MATEMATIČKI FAKULTET
UNIVERZITETA U BEOGRADU

MASTER RAD

Kriptoanaliza Vižnerove šifre

STUDENT: Ana Draganović
BROJ INDEKSA: 1062/06

MENTOR: dr Miodrag Živković

Beograd, 13.11.2009.

Sadržaj

1 Uvod

Vižnerova šifra je jedna od najpoznatijih klasičnih šifarskih sistema. Nastala je u XVI veku, a sve do XIX veka smatrano je da ju je nemoguće razbiti.

Tema ovog rada je analiza i implementacija jednog algoritma za kriptoanalizu Vižnerove šifre. Za potrebe rada korišćeni su primeri na dva prirodna jezika (engleskom i srpskom), odnosno tri pisma (engleski alfabet, azbuka i abeceda našeg jezika).

Algoritam kriptoanalize je zasnovan na primeni učestanosti pojavljivanja slova prirodnog jezika, odnosno na pojmu indeksa koincidencije prirodnog jezika. Za potrebe rada izvršena su eksperimentalna izračunavanja na uzorku tekstova pomenutih prirodnih jezika, kako bi se odredile njihove statističke karakteristike. Dobijene karakteristike jezika koriste se u okviru kriptoanalize i prikazane su u radu.

U cilju dobijanja pouzdanijih rezultata kriptoalgoritma, korišćene su tri metode, koje se mogu primeniti u određivanju dužine Vižnerovog ključa i prikazano je poređenje njihove uspešnosti, na osnovu rezultata eksperimentalnih izračunavanja. Najzad, izvršena je analiza uspešnosti algoritma kroz različite odnose dužine teksta i dužine ključa.

Program koji implementira pomenuti algoritam, pored korisničkog dela, predviđenog za demonstraciju rezultata kriptoanalize, sadrži metode za sva eksperimentalna izračunavanja prikazana u radu. Program je višejezičan i može se primeniti za bilo koji prirodni jezik, ako se pripreme odgovarajuće definicije spiska karaktera i baza tekstova na tom jeziku. Dodatna funkcionalnost programa je mogućnost interakcije korisnika u određivanju dužine i pojedinačnih slova Vižnerovog ključa.

U poglavlju 1 definišu se osnovni pojmovi vezani za oblast kriptologije. Navode se tipovi kriptosistema i daje pregled nekih od tehnika šifrovanja u klasičnoj kriptografiji, sa pregledom razvoja kriptografije kroz istoriju.

U poglavlju 2 daje se formalna definicija kriptosistema i definicije šifre pomeranja i Vižnerove šifre, sa primerima koji ilustruju postupke šifrovanja i dešifrovanja.

Poglavlje 3 sadrži prikaz osnovnih tehnika kriptoanalize. Razrađuje se analiza učestanosti pojavljivanja slova, opisuju izvedeni eksperimenti i prikazuju dobijeni rezultati, koji predstavljaju statističke karakteristike korišćenih prirodnih jezika. Detaljno se analizira algoritam kriptoanalize Vižnerove šifre kroz postupke određivanja dužine ključa i određivanje slova ključa, sa uvidom u druge poznate metode.

U poglavlju 4 se navode postupci korišćeni za procenu uspešnosti algoritma, opisuju eksperimenti i izlažu dobijeni rezultati.

Poglavlje 5 sadrži detaljan opis programa koji predstavlja implementaciju opisanih metoda iz poglavlja 3, sa korisničkim uputstvom.

1.1 Uvod u kriptologiju

Kriptografija *(*engl*. cryptography)* je veština i nauka čuvanja bezbednosti poruka. Naziv kritografija potiče od grčke reči ***kriptos***, što znači skriven. Ona omogućava da subjekt A (pošiljalac) sigurno pošalje svoju poruku subjektu B (primaocu), tako da nepozvana treća strana, subjekt C (napadač) ne može da dođe do njenog sadržaja.

Poruka koja se šalje naziva se još i ***otvoreni tekst*** (engl. *plaintext*). To je informacija u bilo kom obliku (tekstualni dokument, niz bitova, digitalni zapis,...).

Šifrovanje *(*engl*. encryption)* je proces maskiranja poruke, koji za cilj ima sakrivanje njene sadržine. Šifrovana poruka naziva se ***šifrat*** (engl. *ciphertext*).

Dešifrovanje *(*engl*. decryption)* je proces vraćanja šifrovane poruke u otvoreni tekst.

Označimo otvoreni tekst sa P, šifrat sa C, funkciju šifrovanja sa E, funkciju dešifrovanja sa D.

Proces šifrovanja poruke matematički se zapisuje

$$E(P) = C$$

a dešifrovanja šifrata

$$D(C) = P$$

Kako je cilj šifrovanja i dešifrovanja prenošenje originalne poruke treba da važi

$$D(E(P)) = P$$

Kriptografija ima za cilj:

- Zaštitu tajnosti informacija (sprečavanje otkrivanja njihovog sadržaja)

- Integritet informacija (sprečavanje neovlašćene izmene informacija)

- Autentičnost informacija (definisanje i provera identiteta pošiljaoca)

Kriptografski algoritam, poznat još kao ***šifra*** *(*engl*. cipher*) je matematička funkcija koja se koristi za šifrovanje i dešifrovanje (u osnovi su to dve srodne funkcije: jedna za šifrovanje, druga za dešifrovanje).

Kriptoanaliza (engl. *cryptanalysis*) je nauka razbijanja i čitanja šifrovanih poruka. Pokušaj kriptoanalize naziva se ***napad***. Uspešna kriptoanaliza naziva se ***dekriptiranje***. Cilj kriptoanalize je pronalažanje slabosti date kriptografske šeme u cilju njenog razbijanja.

Kriptoanaliza se preduzima od strane zlonamernih napadača sa ciljem obaranja sistema ili od strane samih dizajnera, radi provere sigurnosti i eventualne ranjivosti sistema. U tom smislu, cilj kriptoanalize ne mora obavezno biti obaranje sistema i otkrivanje sadržaja skrivene poruke.

Kriptologija *(*engl. *cryptology)* je oblast matematike koja obuhvata i kriptografiju i kriptoanalizu.

1.2 Algoritmi i ključevi

Ukoliko je sigurnost informacije zasnovan na tajnosti korišćenog algoritma za šifrovanje, radi se ***ograničenim*** *(*engl. *restricted)* algoritmima [1].

Nedostatke ograničenih algoritama savremena kriptografija rešava korišćenjem ***ključa*** *(*engl. *key)* za šifrovanje i dešifrovanje. Označimo ga sa K. Sigurnost ovih algoritama zasniva se na tajnosti ključa, i ne zavisi od detalja algoritma šifrovanja. Napadač, iako je upoznat sa algoritmom korišćenim za šifrovanje, neće moći da otkrije sadržaj poruke bez poznavanja ključa kojim je poruka šifrovana.

Funkcije šifrovanja i dešifrovanja ovih algoritma zavise od izbora ključa, pa se mogu zapisati:

$$E_K(P) = C$$

$$D_K(C) = P$$

i za njih važi $$D_K(E_K(P)) = P$$

Kriptosistem *(*engl. *cryptosystem)* predstavlja algoritam sa svim mogućim otvorenim tekstovima, ključevima i šifratima.

Postoje dva osnovna tipa algoritama zasnovanih na ključu:

- **simetrični algoritmi** (engl. *symmetric algorithms*)
- **algoritmi sa javnim ključem** (engl. *public-key algorithms*) ili **asimetrični algoritmi** (engl. *asymmetric algorithms*)

Simetrični algoritmi su algoritmi kod kojih se ključ za dešifrovanje može izvesti iz ključa za šifrovanje i obrnuto (često su ova dva ključa jednaka). Ovi algoritmi poznati su još i kao algoritmi sa tajnim ključem ili algoritmi sa jednim ključem. Da bi pošiljalac i primalac komunicirali, moraju se najpre dogovoriti oko upotrebe ključa. Problemi simetričnih kriptosistema su:

- Ključevi se moraju distribuirati u tajnosti. Sigurnost ovih algoritama zasniva se na tajnosti ključa, i oni su vredni koliko i poruke koje šifruju, jer poznavanje ključa daje uvid u sve poruke.
- Napadač može da se pretvara da je jedan od učesnika u komunikaciji i da proizvodi lažne poruke koje bi se šifrovale tim ključem. Jednom otkriveni ključ može da dešifruje sve poruke koje su njime šifrovane.
- Ukoliko svaki par korisnika u mreži upotrebljava poseban ključ, ukupan broj ključeva se uvećava sa rastom broja korisnika. Tako npr. za mrežu od n korisnika potrebno je $n(n-1)/2$ ključeva. Ovaj problem se može minimizirati tako što bi broj korisnika u mreži bio mali, ali to nije uvek moguće.

Simetrični algoritmi mogu se uslovno podeliti u dve kategorije: **lančane šifre** i **blokovske šifre.** Lančane šifre obrađuju otvoreni tekst bit po bit. Blokovske šifre rade na otvorenom tekstu podeljenom na grupe bitova - blokove. Tipična veličina bloka je 64 bita – dovoljno velika da otežа analizu, a dovoljno mala da bude praktična.

Algoritmi sa javnim ključem osmišljeni su tako da se za šifrovanje i dešifrovanje koriste različiti ključevi. Ključ za dešifrovanje ne može biti (bar ne u razumnom vremenskom roku[1]) izveden od ključa za šifrovanje. Stoga je čest slučaj da je ključ za šifrovanje javan i poznat svima, a samo strana koja ima odgovarajući ključ za dešifrovanje može doći do poruke. Ovaj postupak liči na korišćenje poštanskog sandučeta. Ubacivanje pošte u sanduče analogno je šifrovanjem pomoću javnog ključa - svako to može da učini. Uzimanje pošte iz sandučeta analogno je dešifrovanjem pomoću privatnog ključa - samo vlasnik sandučeta to može elegantno da uradi.

Ovaj postupak je matematički zasnovan na **jednosmernim funkcijama sa zamkom** (engl. *trapdoor one-way function*) [1]. One su zasnovane na pojmu **jednosmernih funkcija** (engl. *one-way function*). To su funkcije koje je relativno lako izračunati, ali je mnogo teže naći inverznu funkciju. Dobar primer jednosmerne funkcije jeste razbijanje tanjira - jednostavno je smrskati tanjir na hiljade delova, ali nije nimalo lako ponovo ga sastaviti.

Jednosmerne funkcije sa zamkom su poseban tip jednosmernih funkcija, sa skrivenom zamkom. Izračunavanje u jednom smeru je i dalje jednostavno, ali je teško nalaženje inverzne funkcije. Međutim, ukoliko znate tajnu informaciju (zamku) lako se može izračunati i drugi smer.

1.3 Klasična kriptografija

Klasična kriptografija odnosi se na algoritme koji šifruju tekstualne poruke [4]. Različiti kripografski algoritmi su ili zamenjivali slova teksta jedno drugima ili su ih premeštali. Tako se izdvajaju dva opšta tipa klasičnih šifara:

- **šifra zamene** ili supstituciona šifra (engl. *supstitution cipher*)
- **šifra premeštanja** ili transpoziciona šifra (engl. *transposition cipher*)

Kombinovanjem šifri premeštanja i zamene dobija se tzv. ***kombinovana šifra***.

Sa pojavom računara, algoritmi više ne rade sa slovima, već sa bitovima. Međutim, filozofija šifrovanja je ostala ista. Mnogi algoritmi još uvek kombinuju elemente zamene i premeštanja [4].

[1] Ukoliko je vreme potrebno za razbijanje šifre duže od vremena tokom kojeg je neophodno da šifrovani podaci ostanu tajna, algoritam se smatra bezbednim.

1.3.1 Šifra zamene

Šifra zamene svako slovo u otvorenom tekstu zamenjuje drugim slovom ili slovima u šifratu. U klasičnoj kriptografiji postoje četiri tipa šifara zamene [4]:

- **Šifra proste zamene** ili **monoalfabetska šifra** je šifra u kojoj je svako slovo otvorenog teksta zamenjeno odgovarajućim slovom u šifratu.
- **Homofonska šifra zamene** je šifra u kojoj jedno slovo otvorenog teksta može biti zamenjeno jednim ili nekolikim slovima u šifratu. Na primer, slovu „A" može da odgovara brojevima 5, 13, 25, slovo „B" brojevima 7, 19, slovo „C" broju 88,... itd.
- **Poligramska šifra zamene** šifruje blokove znakova u grupama. Na primer „ABA" može odgovarati bloku „RTQ", „ABB" može odgovarati bloku „SLL" itd.
- **Polialfabetska šifra zamene** sastoji se od više monoalfabetskih šifara. Na primer, može se koristiti pet različitih šifara proste zamene, a koja od njih će biti korišćena, zavisi od rednog broja slova u otvorenom tekstu.

Šifra proste zamene je najjednostavnija i najviše korišćena šifra u istoriji kriptografije. Specijalan slučaj šifre proste zamene je **šifra pomeranja** (engl. *shift cipher*), gde se svaki znak poruke zamenjuje znakom pomerenim za k mesta udesno (po modulu dužine azbuke). Primer ovakve šifre je čuvena Cezarova šifra[2], koja svaki znak poruke zamenjuje znakom koji se nalazi tri mesta udesno (po modulu veličine azbuka). Na primer, u slučaju engleskog alfabeta, slovo A se zamenjuje sa D, B sa E, C sa F ,..., X sa A, Y sa B, Z sa C.

Polialfabetske šifre zamene imaju više ključeva sa jednim slovom, a svaki od njih se koristi za šifrovanje tekućeg slova poruke (prvi ključ šifruje prvo slovo, drugi ključ drugo i tako redom). Nakon što se svi ključevi iskoriste, pristupa se njihovom recikliranju, odnosno za ključ dužine m, svako m-to slovo poruke će biti šifrovano istim ključem. Ovo se naziva ciklus (period) šifre.

1.3.2 Šifra premeštanja

Šifra premeštanja je šifra kod koje slova otvorenog teksta ostaju ista, ali im se menja raspored. Zato se ova šifra naziva još i šifra permutacije.

Svako slovo otvorenog teksta P zamenjuje se nekim slovom šifrata C, prema odgovarajućoj funkciji permutacije k.

$$E_k(P) = k(P)$$

Za dešifrovanje se koristi inverzna funkcija permutacije k^{-1}.

$$D_k(C) = k^{-1}(C)$$

Ovu šifru je, međutim, lako razbiti korišćenjem osnovnih tehnika kriptoanalize. Obrada šifrata drugom šifrom premeštanja u velikoj meri povećava sigurnost.

[2] Cezarova šifra je i danas sinonim za šifre proste zamene gde svakom slovu poruke odgovara tačno jedno slovo šifrata.

Iako mnogi savremeni algoritmi koriste transpoziciju, ona zahteva mnogo memorije i ponekad se mora ograničiti dužina poruke. Zato se šifra zamene mnogo češće koristi od šifre premeštanja.

1.3.3 Rotorske mašine

Oko 1920. godine izumljene su razne sprave za automatizaciju procesa šifrovanja. Mnoge od njih su se zasnivale na konceptu **rotora** – mehaničkog diska za izvođenje raznih vrsta zamene. Rotorska mašina ima tastaturu i niz rotora i radi na principu Vižnerove šifre (1.5.4). Svaki rotor sadrži proizvoljnu permutaciju slova korišćene azbuke, broj pozicija je jednak veličini azbuke i obavlja običnu zamenu. Izlazne iglice jednog rotora povezane su sa ulaznim sledećeg. Kombinacija nekoliko rotora i zupčanika koji ih pomeraju, čini da rezultat rada mašine bude siguran [1].

Najpoznatija rotorska mašina je Enigma[3]. Enigmu su koristili Nemci za vreme II svetskog rata. Nemačka Enigma je upotrebljavala tri rotora, po izboru od ukupno pet, komutator koji je neznatno permutovao otvoreni tekst, i reflektujući rotor zbog kojeg je svaki rotor po dva puta operisao sa svakim slovom poruke.

1.4 Jednokratne beležnice

Dugo je važilo pogrešno ubeđenje da je svaki kriptografski sistem moguće razbiti. Sa tim u vezi, Klod Šenon[4] (Claude Shannon) je u Bell laboratorijama tokom II svetskog rata dokazao da postoje šifre koje je nemoguće razbiti. One se nazivaju **jednokratne beležnice** ili **OTP šifre**[5] i njihova neranjivost zasniva se na činjenici da je ključ kojim se poruka šifruje potpuno slučajan niz karaktera, čija je dužina ne manja od dužine poruke, da se svaki ključ koristi samo jedanput i da je nedostupan napadaču.

U suštini, jednokratna beležnica je veliki, neponavljajući skup istinski nasumičnih ključeva, napisanih na parčetu papira i zalepljenih zajedno u beležnicu. Pošiljalac koristi svako slovo ključa iz beležnice da bi šifrovao tačno jedno slovo otvorenog teksta. Svako slovo ključa koristi se jednom, za samo jednu poruku. Nakon što se jedna poruka pošalje, odgovarajući list se uništava, tako da se za svaku novu poruku koristi novi list sveske, tj. novi ključ [1].

Primalac ima identičnu beležnicu i koristi svaki ključ iz beležnice da bi dešifrovao svako slovo šifrata. Po dešifrovanju poruke, odgovarajući list se uništava.

Ovaj sistem je apsolutno siguran pod pretpostavkom da napadač nema pristup jednokratnoj beležnici korišćenoj za šifrovanje poruke. Dati šifrat poruke može odgovarati bilo kojem otvorenom tekstu slične dužine.

---

[3] Enigma na grčkom znači zagonetka.

[4] Klod Šenon (1916-2001) bio je američki naučnik i inženjer. Smatra se tvorcem moderne kriptografije. 1949. Njegov članak "Komunikaciona teorija tajnosti sistema" uzdigla je kriptografiju do statusa nauke.

[5] OTP (engl. One Time Pad).

Pošto je skup ključeva nasumično generisani skup slova, napadač nema informaciju neophodnu za kriptoanalizu šifrata. Kako je svaki otvoreni tekst poruke podjednako moguć, ne postoji način na koji bi kriptoanalitičar otkrio koji je otvoreni tekst poruke tačan.

Sigurnost jednokratne beležnice zasniva na dve pojedinosti:

- slova - ključevi moraju se generisati nasumično
- jednom iskorišćeni redosled ključeva više nikada ne sme biti korišćen

Jednokratne beležnice se i danas primenjuju u onim komunikacionim kanalima u kojima je potrebno obezbediti ultravisok nivo bezbednosti [1].

1.5 Kriptografija kroz istoriju

1.5.1 Sakrivanje poruka

Neprestana borba između šifranata i dekriptanata dovela je do niza naučnih dostignuća. Šifranti teže da naprave što otpornije šifre kako bi zaštitili komunikacije, dok dekriptanti neprestano iznalaze nove metode za njihovo razbijanje. U svojim naporima da ugroze, odnosno sačuvaju tajnost poruke, obe strane su posezale za različitim disciplinama i tehnologijama, od matematike do lingvistike, od teorije informacija do kvantne fizike. Za uzvrat, šifranti i dekriptanti su obogatili ove nauke, a njihov rad je ubrzao tehnološki razvoj, što se vidi na primeru današnjih računara [2].

