

KRIPTOGRAFIJA SA PRIMJENOM

VJEŽBA 9

LABORATORIJSKE VJEŽBE U MATLABU

Roko Rogulj

VIGENEROVA ŠIFRA

Kod supstitucijske šifre svakom slovu otvorenog teksta odgovara jedinstveno slovo šifrata. Takve šifre se nazivaju *monoalfabetske*. Sada će se prikazati *Veigenerova* šifra koja spada u *polialfabetske* šifre. Naime, kod nje se svako slovo otvorenog teksta može preslikati u jedno od m mogućih slova (gdje je m duljina ključa), u ovisnosti o svom položaju unutar otvorenog teksta.

Definicija: Neka je m prirodan broj. Za ključ

$$K = (k_1, k_2, \dots, k_m),$$

definiramo

$$e_K(x_1, x_2, \dots, x_m) = (x_1 + k_1) \bmod 26, (x_2 + k_2) \bmod 26, \dots, (x_m + k_m) \bmod 26$$

$$d_K(y_1, y_2, \dots, y_m) = (y_1 - k_1) \bmod 26, (y_2 - k_2) \bmod 26, \dots, (y_m - k_m) \bmod 26$$

Za veličine ključa manje od veličine teksta ključ se ponavlja u nedogled. U primjeru ispod, pokazana je enkripcija koristeći otvoreni tekst '*hereishowitworks*' i ključ '*vector*'. Slova u ovim varijablama pretvaramo u brojeve koji odgovaraju položaju samoga slova u abecedi ($a=1, b=2, c=3$ itd...). Tako bi primjerice ključ '*vector*' preslikali u vrijednosti **[21, 4, 2, 19, 14, 17]**. Kroz ovu laboratorijsku vježbu koristit će se engleska abeceda.

Primjer 1:

(plaintext)	h	e	r	e	i	s	h	o	w	i	t	w	o	r	k	s
(key)	21	4	2	19	14	17	21	4	2	19	14	17	21	4	2	19
(ciphertext)	c	i	t	x	w	j	c	s	y	b	h	n	j	v	m	l

U slučaju da nije poznat cijeli ili dio otvorenog teksta najbolji napad na *Veigenerovu* šifru zasniva se na tome da frekvencija slova u engleskome jeziku nije homogena već neka slova se ponavljaju češće od drugih. Frekvencije slova engleskog jezika prikazane su u tablici ispod.

a	b	c	d	e	f	g	h	i	j
.082	.015	.028	.043	.127	.022	.020	.061	.070	.002
k	l	m	n	o	p	q	r	s	t
.008	.040	.024	.067	.075	.019	.001	.060	.063	.091
	u	v	w	x	y	z			
	.028	.010	.023	.001	.020	.001			

Naravno, postoje varijacije frekvencija unutar tekstova. Primjerice knjiga *'Gadsby'* koju je napisao *Ernest Vincent Wright* ne sadrži slovo e. Još je impresivnija knjiga *'La Disparation'* autora *George Pereca*, napisana na francuskom, koja nema slovo e (autor nije mogao koristiti priloge ili ženske imenice). Općenito se može zaključiti da je gornja tablica dobra pretpostavka za prosječne tekstove koji imaju više stotina riječi.

Kod jednostavne supstitucijske šifre svako pojedinačno slovo otvorenog teksta preslikano je u jedinstveno slovo šifrata. Jednostavnom frekvencijskom analizom bi utvrdili korišteni ključ za dovoljno dugi šifrat. *Vigenerova* šifra ima karakteristično svojstvo preslikavanja slova otvorenog teksta na više različitih slova u šifratu tako da frekvencije određenih slova u šifratu nisu nužno preslika frekvencija otvorenog teksta. Zbog toga je izrazito teže deducirati bilo koje informacije iz frekvencijske analize *Vigenerove* šifre. Također frekvencijska karakteristika slova u *Vigenerovoj* šifri je izrazito homogena i približno jednaka $1/26$ za svako slovo u abecedi. U primjeru ispod dana je frekvencija slova za prikazani šifrat.

