noSVEUČILIŠTE U ZAGREBU

FILOZOFSKI FAKULTET

ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI

Ak. god. 2014/2015.

Ian Christian Hanser

Aplikacija za zakrivanje podataka

Završni rad

Mentor: dr. sc. Vjera Lopina

Zagreb, 2015.

Contents

Popis slika

1.Uvod

Još od razvitka pisma je postojala potreba za sigurnom i nesmetanom komunikacijom.Sam razlog toj istoj potrebi je ljudska želja za privatnošću. Također je važno napomenuti da se kroz povijest mijenjao sam medij na kojemu se nalazi poruka(u početku je to bio pregament/papir, dok se danas najčešće koriste računala i mobilni uređaji.)Upravo radi te potrebe došlo je do razvoja kriptografije čiji je glavni cilj osigurati siguran kanal pošiljatelju poruke i primatelju iste.

U ovome seminarskom radu ćemo objasniti samu definiciju kriptografije kao znastvene discipline, njezinu primjenu u praksi i ciljeve kojima se kriptografija bavi. Također ćemo detaljnije objasniti tradicionalne sustave za raskrivanje i raskrivanje, razliku između transpozicijskih i supstitucijskih sustava za zakrivanje.

Cilj samoga rada je čitateljima objasniti kako funkcioniraju Cezarov sustav, Vigenerov sustav i sustav zakrivanja dvostrukim stupačnim slovoredom uz pomoć ključa. Osim samih sustava za zakrivanje, ovaj rad će također objasniti kako je moguće kreirati aplikaciju za zakrivanje i raskrivanje podataka. Za razumijevanje je potrebno znanje programskog jezika Python pošto je i sama aplikacija zakodirana u njemu.

2.Osnovni pojmovi

Jasnopis(eng. *plaintext*)- označava poruku u onome obliku u kojemu ju je moguće pročitati, odnosno označava poruku prije zakrivanja.

Zakritak(eng. *ciphertext*)- označava poruku u onome obliku u kojemu se nalazi poslje zakrivanja.

Ključ(eng. *key*) se koristi za zakrivanje poruke(sam način na koji se koristi ovisi o sustavu koji je u pitanju.) Također je važno napomenuti da se pošiljatelj i primatelj trebaju dogovoriti oko ključa prije samog zakrivanja i slanja poruke kako bi komunikacija tekla bez ikakvih pogrešaka.

Šifriranje ili zakrivanje(eng. *encryption*) je postupak u kojemu čitljivu poruku pokušavamo zakriti uz pomoć određenog kriptografskog algoritma.

Dešifriranje ili raskrivanje(eng. *decryption*) je postupak pri kojemu primatelj pokušava zakrivenu poruku učiniti čitljivom na temelju kriptografskog algoritma sa kojime je ta ista poruka zakrivena.

jasnopisni slovored- predstavlja abecedu jezika na kojemu želimo zakriti poruku.

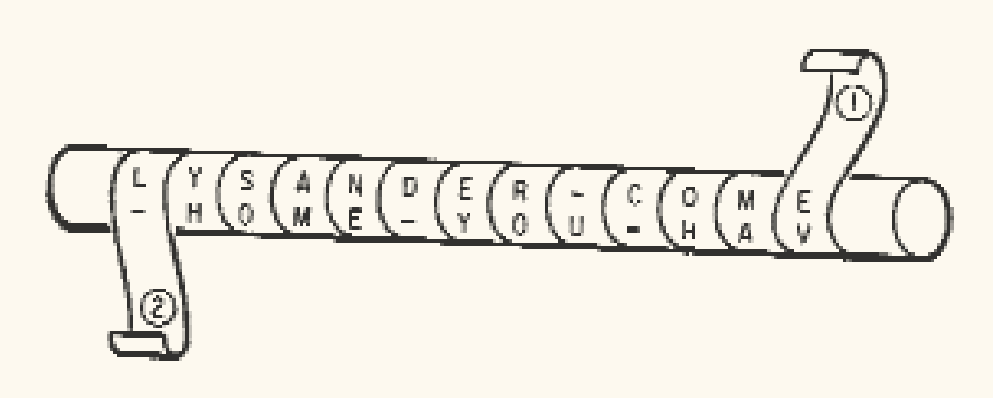
Zakriti slovored- predstavlja abecedu koja je rezultat primjene algoritma koji mjenja poziciju slovima u becedi.

Kvadratna tablica(lat. *Tabula recta*) je grafički prikaz abeceda gdje se sa svakim novim redom vrši pomak u ljievo za jedan.

3. O kriptografiji

Kriptografija je znnstvena disciplina koja se bavi proučavanjem i razvijanjem metoda pomoću kojih je moguće slati poruke u zakrivenome obliku, odnosno, u takvome obliku da ih može pročitati samo osoba kojoj je ta poruka namjenjena[1].

Određene metode možemo primjetiti već u 5. stoljeću prije Krista u Sparti. Spartanci su u to vrijeme koristili "napravu" za šifriranje koja se zvala Skital(lat. *scital*) i koristila se isključivo u vojne svrhe. To je u zapravo bio drveni štap oko kojega je pošiljatelj namotao vrpcu od životinjske kože ili, najčešće, pergamenta. Nakon namotavanja je jedino bilo potrebno okomito napisati poruku i odmotati vrpcu koja bi u ovome obliku sadržavala izmješana slova. Tu poruku je mogla pročitati jedino osoba koja je posjedovala štap iste debljine, inaće poruka ne bi imala smisla. Iako ovdje ne možemo uočiti nekakvu metodu za zakrivanje u matematičkom smislu, možemo zaključiti kako su ljudi krenuli osmišljati načine kako bi mogli komunicirati bez straha od presretanja poruke[2].