Ideja za prikrivanjem značenja poruke može se vezati čak za period starih Egipćana, 4000 godina pre nove ere. Slova uklesana na grobnicama nekih vladara, koja ne odgovaraju klasičnim hijeroglifama, smatraju se prvim oblicima kriptografije. Međutim, ovi drevni zapisi zapravo nisu pravi šifrati korišćeni radi tajnosti, već predstavljuju drugi, takođe bitan aspekt kriptografije: namerno transformisanje načina pisanja.

Vremenom, siguran i zaštićen prenos poruka postajao je sve važniji, pre svega u vojnoj komunikaciji. Kuriri su slani da prenose različite poruke svojim trupama, i bilo je važno da u slučaju da ih neprijatelj presretne, to ne znači da će presresti samu poruku.

Prema rimskom filozofu i državniku Ciceronu, tajna pisma pominjao je još Herodot, „otac istorije“. Prema Herodotu, upravo je veština tajnog pisma spasla Grke od potpadanja pod vlast Kserksa, vladara Persije. Strategija njihovog tajnog komuniciranja zasnivala se samo na skrivanju poruka. Ovaj princip komunikacije naziva se **steganografija** (*steganos*-prikriven i *grafein*-pisati). Od Herodotovog vremena proteklo je dve hiljade godina. Za to vreme širom sveta korišćeni su razni oblici steganografije: sakrivanje poruke uklesane u drvetu polivajući drvo voskom, tetoviranje glave robova i čekajući da im poraste kosa, pisanje nevidljivim mastilom i slično [2].

Dug opstanak steganografije ukazuje da ona svakako nudi određeni nivo zaštite, ali ima i određene slabosti. Ako neprijatelj presretne glasnika i pronađe skrivenu poruku, njen sadržaj je automatski otkriven.

1.5.2 Počeci klasične kriptografije

Paralelno sa razvojem steganografije odvijala se evolucija kriptografije. Nasuprot steganografije, cilj kriptografije nije da u postupku šifrovanja sakrije samu poruku, već njeno značenje. Mada steganografija i kriptografija nisu ni u kakvoj vezi, poruku je moguće i šifrovati i sakriti kako bi se postigla maksimalna sigurnost. Na primer, mikrotačka[6] je oblik steganografije koji je postao popularan tokom Drugog svetskog rata.

Spartanci su prvi uveli sistem vojne kriptografije. Oni su konstruisali uređaj **skitale** u petom veku pre nove ere. Ovaj uređaj zasnivao se na jednoj vrsti šifre premeštanja. Skitale je drveni štap oko koga je obmotana traka od kože. Pošiljalac ispisuje poruku duž štapa, a zatim odmotava traku koja naizgled nosi besmisleni niz slova. Poruka je šifrovana, jer se ona mogla pročitati jedino ako se delovi teksta postave u pravilnom redosledu.

Prva šifra, priznata u današnjem smislu je šifra zamene i ona dolazi od Rimljana. To je pomenuta Cezarova šifra.

Zbog svoje jednostavnosti i pouzdanosti, šifra zamene je dominirala veštinom tajnog pisanja tokom celog prvog milenijuma nove ere. Šifranti su razvili sistem koji je garantovao bezbedne komunikacije, tako da nije bilo potrebe da se na tom polju daje istražuje. Šifarski sistemi koje su koristili bili su prilično jednostavni i zasnivali su se na šiframa zamene i steganografima. Mnogi naučnici starog veka smatrali su da je, zbog ogromnog broja mogućih ključeva, šifra zamene nerešiva. To je tokom dugog niza vekova bilo tačno. Do proboja u ovoj oblasti, koji je zahtevao briljantno poznavanje lingvistike i statistike, došlo se na istoku.

1.5.3 Arapska kriptoanaliza

Bogatstvo islamske kulture najvećim delom je rezultat imućnog i mirnog društva. Država se oslanjala na efikasan administrativni sistem, a njeni činovnici na bezbedne komunikacije zaštićene šifrovanjem. Zabeleženo je da su se, osim osetljivih državnih dokumenata, šifrovali i podaci o porezima, što ukazuje na vrlo rasprostranjenu svakodnevnu upotrebu kriptografije. Međutim, osim što su šifrovali poruke, Arapi se smatraju osnivačima kriptoanalize.

Arapi su imali najbolje uslove za razvoj kriptoanalize, jer su dostigli visok nivo u matematici, statistici i lingvistici. U verskim školama, u Basri, Kufi i Bagadadu, prikazanja Muhamedu, opisana u Kuranu proučavala su se do detalja. Njihova namera bila je da izvedu hronološki red pojavljivanja reči u Kuranu, te su prebrojavali učestalost pojavljivanja određenih reči (teolizi su smatrali da učestalo pojavljivanje novijih reči u opisu prikazanja, ukazuje na to da se ono dogodilo kasnije). Došli su do zaključka da se neke reči pojavljuju češće od ostalih. Nastavili su da proučavaju zapise na nivou fonetike i zaključili da su neka slova mnogo frekventnija od ostalih i izdvojili su slova koja se pojavljuju, odnosno ne

[6] Nemački agenti uspevali su da fotografisanjem smanje celu stranu teksta na veličinu prečnika manjeg od milimetra, a zatim da je stave na tačku u rečenici nekog sasvim bezazlenog pisma. Kada su Američki agenti pronašli prvu mikrotačku i prekinuli komunikaciju, nemački agenti su šifrovali svoje poruke pre nego što bi ih pretvorili u mikrotačku.

pojavljuju zajedno (pronalazili su n-grame u jeziku). Ovo naizgled bezopasno proučavanje je dovelo do prvog velikog otkrića u kriptoanalizi - **analize učestanosti.** Nije poznato ko je prvi izneo da se razlike u učestanostima slova mogu iskoristiti u razbijanju šifre proste zamene, ali prvi poznati opis ove metode potiče od arapskog naučnika Al Kindija, koji je živeo u devetom veku [2].

Al Kindijeva tehnika, koja se danas naziva analiza učestanosti, pokazuje da nije potrebno proveravati milijarde mogućih ključeva, već je sadržaj šifrovane poruke moguće otkriti na osnovu učestanosti pojavljivanja slova u kriptogramu.

1.5.4 Polialfabetska šifra zamene

Jedini sistemi korišćeni za šifrovanje do XV veka bili su izvođeni iz šifre proste zamene. Razvoj analize učestanosti, najpre u arapskom svetu, a potom i u Evropi, ugrozio je njenu sigurnost.

Ideju za jačim i sigurnijim šiframa uveo je Leon Batista Alberti[7] (Leon Battista Alberti) u XV veku, mada se takva šifra nije pojavila sve do kraja XVI veka.

U to vreme, za šifrovanje jedne poruke šifrom zamene, bio je potreban samo jedan šifrovani alfabet. Alberti je predložio da se koriste dva ili više alfabeta, koji bi se tokom šifrovanja upotrebljavali naizmenično, zbunjujući potencijalnog kriptoanalitičara.

Primer 1.1

Primena Albertovog sistema šifrovanja za engleski alfabet [2]:

Početni alfabet	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
Šifrovan alfabet 1	F	Z	B	V	K	I	X	A	Y	M	E	P	L	S	D	H	J	O	R	G	N	Q	C	U	T	W
Šifrovan alfabet 2	G	O	X	B	F	W	T	H	Q	I	L	A	P	Z	J	D	E	S	V	Y	C	R	K	U	H	N

Prema Albertovoj zamisli, poruka se šifruje tako što se naizmenično koriste jedan, pa drugi alfabet. Na primer, da bi se šifrovala poruka **hello,** zamena za prvo slovo uzima se iz prvog alfabeta, za drugo slovo iz drugog, i tako redom. Tako se dobija šifrovan tekst **AFPAD**.

Primer 1.2

Šifrovanje poruke primenom Albertovog sistema:

poruka	H	e	l	l	o
šifrat	A	F	P	A	D

Osnovna prednost Albertovog sistema jeste to što se isto slovo iz početnog teksta ne pojavljuje uvek kao isto slovo u šifratu.

[7] Firentinski matematičar Leon Batista Alberti živeo je u 15. veku. Bio je pesnik, lingvista, arhitekta i filozof. U kriptografiji je poznat kao osnivač polialfabetskih kriptosistema, koje je u to vreme bilo nemoguće razbiti.

Iako je naišao na najznačajnije otkriće u kriptografiji za poslednjih hiljadu godina, polialfabetsku šifru zamene, Alberti nije uspeo da svoj koncept razvije u potpuno zaokružen sistem šifrovanja. Taj zadatak obavili su Johan Tritemijus (Johannes Trithemius), Đovani Porta (Giovanni Porta) i konačno ga je zaokružio Blez de Vižner[8] (**Blaise de Vigenère).** Iako su Alberti, Tritemijus i Porta dali značajan doprinos, šifra je nazvana po Vižneru, u čast čoveka koji ju je konačno uobličio.

Snaga Vižnerove šifre leži u tome što za alfabet veličine ***n***, ona ne koristi jedan ili dva, već ***n*** različitih šifrovanih alfabeta.

Primer 1.3

Vižnerov kvadrat za engleski alfabet:

Početni alfabet	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
Alfabet 1	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
Alfabet 2	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
Alfabet 3	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
Alfabet 4	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
Alfabet 5	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
Alfabet 6	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
Alfabet 7	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
Alfabet 8	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
Alfabet 9	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
Alfabet 10	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
Alfabet 11	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
Alfabet 12	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
Alfabet 13	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
Alfabet 14	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
Alfabet 15	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
Alfabet 16	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

[8] Blez de Vižner (1523-1596) bio je francuski diplomata i kriptograf, koji se nakon uspešne diplomatske karijere posvetio nauci i dao veliki doprinos u razvoju kriptografije.

Alfabet 17	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
Alfabet 18	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
Alfabet 19	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
Alfabet 20	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
Alfabet 21	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
Alfabet 22	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
Alfabet 23	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
Alfabet 24	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
Alfabet 25	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
Alfabet 26	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

Razmotrimo Vižnerov kvadrat na primeru engleskog alfabeta:

Prvi red Vižnerovog kvadrata predstavlja početni alfabet, a ostali redovi predstavljaju 26 šifrovanih alfabeta, od kojih je svaki pomeren za jedno mesto udesno. Tako je prvi šifrovani alfabet Cezarova šifra pomeranja za jedan, drugi - Cezarova šifra pomeranja za dva i tako redom. Vižnerova šifra podrazumeva da se za šifrovanje različitih slova iz poruke koriste različiti redovi Vižnerovog kvadrata (odnosno različiti šifrovani alfabeti).

Da bi odgonetnuo poruku, primalac mora da zna koji je red Vižnerovog kvadrata korišćen za svako slovo. To se postiže korišćenjem ključne reči, tj. ključa.

Primer 1.4

Šifrovanje teksta pomoću Vižnerovog kvadrata:

Poruka	t	h	i	s	i	s	s	o	m	e	m	e	s	s	a	g	e
Ključ	K	E	Y	K	E	Y	K	E	Y	K	E	Y	K	E	Y	K	E
Šifrat	D	L	G	C	M	Q	C	S	K	O	Q	C	C	W	Y	Q	I

U primeru šifrujemo poruku „this is some message“ ključem „KEY“. Najpre se ključ više puta ispisuje ispod poruke, sve dok se svakom slovu u poruci ne pridruži jedno slovo ključa. Zatim se šifrat formira na sledeći način: prvom slovu poruke **t** odgovara ključno slovo **K**, koje definiše jedan red u Vižnerovom kvadratu - red sa alfabetom 10, koji predstavlja alfabet u kojem ćemo pronaći zamenu za slovo **t**. U preseku kolone u čijem je zaglavlju slovo **t** i reda koji počinje sa **K** nalazi se slovo **D**. Prema tome, slovo **t** iz poruke u šifrovanom tekstu predstavljaće slovo **D**. Postupak se ponavlja za preostala slova poruke.

Velika prednost Vižnerove šifre jeste njena otpornost na analizu učestanosti. Pored toga, ova šifra ima ogroman broj ključeva. Pošiljalac i primalac se mogu dogovoriti da to bude bilo

koja reč iz rečnika, kombinacija reči ili izmišljena reč. Kriptoanalitičar ne može isprobati sve moguće ključeve, prosto jer je njihov broj gotovo neograničen.

Suprotno monoalfabetskim šiframa, Vižnerova šifra svrstava se u klasu polialfabetskih šifara, jer se u istoj poruci koristi više šifrovanih alfabeta. Upravo je polialfabetska priroda ono što Vižnerovu šifru čini snažnom i istovremeno mnogo složenijom.

Ovaj sistem je dobro funkcionisao i imao je svojstva sigurnog kriptosistema. Međutim, iako je šifra od početka delovala neranjivo na analizu učestanosti, ipak je njena upotreba u naredna tri veka bila retka, uglavnom zbog kompleksnosti korišćenja. Za razliku od Vižnerove šifre, monoalfabetske šifre bile su dovoljno dobre za većinu potreba: bile su brze, jednostavne za korišćenje i sigurne ukoliko napadač nema znanja iz kriptoanalize.

Ipak, vremenom je Vižnerova šifra dobila primenu u mnogo ozbiljnijim vojnim i državnim komunikacijama i važilo je mišljenje da ju je nemoguće razbiti.

Kako se pretpostavlja, Čarls Bebidž[9] (Charles Babbage) je prvi uspešno izveo kriptoanalizu jedne Vižnerove šifre, sredinom XIX-tog veka. Bebidžov rad na razbijanju Vižnerove šifre bio je u okviru engleske vojske, stoga tajan i njegov razultat objavljen je tek nekoliko godina kasnije [2]. Zato su sve zasluge za postupak razbijanja Vižnerove šifre pripale pruskom oficiru **Kaziskom**[10] (Friedrich Kasiski), koji je došao do istog otkrića, ali nekoliko godina nakon Bebidža.

U XX veku dolazi do izuma uređaja koji je predstavljao pravi izazov za kriptoanalitičare u Drugom svetskom ratu. Ovaj uređaj, nazvan **Enigma** znatno je ojačao poziciju kriptografije u vojnim i civilnim obaveštajnim službama. Tvorac Enigme bio je nemac Artur Šerbijus[11] (Arthur Scherbius) i svoj patent objavio je 1918. Međutim, on nije bio usamljen u ideji proučavanja principa rotorske mašine i paralelno se sa njegovim, u Evropi i Americi javljaju se slični izumi. 1925. godine počela je serijska proizvodnja Enigme u Nemačkoj, a u upotrebi je od 1926. godine.

Grupa poljskih naučnika, od kojih najistaknutiji Marijan Rejevski (Marian Rejewski), uspela je 1933. godine da pokaže da se Enigma može razbiti, napravivši mašinu nazvanu Bomba, koja je imala brži proces šifrovanja od Enigme. Ali tek 1939. Alan Tjuring[12] (Alan Turing) i

---

[9] Čarls Bebidž (1791-1871) bio je engleski matematičar, inženjer, izumitelj, poznat po tome što je nacrtao šemu modernog računara. Izumeo je mehaničku mašinu „Diferencijalna mašina“, koja je trebala da automatizuje proces izračunavanja matematičkih tablica, izvodeći osnovne računske operacije. Ovaj projekat nikada nije završio. Radio je na ideji analitičke mašine- preteče današnjih računara.

[10] Fridrih Kaziski bio je Pruski oficir, arheolog i kriptoanalitičar. 1863. u knjizi “Tajno pisanje i umetnost dešifrovanja” prvi put su iznete motode napada na polialfabetske šifre zamene, a posebno na Vižnerovu šifru.

[11] Artur Šerbijus (1878-1929) bio je nemački elektroinženjer i pre Enigme patentirao je brojne verzije uređaje za šifrovanje: Model A, B, C. Enigma je bila u upotrebi od 1926-te i odigrala veliki značaj u obaveštajnim vojnim poslovima tokom II svetskog rata.

[12] Alan Tjuring (1912-1954) bio je britanski matematičar, programer i kriptoanalitičar. Smatra se ocem računarstva. Postavio je formalnu definiciju algoritma, poznatu kao Tjuringova mašina, koji je i danas osnova teorije računara.

Gordon Velhman[13] (Gordon Welchman) razvili su englesku verziju Bombe, nezavisno od poljske verzije mašine, kojom su uspeli da dešifruju nemačke poruke. Time je Enigma i u praksi razbijena.

1.6 Savremena kriptografija

Moderna kriptografija počinje sa Klodom Šenonom, koji je uveo OTP algoritme (1.4) u kriptografiju i dokazao da je jedini siguran kriptosistem onaj kod kojeg je ključ slučajan, dugačak bar koliko i poruka i koristi se ne više od jednom. Šenonova teorija je da šifrat ne sme dovesti napadača do sadržaja polazne poruke. To znači da informacija, sadržana u poruci mora biti savršeno preneta kroz šifrovanu poruku. Ovo je dovelo do novih standarda i deja u kriptografiji, čime se Šenon smatra tvorcem moderne kriptografije.

Koncept javnog ključa pojavio se 1975. Ovi sistemi su zamišljeni tako da je ključ kojim se dešifruje tajan (poznat samo osobi do koje sadržaj poruke treba da stigne), a ključ kojim se šifruje je javan i poznat svima. Međutim, njegovi autori nisu uspeli da pronađu metodu koja bi funkcionisala u sistemima sa javnim ključem. Funkcija koja bi se lako mogla izračunati u jednom smeru, ali teško u suprotnom nazvana je jednosmerna funkcija. Sedamdesetih godina XX veka javile su se ideje za ovu funkciju, ali kao strogo poverljivi radovi nisu objavljeni.

Postoji nekoliko standardnih algoritama koji se danas koriste.

RSA[14] sistem je najpoznatiji asimetrični šifarski sistem. Predstavljen je 1977. Sigurnost RSA zasniva se na faktorizaciji velikih brojeva. Do sada njegova sigurnost nije ni dokazana ni opovrgnuta [4].

DES[15] sistem korišćen je kao standard u kriptografiji preko dvadeset godina. DES sistem je blokovski sistem. 1976. godine ovaj algoritam je usvojen kao standard za šifrovanje američkih nevojnih vladinih komunikacija. 1998. godine demonstrirana je nesigurnost ovog sistema korišćenjem računara za nalaženje ključa.

AES[16] je rezultat trogodišnjeg javnog konkursa od strane Nacionalnog instituta za standarde i tehnologiju (engl. National Institute Of Standards and Technology), koji je započeo 1997. 1999. izabrano je 5 kandidata, a **Rijndael**, šifra koju su razvili Belgijanci Džoan Demen (Joan Daemen) i Vinsent Rijmen (Vincent Rijmen) proglašena je za novi standard. Ovo je blokovska šifra, sa fiksnom dužinom bloka od 128 bitova.

[13] Gordon Velhman (1906-1985) bio je britanski matematičar, osnivač i predvodnik tzv. odreda „barake šest“, koji je radio na razbijanju nemačke Enigme tokom II svetskog rata.