Primjer 2:

jhbsbzomqropvqgquvhvnpseyiqscslzgyvxhgvuouqwlzshjwtsimfykiuibbpgsteysfrvhzkofrqzm
xugqglqcovtypbofprynpdigqiebvplghtcugsmzbygkqhzhzkofrqwibbiuewpgfeuihboevkiyixeopt
hzkhtdtoooguaeawbxkwlfwejpmmuldigsvvkotulwxcsgvmtxvfeksvaoujhrbulftrjyixeqpthzkht
dtoaipwgppsouqvlixeeomjsdigeyabyoaryykrpuhrucwgcusmxkscjvhxrtylpsqxtvdratibrsihz


a	b	c	d	e	f	g	h	i	j
6	14	7	6	14	9	17	17	20	7
k	l	m	n	o	p	q	r	s	t
12	11	8	2	19	17	15	13	18	16
		u	v	w	x	y	z		
		18	17	10	11	14	10		

Može se primijetiti da ne postoji slovo čija je frekvencija izrazito veća od drugih jer kako je otprije rečeno svako slovo otvorenog teksta je preslikano na više slova šifrata.

Razbijanje *Vigenerove* sastoji se od dva koraka: pronalaženje duljine ključa i pronalaženje vrijednosti ključa.

PRONALAŽENJE DULJINE KLJUČA

Neka je šifrat zapisan na dvije duge trake. Neka se trake postave jedna iznad druge uz pomak za određeni broj mjesta. U tablici ispod prikazan je šifrat i njegov pomak od dva mjesta.

pomak		v	q	h	q	w	v	w	v	h	m	u	s
šifrat		v	q	h	q	w	v	w	v	h	m	u	s
podudaranje					☺			☺					

U gornjem primjeru imamo podudaranje slova 'w' i 'q' za pomak od 2. Za pronalaženje duljine ključa potrebno je prebrojiti sva podudaranja za sve moguće vrijednosti pomaka. Najveća vrijednost podudaranja za određeni pomak trebala bi određivati veličinu ključa. U praksi se pokazalo da je duljina ključa najčešće jedan od višekratnika najveće vrijednosti podudaranja. U primjeru ispod izbrojena su podudaranja do 10 pomaka na danome šifratu.

Primjer 3:

*dtbtapiuewiasvpptwsaigpskftqenfydwlghhgajxhpftaqsuzoryzkwllloeucwukrrjppiyukqkpohr
vtbvlugtrqhgshhcnhwabvhxvplrafnomnjdoiuppviutgwlltgqhlsqieufpfvfqvhhtfuvenfukeyfu
wllegfskjpjldjqmxvgrrszylxojpwiueggvldksmlovvxvqthgsvfhejdgvwmsqpekwwguwhskhwaigfvf
qvrkybrkcsjvhvhuviugvhrbtgvxofpdqltcomjfhrrvaigviuegufvchrvaigrlafpgiksgfmwjgqxhofhzi
gquxofcgzlsudvf*

pomak	1	2	3	4	5	6	7	8	9	10
podudaranja	14	12	14	13	22	14	14	17	17	27

Može se primijetiti da su za pomake 5 i 10 najveće vrijednosti podudaranja te se može pretpostaviti da su to moguće veličine ključeva. Zbog toga što je 5 višekratnik od 10 pretpostavit će se da je duljina ključa 5.

PRONALAŽENJE KLJUČA

Neka matrica \mathbf{A}_1 predstavlja frekvencije svih slova u engleskom jeziku tako da je

$$\mathbf{A}_1 = [0.082, 0.015, 0.028, \dots, 0.020, 0.001].$$

Općenito, neka matrica \mathbf{A}_i predstavlja pomicanje elemenata matrice \mathbf{A}_0 za i mjesta u desno.

Primjerice matrica \mathbf{A}_2 bi iznosila

$$\mathbf{A}_2 = [0.020, 0.001, 0.082, 0.015, \dots]$$

Sljedeće je potrebno definirati matrice frekvencija za pojedine elemente šifrata sukladno samoj veličini ključa. Potrebno je napraviti n matrica \mathbf{W} gdje je n pretpostavljena veličina

ključa u šifratu. Tako na šifratu iz primjera 3, gdje je pretpostavljena veličina ključa 5, definiralo bi se pet matrica W_1, W_2, W_3, W_4 i W_5 .