Slika 1)Skital

Ako pobliže želimo promotriti zadatke kriptologije, kao primjer možemo uzeti primjer sa tri osobe i to: Alice, Bob i Eve. Prva dva imena su nastala od prva dva slova abecede(A i B), dok je treće dobilo naziv po engleskom nazivu za osobu koja prisluškuje(*eavesdropper.*) Zadatak kriptografije je slijedeći: omogućiti Alice i Bobu sigurno uspostavljanje komunikacije bez straha da će Eve nadzirati komunikacijski kanal, te u konačnici presresti njihove poruke.[3]

[1] A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007.
[2] Beckett, B. Intruduction to Cryptology. Blackwell Scientific Publications, 1988.
[3] Shannon, C. Communication Theory of Secrecy Systems. 1949., 656-715

Jedan od glavnih komponenata kriptografije je zapravo kriptografski algoritam. On predstavlja matematičku funkciju koja se koristi za zakrivanje i raskrivanje poruke. Sam algoritam se sastoji od dvije funkcije, i to: funkcije za zakrivanje i funkcije za raskrivanje. Ovisno o kompleksnosti funkcije možemo reći koliko je teško neku poruku zapravo i raskriti, te da li će sam algoritam biti javno objavljen. Naime, ako je određeni kriptografski algoritam zastario ili se smatra prejednostavnim da bi se koristio za zakrivanje povjerljivih informacija, ta funkcija će biti objavljena javnosti. Također, ako se određeni sustav za zakrivnje zasniva na tajnosti ključa, isti sustav će biti dostupan javnosti.

3.1Razlika između šifre i koda

Kada govorimo o zakrivanju općenito, često se pojavljuju dva različita pojma: *Šifra* i *kod.* Osnovna razlika je u jedinici nad kojom se vrši zakrivanje. Na primjer, u slučaju šifara mi ćemo zapravo gledati svako slovo koje nalazi u poruci koja je namjenjena za slanje, te će se sukladno sa time, samo zakrivanje odvijati na razini slova. U slučaju kodova ćemo osim samih slova gledati i cijele riječi koje se često pojavljuju, te u skladu sa time će nam biti potrebna i *kodna knjiga*.

Kodna knjiga se u pravilu sastoji od popisa često korištenih riječi u svakodnevnome govoru ili popisa riječi vezanih uz određenu temu(npr. za vrijeme ratnoga stanja postoji jako velika potreba za slanjem tajnih poruka, te će ta kodna knjiga sadržavati određene vojne pojmove, te način kako ih zakriti.) i njihovih zakrivenih oblika(oni mogu varirati od jednog simbola do mješavine simbola i slova.)

Jedna od prednosti šifara naspran kodova je ta da ako se u slučaju kodova izgubi kodna knjiga ili ista padne u ruke neprijatelja, korištenje tog istog koda postaje besmisleno, dok u slučaju šifara ako se probije ključ, samo je potrebno stvoriti novi.Još jedna od problematika kodova je što za pisanje kodne knjige treba utrošiti dosta vremena, te opet nije sigurno da će sve riječi potrebne za komunikaciju biti zapisane. Uprvo radi toga se dosta često zna i koristiti mješavina šifre i koda koja korsiti i kodnu knjigu i algoritam za zakrivanje.[4]

4.Supstitucijske šifre

Supstitucijske šifre predstavljaju postupk u kojemu se svaki znak jasnopisa zamjenjuje sa odgovarajućim znakom u zakritom slovoredu. Među samim supstitucijskim šiframa također

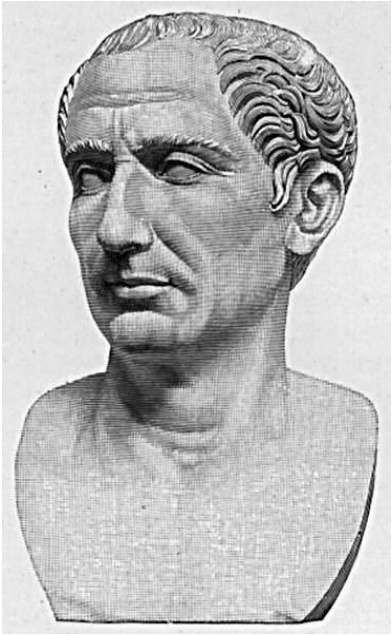
[4] Beckett, B. Intruduction to Cryptology. Blackwell Scientific Publications, 1988.

postoji dodatna podpodjela na *polialfabetske* i *monoalfabetske* sustave. Glavna razlika između njih je broj abeceda koji se koristi u šifriranju. [5]

Neki od primjera monoalfabetskih supstitucijskih šifara, ili šifara koje koriste isključivo jednu abecedu pri šifiranju jesu: Masonsko pismo(*eng. pigpen cipher*), Cezarova šifra, Abtash šifra i ROT13 šifra. Razlikujemo ih po nazivu, no u pravilu su sve šifre veoma slične jer funkcioniraju na sličnome principu, te ćemo navedenih šifara najdetaljnije obraditi Cezarovu šifru.

Osim monoalfabetskih sustava također razlikujemo i polialfabetske sustave, odnosno sustave u kojemu se koristi veći broj abeceda pri postupku šifriranja. Neki od polialfabetskih šifara su: Vigenereova šifra, Gronsfeldova šifra, Beaufortova šifra i autokey šifra. Od navedenih polialfabetskih šifara ćemo najdetaljnije obraditi Vigenereovu šifru posto su ostale šifre dosta slične.

4.1Cezarova šifra



Sama šifra je dobila naziv po rimskom vojskovođi Gaju Juliju Cezaru koji je istu koristio u vojne svrhe i za komunikaciju sa prijateljima. Sam sustav šifriranja je funkcionirao tako da se svako slovo jasnopisa zamjeni slovom koje se nalazi tri mjesta dalje u abecedi. Ukoliko se slovo jasnopisa nalazi na samome kraju abecede, te je nemoguće ostvariti pomak za tri mjesta, pretpostavljamo da se abeceda ponavlja ciklički, te ćemo to slovo zapravo smjestiti na početku zakritog slovoreda.[6]

Naravno, potrebno je napomenuti kako Caesarova šifra nije isključivo vezana uz broj tri, te da veličina pomaka ovisi o samom pošiljatelju poruke koji će osim poruke odrediti i pomak. Osim samoga zakrivanja, odraditi ćemo i raskrivanje, razbijanje šifre grubom silom i postupak kriptoanalize nad Caesarovom šifrom.