[14] naziv RSAje nastao od inicijala izumitelja ovog algoritma: Ron **R**ivest, Adi **S**hamir and Len **A**dleman.

[15] Data Encryption Standard

[16] Advanced Encryption Standard

2 Kriptosistemi

2.1 Formalna definicija kriptosistema

Definicija 2.1: Kriptosistem je petorka (***P,C,K,ε,D***), za koju su ispunjeni sledeći uslovi:

- ***P*** je konačan skup mogućih poruka
- ***C*** je konačan skup mogućih šifrata
- ***K*** je konačan skup mogućih ključeva

Za svako $k \in K$ postoji pravilo šifrovanja $e_k \in \varepsilon$ i odgovarajuće pravilo dešifrovanja $d_k \in D$. Svako $e_k: P \rightarrow C$ i $d_k: C \rightarrow P$ su funkcije, takve da važi $d_k(e_k(p))=p$, za svaku poruku $p \in P$.

Kako se i poruka i šifrat zapisuju nekim konačnim skupom simbola - azbukom veličine n, u daljem tekstu koristi se pojam kongruencije po modulu[17] broja n i skup $\boldsymbol{Z_n}$[18] .

Skup Z_n predstavlja celobrojne vrednosti koje se jednoznačno dodeljuju svakom slovu primenjene azbuke. On je zatvoren za operacije sabiranja i oduzimanja, stoga se one u skupu Z_n odvijaju po modulu n.

Na primer, u slučaju engleskog alfabeta, posmatra se skup Z_{26}, u slučaju srpske azbuke (ćirilično pismo)[19] skup Z_{30}, a u slučaju srpske abecede (latinično pismo)[20] skup Z_{27}.

Tabela 2.1

TABELA VREDNOSTI ZA ENGLESKI ALFABET													
slovo	A	B	C	D	E	F	G	H	I	J	K	L	M
kod	0	1	2	3	4	5	6	7	8	9	10	11	12
slovo	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
kod	13	14	15	16	17	18	19	20	21	22	23	24	25

[17] Dva cela broja a i b su kongruentna po modulu m, ako je njihova razlika a-b sadržalac broja m.

[18] Z_n je skup čiji su elementi $0 \ldots n-1$

[19] Srpska azbuka ima 30 slova

[20] Srpski jezik za latinični zapis koristi 27 karaktera, jer se slova abecede: *nj*, *lj* i *dž* zapisuju sa po dva karaktera.

Tabela 2.2

TABELA VREDNOSTI ZA SRPSKU AZBUKU															
slovo	А	Б	В	Г	Д	Ђ	Е	Ж	З	И	Ј	К	Л	Љ	М
kod	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
slovo	Н	Њ	О	П	Р	С	Т	Ћ	У	Ф	Х	Ц	Ч	Џ	Ш
kod	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29

Tabela 2.3

TABELA VREDNOSTI ZA SRPSKU ABECEDU														
slovo	A	B	C	Č	Ć	D	Đ	E	F	G	H	I	J	K	L
kod	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
slovo	M	N	O	P	R	S	Š	T	U	V	Z	Ž			
kod	15	16	17	18	19	20	21	22	23	24	25	26			

2.1.1 Šifra pomeranja

Šifra pomeranja (engl. *shift cipher*) je monoalfabetska šifra u kojoj je svako slovo poruke zamenjeno odgovarajućim slovom šifrata, „pomerenim“za k mesta udesno.

Definicija 2.2: Neka je $P=C=K=Z_n$. Za $0 \leq k \leq n-1$ definiše se

$$e_k(p) = (p+k) \bmod n$$

$$d_k(c) = (c-k) \bmod n$$

gde $p,c \in Z_n$ [5].

Postupak šifrovanja poruke pomeranjem za k mesta, na azbuci veličine n, izvodi se u koracima:

- izabere se ključ k
- poruka se konvertuje u niz nenegativnih celih brojeva, koristeći jednu od odgovarajućih tabela za taj jezik (npr. Tabela 2.1, 2.2, 2.3)
- vrednost ključa ispisuje se ispod svake vrednosti slova poruke
- svaki od brojeva u nizu sabira se sa vrednošću ključa, po modulu n
- dobijeni niz brojeva se konvertuje nazad u tekst, zamenom svakog broja odgovarajućim slovom iz tabele

Primer 2.1

Šifrovanje poruke pomeranjem za k=2, na srpskom latiničnom pismu:

poruka	s	k	r	i	v	e	n	a	p	o	r	u	k	a
poruka_kod	20	13	19	11	24	7	16	0	18	17	19	23	13	0
Ključ	C	C	C	C	C	C	C	C	C	C	C	C	C	C
ključ_kod	2	2	2	2	2	2	2	2	2	2	2	2	2	2
šifrat_kod	22	15	21	13	26	9	18	2	20	19	21	25	15	2
Šifrat	T	M	Š	K	Ž	G	P	C	S	R	Š	Z	M	C

Dešifrovanje se radi na sličan način, sa razlikom što se od vrednosti slova otvorenog teksta oduzima vrednost ključa (po modulu n).

Primer 2.2

Dešifrovanja šifrata pomeranjem za k=2, na srpskom latiničnom pismu:

Šifrat	T	M	Š	K	Ž	G	P	C	S	R	Š	Z	M	C
šifrat_kod	22	15	21	13	26	9	18	2	20	19	21	25	15	2
Ključ	C	C	C	C	C	C	C	C	C	C	C	C	C	C
ključ_kod	2	2	2	2	2	2	2	2	2	2	2	2	2	2
poruka_kod	20	13	19	11	24	7	16	0	18	17	19	23	13	0
poruka	s	k	r	i	v	e	n	a	p	o	R	u	k	a

Primetimo da za azbuku veličine n, šifra pomeranja ima samo n različitih ključeva. Dobar kriptoanalitičar može sa lakoćom rekonstruisati poruku samo na osnovu šifrata.

2.1.2 Vižnerova šifra

Vižnerova šifra je polialfabetska šifra zamene. To znači da jednom slovu otvorenog teksta može odgovarati više od jednog slova šifrata. Ovaj kriptosistem je pre svega OTP sa periodičnim ključem. To znači da se ključ ponavlja kroz otvoreni tekst sve dok se ne dođe do kraja teksta.

Formalna definicija Vižnerove šifre za proizvoljnu azbuku veličine n glasi:

Definicija 2.3: Neka je $P=C=K=(Z_n)^m$. Za ključ $k=(k_1, k_2, ..., k_m)$, dužine m definiše se:

$$e_k(p_1, p_2,...p_m)=(p_1+ k_1, p_2+ k_2, ..., p_m+ k_m) \text{ i}$$

$$d_k(c_1, c_2,...c_m)=(c_1- k_1, c_2- k_2, ..., c_m- k_m)$$

gde se sve operacije izvode u skupu Z_n, dakle po modulu n [5].

Analogno se definiše Vižnerov kriptosistem za engleski alfabet i srpska pisma latinicu i ćirilicu, pri čemu se sve operacije izvode redom u skupu Z_{26}, Z_{27}, odnosno Z_{30}.

Praktično, neka je poruka koju šifrujemo dužine d, korišćena azbuka veličine n i dužina ključa kojim se poruka šifruje m.

Tada šifrovanje Vižnerovom šifrom podrazumeva sledeće korake: najpre se i ključ i otvorena poruka konvertuju u niz celih brojeva korišćenjem jedne od odgovarajućih tabela (npr. Tabela 2.1, 2.2, 2.3). Zatim se otvoreni tekst (dužine d) deli na $[d/m]$ blokova dužine m i svakom bloku se dodaje dati ključ, po modulu dužine azbuke n. Ako dužina otvorenog teksta nije delilac broja m, tj. ceo ključ se ne može ponoviti ceo broj puta u poruci, samo deo ključa se koristi za šifrovanje poslednjeg bloka poruke.

Najzad se dobijeni niz brojeva konvertuje nazad u tekst, ponovo zamenom vrednosti odgovarajućim slovima iz tabela.

Primer 2.3

Šifrovanje poruke Vižnerovom šifrom na srpskom ćiriličnom pismu:

poruka	с	к	р	и	В	е	н	а	п	о	Р	у	к	а
poruka _kod	20	11	19	9	2	6	15	0	18	17	19	23	11	0
ključ	К	Љ	У	Ч	К	Љ	У	Ч	К	Љ	У	Ч	К	Љ
ključ_kod	11	13	23	27	11	13	23	27	11	13	23	27	11	13
šifrat_kod	1	24	12	6	13	19	8	27	29	0	12	20	22	13
šifrat	Б	Ф	Л	Е	Љ	Р	З	Ч	Ш	А	Л	С	Ћ	Љ

Dešifrovanje šifrata podrazumeva sledeće korake: najpre se šifrat konvertuje u niz brojeva, korišćenjem jedne od odgovarajućih tabela. Zatim se, kao kod šifrovanja, šifrat podeli na blokove brojeva, dužine m. Od svakog bloka dužine m se oduzme vrednost ključa, dok se u slučaju poslednjeg (ukoliko dužina poruke d nije celobrojni delilac broja m) koristi samo deo ključa (dužine d mod m). Najzad, dobijeni niz brojeva se konvertuje u slova i dobija se polazna poruka.

Primer 2.4

Dešifrovanje Vižnerovog šifrata na srpskom ćiriličnom pismu:

šifrat	Б	Ф	Л	Е	Љ	Р	З	Ч	Ш	А	Л	С	Ћ	Љ
šifrat_kod	1	24	12	6	13	19	8	27	29	0	12	20	22	13
ključ	К	Љ	У	Ч	К	Љ	У	Ч	К	Љ	У	Ч	К	Љ
ključ_kod	11	13	23	27	11	13	23	27	11	13	23	27	11	13
poruka_kod	20	11	19	9	2	6	15	0	18	17	19	23	11	0
poruka	с	к	р	и	В	е	н	а	п	о	р	у	к	а

Na azbuci veličine n, broj svih mogućih ključeva dužine m iznosi n^m [21]. Tako, u slučaju engleskog alfabeta, broj svih mogućih ključeva dužine m iznosi 26^m. Slično, u slučaju srpske latinice broj svih mogučih ključeva je 27^m, a u slučaju ćirilice 30^m.

[21] Iz skupa od n elemenata, niz dužine m se može izabrati na n^m načina (m varijacija n-točlanog skupa).

3 Kriptoanaliza

3.1 Osnovne tehnike kriptoanalize

Osnovni cilj kriptoanalize je da otkrije ključ ili da otkrije sadržaj polazne poruke, bez znanja ključa korišćenog za šifrovanje.

Pokušaj kriptoanalize naziva se napad, a uspešan napad dekriptiranje. Kriptoanalitičar se, u ovom smislu, naziva i napadač.

Osnovna pretpostavka pri napadu je da napadač poznaje algoritam šifrovanja. Ova pretpostavka naziva se ***Kerkhofov princip*** (Auguste Kerckhoffs)[22] i glasi [1]:

Kriptosistem bi trebalo da bude siguran i ako je javno sve sem ključa.

Ova pretpostavka je razumna u praksi, jer ako sigurnost sistema zavisi od tajnosti algoritma, onda se takav sistem ne može smatrati sigurnim.

Osnovni modeli napada na kriptosisteme su [4]:

- Napad na osnovu šifrata

Napadač poseduje šifrate više poruka, šifrovane istim algoritmom. Dakle, dato je $c_1=e_k(p_1)$, $c_2=e_k(p_2)$,... $c_n=e_k(p_n)$, a treba odrediti ključ k ili p_1, p_2,... p_n. Svaki sistem koji je osetljiv na ovaj napad smatra se potpuno nesigurnim sistemom.

- Napad na osnovu poznatog otvorenog teksta

Napadač poseduje tekst poruke (otvorenog teksta) i odgovarajući šifrat. Dakle, dato je p i $c=e_k(p)$. Napadač treba da odredi ključ k ili da otkrije algoritam koji daje ispravnu poruku proizvoljno izabranog šifrata.

- Napad na osnovu izabranog otvorenog teksta

Napadač može da izabere poruke koje će biti šifrovane tim kriptosistemom. Dakle napadač zna parove: p_1, $c_1=e_k(p_1)$; p_2, $c_2=e_k(p_2)$,..., p_n, $c_n=e_k(p_n)$, pri čemu sam bira p_1,...p_n.

- Napad na osnovu izabranog šifrata

Napadač može da bira šifrate i pribavi njijma odgovarajuće otvorene tekstove. Dakle, napadač zna parove c_1, $p_1=d_k(c_1)$, c_2, $p_2=d_k(c_2)$,..., c_n, $p_n=d_k(c_n)$.

[22] Holanđanin Avgust Kerkhof (1835-1903) bio je lingvista i kriptograf. Poznat je po članku *Vojna kriptografija* u kome je objavio jednu od osnovnih kriptografskih aksioma, poznatu kao Kerkofov princip.

3.2 Statistika jezika

Arapi se smatraju osnivačima kriptoanalize. Oni su utemeljili princip razbijanja monoalfabetskih šifara napadom samo na šifrat. Taj metod zasniva se na poznavanju statističkih osobina jezika, na kome je poruka pisana. Deo iz Al Kindijeve knjige „Manuskript o dešifrovanju kriptografskih poruka“, ukratko objašnjava recept za razbijanje šifara proste zamene analizom učestanosti [2]:

Jedan od načina da odgonetnemo šifrovanu poruku, ako znamo na kom je jeziku, jeste da pronađemo neki tekst na tom jeziku, dugačak otprilike jednu stranu, i da prebrojimo koliko se puta koje slovo pojavljuje. Slovo koje se najčešće pojavljuje nazivamo „prvim“, ono koje se pojavljuje malo ređe nazivamo „drugim“ tako dalje, sve dok ne poslažemo sva slova iz uzorka teksta.

Zatim u šifrovanom tekstu koji želimo da odgonetnemo takođe klasifikujemo simbole. Pronalazimo simbol koji se najčešće pojavljuje i zamenjujemo ga „prvim“ slovom iz uzorka teksta, sledeći najčešći simbol „drugim“ i tako dalje, sve dok ne zamenimo sve simbole iz kriptograma.

Razumljivo, u praksi se ovaj princip ne može slepo primeniti, jer je potrebno mnogo više od jedne strane teksta da bi se izveo zaključak o učestanosti pojavljivanja slova nekog jezika.

3.2.1 Analiza učestanosti pojavljivanja slova

U kriptologiji, **analiza učestanosti** (engl. *frequency analysis*) je proučavanje učestanosti pojavljivanja slova ili grupa slova u šifratu. Ovaj metod se koristi u razbijanju klasičnih šifara.

Analiza učestanosti zasniva se na činjenici da je broj pojavljivanja određenog slova ili grupe slova u proizvoljnom tekstu (pisanom na nekom prirodnom jeziku) različita. Tako je na primer, prema statistici, u tekstovima na engleskom jeziku, slovo „E“ najčešće slovo.

Ne postoji, međutim, tačna raspodela učestanosti pojavljivanja slova u tekstovima prirodnih jezika. Analize pokazuju da učestanosti slova veoma variraju u zavisnosti od autora tekstova i tema. Zato je jedini način preciznog određivanja prosečne učestanosti pojavljivanja slova nekog prirodnog jezika u obradi velike baze tekstova pisanih na tom prirodnom jeziku, što uz današnje hardverske mogućnosti ne predstavlja veliki posao. Veći problem je odrediti koji tekstovi su reprezentativni za taj jezik.

Uvedimo oznake:

d – dužina teksta na nekom prirodnom jeziku

f_i - broj pojavljivanja slova i u tekstu dužine d

Učestanost pojavljivanja slova i u tekstu dužine d, računa se formulom

$$p_i = \frac{f_i}{d} \tag{1}$$

Prosečna učestanost $\overline{\mu}_{[i]}$ pojavljivanja slova i u nekom prirodnom jeziku može se izračunati formulom

$$\overline{\mu}_{[i]} = \frac{\sum_{k=1}^{m} p_{k[i]}}{m} \tag{2}$$

gde je m - broj uzoraka iz baze tekstova, a $p_{k[i]}$ učestanost slova i u tekstu k.

Standardna devijacija $s_{[i]}$, za svako slovo jezika i računa se prema formuli [6]

$$s_{[i]} = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(p_{k[i]} - \overline{\mu}_{[i]})^2} \tag{3}$$

Prosečnu učestanost pojavljivanja $\overline{\mu}_{[i]}$, izračunatu na osnovu (2), treba zaokružiti na broj decimalnih mesta, koji odgovara minimalnom redu veličine standardne devijacije uzorka.

Dakle, za razbijanje klasičnih kriptosistema, napadač mora da poznaje statističke osobine jezika u kome je šifrovana poruka pisana. Kako se u radu koriste poruke pisane na engleskom i srpskom jezikom (ćiriličnim i latiničnim pismom), to će za dalju analizu biti potrebne statistike ovih jezika.

Ekperiment 1:

Da bi se odredila verovatnoću pojavljivanja slova prirodnog jezika, potrebna je baza tekstova napisanih na tom jeziku. U eksperimentima se korist dvanaest tekstova različite tematike, dužine 100 000 slova, za svaki od jezika.

Prosečne učestanosti pojavljivanja slova računate su na osnovu (2), a ocena greške na osnovu (3).

3.2.2 Statistika engleskog jezika

Tabela 3.1

Tabela učestanosti pojavljivanja slova u tekstovima na engleskom jeziku, standardna devijacija i njen red veličine, na osnovu rezultata Eksperimenta 1:

slovo	učestanost pojavljivanja	standardna devijacija	red veličine	slovo	učestanost pojavljivanja	standardna devijacija	red veličine
A	0.08169333	0.00319474	10^{-3}	N	0.06958083	0.00350263	10^{-3}
B	0.01616083	0.00174612	10^{-3}	O	0.07495666	0.00428945	10^{-3}
C	0.02412333	0.00450282	10^{-3}	P	0.01750083	0.00216160	10^{-3}
D	0.04583250	0.00523104	10^{-3}	Q	0.00095378	0.00028651	10^{-4}
E	0.12418751	0.00507378	10^{-3}	R	0.05786757	0.00413469	10^{-3}
F	0.02245166	0.00260298	10^{-3}	S	0.06205083	0.00304212	10^{-3}
G	0.02203666	0.00326925	10^{-3}	T	0.09264583	0.00452281	10^{-3}

H	0.06451753	0.00669752	10^{-3}	U	0.02830916	0.00212901	10^{-3}
I	0.06663083	0.00375137	10^{-3}	V	0.00912166	0.00112065	10^{-3}
J	0.00114754	0.00048264	10^{-4}	W	0.02294416	0.00341597	10^{-3}
K	0.00866833	0.00284978	10^{-3}	X	0.00178567	0.000537062	10^{-4}
L	0.04118509	0.00365313	10^{-3}	Y	0.01909545	0.00229166	10^{-3}
M	0.02383252	0.00201747	10^{-3}	Z	0.00072867	0.00042742	10^{-4}

Minimalni red veličine standardne devijacije uzorka iznosi 10^{-4}, pa je zaokruživanje dobijenih učestanosti pojavljivanja slova izvršeno na 4 decimalna mesta.