Matrica W_i predstavlja frekvencije slova u šifratu na pozicijama koje su kongruentne s $i \bmod V_k$, gdje je V_k pretpostavljena veličina ključa. Primjerice, matrica W_1 bi predstavljala frekvencije slova u danome šifratu na pozicijama 1, 6, 11 ...

Za dani šifrat iz primjera 3 matrica vrijednost matrice W_1 je

$$W_1 = [0.0429, 0.0286, 0.0571, 0.0429, \dots].$$

Ova izračunata matrica bi trebala aproksimirati jedan od vektora A_i , gdje je i pomak izazvan vrijednosti prvoga elementa traženog ključa. Ako se izračunaju sve vrijednosti skalarnog produkta $W \times A_i$ za $1 \leq i \leq 26$, najveća vrijednost za i će se podudarati s točnom vrijednosti ključa za prvu poziciju. Za danu matricu W_1 vrijednosti skalarnog produkta s matricom A_i gdje je $1 \leq i \leq 26$ bi iznosile:

.0624	.0267	.0395	.0372	.04741	.0279	.0319	.0504	.0378	.0351
.0367	.0395	.0246	.0415	.0427	.0362	.0322	.0457	.0526	.0397
		.0322	.0299	.0364	.0372	.03352	.0406		

Najveća vrijednost nalazi se na prvome mjestu 0.0624 te prikazuje skalarni produkt matrica $W_1 \times A_1$. Iz ovoga se može zaključiti da je prvi član traženoga ključa 1.

Ukoliko se ponovi postupak traženja skalarnog produkta preostalih W matrica s matricama A , dobivene najveće vrijednosti bi predstavljale traženi ključ. Za šifrat iz primjer 3 ključ bi iznosio [1,2,3,4,7]. Otvoreni tekst naveden je ispod.

cryptographypriortothemodernagewaseffectivelysynonymouswithencryptionconvertinginformationfromareadablestatetounintelligiblenonsensethesenderofanencryptedmessagesharesthedecodingtechniqueonlywithintendedrecipientstoprecludeaccessfromadversariesthecryptographyliteratureoftenusethenamesaliceforthesenderbobfortheintendedrecipientandevefortheadversary

VJEŽBE U MATLABU

ZADATAK 1

Napišite funkciju oblika

```
function output = vigenereE(plaintext, key)
```

koja vrši enkripciju nad argumentom '**plaintext**' koristeći ključ '**key**' u Vigenereovom kodu.

Primjer poziva funkcije : `vigenereE('crypto', [2 1 2 2])`

ZADATAK 2

Napišite funkciju oblika

```
function output = vigenereD(ciphertext, key)
```

koja vrši dekripciju nad argumentom '**ciphertext**' koristeći ključ '**key**'.

Primjer poziva funkcije : `vigenereD('zgetzh', [2 1 2 2])`

ZADATAK 3

Napišite funkciju oblika

```
function output = frequency(input)
```

koja računa frekvenciju pojavljivanja slova engleske abecede ulaznog znakovnog niza.

ZADATAK 4

Napišite funkciju oblika

```
function output = coincidence(input)
```

koja računa podudaranja u znakovnom nizu kao što je to opisano u primjeru 3. Funkcija ispisuje vrijednosti svih podudaranja za sve moguće pomake.

ZADATAK 5

Napišite funkciju oblika

```
function output = freqW(input, position, kd)
```

koja računa W_i odnosno, frekvencije slova znakovnog niza '**input**' za indekse slova koji su kongruentni **position (mod kd)**.

ZADATAK 6

Napišite funkciju oblika

```
function output = WA(W,A)
```

koja računa sve vrijednosti skalarnog produkta $\mathbf{W} \times \mathbf{A}_i$ za $1 \leq i \leq 26$. Funkcija vraća najveću izračunatu vrijednost.