[5] A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007.
[6] http://practicalcryptography.com/ciphers/caesar-cipher/

4.1.1Cezarova šifra-zakrivanje

Recimo da želimo zakriti sljedeću poruku: "*Što je danas lijep i sunčan dan.*" Kao pošiljatelj poruke imamo pravo odrediti koliki će biti pomak koji će se koristiti pri zakrivanju. Pretpostavimo da je pomak jednak broju 3, u tome slučaju ćemo ispod svakoga slova napisati ono koje je pomaknuto za tri mjesta udesno. Stoga će jasnopisni slovored(*prvi red*) i zakritni slovored(*drugi red*) izgledati ovako:

A	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K	L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž
Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K	L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž	A	B	C

Samo zakrivanje se izvršava tako da se svako slovo jasnopisa zamjeni sa odgovarajućim slovom iz zakritnog slovoreda.

Primjer zakrivanja:

Jasnopis:

Što je danas lijep i sunčan dan.

Zakritak:

Vzs Ljh ečpču nlLjhš l užpDžčp ečp.

Iz gore navedenog primera također možemo primjetiti kako su dvoslovi kao što su lj nj i dž napisani kao: Lj Dž i Nj. Razlog tome je taj što se u zakritku nakon zakrivanja mogu na primjer pojaviti slovo l i j jedno pored drugoga, te upravo da bismo izbjegli zabune oko dvoslova, njih pišemo na prethodno spomenuti način.

4.1.2Cezarova šifra-raskivanje

Logika koja se koristi pri raskrivanju je ista kao i kod zakrivanja. Ovdje se isto uzima pomak, te se na temelju njega stvara jasnopisni slovored. Jedina razlika između zakrivanja i raskrivanja je ta što se kod raskrivanja slova pomiču u lijevo, a ne u desno.[7]

4.1.3Razbijanje Cezarove šifre grubom silom

Sada kada smo objasnili kako funkcionira zakrivanje uz pomoć Cezarove šifre uz korištenje pomaka, objasniti ćemo kako je moguće zakrivenu poruku raskriti bez poznavanja pomaka.Jedan od načina je uz pomoć "grube sile."[8]

[7] http://practicalcryptography.com/ciphers/caesar-cipher/
[8] http://practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-caesar-cipher/

Budući da je prostor ključeva jako mali(u ovome slučaju jednak 30,) ovu šifru možemo probiti u jako kratkome vremenu.To ćemo učiniti tako da uzmemo prvih pet slova zakritka te ćemo u svakome novome redu pomak povećati za jedan. Pošto se u ovome slučaju radi o raskrivanju pomak se neće pomicati u desno već u lijevo.

VLITČ

UKHŠC

TJGSB

ŠIFRA =>Ispravno

Pošto smo našli jedan slučaj gdje je zakritak nakon raskrivanja smislen, taj isti pomak ćemo iskoristiti i na ostatku rečenice.

Zakritak:

Vlitc tčbćLj hpč jzžćso ulnso

Jasnopis:

Šifra razbijena grubom silom.

4.1.4.Kriptoanaliza Cezarove šifre

Začetci frekvencijskih analiza se mogu naći već u 14. stoljeću u djelu Ibn-ad Duraihima(podrijetlom iz Arabije.) Osim velikoga doprinosa području fekvencijske analize važno je napomenuti kako je on prvi dao detaljne opise na temelju osam supstitucijskih šifara Također je bio prva osoba koja je predlagala korištenje *tabule recte* čak dva stoljeća prije nastanka Vigereove šifre.

Zanimljivo je pogledati da su se kriptografi u Europi počeli baviti kriptoanalizom tek u 15. stoljeću i u Italiji. U to vrijeme su najfrekventnija slova ili česte dvoslove i riječi zamjenjivli sa posebnim simbolima kako bi zbunili onoga koji pokušava presresti poruku. Upravo po

tome možemo zaključiti da su bili svijesni kako se supstitucijska šifra može vrlo lako probiti uz pomoć frekvencijske analize.[9]

Frekvencijska analiza se odvija tako da prvo priakupimo podatke koliko se često pojavljuje svako slovo u jeziku na kojemu je pisan tekst. Nakon analize pojava slova u tom jeziku je potrebno analizirati i zakritak koji želimo raskriti. Osim samih podataka o čestoći pojave slova, također nam mogu biti korisni i podatci o *bigramima*(dvoslovi) i *trigramima*(troslovi.)

A	I	O	E	N	S	R	J	T	U	D	K	V	L	M	P	C	Z	G	B	H	F
115	98	90	84	66	56	54	51	48	43	37	36	35	33	31	29	28	23	16	15	8	3

Frekvencija slova u hrvatskom jeziku(u promilima)[10]

Još jedan podatak koji nam može biti vrlo koristan pri kriptoanalizi jest frekvencija bigrama u hrvatskome jeziku.

JE(2.7%), NA(1.5%), RA, ST, AN, NI, KO, OS, TI, IJ, NO, EN, PR(1.0%).[11]

Zakritak:

Ekšjekščdk ćj zfkfNjšvjfk očNjičmDžčfk fkNjškDžk mnLjupkvkfćje rnLjćjvk č bjLjejšnčćNjdčc LjofLjNjk.

Sada ćemo za primjer napraviti kriptoanalizu jedne poruke koja je zakrivena **supstitucijskom šifrom** na hrvatskome jeziku. Analizu ćemo vršiti tako da za svako slovo u zakritku ispišemo koliko se puta određeno slovo pojavilo, koji se sve bigrami pojavljuju, te za svaki bigram broj njegovih pojava u zakritku. Na temelju toga ćemo moći zaključivati koje slovo u zakritku predstavlja koje slovo u jasnopisu.