Tabela 3.2

Tabela učestanosti pojavljivanja slova u tekstovima na engleskom jeziku, na osnovu rezultata Eksperimenta 1:

slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja
A	0.0817	H	0.0645	O	0.0749	V	0.0091
B	0.0161	I	0.0666	P	0.0175	W	0.0231
C	0.0242	J	0.0012	Q	0.0010	X	0.0018
D	0.0456	K	0.0086	R	0.0580	Y	0.0192
E	0.1239	L	0.0413	S	0.0620	Z	0.0007
F	0.0225	M	0.0238	T	0.0928		
G	0.0220	N	0.0696	U	0.0282		

Grafik 3.1

PRIMENJENI JEZIK [english]

DUŽINA PISMA [26]

DUŽINA TEKSTA [100000]

BROJ UZORAKA [12]

3.2.3 Statistika srpskog jezika koristeći latinično pismo

Na sličan način dobijamo tabelu učestanosti pojavljivanja slova u srpskom jeziku na latiničnom pismu:

Tabela 3.3

Tabela učestanosti pojavljivanja slova u tekstovima na srpskom jeziku, na osnovu rezultata Eksperimenta 1:

slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja
A	0.1202	E	0.0885	L	0.0336	Š	0.0130
B	0.0156	F	0.0013	M	0.0356	T	0.0438
C	0.0068	G	0.0179	N	0.0618	U	0.0425
Č	0.0066	H	0.0065	O	0.0992	V	0.0377
Ć	0.0111	I	0.0937	P	0.0285	Z	0.0172
D	0.0389	J	0.0427	R	0.0452	Ž	0.0060
Đ	0.0022	K	0.0344	S	0.0494		

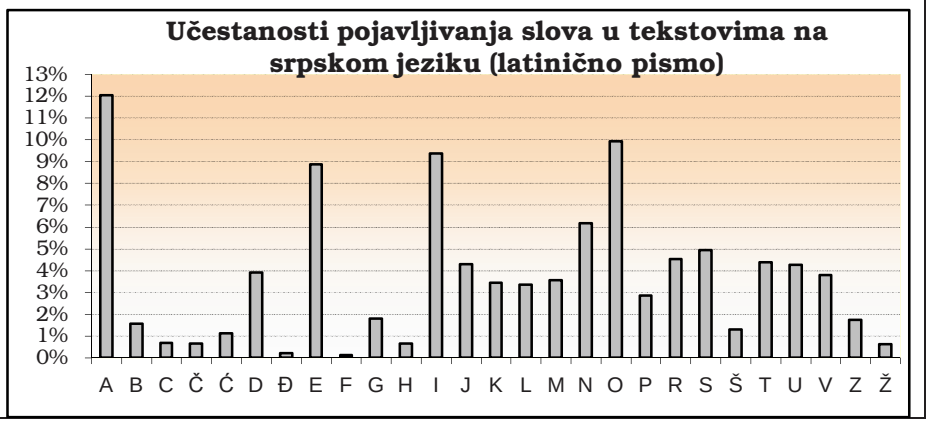
Grafik 3.2

PRIMENJENI JEZIK [srpski (latinica)]

DUŽINA PISMA [27]

DUŽINA TEKSTA [100000]

BROJ UZORAKA [12]



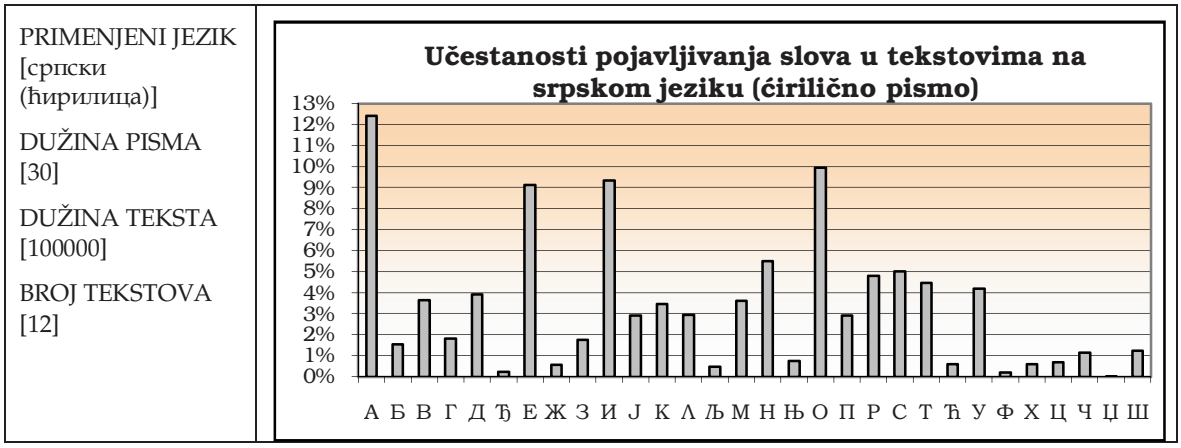
3.2.4 Statistika srpskog jezika koristeći ćirilično pismo

Tabela 3.4

Tabela učestanosti pojavljivanja slova u tekstovima na srpskom jeziku, na osnovu rezultata Eksperimenta 1:

slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja	slovo	učestanost pojavljivanja
А	0.1240	З	0.0177	Њ	0.0076	Ф	0.0020
Б	0.0156	И	0.0935	О	0.0996	Х	0.0061
В	0.0366	Ј	0.0291	П	0.0292	Ц	0.0069
Г	0.0181	К	0.0347	Р	0.0482	Ч	0.0115
Д	0.0393	Л	0.0294	С	0.0502	Џ	0.0003
Ђ	0.0023	Љ	0.0048	Т	0.0447	Ш	0.0125
Е	0.0913	М	0.0362	Ћ	0.0061		
Ж	0.0056	Н	0.0551	У	0.0421		

Grafik 3.3



3.3 Kriptoanaliza šifre pomeranja

Šifru pomeranja moguće je razbiti koristeći napad na osnovu šifrata (poznavajući statistiku jezika u kome je poruka pisana).

Šifra proste zamene (specijalno, šifra pomeranja) svako slovo poruke zamenjuje tačno jednim slovom šifrata. Iako se na primer najfrekventnije slovo engleskog jezika 'E' u šifratu

više neće pojavljivati sa verovatnoćom 0,1239 (jer će biti zamenjeno nekim drugim slovom), slovo koje zamenjuje slovo 'Е' će se pojavljivati sa tom verovatnoćom. Tako će skup verovatnoća pojavljivanja slova jezika ostati isti i u šifratu.

Dakle, za pokušaj razbijanja šifre proste zamene, mogu se odrediti učestanosti pojavljivanja različitih slova u šifratu i dobijene vrednosti uporediti sa vrednostima iz tabela 3.2, 3.3, 3.4. Zamenom slova šifrata odgovarajućim slovima, na osnovu tabela učestanosti pojavljivanja slova za taj jezik, teoretski se može dobiti polazna poruka.

Primer 3.1

Tekst *P*: „сигурност система је у тајности кључа" šifrovan je šifrom pomeranja za ključ „К", na srpskom jeziku (ćirilično pismo).

Dobijeni šifrat *C* je „БСМДАЦЏБВБСБВОХКТОДВКТЦЏБВСЋФДЗК". Tablica učestanosti slova azbuke šifrata i poruke pokazuje da je skup verovatnoća slova šifrata odgovara skupu verovatnoća polazne poruke (npr. slovo „Б" u šifratu odgovara slovu „с" u poruci, a vrednosti učestanosti njihovog pojavljivanja su jednake).

Tabela učestanosti pojavljivanja slova u poruci i šifratu

slovo	učestanost u šifratu	učestanost u poruci	slovo	učestanost u šifratu	učestanost u poruci	slovo	učestanost u šifratu	učestanost u poruci
А	0.0313	0.0938	Ј	0.0000	0.0625	С	0.0938	0.1563
Б	0.1563	0.0000	К	0.0938	0.0313	Т	0.0625	0.1250
В	0.1250	0.0000	Л	0.0000	0.0000	Ћ	0.0313	0.0000
Г	0.0000	0.0313	Љ	0.0000	0.0313	У	0.0000	0.0938
Д	0.0938	0.0000	М	0.0313	0.0313	Ф	0.0313	0.0000
Ђ	0.0000	0.0000	Н	0.0000	0.0625	Х	0.0313	0.0000
Е	0.0000	0.0625	Њ	0.0000	0.0000	Ц	0.0625	0.0000
Ж	0.0000	0.0000	О	0.0625	0.0625	Ч	0.0000	0.0313
З	0.0313	0.0000	П	0.0000	0.0000	Џ	0.0625	0.0000
И	0.0000	0.0938	Р	0.0000	0.0313	Ш	0.0000	0.0000

Učestanosti slova u tekstovima na srpskom jeziku		Učestanosti slova u šifratu	
А	0.1240	Б	0.1563
О	0.0996	В	0.1250
И	0.0935	Д	0.0938
Е	0.0913	К	0.0938
Н	0.0551	С	0.0938
С	0.0502	О	0.0625
Р	0.0482	Т	0.0625

Prva i druga kolona prikazane tabele (levo) sadrže slova azbuke i njihove učestanosti pojavljivanja (na osnovu vrednosti iz Tabele 3.4), sortirane prema vrednostima učestanosti u opadajućem poretku. Treća i četvrta kolona sadrže slova šifrata i njihove učestanosti pojavljivanja, takođe sortirane u opadajućem poretku prema vrednosti učestanosti.

Na osnovu prikazane tabele može se primetiti da se Al Kindijev recept za kriptoanalizu ne može "slepo"

Т	0.0447	Ц	0.0625
У	0.0421	Џ	0.0625
Д	0.0393	А	0.0313
В	0.0366	З	0.0313
М	0.0362	М	0.0313
К	0.0347	Ћ	0.0313
Л	0.0294	Ф	0.0313
П	0.0292	Х	0.0313
Ј	0.0291	Г	0.0000
Г	0.0181	Ђ	0.0000
З	0.0177	Е	0.0000
Б	0.0156	Ж	0.0000
Ш	0.0125	И	0.0000
Ч	0.0115	Ј	0.0000
Њ	0.0076	Л	0.0000
Ц	0.0069	Љ	0.0000
Х	0.0061	Н	0.0000
Ћ	0.0061	Њ	0.0000
Ж	0.0056	П	0.0000
Љ	0.0048	Р	0.0000
Ђ	0.0023	У	0.0000
Ф	0.0020	Ч	0.0000
Џ	0.0003	Ш	0.0000

primeniti. Naime, očigledno je da najučestanijem slovu šifrata (dužine 32 slova) ne mora odgovarati najučestanije slovo jezika, računato na uzorku tekstova dužine nekoliko miliona slova. Šifrovani tekst suviše je kratak da bi samo zamena vrednosti iz tabele učestanosti pojavljivanja slova dala traženo rešenje.

Neka su sva slova otvorenog teksta *P* označena u prvoj i drugoj koloni tabele. Može se primetiti da najučestanijem slovu šifrata „Б“ odgovara tek šesto slovo po vrednosti u prvoj koloni.

Kriptoanalitičar bi nastavio analizu, razmatrajući prvih nekoliko slova koja se češće javljaju u šifratu, kao zamenu za najučestanija slova srpke ćirilice “А”, “О”,“ *И*”,”Е”,… Zatim bi analiza morala da krene u drugom pravcu, na razmatranje koliko se često ova slova pojavljuju pored drugih slova. U tom slučaju, potrebna je i analiza bigrama[23], pa čak i trigrama korišćenog jezika, čime se analiza dodatno komplikuje.

Analizom učestanosti, u ovom primeru, teško je doći do teksta poruke, napadom samo na šifrat.

Međutim, u opštem slučaju, kod dovoljno dugih šifrata, analiza učestanosti daje odlične rezultate.

3.4 Napad na Vižnerovu šifru

Vižnerova šifra je vrsta polialfabetske šifre, stoga ona menja skup verovatnoća slova u šifratu u odnosu na skup verovatnoća slova poruke. Ova činjenica otežava analizu učestanosti, pa je za njeno razbijanje potrebno uložiti veći napor.

Kriptoanaliza Vižnerove šifre izvodi se u dve faze:

- određivanje dužine ključa *m*

[23] Bigrami su parovi slova, trigrami – trojke slova

- određivanje ključa $k=(k_1, ..., k_m)$

Poznavanje dužine ključa svodi razbijanje Vižnerove šifre na problem dekriptiranja monoalfabetske šifre pomeranja.

Posmatrajmo šifrat $x=x_0x_{2...}x_{d-1}$ šifrovan ključem k dužine *m*.

Ako je dužina ključa *m*, tada su slova

$x_0,$	$x_m,$	$x_{2m}, \ldots ,$	šifrovana prvim slovom ključa,
$x_1,$	$x_{m+1},$	$x_{2m+1}, \ldots ,$	šifrovana drugim slovom ključa
	.		
	.		
	.		
$X_{m-1,}$	$x_{2m-1},$	$x_{3m-1}, \ldots ,$	šifrovana m-tim slovom ključa

Dakle, ako se šifrat x razbije na *m* isečaka:

$$\begin{array}{llllll} C_1 = & x_0 & x_m & x_{2m} & \ldots \\ C_2 = & x_1 & x_{m+1} & x_{2m+1} & \ldots \\ & & . & & \\ & & . & & \\ & & . & & \\ C_m = & x_{m-1} & x_{2m-1} & x_{3m-1} & \ldots \end{array} \qquad (4)$$

svaki od isečaka C_i, $i = 1..m$ šifrovan je i - tim slovom ključa. To znači da se problem nalaženja Vižnerovog ključa svodi na nalaženje *m* ključeva šifre pomeranja za svaki od isečka C_i. Ovo proističe iz činjenice da je svaki podtekst C_i šifrovan monoalfabetskom šifrom pomeranja za *i*- to slovo ključa.

3.4.1 Određivanje dužine ključa

Da bi se odredila dužina ključa, koristi se osnovna osobina Vižnerove šifre***: ključ se ponavlja u šifratu***.

Poznate su dve tehnike za određivanje dužine ključa:

- metod Kaziskog

- Fridmanov[24] napad

3.4.1.1 Metod Kaziskog

Metod Kaziskog omogućava nalaženje dužine ključa kojim je poruka šifrovana na osnovu činjenice da će se isti delovi teksta poruke šifrovati u iste delove šifrata ako se nalaze na udaljenosti δ, gde je a $\delta = k \cdot m$, za dužinu ključa m.

Algoritam koji opisuje metod Kaziskog je:

- nalaženje parova identičnih segmenata dužine najmanje tri
- registrovanje rastojanja između dva identična segmenta Δ_1, Δ_2,...
- određivanje dužine ključa m kao zajedničkog delioca brojeva Δ_1, Δ_2,...

Ovaj algoritam je prilično koristan i pouzdan kada se primenjuje na dužim šifratima sa kraćim ključem, jer je verovatnoća slučajnog ponavljanja tri ili četiri istih slova teksta veoma mala.

Primer 3.2 dužina teksta=13 dužina ključa=4

pozicija	0	1	2	3	4	5	6	7	8	9	10	11	12
otvoreni tekst	t	o	b	e	o	R	n	o	t	t	o	b	e
ključ	m	a	n	m	a	N	m	a	n	m	a	n	m
šifrat	f	o	o	q	o	I	z	o	g	f	o	o	q

U šifratu se dva puta pojavljuje segment "fooq", na poziciji 0 i 9. Prema algoritmu očekuje se da broj slova između početaka ponavljenog segmenta bude umnožak dužine ključa. Prema tome, u datom primeru očekuje se da dužina ključa bude 3 ili 9, pošto su to jedini činioci broja 9.

Jasno je da se u šifratu mogu pojaviti ponavljanja na rastojanju koje nije umnožak broja m. Delovi same poruke se ponavljaju, što dovodi i do ponavljanja delova u šifratu, bez obzira na odnos dužine poruke i ključa.

Međutim, ovaj metod u opštem slučaju ne daje rezultate kod tekstova šifrovanih dužim ključem, odnosno, ključem koji se ponavlja mali broj puta u šifratu.

[24] Vilijam Fridman (William Frederick Friedman) (1891-1969) smatra se za najznačajniju ličnost američke kriptoanalize dvadesetog veka. Deo života radio je kao kriptoanalitičar za američku nacionalnu sigurnosnu agenciju (NSA). Učestvovao je u razbijanju mnogih šifara tokom I i II svetskog rata. „Indeks koncidencije i njegove primene u kriptografiji" jedno je od najznačajnijih publikacija moderne kriptologije do tada.

Primer 3.3 dužina teksta = 23 dužina ključa = 5

pozicija	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
otvoreni tekst	o	v	o	j	e	t	e	k	s	t	K	o	j	i	š	i	f	r	U	J	e	m	o
Ključ	k	l	j	u	č	k	l	j	u	č	K	l	j	u	č	k	l	j	U	Č	k	l	j
Šifrat	ć	i	c	f	i	f	š	z	n	ž	Ž	č	v	e	z	v	t	č	R	N	s	c	c

U ovom slučaju, otvoreni tekst je dužine $d=23$ i šifrovan je ključem dužine $m=5$. Može se primetiti da nema blokova teksta dužine bar tri koji se ponavljaju.

Ako sada isti tekst šifruje kraćim ključem:

Primer 3.4 dužina teksta = 23 dužina ključa = 3

pozicija	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
otvoreni tekst	o	v	o	j	E	t	e	k	s	T	K	o	j	i	š	i	f	r	U	j	e	m	o
Ključ	t	r	k	t	R	k	t	r	k	T	R	k	t	r	k	t	r	k	T	r	k	t	r
Šifrat	j	n	ć	e	Ž	f	c	d	đ	O	D	ć	e	ć	e	đ	a	d	P	č	s	h	g

Na drugoj, jedanaestoj i trinaestoj poziciji pojavljuje se isti segment „će“. Kako je $\Delta_1=9$, $\Delta_2=11$, $\Delta_3=2$, može se zaključiti da se neki od segmenata slučajno pojavljuje, te da je dužina ključa 2, 3, 9 ili 11, što daje dobru pretpostavku za određivanje dužine ključa.

Prikazani primeri pokazuju da metod Kaziskog, u opštem slučaju, ne daje očekivane rezultate. Posebno, primenom metode kod kraćih šifrata sa dužim ključem, dužina nije jednoznačno određena.

3.4.1.2 Fridmanov napad

Fridmanov napad smatra se najuspešnijom tehnikom za razbijanje polialfabetskih šifara. Ovaj metod zasniva se na metodi **indeksa koincidencije.**

Indeks koincidencije je, grubo rečeno, verovatnoća da će dva proizvoljna elementa n-točlanog niza biti jednaka [5].

Definicija 3.1: Neka je $X=x_1\, x_2 \ldots\, x_d$ niz simbola. Indeks koincidencije niske X, u oznaci $I_c(X)$ je verovatnoća da dva različita, nezavisna, slučajno izabrana (sa ravnomernom verovatnoćom) elementa niske X budu jednaka.

$$I_C(X)=p(x_j = x_k)$$

gde $j,k \in \{1,..d\}$.

