	29	4	42	17	55	30	5	43	18	56	31	6

y	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62
$d_k(y)$	44	19	57	32	7	45	20	58	33	8	46	21	59	34	9	47	22	60	35	10	48

This results in decoding the characters of the alphabet:

Table 5. Decoding the characters.

$char$	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
$d_k(char)$	X	9	k	L	x	Y	␣	l	M	y	Z	A	m	N	z	a	B	n	O	0	b

$char$	V	W	X	Y	Z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
$d_k(char)$	C	o	P	1	c	D	p	Q	2	d	E	q	R	3	e	F	r	S	4	f	G

$char$	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	␣
$d_k(char)$	s	T	5	g	H	t	U	6	h	I	u	V	7	i	J	v	W	8	j	K	w

We will find the encoding of the SeaConf 2024 message.
Because:
$e_k(S) = m$, $e_k(e) = j$, $e_k(a) = P$, $e_k(C) = V$, $e_k(o) = W$, $e_k(n) = R$, $e_k(f) = o$, $e_k(\text{␣}) = G$, $e_k(2) = d$, $e_k(0) = T$, $e_k(2) = d$ and $e_k(4) = n$
it turns out that the encoded SeaConf 2024 message is the mjPVWRoGdTdn message.
Because:
$d_k(m) = S$, $d_k(j) = e$, $d_k(P) = a$, $d_k(V) = C$, $d_k(W) = o$, $d_k(R) = n$, $d_k(o) = f$, $d_k(G) = \text{␣}$, $d_k(d) = 2$, $d_k(T) = 0$, $d_k(d) = 2$ and $d_k(n) = 4$
it turns out that the mjPVWRoGdTdn message decode is the SeaConf 2024 message.

4. Automatic encryption and decryption

For automatic encryption and decryption, the author wrote a program in Matlab. The program has the following functions:
Signature function function affine(a,b,n,mess) is the main function of the program. The function has the input arguments:

a	the coefficient of x from the encryption function $e_k\colon Z_{63} \to Z_{63}$, $e_k(x) = a \cdot x + b \ (mod\ 63)$.
b	the free term from the encryption function
n	value 63 for this paper
mess	character string containing the plaintext to be encrypted and then decrypted;

The function performs the encryption and decryption of the plaintext message using the key in the ring of residue classes modulo n, $(Z_n, +,\cdot)$.
Signature function function function charcod=ek(a,b,n,c,ychar,ycchar)
The function performs the encryption of the plain text message using the key in the ring of residue classes modulo n, $(Z_n, +,\cdot)$.
Signature function function chardecod=dk(a,b,n,codc,ychar,ycchar)
The function performs the decryption of the plaintext message using the key in the ring of residue classes modulo n, $(Z_n, +,\cdot)$.
Signature function function val=inv(a,n)
The function determines the inverse of the scalar a in the ring of residue classes modulo n, $(Z_n, +,\cdot)$.
The source program is:

```
function affine(a,b,n,mess)
 val=inv(a,n);
 if val ~= 0
   for i=1:n
       y(i)=i-1;
       if y(i)<=25
          ychar(i)=char(y(i)+65);
       end
```

```
        if y(i)>25 && y(i)<=51
            ychar(i)=char(y(i)+71);
        end
        if y(i)>=52 && y(i)<=61
            ychar(i)=char(y(i)-4);
        end
        if y(i)==62
            ychar(i)=char(y(i)-30);
        end
        yc(i)=mod(a*(i-1)+b,n);
        if yc(i)<=25
            ycchar(i)=char(yc(i)+65);
        end
        if yc(i)>25 && yc(i)<=51
            ycchar(i)=char(yc(i)+71);
        end
        if yc(i)>=52 && yc(i)<=61
            ycchar(i)=char(yc(i)-4);
        end
        if yc(i) == 62
            ycchar(i)=char(yc(i)-30);
        end
    end
    disp(ychar);
    disp(ycchar);
    pause;
    k=length(mess);
    for i=1:k
      messcod(i)=ek(a,b,n,mess(i),ychar,ycchar);
    end
    message=['The coded message is : ',messcod];
    disp(message);
    for i=1:k
      messdecod(i)=dk(a,b,n,messcod(i),ychar,ycchar);
    end
    message=['The decoded message is : ',messdecod];
    disp(message);
  else
    message=[' a = ',num2str(a),' it is not reversible in the ring Z',num2str(n)];
    disp(message);
  end;
end

function charcod=ek(a,b,n,c,ychar,ycchar)
for i=1:n
    if c==ychar(i)
        charcod=ycchar(i);
    end
end
end
```

```
function chardecod=dk(a,b,n,codc,ychar,ycchar)
for i=1:n
   if codc==ycchar(i)
    chardecod=ychar(i);
   end
end
end
function val=inv(a,n)
  val=0;
  for i=1:n-1
      if mod(i*a,n)==1
        val=i;
      end
  end
end
```

5. Running example

To run the program, I used the data from example 1.

```
>> a=5
a =
     5
>> b=11
b =
    11
>> n=63
n =
    63
>> mess='SeaConf 2024'
mess =
SeaConf 2024
>> affine(a,b,n,mess)

The coded message is : mjPVWRoGdTdn
The decoded message is : SeaConf 2024
```

6. Conclusions and further developments

In this work the author generalized the affine encryption system by extending the alphabet over which the messages are defined.

The author wrote an Matlab program that performs the encryption and decryption operations of the messages defined on this extension.

The presented examples prove the correctness of the implementation of the algorithm.

In the future, the author will expand the alphabet and develop the program by adding punctuation characters and other separators.

References

[1] Atanasiu A. - Teoria codurilor corectoare de erori, *Editura Univ. Bucureşti*, 2001.

[2] Atanasiu, A. – Securitatea informaţiei Vol. 1 (Criptografie), *Editura Infodata, Cluj*, 2008.

[3] Horowitz E., Sahni S. - Fundamentals of Computer Algoritms, *Computer Science Press*, 1985.

[4] Konheim A. - Computer Security and Cryptography, *Wiley Interscience*, 2007.

[5] Knuth E., D. – Tratat de programarea calculatoarelor, vol. 1, Algoritmi fundamentali, *Editura*

Tehnică, Bucureşti, 1974.

[6] Knuth E., D. - Tratat de programarea calculatoarelor, vol. 2, Sortare şi căutare, *Editura Tehnică, Bucureşti*, 1976.

[7] Knuth D. – Arta programării calculatoarelor, vol 3 (Algoritmi seminumerici*), Editura Tehnică, Bucureşti*, 1982.

[8] Paul Vasiliu, Tiberiu Pazara, Automatic determination of the elements $U(Z_n)$ and the inverse of a square matrix in the ring of residue classes modulo n, SEA - CONF 2023 9th INTERNATIONAL CONFERENCE, May 18st – 19nd, 2023 Constanța.

[9] Paul Vasiliu, A generalization of the Hill encryption system. automatic encryption and decryption Automatic determination of the elements $U(Z_n)$ and the inverse of a square matrix in the ring of residue classes modulo n, SEA - CONF 2021 9th INTERNATIONAL CONFERENCE, May 18st – 19nd, 2023 Constanța.

[10] Vasiliu P., Programare în Matlab, *Ed. ANMB, Constanţa* 2015.