[9] A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007.
[10] Tablica preuzeta iz knjige: A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007. str. 10
Podatci preuzeti iz knjige:[11] A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007. str. 11

	A												
1	**B**	J											
2	**C**	Lj											
7	**Č**	D	Nj	M	F	Ć	C	B					
4	**Ć**	J	J	J	Nj								
2	**D**	K	Č										
2	**Dž**	Č	K										
	Đ												
3	**E**	K	K	I									
7	**F**	K	Nj	K	K	K	Ć	Lj					
	G												
	H												
1	**I**	Č											
7	**J**	E	F	E	V	Lj	Š	Z					
13	**K**	Š	Š	F	Nj	Dž	V	F	Ć	O	F	M	Č
	L												
5	**Lj**	U	Ć	E	O	Nj							
2	**M**	Dž	N										
3	**N**	Lj	Lj	Č									
5	**Nj**	Š	I	Š	D	K							
2	**O**	Č	F										
1	**P**	K											
1	**R**	N											
	S												
5	**Š**	J	Č	V	K	N							
	T												
1	**U**	P											
3	**V**	J	K	K									
1	**Z**	F											
	Ž												

Najčešći bigrami:

FK(4), KF(3), ĆJ(3), NjŠ, EK, NLj

Pretpostavimo da slovo **K** predstavlja slovo **A**. Također po toj logici možemo primjetiti da dva najfrekventnija bigrama sadržavaju slovo K, odnosno, A kada izvršimo raskrivanje. Također je primjetno da su ta dva bigrama recipročna te najvjerojatnije odgovaraju bigramima: NA i AN, po ćemu zaključujemo da Slovo **F** predstavlja slovo **N**. Sljedeći najfrekventniji bigram koji nalazimo jest ĆJ, te je slovo **J** jedno od najfrekventnijih slova po ćemu možemo zaključiti da slovo **Ć** predstavlja slovo **J,** te slovo **J** predstavlja slovo **E.** Sljedeće što ćemo pokušati otkriti je zakritak bigrama ST(razlog tome je zto što je najčešći od neotkrivenih.) Možemo pretpostaviti da je odgovarajući bigram NjŠ odnosno da slovo **NJ** prestavlja **S** i **Š** predstavlja **T.** Također možemo vidjeti da nismo odgonetnuli slova I i O, to

nebi trebalo biti teško pošto su to samoglasnici i spadaju među najfrekventnija slova hrvatskog jezika. Po tome možemo pretpostaviti da su slova **Č** i **Lj** zapravo **I** i **O.** Za slovo Č ćemo pretpostaviti da odgovara slovu **I** pošto pri kraju zakritka vidimo izolirano slovo **Č.** Kada sve uvrstimo, dobivamo sljedeće slovorede i napola raskrivenu poruku:

A	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K	L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž
K								J				Č	Ć					F		Lj			Nj		Š				

EKŠJEKŠČDK ĆJ ZFKFNjŠVJFK OČNjIČMDžČFK FKNjŠKDžK MNLjUPKVKFĆJE

ate ati a je nanst ena is i ina nasta a o a anje

RNLjĆJVK Č BJLjEJŠNČĆNjDČC LjOFLjNjk

oje a i eo et ijs i o nosa

Jedino što nam je preostalo je pogledati ako prepoznajemo neku riječ koja se pojavljuje u gornjem zakrtitku (na primjer, zadnja riješ je odnosa.) Kada ju prepoznamo na temelju novoraskrivenih slova ćemo nastavljati postupak sve dok ne otkrijemo poruku u potpunosti. Tako dobivamo konačne slovorede i poruku:

A	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K	L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž
K	R	I	P	T	O	L	G	J	A	B	C	Č	Ć	D	Dž	Đ	E	F	H	Lj	M	N	Nj	S	Š	U	V	Z	Ž

Raskrivena poruka:

Matematika je znanstvena disciplina nastala proučavanjem brojeva i geometrijskih odnosa.

4.2Supstitucijska šifra sa ključem

Ova šifra spada pod monoalfabetske sustave za zakrivanje što znači da se u postupku raskrivanja i zakrivanja služi isključivo jednim zakritnim slovoredom. Sam princip zakrivanja i raskrivanja je identičan kao kod Cezarovog sustava uz razliku da ova šifra umjesto pomaka koristi ključ.

4.2.1Suptitucijska šifra sa ključem-zakrivanje

Prvi korak pri zakrivanju je, naravno, kreacija zakritnog slovoreda na temelju ključa. To činimo tako da prvo ispišemo ključ, te odstranimo sva slova koja se pojavljuju više od jednom. [12]

ključ prije odstranjivanja ponavljajućih slova:

MADAGASKAR

ključ nakon odstranjivanja ponavljajućih slova:

MADGSKR

Nakon kreacije ključa je potrebno kreirati i zakritni slovored, to činimo tako da na početak slovoreda ispišemo ključ, nakon kojega je potrebno dodavati slova abecede počinjujući sa početka. Jedini uvjet kada u zakritni slovored nećemo dodati slovo je ako se ono već nalazilo u ključu. Tako zakritni slovored koji se temelji na ključu "MADGSKR" izgleda:

A	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K	L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž
M	A	D	G	S	K	R	B	C	Č	Ć	Dž	Đ	E	F	H	I	J	L	Lj	N	Nj	O	P	Š	T	U	V	Z	Ž

Primjenjujući postupak zakrivanja koji smo savladali još kod Cezarove šifre:

Jasnopis:

ZakrivaNje uz pomoć kLjuča je jednostavno.

zakritak:

ZmfođvmLjc uz Njnjns fiugm ec ecklnptmvln.

[12] A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007.

4.3Vigenereova šifra

Temelj Vigenereove šifre je sam ključ, te se upravo radi njega ova šifra klasificira kao polialfabetski sustav za zakrivanje.Naime, u ovome sustavu vrijedi pravilo da svako slovo jasnopisa može postati *m* broj slova, gdje je m zapravo duljina ključa.

Sama šifra je dobila naziv po francuskome diplomatu Blaiseu de Vigenereu koji je 1586. godine objavio svoju knjigu "*Traicte de Chiffres*" u kojoj se nalazilo svo dotadašnje znanje o kriptografiji(ali ne i o kriptoanalizi.)[13]

Pri zakrivanju i raskrivanju je moguće koristiti matematičku funkciju(koju ćemo objasniti u sljedećem odjeljku) ili tablicu(*tabula recta*) koja predstavlja moguće kombinacije za svako moguće slovo jasnopisa i svako slovo ključa.

4.3.1 Vigenerova šifra-zakrivanje

Prije samog zakrivanja je potrebno definirati ključ koji može biti jedna riječ ili čak cijela rečenica, jedini uvjet je da ključ treba biti kraći od samoga zakritka. Zakrivanje se odvija "slovo po slovo" te funkcionira na slijedeći način.