Posmatrajmo azbuku veličine n. Neka su učestanosti pojavljivanja slova azbuke u niski X redom: $f_0,\ldots,f_{n-1}$. Tada je:

$$I_C(X) = \frac{broj\ koincidencija\ u\ X}{broj\ mogucih\ parova\ u\ X}$$

Ako je broj pojavljivanja i-tog slova azbuke u niski X f_i, tada postoji $\binom{f_i}{2}$ načina za izbor dva i-ta slova azbuke u toj niski. Iz niske dužine d dva proizvoljna elementa mogu se izabrati na $\binom{d}{2}$ načina. Primenom formule za binomni koeficijent izvodimo formulu za indeks koincidencije:

$$I_C(X) = \frac{\sum_{i=0}^{n-1}\binom{f_i}{2}}{\binom{d}{2}} = \frac{\sum_{i-0}^{n-1}\frac{f_i!}{(f_i-2)!2!}}{\frac{d!}{(d-2)!2!}} = \frac{\sum_{i=0}^{n-1}\frac{f_i(f_i-1)}{2}}{\frac{d(d-1)}{2}}$$
$$= \frac{\sum_{i=0}^{n-1} f_i(f_i-1)}{d(d-1)} = \frac{\sum_{i=0}^{n-1} f_i^{\,2} - d}{d^2(1-\frac{1}{d})} = \frac{\sum_{i=0}^{n-1}(\frac{f_i}{d})^2 - \frac{1}{d}}{1-\frac{1}{d}} \xrightarrow{d\to\infty} \sum_{i=0}^{n-1}(\frac{f_i}{d})^2 = \sum_{i=0}^{n-1} p_i^{\,2} \qquad (5)$$

gde je p_i izračunato na osnovu (1).

Za dovoljno veliku dužinu teksta, indeks koincidencije jednak sumi kvadrata učestanosti pojavljivanja slova te azbuke. Dakle, indeks koincidencije je karakteristika jezika.

Treba naglasiti da u slučaju kraćih tekstova (npr. 100-200 slova), I_C treba računati korišćenjem formule

$$I_C = \frac{\sum_{i=0}^{n-1} f_i(f_i-1)}{d(d-1)} \qquad (6)$$

a ne korišćenjem aproksimacije $\sum_{i=0}^{n-1} p_i^{\,2}$, jer je greška koja se javlja reda $\frac{1}{d}$, za dužinu teksta d.

Obradom velikog broj reprezentativnih tekstova nekog prirodnog jezika, može se izračunati vrednost koja predstavlja indeks koincidencije upotrebljenog jezika.

Indeks koincidencije jezika predstavlja srednju vrednost izračunatih vrednosti $I_{C[k]}$, (k=1..m) izračunatih na osnovu (5), iz uzorka veličine m.

$$\overline{I_C} = \frac{\sum_{k=1}^{m} I_{C[k]}}{m} \tag{7}$$

Red veličine greške izračunate prosečne vrednosti jednak je redu veličine standardne devijacije uzorka:

$$s = \sqrt{\frac{1}{m}\sum_{k=1}^{m}(I_{C[k]} - \overline{I_C})^2} \tag{8}$$

Eksperiment 2:

Za svaki od jezika i pisma izabrano je dvanaest tekstova, čija je dužina veća od 100 000 slova (tipični tekstovi za taj jezik). Zatim je iz svakog teksta izdvojen isečak dužine 100 000 slova. Za svaki tekst iz uzorka izračunat je njegov indeks koincidencije $I_{C[k]}$, (*k=1,..,12*), na osnovu (6). Prosečna vrednost indeksa koincidencije za dati uzorak izračunata je na osnovu (7).

Tabela 3.5

Tabela vrednosti indeksa koincidencije za engleski i srpski jezik, na osnovu rezultata Eksperimenta 2:

Jezik	Indeks koincidencije (I_C)
engleski	$\overline{I_C(X)} \approx 0{,}065$
srpski (latinica)	$\overline{I_C(X)} \approx 0{,}064$
srpski (ćirilica)	$\overline{I_C(X)} \approx 0{,}064$

Za izabrani uzorak, red veličine standardne devijacije *s* na osnovu (8) iznosi 10^{-3}, pa je zaokruživanje izvršeno na 3 decimalna mesta.

Napomena: Ako se posmatra primer 3.1, vrednosti indeksa koincidencije za šifrat i poruku iznose $I_C(P) = I_C(C) \approx 0{,}058$. Može se primetiti da ove vrednosti odstupaju od izračunatog indeksa koincidencije srpskog jezika. Razlog za to je, kao što je pomenuto, u maloj dužini šifrovane poruke, u odnosu na veliki broj obrađenih uzoraka za računanje statistike.

Razmotrimo sada tekst sastavljen od niza slučajnih, nezavisno izabranih znakova iz azbuke veličine *n*, sa ravnomernom raspodelom verovatnoće (u daljem tekstu: **slučajan tekst**):

$$I_C(X) = \sum_{i=0}^{n-1} \left(\frac{1}{n}\right)^2 = n\frac{1}{n^2} = \frac{1}{n}$$

Prema tome, vrednosti indeksa koincidencije slučajnog teksta za pomenute azbuke iznose:

Tabela 3.6

Tabela vrednosti indeksa koincidencije slučajnih tekstova:

Dužina pisma	Indeks koincidencije (I_C)
slučajni tekst nad azbukom veličine 26	$I_C(X) \approx 0{,}038$
slučajni tekst nad azbukom veličine 27	$I_C(X) \approx 0{,}037$
slučajni tekst nad azbukom veličine 30	$I_C(X) \approx 0{,}033$

Vratimo se na određivanje dužine Vižnerovog ključa.

Ako se pretpostavi da je dužina ključa *m* i šifrat se razbije na *m* isečaka C_i, na osnovu (4), tada važi:

- ako je *m* dužina ključa, tada za svako C_i indeks koincidencije $I_C(C_i)$ je približno jednak indeksu koincidencije prirodnog jezika poruke (skup slova C_i šifrovan je monoalfabetskom šifrom)
- ako *m* nije dužina ključa, tada će vrednosti $I_C(C_i)$ imati vrednosti bliske indeksu koincidencije slučajnog teksta

Primer 3.5

Poruka *P* se šifruje ključem *K* : „ KINDI “.

P : „Jedan od načina da odgonetnemo šifrovanu poruku, ako znamo na kom je jeziku, jeste da pronađemo neki tekst na tom jeziku, dugačak otprilike jednu stranu, i da prebrojimo koliko se puta koje slovo pojavljuje. Slovo koje se najčešće pojavljuje nazivamo prvim, ono koje se pojavljuje malo ređe nazivamo drugim tako dalje, sve dok ne poslažemo sva slova iz uzorka teksta. Zatim u šifrovanom tekstu koji želimo da odgonetnemo takođe klasifikujemo simbole. Pronalazimo simbol koji se najčešće pojavljuje i zamenjujemo ga prvim slovom iz uzorka teksta, sledeći najčešći simbol drugim i tako dalje, sve dok ne zamenimo sve simbole iz kriptograma.“

Tabela vrednosti indeksa koincidencija za dužine ključa 1..8 :

pretpostavljena dužina ključa m	$I_C(C_1)$	$I_C(C_2)$	$I_C(C_3)$	$I_C(C_4)$	$I_C(C_5)$	$I_C(C_6)$	$I_C(C_7)$	$I_C(C_8)$
[1]	0.044							
[2]	0.043	0.049						
[3]	0.044	0.041	0.043					
[4]	0.043	0.046	0.043	0.048				
[5]	**0.080**	**0.057**	**0.056**	**0.080**	**0.049**			
[6]	0.046	0.048	0.044	0.050	0.037	0.045		
[7]	0.042	0.045	0.043	0.044	0.040	0.041	0.044	
[8]	0.042	0.040	0.036	0.049	0.039	0.047	0.045	0.045

Može se primetiti da izdvojeni peti red tabele, sadrži najveće vrednosti indeksa koincidencije, tj. vrednosti najbliže vrednosti indeksa koincidencije odgovarajućeg prirodnog jezika. U ostalim slučajevima, vrednosti su bliže vrednosti indeksa koincidencije slučajnog teksta.

Postavlja se pitanje kako matematički odrediti za koju dužinu ključa m, isečci C_i ($i=1..m$) imaju najpribližniju vrednost indeksa koincidencije prirodnog jezika.

U cilju najefikasnijeg nalaženja dužine ključa, mogu se razmotriti tri metode: [6].

M_1 – metod minimalnog proseka odstupanja

M_2 – metod minimalnog srednjeg kvadratnog odstupanja

M_3 – metod maksimalnih proseka

Dužina ključa m traži se u unapred zadatom opsegu m_{min}, $m_{max.}$ To znači da se razmatra $m_{\max} - m_{\min} + 1$ redova tabele.

Tabela 3.7

Prikaz tabele iz primera 3.5, sa zadatim opsegom $m_{min}=4$, $m_{max}=8$

Pretpostavljena dužina ključa m	$I_C(C_1)$	$I_C(C_2)$	$I_C(C_3)$	$I_C(C_4)$	$I_C(C_5)$	$I_C(C_6)$	$I_C(C_7)$	$I_C(C_8)$
[4]	0.043	0.046	0.043	0.048				
[5]	**0.080**	**0.057**	**0.056**	**0.080**	**0.049**			
[6]	0.046	0.048	0.044	0.050	0.037	0.045		
[7]	0.042	0.045	0.043	0.044	0.040	0.041	0.044	
[8]	0.042	0.040	0.036	0.049	0.039	0.047	0.045	0.045

Metod minimalnog proseka odstupanja (M_1)

Za svaki red (potencijalnu dužinu ključa u zadatom opsegu) tabele računa se prosek indeksa koincidencije za taj red i određuje minimalno odstupanje *d* od indeksa koincidencije odgovarajućeg prirodnog jezika.

$$d_i = \frac{\sum_{j=1}^{i} \left| \overline{I_C} - I_{C[i][j]} \right|}{i}$$
$$d = \min\{ d_i : i = m_{\min}, m_{\max} \} \tag{9}$$

Metod minimalnog srednje kvadratnog odstupanja (M_2)

Za svaki red tabele računa se prosek kvadrata odstupanja indeksa koincidencije za taj red i određuje minimalno odstupanje *d* od indeksa koincidencije odgovarajućeg prirodnog jezika.

$$d_i = \frac{\sum_{j=1}^{i} \left(\overline{I_C} - I_{C[i][j]} \right)^2}{i}$$
$$d = \min\{ d_i : i = m_{\min}, m_{\max} \} \tag{10}$$

Metod maksimalnog proseka (M_3)

Za svaki red tabele računa se prosek indeksa koincidencije za taj red i određuje maksimalni prosek.

$$d_i = \frac{\sum_{j=1}^{i} I_{C[i][j]}}{i}$$
$$d = \max\{ d_i : i = m_{\min}, m_{\max} \} \tag{11}$$

Primenom jedne od navedene tri metode, na osnovu vrednosti indeksa koincidencije iz opsega $m_{min}, m_{max,}$ izračunava se najverovatnija dužina ključa *d*.

3.4.2 Određivanje ključa

Ako se pretpostavi da je određena dužina Vižnerovog ključa *m* ispravna, tada je svaki od dobijenih isečaka šifrata C_1, ..., C_m nastao primenom šifre pomeranja na isečke P_1, ..., P_m polazne poruke za *m* različitih ključeva k_i dužine jedan. Poslednji korak kriptoanalize je da se za svaki dobijeni isečak C_i odredi ključ k_i, $i = 1..m$.

Uvodimo pojam ***uzajamni indeks koincidencije*** dva teksta:

Definicija 3.2: Uzajamni indeks koincidencije dve niske *X* i *Y*, u oznaci $MI_C(X,Y)$[25] je verovatnoća da će se slova niske *X* podudariti sa slovima niske *Y*.

Razmotrimo niske X i Y na azbuci veličine *n*. Neka f_i , *i=0..n-1,* predstavljaju broj pojavljivanja slova *i* u tekstu X i neka f_i', *i=0..n-1,* predstavljaju broj pojavljivanja slova *i* u tekstu Y. Neka su *d* i *d'* dužine teksta *X*, dnosno *Y*. Tada je [3]

$$MI_C(X,Y) = \frac{\sum_{i=0}^{n-1} f_i f_i'}{dd'} \tag{12}$$

Primer 3.6

Razmotrimo dva teksta na srpskom jeziku:

T_1: „A onde, na pučini, potpuno sam, daleko od čamca i obale, vežbao je galeb Džonatan Livingston. Na visini od trideset metara pružio je svoje opnaste nožice i uzdigao kljun silno se upinjući da krilima izvede bolno težak okret izvijenog tela. “

T_2: „Ne, neću se vratiti. Otišao sam još pre dvanaest godina, a ovde, u Estersundu, već sam osam. Imam porodicu. Vidite onog beloglavog dečaka tamo? Da, liči. I jeste Šveđanin. Moja žena ga je rodila kao devojka, pre mene; drugi su ovde običaji. Dao sam mu svoje prezime. “

Uzajamni indeks koincidencije može se izračunati prebrojavanjem slova na istim pozicijama u oba teksta. U ovom primeru, za tekstove dužine 205, broj takvih slova je 13, pa je njihov uzajamni indeks koincidencije

$$MI_C(T_1,T_2) = \frac{13}{205} \approx 0{,}063$$

Razmotrimo sada dva slučajna teksta S_1 i S_2 nad azbukom veličine *n***.** Kako je verovatnoća pojavljivanja svakog slova u slučajnom tekstu jednaka, uzajamni indeks koincidencije slučajnih tesktova S_1 i S_2 iznosi

$$MI_C(S_1,S_2) = \frac{1}{n}\cdot\frac{1}{n} + \ldots + \frac{1}{n}\cdot\frac{1}{n} = \frac{n}{n^2} = \frac{1}{n}$$

Kako je $MI_C(S_1,S_2)=I_C(S_1)$, to su vrednosti uzajamnog indeksa koincidencije slučajnih tekstova na engleskom i oba srpska pisma date Tabelom 3.6.

Razmotrimo sada slučajni tekst *S*, dužine *d* i tekst T iz prirodnog jezika, dužine *d'* na azbuci dužine *n***.**

Ako f_i', *i=0..n-1* predstavljaju broj pojavljivanja slova *i* u tekstu *T*, tada je njihov uzajamni indeks koincidencije MI_c(S, T)

[25] MI_C oznaka potiče iz engleskog jezika:Mutual Index of Coincidence

$$MI_C(S,T) = \frac{1}{n} \cdot \frac{f_0'}{d'} + \ldots + \frac{1}{n} \cdot \frac{f_{n-1}'}{d'} = \frac{1}{n} \cdot (p_0' + \ldots p_{n-1}') = \frac{1}{n} = I_C(S) \quad (13)$$

Na osnovu Tabele 3.6 važi:

MI_C(engleski tekst,slučajan tekst) = I_C(slučajan tekst) ≈ 0,038
MI_C(srpski tekst na latinici, slučajan tekst) = I_C(slučajan tekst) ≈ 0,037
MI_C(srpski tekst na čirilici, slučajan tekst) = I_C(slučajan tekst) ≈ 0,033

(14)

Najzad, razmotrimo dva teksta T_1 i T_2, dužine d i d', iz prirodnog jezika, na azbuci veličine n. Neka f_i, $i=0..n-1$ predstavljaju broj pojavljivanja slova i u tekstu T_1, a f_i', $i=0..n-1$ broj pojavljivanja slova i u tekstu T_2. Tada je na osnovu (5)

$$MI_C(T_1,T_2) = \frac{\sum_{i=0}^{n-1} f_i f_i'}{dd'} = \sum_{i=0}^{n-1} p_i p_i' \approx \sum_{i=0}^{n-1} p_i^2 \approx I_C(T_1) \quad (15)$$

Tabela vrednosti uzajamnih indeksa koincidencije za tekstove na engleskom i srpskom jeziku, na osnovu vrednosti iz Tabele 3.5:

MI_C(engleski tekst,engleski tekst)≈I_C(engleski tekst)=0,065
MI_C(srpski tekst na latinici, srpski tekst na latinici)≈I_C(srpski tekst na latinici)=0,064
MI_C(srpski tekst na čirilici, srpski tekst na ćirilici)≈I_C(srpski tekst na ćirilici)=0,064

(16)

Uzajamni indeks koincidencije $MI_C(X,Y)$ može se primeniti u analizama periodičnih šifara, pod pretpostavkom da je uspešno određena dužina ključa m.

Na osnovu vrednosti dobijenih statističkom obradom, može se zaključiti da važi

$$MI_C(S,X) << I_C(X),$$

ako je X tekst na nekom prirodnom jeziku, a S slučajan tekst

i

$$MI_C(T,X) \approx I_C(X),$$

ako su i X i T tekstovi na istom prirodnom jeziku.

Praktično se ova činjenica može upotrebiti za nalaženje ključa $k=k_1...k_m$ na sledeći način:

Neka je C_i ($i=1..m$), i-ti podniz dobijen razbijanjem šifrata po modulu m, na osnovu (4).

Svaki isečak C_i šifrovan je šifrom pomeranja za ključ k_i.

Neka f_i', $i=0..n-1$ predstavljaju broj pojavljivanja slova i (iz azbuke veličine n) u tekstu C_i i neka je d' dužina teksta C_i.

Neka je *T tipičan tekst*[26] dužine d, iz prirodnog jezika.

Neka su $p_i \; i = 0,..,n-1$ su verovatnoće pojavljivanja slova te azbuke u tipičnom tekstu T, na osnovu (1).

Neka su C_i^z, $1 \le i \le m$, $0 \le z \le n-1$ tekstovi dobijeni dešifrovanjem slova isečka C_i pomeranjem za ključ z. Tada važi:

$$MI_C(T, C_i^z) \approx \frac{\sum_{j=0}^{n-1} p_j f'_{(j-z) \bmod n}}{d'}$$

Na osnovu vrednosti navedenim u (13) i (15) , očekuje se da

$$MI_C(T, C_i^z) \approx I_C(T) \tag{17}$$

ako je $z = k_i$

$$MI_C(T, C_i^z) << I_C(T) \tag{18}$$

u svakom drugom slučaju [3].

Drugim rečima, uzajamni indeks koincidencije biće približno jednak indeksu koincidencije tog jezika, ako se dešifruje tačnim slovom ključa, odnosno ako se dešifrovanjem dobija tekst na prirodnom jeziku, a ne neki slučajni niz slova.

Ovo upravo znači da redom, za $z = 0,...,n-1$ treba izračunati vrednosti

$$MI^z \approx \frac{\sum_{j=0}^{n-1} p_j f'_{(j-z) \bmod n}}{d'}$$

i pronaći pomeraj h, takav da važi

$MI^h = \min$ *odstupanja* MI^z *od* $I_C(T)$, *za* $0 \le z \le n-1$

Poslednji korak je odrediti $k_i = h$, čime se nalazi i-to slovo ključa Vižnerove šifre.