Svakome od slova u abecedi ćemo dodjeliti jedan broj(počinjući sa 0 i završavajući sa 29.)

A	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

[13] http://www.cryptomuseum.com/crypto/vigenere/

L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

Ako želimo zakriti poruku:" *Poruka je zakrivena Vigenereovim sustavom*" koristeći ključ:"*kripto*" i matematičku funkciju prvo ćemo ispisati jasnopis i ispod njega ključ koji se ponavlja.Sljedeće što ćemo učiniti je da uzmemo broj koji odgovara slovu jasnopisa i broj koji odgovara slovu ključa. Sljedeći korak je zbrojiti ta dva broja i isčitati kojem slovu taj broj odgovara u prethodno navedenoj tablici. U slučaju da je zbroj tih brojeva veći od 29, od njega je jedino potrebno oduzeti 29 i pronaći odgovarajuću poziciju u tablici.

Jasnopis:

PORUKA JE ZAKRIVENA VIGENEREOVIM SUSTAVOM

KRIPTO KR IPTOKRIOT OKRIPTOKRIPT OKRIPTOKR

Zakritak:

DIĆMFO VA GPFIUNJOFT MUCOFČIRIFČI JGLĐPRGBDIĆ

Kada bi tu istu poruku htjeli raskriti uz pomoć *tabule recte,* prvi korak bi bio ispis jasnopisa i ponjavljajućeg ključa. Nakon toga je jedino potrebno pronaći slovo ključa u gornjem redu i slovo jasnopisa u lijevom stupcu i potražiti mjesto gdje se ta dva slova presjecaju i isčitati to slovo kao slovo zakritka.

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

slika 2) Tabula recta koja se koristi za zakrivanje na engleskome jeziku

4.3.2 Vigenerova šifra-raskrivanje

Raskrivanje je u principu isto u z jednu malu razliku. Naime, kod raskrivannja je jedina razlika ta što umjesto da zbrajamo poziciju ključa i poziciju slova u jasnopis, u ovome slučaju ćemo za svako slovo zakritka oduzeti poziciju slova ključa od pozicije slova zakritka.

Također kod raskrivanja uz pomoć tablice je potrebno prvo pronaći slovo koje se nalazi u kjuču u gornjem redu. Nakon toga je potrebno pratiti stupac koji se nalazi ispod tog slova sve dok ne naiđemo na slovo zakritka. Kada naiđemo na slovo zakritka samo je potrebno isčitati slovo jasnopis iz lijevog stupca.

4.4Šifra sa autoključem

Šifra sa autoključem(eng *autokey cipher*) je u pravilu veoma slična Vigenerovoj šifri uz jednu malu razliku. Umjesto da kroz cijelu poruku koristimo originalni ključ, ovdje ćemo stvari malo zakomplicirati. Prvo ćemo poruku razdvojiti u blokove koji su iste dužine kao i naš

ključ.Stoga poruku "Autoključ se temelji na Vigenereovoj šifri." uz ključ:KRIPTO zakrivamo na sljedeći način. Prvo poruku djelimo na blokove duljine 6 slova, te ključ nadopunjavamo tako da mu pridodajemo prethodni blok jasnopisa.

Jasnopis:

AUTOKLJ UČSETE MELJINA VIGENE REOVOJ ŠIFRI

KRIPTO AUTOKLJ UČSETE MELJINA VIGENE REOVOJ

Zakritak:

knđhfDž užnzfš jhepđuu kotpau Njoadep LjožNjc

5.Transpozicijske šifre

Prethodno smo spominjali supstitucijske šifre za zakrivanje koji se oslanjaju na zamjenu slova koja sse nalaze u zakritku. Sljedeće što ćemo obraditi su transpozicijske šifre. One se oslanjaju na promjenu razmještaja slova u jasnopisu. Samu podjelu na supstitucijske i transpozicijske šifre je ustanovio Giovanni Porta u 16. stoljeću.

Najčešća vrsta transpozicijske šifre je *stupačna transpozicija* u kojoj se jasnopis upisuje u pravokutnik po redcima koji su iste duljine kao i ključ. U slučaju da se posljednji redak ne ispuni do kraja, prazna mjesta možemo ispuniti nulom, znakom X ili ih jednostavno možemo ostaviti praznima.[14]

5.1Dvostruki stupačni slovored up pomoć ključa-zakrivanje

Kao što smo spomenuli ovaj sustav se zasniva na mjenjanju položaja svakog slova jasnopisa po određenome pravilu. U ovome slučaju je potrebno imati ključ koji ćemo analizirati i ispod svakoga slova napisati red pojavljivanja sljedeći abecedni red.

Ključ:

MAJSTOR

3 12 6745

Jasnopis:

Ova poruka je zakrivena uz pomoć transpozicije i kLjuča.

[14] http://crypto.interactive-maths.com/columnar-transposition-cipher.html

Sljedeće što je potrebno napraviti je ispisati naš jasnopis u tablicu koja će imati retke duljine 7 slova.

3	1	2	6	7	4	5
O	V	A	P	O	R	U
K	A	J	E	Z	A	K
R	I	V	E	N	A	U
Z	P	O	M	O	Ć	T
R	A	N	S	P	O	Z
I	C	I	J	E	I	K
Lj	U	Č	A	/	/	/

Kada smo ispisali poruku u tablicu kao što je navedeno gore je potrebno ta slova ispremještati. To činimo tako da stupce pišemo u redove druge tablice slijedeći brojeve koji se nalaze iznad svakoga stupca.