[26] Tipičan tekst je tekst na prirodnom jeziku, čiji indeks koincidencije odgovara indeksu koincidencije tog jezika.

4 Metode za procenu uspešnosti kriptoanalize

4.1 Interval poverenja

Nakon određivanja potencijalnog Vižnerovog ključa, vrši se dešifrovanje šifrata pretpostavljenim ključem i dobija mogući tekst poruke. Na osnovu njegovog indeksa koincidencije, oslanjajući se na statističke karakteristike primenjenog jezika, na osnovu (14) i (16), može se utvrditi da li je dobijeni tekst "smislen" za taj jezik i u tom slučaju ćemo smatrati da je pretpostavljeni ključ ispravan.

Za potencijalni tekst poruke, treba dakle, odrediti metodu koja proverava da li važi (17) i u tom slučaju smatramo da je dekriptiranje uspešno, ili (18) i u tom slučaju kriptoanaliza nije uspešna.

U tom cilju, izračunaćemo 99%-ni interval poverenja za vrednost indeksa koincidencije za svaki od primenjenih prirodnih jezika.

Neka je m - broj uzoraka u eksperimentu; $I_{C[i]}$, $(i=1..m)$ vrednosti indeksa koincidencije uzoraka, na osnovu (6); $\overline{I_C}$ - ocena matematikog očekivanja obrađenih uzoraka

$$\overline{I_C} = \frac{\sum_{i=1}^{m} I_{C[i]}}{m} \tag{19}$$

s^2 - ocena varijanse uzorka, a s - standardna devijacija

$$s^2 = \frac{1}{m-1}\sum_{i=1}^{m}(I_{C[i]} - \overline{I_C})^2 \tag{20}$$

Izraz (20) može se transformisati u formulu [6]

$$s^2 = \frac{1}{m-1}\sum_{i=1}^{m} I_{C[i]}{}^2 - \frac{m}{m-1}\overline{I_C}^2 \tag{21}$$

Tada je dvostrani interval poverenja za indeks koincidencije I_C [6]:

$$p\left(I_C \in \left[\overline{I_C} - \varepsilon_{1-\frac{\alpha}{2}}\frac{s}{\sqrt{m}}, \overline{I_C} + \varepsilon_{1-\frac{\alpha}{2}}\frac{s}{\sqrt{m}}\right]\right) = 1-\alpha \tag{22}$$

gde je $\alpha = 0{,}01$ u slučaju 99% intervala poverenja, a $\varepsilon_{1-\frac{\alpha}{2}} = \varepsilon_{0{,}995}$ kvantil reda 0,995 Studentove $t(m-1)$ raspodele.

Na osnovu (22) izvodi se formula za izračunavanje donje granica za interval poverenja

$$\overline{I_C} - \varepsilon_{1-\frac{\alpha}{2}}\frac{s}{\sqrt{m}} \tag{23}$$

i gornje granice za interval poverenja

$$\overline{I_C} + \varepsilon_{1-\frac{\alpha}{2}} \frac{s}{\sqrt{m}} \tag{24}$$

Eksperiment 3:

Da bi se odredio interval poverenja za engleski i srpski jezik i njihova pisma, korišćen je uzorak od $m = 12$ tekstova za svaki od jezika. Svaki tekst iz uzorka skraćen je na dužinu od 100 000 slova. Indeks koincidencije svakog od uzorka izračunava se na osnovu (6), a srednja vrednost za taj jezik na osnovu (19). Standardna devijacija uzorka izračunava se na osnovu (21), a donja i gornja granica za interval poverenja na osnovu (23) i (24) gde je ε_{995} =3,106 kvantil iz tablice Studentove raspodele [6].

Tabela 4.1

Prikaz 99% -nih intervala poverenja I_C za engleski i srpski jezik, na osnovu rezultata Eksperimenta 3:

Primenjeni jezik	Donja granica intervala poverenja	Gornja granica intervala poverenja
Engleski	0.06390354998108362	0.06677919660718147
Srpski (latinica)	0.06244808462189807	0.0650393907276089
Srpski (ćirilica)	0.062493024958986114	0.06477173446953581

Izračunati intervali poverenja koriste se za procenu rezultata (17) i (18). Kada se šifrat dešifruje pretpostavljenim Vižnerovim ključem, dobija se tekst za koji se proverava da li je vrednost njegovog indeksa koincidencije u granicama izračunatog intervala poverenja indeksa koincidencije za taj jezik. U zavisnosti od toga, može se zaključiti da li je tekst „smislen“ za taj jezik, odnosno, da li je napad na šifrat uspešan.

4.2 Uspešnost kriptoanalize kroz odnos dužina šifrata, dužina ključa

Primenom algoritma za određivanje Vižnerovog ključa, opisanog u poglavlju 3, moguće je za kraće ključeve i duže šifrate odrediti ključ i doći do teksta poruke. Međutim, u slučaju dužih ključeva, kada je broj ponavljanja ključa u šifratu mali, algoritam ne daje obavezno ispravan rezultat. Eksperiment treba da pokaže procenat uspešnosti algoritma, za različite dužine ključeva i šifrata.

Eksperiment 4:

Rezultati eksperimenta trebalo bi da pokažu uspešnosti opisanih postupaka za kriptoanalizu, kroz različit odnos [dužina poruke, dužina ključa].

Razmatrane dužine ključeva su od 1 do 16, a razmatrane dužine tekstova su od 2^5 do 2^{11}.

Za svaki jezik napravljena je baza tekstova, dodavanjem tipičnih tekstova tog jezika jedan na drugi. Veličina dobijene baze za svaki jezik je reda veličine 10^6 slova.

Za svaki jezik i svaki par [dužina teksta, dužina ključa] izvršeno je 100 merenja.

Jedno takvo merenje kroz odnos [dužina teksta=d, dužina ključa=m], $d \in \{2^i, i = 5,..,11\}$, $m \in \{1,..,16\}$ izvodi se na sledeći način:

Da bi se izbeglo ponavljanja ključa, za svako merenje generiše se slučajan ključ dužine m. Takođe, da bi se izbeglo ponavljanje poruka koje se šifruju, za svako merenje generiše se slučajan broj, koji određuje početak teksta dužine d, koji se izdvaja iz baze za taj jezik. Tako dobijeni tekst, dužine d se šifruje ključem dužine m i dobija šifrat dužine d. Zatim se proverava uspešnost nalaženja ključa, upoređivanjem dobijenog dešifrovanog teksta sa tekstom polazne poruke. Ispravno određen ključ, za par [d, m] se registruje i prelazi se na sledeće merenje.

Rezultati Eksperimenta 4, kojima se ocenjuje uspešnost metoda M_1, M_2, M_3 (opisanih u 3.4.1.2) prikazani su u tabelama.

Tabela 4.2

Uspešnost kriptoanalize metodom minimalnog proseka odstupanja (M_1)

dužina ključa dužina šifrata	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32	92%	29%	6%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
64	96%	45%	35%	18%	7%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
128	97%	69%	76%	51%	66%	20%	21%	4%	6%	2%	0%	0%	0%	0%	0%	0%
256	100%	80%	91%	79%	91%	67%	82%	51%	52%	32%	38%	10%	16%	6%	5%	1%
512	100%	96%	99%	93%	99%	82%	99%	89%	96%	77%	96%	65%	83%	61%	60%	41%
1024	100%	99%	99%	98%	100%	98%	100%	97%	99%	99%	100%	97%	100%	94%	97%	88%
2048	100%	100%	100%	99%	100%	99%	100%	100%	100%	99%	100%	98%	100%	99%	100%	99%

Tabela 4.3

Uspešnost kriptoanalize metodom minimalnog srednjeg kvadratnog odstupanja (M_2)

dužina ključa dužina šifrata	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32	93%	21%	12%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
64	99%	52%	38%	19%	5%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
128	100%	66%	83%	58%	55%	16%	20%	5%	2%	0%	0%	0%	0%	0%	0%	0%
256	100%	85%	97%	78%	93%	57%	81%	39%	48%	22%	37%	15%	8%	1%	3%	0%
512	100%	92%	99%	93%	100%	79%	99%	88%	92%	66%	86%	60%	90%	56%	55%	35%
1024	100%	100%	99%	95%	100%	94%	100%	96%	97%	97%	100%	95%	100%	94%	97%	84%
2048	100%	99%	100%	100%	100%	100%	100%	100%	100%	99%	100%	98%	100%	99%	100%	97%

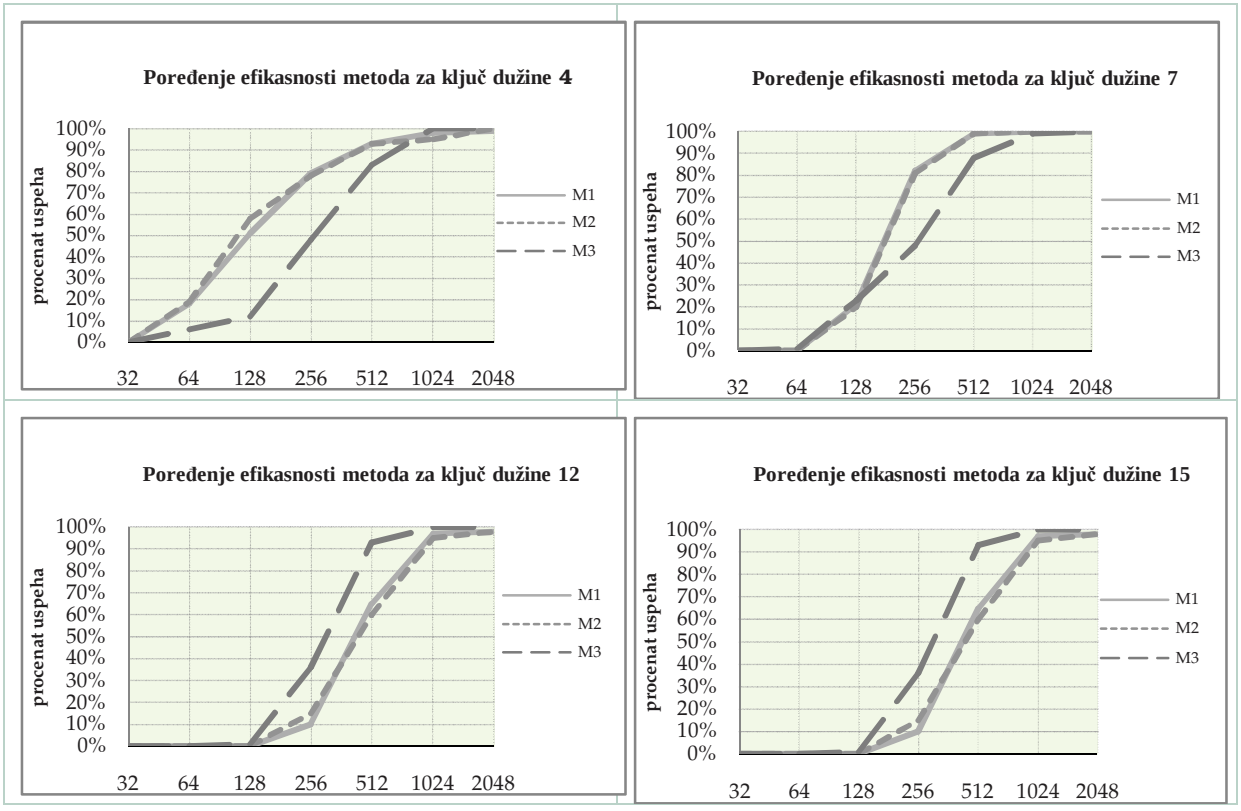
Tabela 4.4

Uspešnost kriptoanalize metodom maksimalnog proseka (M_3)

dužina ključa / dužina šifrata	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
64	0%	0%	0%	6%	3%	1%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%
128	7%	10%	14%	12%	20%	22%	23%	11%	5%	7%	1%	1%	0%	0%	0%	0%
256	25%	27%	33%	48%	46%	65%	48%	41%	71%	56%	48%	36%	23%	14%	2%	5%
512	74%	74%	86%	83%	87%	93%	88%	80%	100%	96%	91%	93%	84%	82%	79%	59%
1024	98%	99%	100%	100%	100%	100%	99%	99%	100%	100%	100%	100%	99%	100%	100%	100%
2048	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Uspešnost navedenih metoda, za proizvoljno odabrane dužine ključeva, na primer *m=4, m=7, m=12, m=15*, prikazana je grafikom**:**

Grafik 4.1



Može se primetiti da se u zavisnosti od izbora metode kriva uspešnosti algoritma značajno menja kroz odnos [dužina teksta, dužina ključa]. Tako, za kraće ključeve na tekstovima iste dužine, metode minimalnog proseka odstupanja (M_1) i srednje kvadratnog odstupanja (M_2), daju približno iste rezultate. To je slučaj kada je ključ ponovljen najveći broj puta u šifratu,

čime se dobijaju precizniji podaci o dužini ključa. Za razliku od pomenutih metoda, metoda maksimalnih indeksa koincidencija (M_3) daje značajno lošije rezultate za ključeve kraće dužine. Međutim, sa povećanjem dužine ključa, na tekstovima iste dužine, ova metoda daje za nijansu bolje rezultate od metoda M_1 i M_2.

Iz tih razloga je za izračunavanje u okviru programa (Poglavlje 5) izabrana metoda minimalnog srednje kvadratnog odstupanja (M_2), koja u svim slučajevima, bez obzira na dužinu ključa, daje pouzdane rezultate.

Navode se rezultati Eksperimenta 4, izvršenog za sva tri prirodna jezika, korišćenjem metode M_2.

Tabela 4.5

Uspešnost kriptoanalize za engleski jezik, na osnovu rezultata Eksperimenta 4:

dužina ključa / dužina šifrata	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32	88%	26%	13%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
64	97%	52%	31%	7%	5%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
128	97%	81%	87%	57%	56%	19%	18%	4%	1%	0%	0%	0%	0%	0%	0%	0%
256	100%	94%	95%	80%	95%	74%	76%	47%	43%	30%	32%	3%	14%	1%	4%	0%
512	100%	98%	100%	96%	99%	88%	99%	86%	94%	74%	95%	71%	85%	57%	63%	44%
1024	100%	100%	99%	99%	98%	98%	100%	96%	100%	97%	100%	97%	99%	99%	100%	94%
2048	100%	99%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%	99%	99%

Grafik 4.2

METODA

[srednje kvadratno odstupanje]

PRIMENJENI JEZIK [english]

DUZINA PISMA [26]

DUZINA TEKSTA [7883011]

INDEKS KOINCIDENCIJE [0.065]

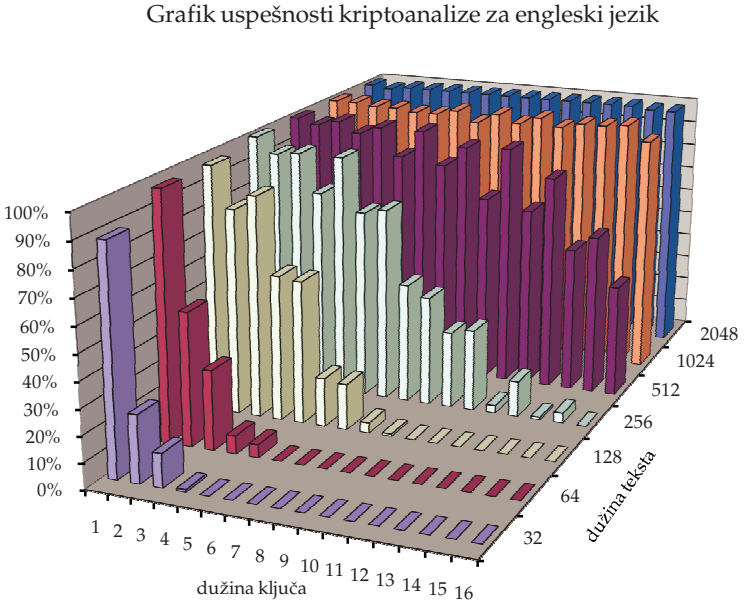


Tabela 4.6

Uspešnost kriptoanalize sa srpski jezik (latinično pismo), na osnovu rezultata Eksperimenta 4:

dužina ključa dužina šifrata	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32	90%	26%	11%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
64	98%	44%	33%	9%	4%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
128	99%	68%	69%	46%	50%	14%	18%	2%	1%	1%	1%	0%	0%	0%	0%	0%
256	100%	87%	94%	76%	90%	63%	79%	33%	43%	21%	37%	5%	12%	2%	1%	4%
512	100%	90%	98%	94%	98%	85%	98%	80%	87%	71%	92%	65%	85%	39%	50%	29%
1024	100%	97%	100%	98%	100%	90%	100%	97%	100%	93%	100%	90%	100%	87%	98%	83%
2048	100%	100%	100%	100%	100%	99%	100%	100%	100%	97%	100%	98%	100%	98%	100%	97%

Grafik 4.3

METODA

[srednje kvadratno odstupanje]

PRIMENJENI JEZIK [srpski (latinica)]

DUZINA PISMA [27]

DUZINA TEKSTA [2330743]

INDEKS KOINCIDENCIJE [0.064]

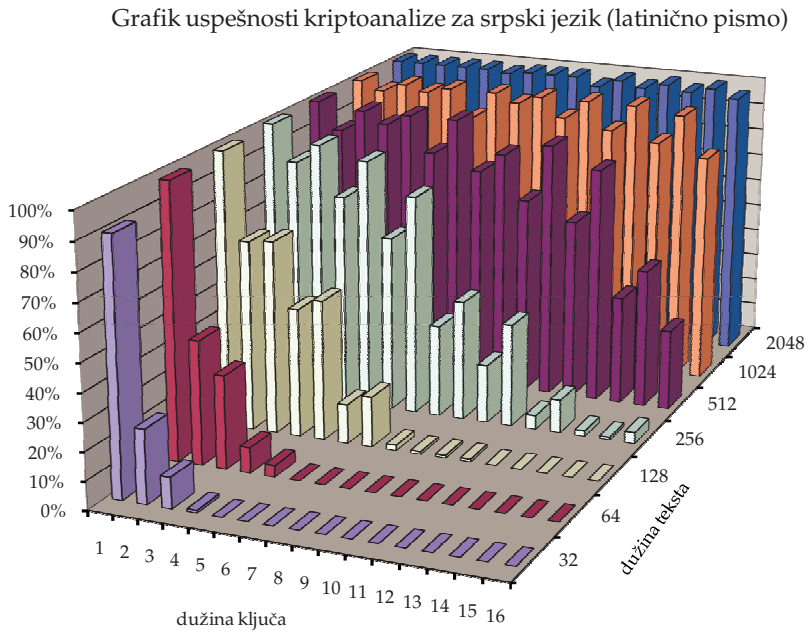


Tabela 4.7

Uspešnost kriptoanalize sa srpski jezik (ćirilično pismo) na osnovu rezultata Eksperimenta 4:

dužina ključa dužina šifrata	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
32	96%	20%	11%	1%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
64	100%	47%	48%	20%	9%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
128	100%	63%	87%	53%	70%	28%	28%	5%	6%	0%	1%	0%	0%	0%	0%	0%
256	100%	82%	97%	72%	94%	65%	82%	51%	53%	29%	47%	13%	21%	4%	6%	1%
512	100%	91%	99%	94%	100%	92%	99%	83%	93%	83%	98%	65%	86%	66%	72%	49%
1024	100%	98%	100%	100%	100%	99%	100%	98%	100%	97%	100%	91%	100%	90%	99%	90%
2048	100%	100%	100%	99%	100%	100%	100%	100%	100%	99%	100%	98%	100%	100%	100%	100%

Grafik 4.4

METODA

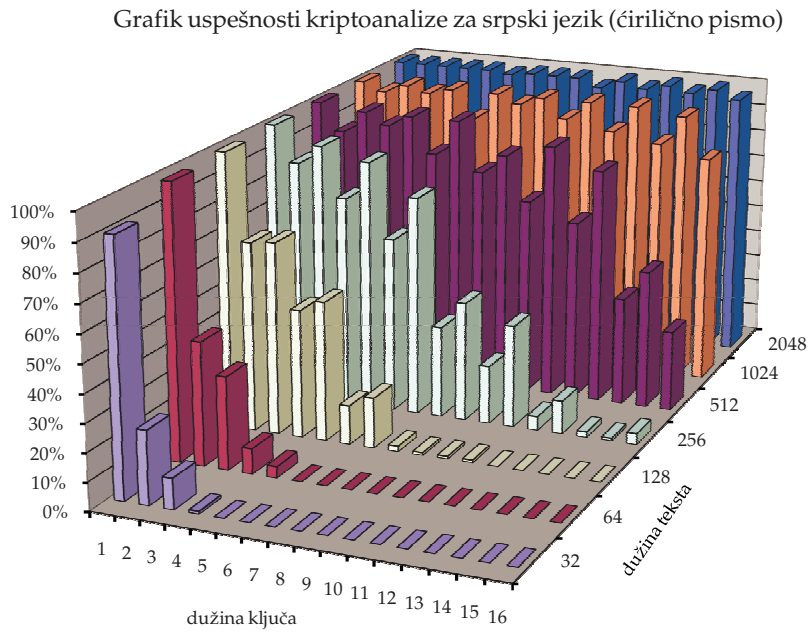
[srednje kvadratno odstupanje]

PRIMENJENI JEZIK [srpski (ćirilica)]

DUZINA PISMA [30]

DUZINA TEKSTA [3053046]

INDEKS KOINCIDENCIJE [0.064]



Na osnovu prikazanih rezultata vidi se da uspešnost algoritma zavisi od odnosa dužine teksta i dužine ključa, odnosno broja ponavljanja ključa u šifratu. U tabelama vrednosti uspešnosti algoritma istaknute su vrednosti kod kojih je procenat uspeha preko 50%.