V	A	I	P	A	C	U
A	J	V	O	N	I	Č
O	K	R	Z	R	I	Lj
R	A	A	Ć	O	I	U
K	U	T	Z	K	P	E
E	M	S	J	A	O	Z
N	O	P	E	/	/	/

Ako želimo dodatno poboljšati sustav, ovdje možemo dodati jedan element supstitucijske šifre i to je da na temelju ključa kreirmo slovored koji ćemo koristiti za zakrivanje gornje tablice.[15]

[15] http://crypto.interactive-maths.com/columnar-transposition-cipher.html

Slovored koji smo kreirali na temelju ključne riječi majstor:

A	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	J	K	L	Lj	M	N	Nj	O	P	R	S	Š	T	U	V	Z	Ž
M	A	J	S	T	O	R	B	C	Č	Ć	D	Dž	Đ	E	F	G	H	I	K	L	Lj	N	Nj	P	Š	U	V	Z	ž

Na temelju gore navedenih slovoreda ćemo zakriti sljedeću poruku:

VaipacuajvonićokrzriLjraacoiukutzkpeemsjoznope

i dobiti:

VmDžLjmjumđvliDžtlenznDžgnmmjlDžueušzeLjcchNjđlzilLjcm

5.2Dvostruki stupačni slovored uz pomoć ključa-raskrivanje

Proces raskrivanja ćemo u ovome slučaju odraditi na istome primjeru koji smo malo prije zakrili. On se odvija tako da prvo poruku:
VmDžLjmjumđvliDžtlenznDžgnmmjlDžueušzeLjcchNjđlzilLjc

raskrijemo koristeći slovorede dobivene na temelju ključa. Nakon čega dobivamo:

VaipacuajvonićokrzriLjraacoiukutzkpeemsjoznope.

Tu ćemo poruku smjestiti ponovno u tablicu tako što ćemo ispisivati poruku red po red sve dok ne dobijemo sljedeću tablicu:

V	A	I	P	A	C	U
A	J	V	O	N	I	Č
O	K	R	Z	R	I	Lj
R	A	A	Ć	O	I	U
K	U	T	Z	K	P	E
E	M	S	J	A	O	Z
N	O	P	E	/	/	/

Zadnji korak u raskrivanju je kreacija tablice koj će predstavljati raskrivenu poruku. To ćemo postići tako da redke u tablici krenemo ispisivati u stupce na temelju ključa.

3	1	2	6	7	4	5
O	V	A	P	O	R	U
K	A	J	E	Z	A	K
R	I	V	E	N	A	U
Z	P	O	M	O	Ć	T
R	A	N	S	P	O	Z
I	C	I	J	E	I	K
Lj	U	Č	A	/	/	/

6.Objašnjenje koda

U ovome poglavlju seminarskoga rada ću objasniti važnije dijelove koda koji tvore aplikaciju koja je dio ovog seminarskog rada. Sama aplikacija je pisana u programskom jeziku Python(V2.7,) te je za shvaćanje koda potrebno predznanje programiranja u pythonu. Što se tiče samih funkcija, nisu korišteni moduli, dok sam za dizajn grafičkog sučelja izabrao modul Tkinter.

6.1GUI

Sljedeći isječci koda će opisivati izgled grafičkoga sučelja. Prvi primjer će pokazivati kreiranje same aplikacije i njezinog glavnog izbornika

```
Lang=getLang()
root = Tk()
root.title('CryptoBase')
root.geometry('280x200')
```

Pokretanje izbornika

Na početku pozivamo sam modul Tkinter te ga spremamo u varijablu *root,* sljedeće što radimo je definicija naslova koji će se prikazivati u glavnoj traci izbornika. Osim naslova još definiramo i veličinu izbornika(u pikselima.)

```
langCheck()
CaesarBtn = Button(root, text='Monoalfabetski supstitucijski sustav zakrivanja',
                   command=otvoriCaesar)
CaesarBtn.place(x=10, y=50)

VigenereBtn = Button(root, text='Polialfabetski supstitucijski sustav zakrivanja',
                     command=OtvoriVigenere)
VigenereBtn.place(x=10, y=80)

UbchiBtn = Button(root, text='Transpozicijski sustav zakrivanja sa ključem'.decode('cp1250'),
                  command=OtvoriUbchi)
UbchiBtn.place(x=10, y=110)

root.mainloop()
```

Izgled glavnog izbornika

Ovaj dio koda se nalazi pri samome kraju koda pošte sam prije definiranja izgleda samog glavnog izbornika ispisao funkcije koje će otvarati druge izbornike(svaki sa jednim sustavom za zakrivanje.) Na početku vidimo pozivanje funkcije langCheck koja se poziva pri otvaranju aplikacije.Sama će funkcija biti objašna u idućem potpoglavlju. Iznad također vidimo pozicioniranje gumbiju koji će otvarati odgovarajući sustav za zakrivanje.Pozicija gumbiju se također definira u pikselima (prva vrijednost za x os, druga za y.)

Osim glavnog izbornika također imamo i 3 zasebna izbornika od kojih svaki predstavlja jedan sustav zakrivanja koji je obrađen u ovome seminarskome radu(Cezarov sustav, Vigenerov sustav i transpozicijski sutav zakrivanja uz pomoć ključa.)

6.1.1Izgled izbornika[16]

Sljedeći kod će objašnjavati izgled izbornika koji sadrži Cezarovu šifru

```
CaesarWin = Toplevel(root)
CaesarWin.title("Monoalfabetski supstitucijski sustavi zakrivanja")
CaesarWin.geometry('600x500')

langCheck()
Lang=getLang()
```

Pokretanje prozora

Ovdje, kao i u prethodnome primjeru definiramo veličinu prozora koji će se pojaviti kada želimo koristiti Cezarov sustav zakrivanja. Također na početku vršimo provjeru jezika na kojemu će podatci biti ispisani.

```
Notelbl=StringVar()
Notelbl.set(Lang[12].decode('cp1250'))
LabelJasnopis = Label(CaesarWin, textvariable=Notelbl)
LabelJasnopis.place(x=10,y=10)
```

Primjer labela

Ovaj isječak koda prikazuje kako stvaramo label(naljepnicu) na kojoj će pisati „Polje za unos“ u Hrvatskom modu rada i „input field“ u Engleskom modu rada.U primjeru vidite kako zapravo pozivam dvanaesti član liste Lang(koja se uz pomoć funkcije langcheck mjenja ovisno o željenom jeziku.)

```
Jasnopis = StringVar()
Jasnopis.set('')
scrollbarJasnopis = Scrollbar(CaesarWin,orient="horizontal")
JasnopisEntry = Entry(CaesarWin,xscrollcommand=scrollbarJasnopis.set,
                      textvariable=Jasnopis)
```

Primjer polja za unos

Ovdje definiramo polje za unos, zadajemo mu poziciju i povezujemo samo polje sa klizačem koji se može pomicati horizontalno radi lakšeg pregleda pri unosu duljih poruka. Na isti način definiramo sve labele i polja u ovome prozoru i u ostalim prozorima, iduće što ćemo objasniti su radio gumbi i veliko polje za ispis podataka.

[16] https://docs.python.org/2/

```
RadioModUpVar=StringVar()
RadioModUpVar.set(Lang[6])
RadioModDnVar=StringVar()
RadioModDnVar.set(Lang[7])
Mod = StringVar()
Mod.set('z')
radioModUp = Radiobutton(CaesarWin, textvariable=RadioModUpVar,
                         variable=Mod, value='z').place(x=260,y=230, width=60, height=17)
radioModDown = Radiobutton(CaesarWin, textvariable=RadioModDnVar,
                           variable=Mod, value='r').place(x=260, y=250, width=60, height=17)
```

Radio gumbi

Na ovaj način se definiraju radio gumbi. Jedina razlika kod njih je što svaki gumb ima svoju vrijednost te se ovisno o njoj poziva odgovarajuća funkcija. Ovaj radio gumb zapravo kontrolira da li se radi o zakrivanju ili raskrivanju poruke.

```
scrollbarPrikaz = Scrollbar(CaesarWin,orient="vertical")
Prikaz = Text(CaesarWin, state='disable',
              yscrollcommand=scrollbarPrikaz.set, font=("Purista",10))
Prikaz.focus()
Prikaz.insert(END,'')
Prikaz.place(x=10, y=300, width=560, height=180)
scrollbarPrikaz.place(x=570, y=300, height=180)
scrollbarPrikaz.config(command=Prikaz.yview)
Prikaz.config()
```

Polje za prikaz

Gornji kod prikazuje postupak kreacije polja za prikaz. Ovdje također prikazujemo kako je moguće promjeniti stanje određenog objekta na sceni. To postižemo tako da za određeni objekt dodamo atribut „disable“ i „normal“. Razlog isključivanja ovog objekta je taj što korisniku želimo onemogućiti mjenjanje podataka u polju za prikaz.

```
CaesarBtn.config(state='disable')

def quitwin():
    CaesarWin.destroy()
    CaesarBtn.config(state='normal')

CaesarWin.protocol("WM_DELETE_WINDOW", quitwin)
```

Isključivanje gumba u glavnome izborniku dok je ovaj prozor aktivan

Gornji kod prikazuje funkciju koja se poziva svaki put kada se ovaj prozor zatvori. Prije toga vidimo kako je gumb u glavnome izborniku isključen

6.2Osnovne funkcije[17]

Sljedeće funkcije prikazumu određene procese koji će se odvijati prije samoga zakrivanja(pošto je postupak zakrivanja objašnjen u prvome dijelu ovoga rada, smatram da nije potrebno ponovno objašnjavati cijeli postupak.)

```
def langCheck():
    Lang=open('Lang.txt','r')
    Jezik=Lang.read()
    if Jezik!='Hrv' and Jezik!='Eng':
        Change=open('Lang.txt','w')
        Change.write('Hrv')
        Change.close()
```

Funkcija langCheck

Gore navedena funkcija se poziva pri otvaranju aplikacije te se ovdje vrši provjera datoteke Lang(koja je dio ove aplikacije.) U slučaju da netko promjeni sadržaj datoteke ova funkcija će automatski u nju zapisat „Hrv“ što aplikacija gleda kao varijablu koja određuje jezik. Ako u datoteci piše „Eng“, funkcija neće prepisati „Hrv“ preko toga pošto aplikacija ima podršku za engleski jezik.

```
def getLang():
    Jezik=open('Lang.txt','r')
    Lan=Jezik.read()
    if Lan=='Hrv':
        Lang=Hrv
    if Lan=='Eng':
        Lang=Eng
    return Lang
```

Odabir jezika

Odabir jezika se izvršava na način da aplikavcija provjerava datoteku u kojoj se nalazi odabrani jezik te sukladno sa time varijabli Lang dodjeljuje listu koja sadrži željene informacije za taj jezik.

[17] https://docs.python.org/2/

```
def supstSwitch():
    switch=Lan.get()
    if switch=='hr':
        Lang=Hrv
    elif switch=='en':
        Lang=Eng
    JasnopisLbl.set(Lang[2].decode('cp1250'))
    ZakritakLbl.set(Lang[3].decode('cp1250'))
    RadioWayUpVar.set(Lang[4].decode('cp1250'))
    RadioWayDnVar.set(Lang[5].decode('cp1250'))
    RadioModUpVar.set(Lang[6].decode('cp1250'))
    RadioModDnVar.set(Lang[7].decode('cp1250'))
    RadioLangHr.set(Lang[9].decode('cp1250'))
    RadioLangEn.set(Lang[10].decode('cp1250'))
    KeyK.set(Lang[11].decode('cp1250'))
    Notelbl.set(Lang[12].decode('cp1250'))
```

Provjera jezika unutar prozora

Za vrijeme rada aplikacije pri svakom novom otvaranju prozora se vrši provjera jezika, te ovisno o varijabli ispisuje sve potrebne podatke na željenom jeziku u odabranome prozoru. Odabir jezika također ima utjecaj na slova koja se mogu koristiti(s obzirom da hrvatska i engleska abeceda ne dijele sva slova.)

```
def ProvjeriDvoslov(unos):
    try:
        niz=list(unos.encode('cp1250'))
    except:
        niz=list(unos)
    unos=[]
    prethodnoSlovo=''
    for slovo, iduce_slovo in zip(niz, niz[1:]+[niz[0]]):
        if slovo+iduce_slovo in Dvoslov[0]:
            prethodnoSlovo=slovo+iduce_slovo
            unos.append(Dvoslov[0][Dvoslov[0].index(prethodnoSlovo)])
        elif slovo+iduce_slovo in Dvoslov[1]:
            prethodnoSlovo=slovo+iduce_slovo
            unos.append(Dvoslov[0][Dvoslov[1].index(prethodnoSlovo)])
        elif prethodnoSlovo in Dvoslov[0] or prethodnoSlovo in Dvoslov[1]:
            prethodnoSlovo=slovo
        else:
            unos.append(slovo)
    return  unos
```