Procenjuje se da je potrebno bar trideset ponavljanja ključa u tekstu poruke, za uspešnost algoritma preko 50%.

5 Opis programa

Program koji implementira algoritam za kriptoanalizu Vižnerove šifre napisan je na programskom jeziku Java, u razvojnim okruženjima Eclipse v3.2.1 i JavaBuilder v3.0.

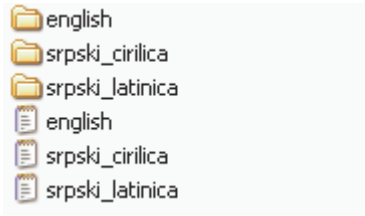
Sastoji iz dve klase: Obrada i GUI. Klasa Obrada sadrži logiku za učitavanje, šifrovanje, dešifrovanje i analizu teksta. U klasi GUI nalazi se grafički korisnički interfejs koji se potpuno oslanja na funkcionalnosti klase Obrada. U obe klase nalazi se statička funkcija main(). Pokretanjem funkcije main() iz klase Obrada pokreće se konzolni program koji je korišćen za izvođenje eksperimenata. Funkcija main() iz klase GUI pokreće grafičko korisničko okruženje.

Program je zamišljen kao višejezičan, a za potrebe ovog rada pripremljene su definicije i primeri za dva prirodna jezika: engleski i srpski (kroz latinično i ćirilično pismo).

5.1 Struktura dokumenata

Dokumenti korišćeni za statistička izračunavanja u delu 4 i dokumenti koji se koriste u radu sa programom organizovani su u dva direktorijuma:

Direktorijum #jezici sadrži tri tekstualna fajla i tri poddirektorijuma:



U tekstualnim fajlovima english.txt, srpski_cirilica.txt, srpski_latinica.txt nalaze se naziv jezika, definicija pisma i izračunata gornja i donja granica intervala poverenja indeksa koincidencije za taj jezik (koje su izračunate u okviru eksperimenta 3 iz 4.1).

Na primer, dokument english.txt sadrži podatke:

english

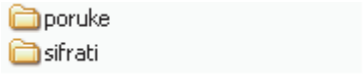
aAbBcCdDeEfFgGhHiIjJkKlLmMnNoOpPqQrRsStTuUvVwWxXyYzZ

0.06390354998108362

0.06677919660718147

U direktorijumima english, srpski_latinica i srpski_cirilica nalaze se tekstualni fajlovi (*.txt) koji sadrže različite tekstove na pomenutim jezicima. To su tekstovi koji su tipični za te jezike i korišćeni su za eksperimente u poglavlju 4.

Direktorijum #dokumenti sadrži dva poddirektorijuma



U oba direktorijuma nalaze se poddirektorijumi english, srpski_latinica i srpski_cirilica, koji sadrže tekstualne fajlove sa porukama, odnosno šifratima na tim jezicima.

Zbog korišćenja slova srpskog ćiriličnog i latiničnog pisma, podaci u *.txt fajlovima predstavljeni su Unicode UTF-16[27] skupom karaktera.

5.2 Obrada.java

Klasa Obrada sadrži sledeća polja privatnog tipa:

naziv_jezika	String koji predstavlja opis korišćenog pisma (english, srpski_latinica, српски_ћирилица).
duzina_pisma	Broj karaktera u azbuci korišćenog jezika.
definicija_pisma	String koji predstavlja definiciju pisma (sva mala i velika slova korišćene azbuke).
mala_slova	Niz koji sadrži sva mala slova korišćene azbuke.
velika_slova	Niz koji sadrži sva velika slova korišćene azbuke.
broj_pojavljivanja	Niz koji predstavlja broj pojavljivanja svakog karaktera azbuke u tekstu poruke ili šifrata.
tekst	StringBuffer koji sadrži tekst poruke, šifrata.
interval_poverenja_donja_granica	Broj koji predstavlja donju granicu intervala poverenja za indeks koincidencije korišćenog jezika, na osnovu (23).
interval_poverenja_gornja_granica	Broj koji predstavlja donju granicu intervala poverenja za indeks koincidencije korišćenog jezika, na osnovu (24).

5.2.1.1 Definisanje jezika i kreiranje poruka i šifrata

Novi objekat klase Obrada kreira se konstruktorima:

[27] UTF-16 je verzija Unicode standarda koja za zapis karaktera koristi najmanje dva bajta. Slova koja se koriste u savremenim prirodnim jezicima sadržana su u osnovnoj jezičkoj ravni (BMP-Basic Multilingual Plane) i predstavljaju se sa dva bajta.

Obrada(File, String)

Obrada(File jezik, String charsetName) – Kreira objekat klase Obrada učitavajući podatke o jeziku i definiciji pisma tog jezika iz objekta tipa File, koristeći kodnu stranu *charsetName*.

Obrada(String, String, String, String)

Obrada(String naziv_jezika, String definicija_jezika, String interval_poverenja_gornja_granica, String interval_poverenja_donja_granica) – Kreira objekat klase Obrada sa navedenim podacima o jeziku.

Objekat klase Obrada može sadržati

- samo definiciju i opis jezika, i u tom slučaju polja *tekst* i *broj_pojavljivanja* ostaju prazna
- tekst poruke ili šifrata, i u tom slučaju se polja *tekst* i *broj_pojavljivanja* popunjavaju metodama:

dodajTekst(char)

dodajTekst(char c) – Dodaje obrađeni karakter *c* na kraj stringa *tekst* tog objekta.

dodajTekst(File)

dodajTekst(File f) – Dodaje obrađeni sadržaj teksta dokumenta *f* na kraj stringa *tekst* tog objekta.

dodajTekst(String)

dodajTekst(String s) – Dodaje obrađeni string *s* na kraj stringa *tekst* tog objekta.

dodajTekst(StringBuffer)

dodajTekst(StringBuffer s) – Dodaje obrađeni string *s* tipa StringBuffer na kraj stringa *tekst* tog objekta.

Ove metode pozivaju privatne metode:

obradiKarakter(char)

obradiKarakter(char c) – Obrađuje karakter *c*, tako što ga odbacuje ukoliko nije iz azbuke definisanog jezika, a ukoliko jeste, u slučaju velikog slova, zamenjuje ga odgovarajućim malim slovom.

popuniPojavljivanja()

popuniPojavljivanja() - Prolazi kroz tekst objekta, prebrojava pojavljivanja slova definisane azbuke u stringu *tekst* tog objekta i ažurira niz *broj_pojavljivanja.*

5.2.2 Metode za šifrovanje poruka i dešifrovanje šifrata

Metode klase Obrada šifruju tekst poruke i dešifruju tekst šifrata

- šifrom pomeranja
- Vižnerovom šifrom

Funkcije koje se koriste za šifrovanje i dešifrovanje su:

obicnoSifrovanje(char, boolean)

obicnoSifrovanje(char c, boolean b) – Šifruje *tekst* tog objekta šifrom pomeranja, na osnovu definicije 3.2, za ključ *c i b = true*, a dešifruje ga ključem *c* za *b = false.*

viznerovoSifrovanje(String, boolean)

viznerovoSifrovanje(String s, boolean b) – Šifruje *tekst* tog objekta Vižnerovom šifrom, na osnovu definicije 3.3, za ključ *s* i *b=true*, a dešifruje ga ključem *s* za *b=false*.

5.2.3 Metode za kriptoanalizu

Metode koje se koriste za dekriptiranje šifre proste zamene i Vižnerove šifre, predstavljaju implementaciju već pomenutih algoritama iz poglavlja 3.

Metode za određivanje indeksa koincidencije i uzajamnog indeksa koincidencije:

indeksKoincidencije()

indeksKoincidencije() – Računa vrednost indeksa koincidencije stringa *tekst* tog objekta, na osnovu (6).

indeksKoincidencije(Obrada)

indeksKoincidencije(Obrada tipican_tekst) - Računa vrednost uzajamnog indeksa koincidencije stringa *tekst* tog objekta i teksta objekta *tipican_tekst*, na osnovu (12).

5.2.3.1 Određivanje ključa dužine jedan

pretpostaviObicanKljuc(Obrada)

pretpostaviObicanKljuc(Obrada tipican_tekst) – Dešifruje *tekst* tog objekta redom svakim slovom *i* definisane azbuke, pozivom metode *obicnoSifrovanje(i, false)* i računa uzajamne indekse koincidencija svakog od dešifrovanih tekstova i teksta objekta *tipican_tekst*, pozivom funkcije *indeksKoincidencije(tipican_tekst)*.

Kao rezultat vraća matricu, čija je prva vrsta niz svih slova definisanog jezika, a druga niz izračunatih vrednosti uzajamnog indeksa koincidencija, sortiranih prema izračunatim vrednostima uzajamnih indeksa koincidencije u opadajućem poretku.

Napomena: Ovakvo rešenje metode pretpostaviObicanKljuc(Obrada tipican_tekst) postoji zbog dodatne funkcionalnosti grafičko-korisničkog dela programa. Naime, metoda za određivanje Vižnerovog ključa svodi se na poziv ove metode za svako slovo ključa. Kako ona vraća niz svih slova azbuke jezika (sa izračunatim vrednostima uzajamnog indeksa koincidencije sa tipičnim tekstom jezika), ovim je data mogućnost interakcije korisnika i dodatni reizbor slova nađenog Vižnerovog ključa, ukoliko korisnik posumnja u tačnost nekog od slova ponuđenog ključa.

5.2.3.2 Određivanje dužine Vižnerovog ključa

Metoda koja implementira postupak određivanje matrice sa vrednostima indeksa koincidencija (videti Primer 3.5), za sve dužine ključa iz zadatog opsega je

- odrediIndekseKoincidencije(int, int)

odrediIndekseKoincidencije(int minimum, int maksimum) - Za svaku od vrednosti *i* iz opsega *[minimum,maksimum]* (zadati opseg dužine ključa) deli *tekst* tog objekta na isečke, na osnovu (4) i računa njihove indekse koincidencija, pozivom metode *indeksKoincidencije()*. Kao rezultat vraća matricu vrednosti indeksa koincidencije isečaka za zadati opseg.

Metode M_1, M_2, M_3 za pretpostavljenje dužine Vižnerovog ključa, opisane u 3.4.1.2. implementirane su sledećim javnim metodama:

- odrediOdstupanjaIndeksaKoincidencije(int, int, Obrada)

odrediOdstupanjaIK (int minimum, int maksimum, Obrada tipican_tekst) – Računa odstupanja vrednosti indeksa koincidencije svakog reda matrice dobijene pozivom metode *odrediIndekseKoincidencije(int minimum, int maksimum)* od indeksa koincidencije teksta objekta *tipican_tekst*, na osnovu (9). Kao rezulat vraća niz vrednosti odstupanja indeksa koincidencije, za sve vrednosti iz zadatog opsega [*minimum,maksimum*].

- odrediKvadrateOdstupanjaIndeksaKoincidencije(int, int, Obrada)

odredKvadrateOdstupanjaIK(int minimum, int maksimum, Obrada tipican_tekst) –Računa kvadrate odstupanja vrednosti indeksa koincidencije svakog reda matrice dobijene pozivom metode *odrediIndekseKoincidencije(int minimum, int maksimum)* od indeksa koincidencije teksta objekta *tipican_tekst*, na osnovu (10). Kao rezulat vraća niz vrednosti kvadrata odstupanja indeksa koincidencije, za sve vrednosti iz zadatog opsega [*minimum,maksimum*].

- odrediProsekeIK(int, int)

odrediProsekeIK(int minimum, int maksimum) – Računa srednju vrednost indeksa koincidencije svakog reda matrice dobijene pozivom metode *odrediIndekseKoincidencije(int minimum, int maksimum)* , na osnovu (11). Kao rezulat vraća niz proseka indeksa koincidencije za sve vrednosti iz zadatog opsega [*minimum,maksimum*].

Metoda za pretpostavljanje dužine Vižnerovog ključa:

- pretpostaviDuzinuViznerovogKljuca(int, int, Obrada, int)

pretpostaviDuzinuViznerovogKljuca(int minimum, int maksimum, Obrada tipican_tekst, int metoda) – Računa najverovatniju dužinu Vižnerovog ključa u opsegu [*minimum,maksimum*] pozivom jedne od opisane tri metode M_1, M_2, M_3 u zavisnosti od vrednosti parametra *metoda=1, 2, 3,* redom. Kao rezultat vraća pretpostavljenu dužinu ključa, na osnovu (9), (10), odnosno (11).

5.2.3.3 Pretpostavljanje Vižnerovog ključa

- pretpostaviVizenrovKljuc(int, Obrada)

pretpostaviViznerovKljuc(int duzina_kljuca, Obrada tipican_tekst) – Za svaku poziciju *i* slova ključa, *i = 1..duzina_kljuca* poziva metodu *pretpostaviObicanKljuc (tipican_tekst)* i vraća trodimenzionalnu matricu, koja predstavlja niz (dužine *duzina_kljuca*) matrica dobijenih opisanom metodom *pretpostaviObicanKljuc (tipican_tekst)*

- najverovatnijiVizenrovKljuc(int, Obrada)

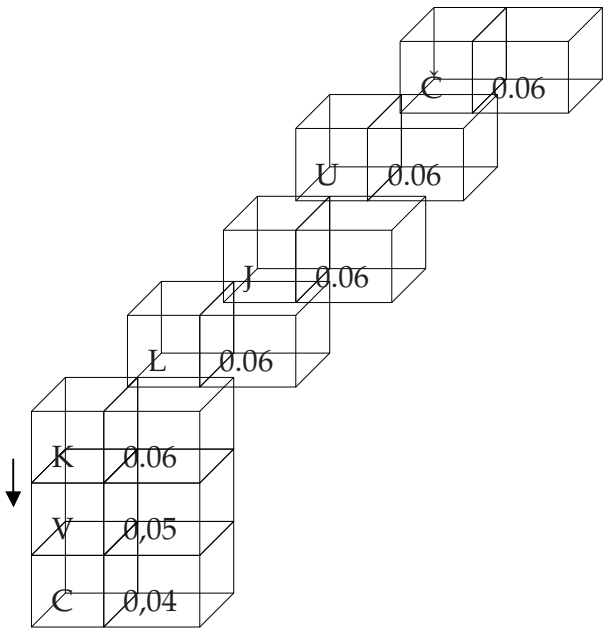
najverovatnijiViznerovKljuc(int duzina_kljuca, Obrada tipican_tekst) – Poziva metodu *pretpostaviViznerovKljuc (duzina_kljuca,tipican_tekst)*. Iz dobijene trodimenzionalne matrice, za svaku poziciju *i* slova ključa, *i = 1..duzina_kljuca* iz dvodimenzionalne matrice uzima sortirani niz slova, koji odgovara najvećim vrednostima indeksa koincidencije za svaku poziciju *i*. Vraća string sačinjen od slova tog niza.

Primer 5.1

Trodimenzionalna matrica, dobijena pozivom metode pretpostaviViznerovKljuc(5,tipican_tekst) prikazana je na slici.

U okviru ove metode pet puta je pozvana metoda pretpostaviObicanKljuc(tipican_tekst), koja u prvom pozivu vraća matricu slova i odgovarajućih indeksa koincidencije - kandidata za prvo slovo ključa sortiranih od najverovatnijeg ka manje verovatnom, u drugom pozivu matricu za drugo slovo ključa itd.

Pozivom metode najverovatnijiViznerovKljuc(5,tipican_tekst) dobija se string KLJUČ.



5.3 Metode korišćene u eksperimentalnim izračunavanjima

Klasa Obrada sadrži nekoliko statičkih metoda koje su korišćene za izvođenje eksperimenata iz poglavlja 4. Dve osnovne statičke metode su *analizaJezika()* i *ocenaMetoda()*. Prva metoda koristi se za analizu odabranog jezika, kroz učestanosti pojavljivanja slova tog jezika. Druga metoda koristi se za procenu uspešnosti algoritma za kriptoanalizu.

5.3.1 Analiza jezika

Izračunavanje učestanosti pojavljivanja slova i indeksa koincidencije za izabrani jezik izvršava se u statičkoj metodi *analizaJezika()*, koja predstavlja implementaciju Eksperimenta 1 iz 3.2.1.

analizaJezika(int upotrebljeniJezik) – Izračunava i ispisuje statističke osobine engleskog, srpskog(latinica) i srpskog(ćirilica) jezika, pozivanjem za vrednost argumenta *upotrebljeniJezik = 1, 2, 3* redom.