Funkcija Provjeri dvoslov

Gore navedena funkcija učitava tekst koji smo upisali u polje za unos te razdvaja sami tekst na stringove duljine jedan ili dva. Ovo činimo u slučaju hrvatskoga jezika radi prisutnosti dvoslova kao što su Lj Nj i Dž. To nam je važno za samu funkciju aplikacije pošto se pri zakrivanju znakovi nj razlikuju od Nj(u prvome slučaju se tretiraju kao dva slova, dok u

druome kao jedno.)U samome programu sva slova ćemo tretirati kao niz, dok ćemo rečenicu gledati kao uređenu listu nizova.

```
def KreirajSlovoredN(key, Lang):
    key=int(key)
    ZakritniSlovoredL=['']*len(Lang[0])
    for char in Lang[0]:
        pozicija=Lang[0].index(char)-key
        pozicija=pozicija%len(Lang[0])
        ZakritniSlovoredL[pozicija]=char
    ZakritniSlovoredU=['']*len(Lang[1])
    for char in Lang[1]:
        char=char
        pozicija=Lang[1].index(char)-key
        pozicija=pozicija%len(Lang[1])
        ZakritniSlovoredU[pozicija]=char
    return ZakritniSlovoredL, ZakritniSlovoredU
```

Kreiranje slovoreda uz pomak

Gornji kod prikazuje kako kreiramo slovored u supstitucijskome sustavu koji koristi pomak. Postupak je slijedeći; za svako slovo dohvaćamo njegovu poziciju u neobrađenoj abecedi te ovisno o pomaku mjenjamo njegovu poziciju u zakritnome slovoredu koji kreiramo. Isti postupak ponavljamo i za velika slova, te sukladno sa time u daljnjemu proramu koristimo dva slovodera(jedan sadrži mala slova, dok drugi velika.)

```
def KreirajSlovoredK(key, Lang):
    saDuplima=ProvjeriDvoslov(key)
    ZakritniSlovoredL=[]
    ZakritniSlovoredU=[]
    for char in saDuplima:
        if char.upper() not in ZakritniSlovoredU:
            ZakritniSlovoredU.append(char.upper())
    for char in Lang[1]:
        if char not in ZakritniSlovoredU:
            ZakritniSlovoredU.append(char.upper())
    for char in ZakritniSlovoredU:
        ZakritniSlovoredL.append(Lang[0][Lang[1].index(char)])
    return ZakritniSlovoredL, ZakritniSlovoredU
```

Kreiranje slovoreda uz ključ

Ovdje primjenjujemo algoritam koji ovisno o zadanome ključu kreira zakritni slovored, to se izvršava tako da prvo odstranimo ponavljajuća slova u samome ključu, te nakon toga svako slovo obrađenoga ključa umećemo na početak slovoreda.Nakon što iskoristimo ključ u potpunosti, nastavljamo dopisivati preostala slova abecede dok ne iskoristimo sva slova abecede.

```
def IspisiZakritak(mod, jasnopis, ZakritniSlovoredL, ZakritniSlovoredU, Lang):
    zakritak=''
    if mod=='z':
        for char in jasnopis:
            if char in Lang[0]:
                zakritak+=ZakritniSlovoredL[Lang[0].index(char)]
            elif char in Lang[1]:
                zakritak+=ZakritniSlovoredU[Lang[1].index(char)]
            else: zakritak+=char
    elif mod=='r':
        for char in jasnopis:
            if char in Lang[0]:
                zakritak+=Lang[0][ZakritniSlovoredL.index(char)]
            elif char in Lang[1]:
                zakritak+=Lang[1][ZakritniSlovoredU.index(char)]
            else: zakritak+=char
    return zakritak
```

Funkcija ispisiZakritak

Ova funkcija radi na način da ovisno o modu ispisuje zakritu ili raskritu verziju poruke. Ova funkcija se primjenjuje samo u supstitucijskim sustavima dok se u transpozicijskom sustavu koristi jednostavniji način ispisa koji je povezan sa samom strukturom tablice.

Gornji kod također imitira postupak šifriranja uz korištenje jasnopisnog slovoreda i zakritnog slovoreda.

7.Zaključak

Iako su navedeni sustavi zastarjeli i danas ih je vrlo lako probiti što je osnovan razlog njihovom prestanku korištenja, njih danas možemo koristit za postavljanje temelja pri učenju kriptologije i kriptografije.U tim sustavima možemo vidjeti začetke logike koja se primjenjuje u zakrivanju i raskrivanju, te različite taktike koje su primjenjivane(pomak, korištenje ključa, transpozicija.)

Naime, smatram da se sa razumjevanjem tradicionalnih sustava za zakrivanje može razviti pobliže objasniti kako funkcionira i moderno zakrivanje danas. Upravo radi prenošenja znanja sam odlučio i napraviti aplikaciju koja sadrži tri različita sustava zakrivanja.

Također bih htio naglasiti da se na ovim sustavima može vježbati i probijanje što nam uvelike može koristiti ako se odlučimo baviti kriptografijom i zaštitom podataka općenito. Upravo sa tim znanjem možemo predviditi moguće propuste u samoj zaštiti podataka

8.Popis literature

1.)A Dujela, M. Maretić, Kriptografija. Zagreb: Element, 2007.

2.)Beckett, B. Intruduction to Cryptology. Blackwell Scientific Publications, 1988.

3.) 1. Bauer, F.L. Decrypted Secrets, Methods and Maxims of Cryptology. Springer, 2002

4.) 7. Shannon, C. Communication Theory of Secrecy Systems. 1949., 656-715

5.) https://docs.python.org/2/

6.) http://practicalcryptography.com/ciphers/caesar-cipher/

7.) http://practicalcryptography.com/cryptanalysis/stochastic-searching/cryptanalysis-caesar-cipher/

8.) http://www.cryptomuseum.com/crypto/vigenere/

9.) http://crypto.interactive-maths.com/columnar-transposition-cipher.html