Ova metoda poziva sledeće statičke metode, koje određuju broj decimalnih mesta na koje izračunate vrednosti treba zaokružiti:

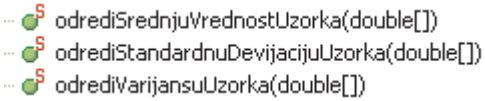
odrediSrednjuVrednostUzorka(double[] d) – Računa ocenu matematičkog očekivanja nezavisnih slučajnih promenljivih koje su elementi niza *d* na osnovu (19) iz 4.1.

odrediStandardnuDevijacijuUzorka(double[] d) – Računa standardnu devijaciju uzorka *d* na osnovu (21) iz 4.1.

Donja i gornja granica intervala poverenja za indeks koincidencije prirodnog jezika računaju se pozivom statičke metode:

odrediIntervalPoverenja(doble[] uzorak) – Izračunava dvostrani interval poverenja za vrednosti indeksa koincidencije iz niza *uzorak* na osnovu (22) i vraća niz dva elementa koji predstavljaju donju i gornju granicu intervala poverenja, na osnovu (23) i (24)

Ova metoda poziva već opisane statičke metode



koje predstavljaju implementaciju formula (19), (21).

5.3.2 Ocena uspešnosti algoritama za kriptoanalizu

Ocena uspešnosti algoritma za kriptoanalizu realizovana je u statičkoj metodi

ocenaMetoda(int, int, int, int, int)	ocenaMetoda(int brojRedova, int maksimalnaDuzinaKljuca, int brojEksperimenata, int upotrebljeniJezik, int upotrebljenaMetoda) – Izračunava i ispisuje procenat uspešnih kriptoanaliza za različite kombinacije dužine teksta i dužine ključa, jednom od opisanih metoda M_1, M_2, M_3 na engleskom, srpskom (latinica), srpskom (ćirilica) jeziku, pozivom za vrednosti argumenata *upotrebljenaMetoda=1,2,3* redom i *upotrebljeniJezik=1,2,3* redom.

Navedena metoda predstavlja implementaciju eksperimenta 4 iz 4.2. U svakom od merenja kao poruka koristi se slučajno odabrani segment teksta iz baze (koja sadrži preko milion karaktera i sastoji se od nekoliko desetina tekstova napisanih na korišćenom prirodnom jeziku), a za ključ slučajno odabrana niska karaktera. Segmenti su birani generisanjem slučajnog broja, pozivom metode *generisiSlucajanBroj(broj_od, broj_do)* koji predstavlja indeks njihove pozicije u bazi za unapred određenu dužinu. Ključevi su birani generisanjem svakog slova posebno, pozivom metode *generisiSlucajanKljuc(dužina_ključa)*. Odabrani segment teksta najpre se šifruje slučajno generisanim ključem, a zatim se dešifruje ključem dobijenim kriptoanalizom šifrata. Pri kriptoanalizi može se koristiti jedna od tri metode za procenu najverovatnije dužine ključa. U slučaju poklapanja polaznog teksta i teksta dobijenog dekriptovanjem, kriptoanaliza se smatra uspešnom.

generisiSlucajanBroj(int, int)	generisiSlucajanBroj(int broj_od,int broj_do) – Vraća generisani slučajan broj u opsegu (*broj_od, broj_do*). Poziva četiri puta Java generator slučajnih brojeva, xor-uje generisane vrednosti i apsolutnu vrednost dobijenog rezultata smešta u zadati opseg *(broj_od, broj_do)*, deljenjem po modulu razlike: broj_do-broj_od.
generisiSlucajanKljuc(int)	generisiSlucajanKljuc(int n) – Vraća slučajan ključ, dužine *n* takav da svako njegovo slovo odgovara slučajnom broju, generisanom pozivom metode *generisiSlucajanBroj(0, duzinaPisma())* .

Nakon određivanja Vižnerovog ključa, vrši se dešifrovanje šifrata pretpostavljenim ključem i dobija se tekst moguće poruke. Na osnovu vrednosti njegovog indeksa koincidencije, oslanjajući se na statističke karakteristike primenjenog jezika, može se utvrditi da li je dobijeni tekst "smislen" za primenjeni jezik, tj. da li je vrednost njegovog indeksa koincidencije pripada intervalu poverenja indeksa koincidencije primenjenog jezika.

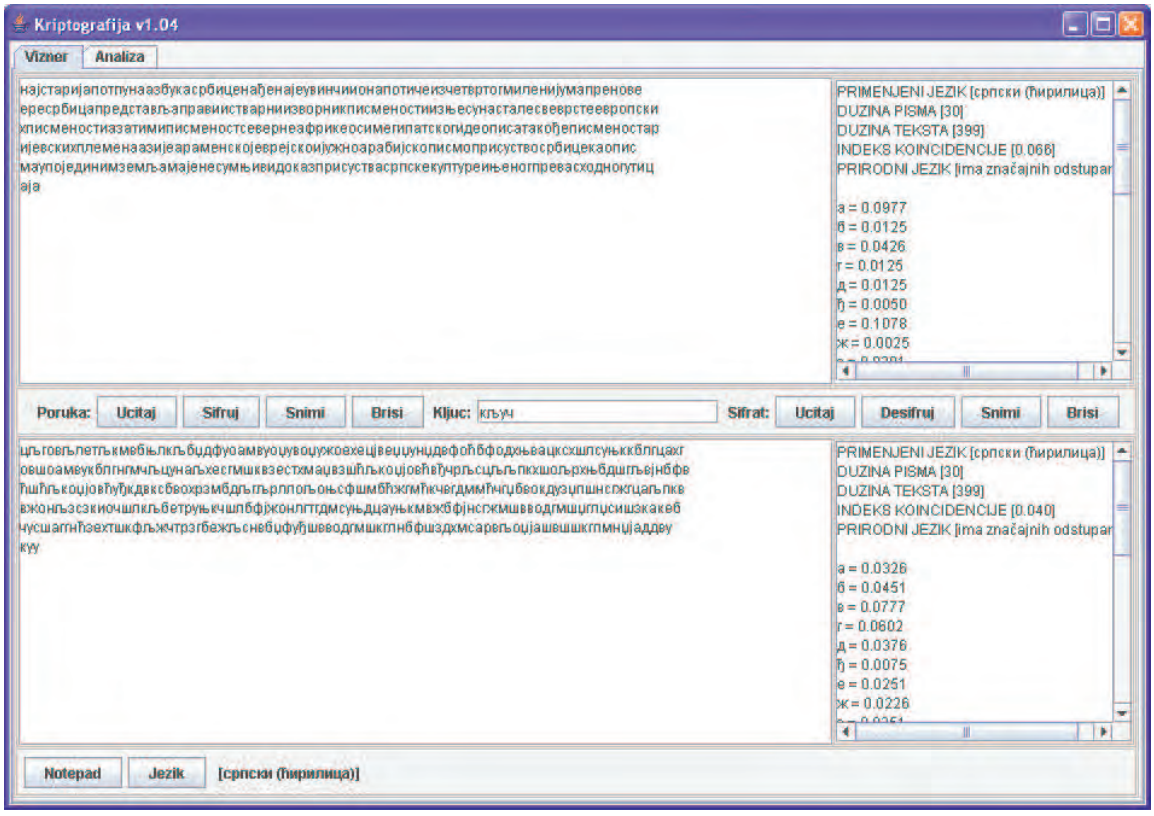
5.4 GUI.java

Grafički deo programa pokreće se pozivom funkcije *main()* klase GUI. Rad u grafičko-korisničkom okruženju podrazumeva prikazivanje prozora aplikacije i omogućava jednostavno korišćenje funkcionalnosti programa.

Osnovni prozor aplikacije sastoji se od dva podprozora: Vizner i Analiza.

5.4.1 Podprozor Vizner

Podprozor Vizner se sastoji iz dva odvojena panela, postavljena sa gornje i donje strane podprozora.



Gornji deo prozora odnosi se samo na poruku, donji samo na šifrat. Između njih se nalaze upravljačka dugmad za poruku i šifrat i polje za tekst ključa. Deo koji se odnosi na osnovnu poruku podeljen je na:

- levi deo, predviđen za prikazivanje teksta poruke
- desni deo, predviđen za prikazivanje statistike poruke

Slično, deo koji se odnosi na šifrat podeljen je na:

- levi deo, predviđen za prikazivanje teksta šifrata
- desni deo, predviđen za prikazivanje statistike šifrata

Osnovni panel Vizner prozora sadrži dugmad:

- Klikom na dugme **Notepad** otvara se prozor Notepad editora, čime se omogućava brzo kreiranje novog tekstualnog dokumenta *.txt. Uneti tekst treba snimiti kao Unicode karakter set.
- Klikom na dugme **Jezik** otvara se dijalog prozor za izbor prirodnog jezika na kome korisnik želi da radi. Podrazumevani direktorijum iz koga se čitaju fajlovi sa definicijom jezika je #jezici, kao što je opisano u 5.1.

  Nakon izbora odgovarajućeg .txt fajla za definisanje jezika, otvara se dijalog prozor za izbor tipičnog teksta za taj jezik (čiji će indeks koincidencije biti predstavnik za taj prirodni jezik). Podrazumevani direktorijum iz koga se čitaju tipični tekstovi je #jezici, koji sadrži tri poddirektorijuma (english, srpski_latinica, srpski_cirilica) i izbor se vrši u zavisnosti od definisanog jezika.

  Izborom jezika, kreira se objekat klase Obrada, za koji je definisan samo jezik, definicija pisma i prethodno izračunati intervali poverenja na dovoljno velikom uzorku, kao što je opisano u 5.1 i na panelu se ispisuje opis izabranog jezika.

Napomena: Nijedno dugme, osim Notepad, na prozoru nije aktivno pre izbora jezika u kome će korisnik raditi.

Nijedno polje, osim polja ključ ne dozvoljava unos, ni kopiranje teksta iz drugih aplikacija. Tekst se, i u slučaju poruke i u slučaju šifrata, može uneti ili izbrisati isključivo korišćenjem kontrolnih dugmadi, a statistika teksta poruke, odnosno šifrata se izračunava pri svakoj promeni teksta.

Kontrole za poruku (levo od polja ključ):

- Klikom na dugme **Učitaj** otvara se prozor za dijalog za izbor teksta koji se želi šifrovati. Podrazumevani direktorijum za poruke je je #dokumenti\poruke, koji sadrži tri poddirektorijuma (english, srpski_latinica, srpski_cirilica) i izbor se vrši u zavisnosti od definisanog jezika. Odmah po učitavanju teksta odabranog dokumenta u levi deo, desni deo se popunjava statistikom jezika, izračunatom za taj tekst.
- Klikom na dugme **Šifruj** učitani tekst se šifruje tekstom prikazanim u polju **ključ**. Tekst koji predstavlja šifrovanu poruku se automatski ispisuje u levom delu predviđenom za šifrat, a njegova statistika u desnom delu.
- Klikom na dugme **Snimi** otvara se dijalog prozor koji omogućava čuvanje teksta trenutno ispisanog u polju za poruku, u zaseban fajl. Podrazumevani direktorijum je

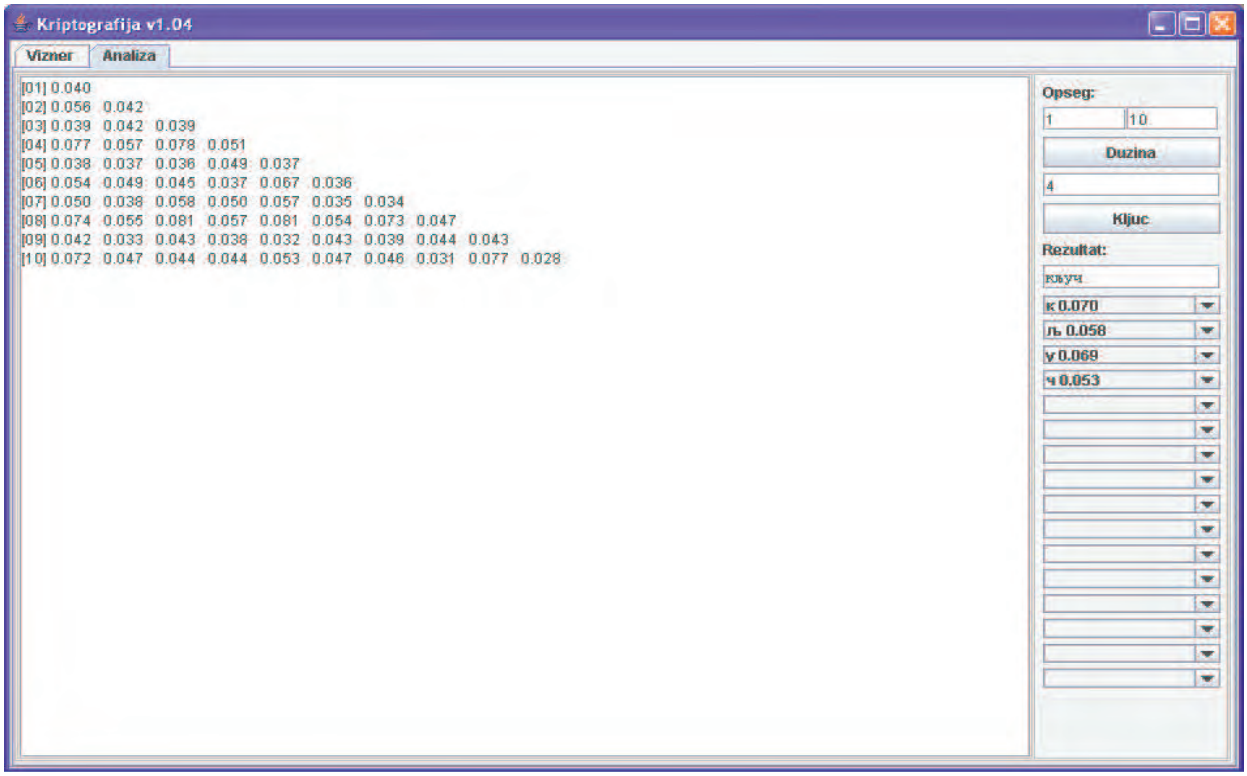
#dokumenti\poruke. Ova funkcionalnost se koristi kod dekriptiranja učitanog šifrata, i omogućava čuvanje više potencijalnih poruka.

- Klikom na dugme **Briši**, sa prozora se brišu svi podaci koji se odnose na taj dokument - i sadržaj teksta i odgovarajuća statistika.

Kontrole za šifrat (desno od polja ključ):

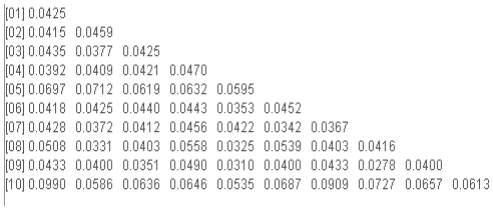
- Klikom na dugme **Učitaj** otvara se prozor za dijalog za izbor teksta koji se želi dešifrovati. Podrazumevani direktorijum je #dokumenti\šifrati, koji sadrži tri poddirektorijuma (english, srpski_latinica, srpski_cirilica) i izbor se vrši u zavisnosti od definisanog jezika. Odmah po učitavanju teksta odabranog dokumenta u levi deo, desni deo se popunjava izračunatom statiskom za taj šifrat.
- Klikom na dugme **Dešifruj** učitani tekst se dešifruje tekstom prikazanim u polju **ključ**. Tekst koji predstavlja dešifrovanu poruku se automatski ispisuje u levom delu predviđenom za tekst poruke, a njegova statistika u desnom delu.
- Klikom na dugme **Snimi** otvara se dijalog prozor koji omogućava čuvanje teksta trenutno ispisanog u polju za šifrat, u zaseban fajl. Podrazumevani direktorijum je #dokumenti\šifrati.
- Klikom na dugme **Briši**, sa prozora Vizner se brišu svi podaci koji se odnose na šifrat, kao i odgovarajuća analiza iz prozora Analiza.

5.4.2 Podprozor Analiza



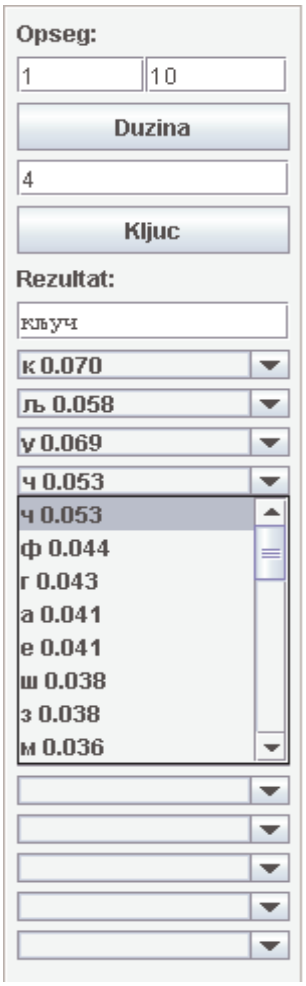
Podprozor Analiza se sastoji iz dva dela:

levo je prostor za ispis rezultata kriptoanalize teksta šifrata iz prozora Vizner (Tabela 3.5).



Prostor za ispis rezultata kriptoanalize sadrži vrednosti indeksa koincidencije za dužine ključa u zadatom opsegu.

desno su polja i kontrolna dugmad za određivanje ključa, kojim je šifrovan tekst šifrata iz prozora Analiza.



Opseg za dužinu ključa sadrži dva tekstualna polja u koja se unose minimalna, odnosno maksimalna vrednost, za koje će se razmatrati dužina ključa. Korisnik može sam zadavati opseg za dužinu ključa, a podrazumevani opseg je [1..10].

Klikom na dugme *Dužina* ispisuje se pretpostavljena dužina ključa na osnovu kriptoanalize, na osnovu(10), u tekstualno polje predviđeno za dužinu ključa.

Korisnik može i sam zadati pretpostavljenu dužinu ključa, direktnim upisom u polje za dužinu ključa.

U delu za ispis rezultata kriptoanalze, ispisuju se vrednosti indeksa koincidencije, izračunate za sve dužine ključa iz zadatog opsega.

Klikom na dugme *Ključ* u tekstualnom polju, označenom kao *Rezltat*, generiše se najverovatniji ključ, dužine koja je ispisana u polju za dužinu ključa.

Istovremeno se popunjava niz padajućih listi, od kojih se svaka lista odnosi na jedno slovo pretpostavljenog ključa. Prikazane vrednosti u svakoj od listi su slovo pretpostavljenog ključa i indeks koincidencije, izračunat dešifrovanjem za to slovo, na osnovu (12). Svaka od listi sadrži sva slova pisma tekućeg jezika i njima odgovarajuće vrednosti indeksa koincidencije, koji su sortirani u opadajućem poretku.

Ovim je data mogućnost interakcije korisnika, koji na osnovu izračunatih vrednosti indeksa koincidencije, može izborom iz padajuće liste da utiče na izbor slova Vižnerovog ključa.

6 Zaključak

Izvršena je detaljna analiza jednog algoritma za napad na Vižnerovu šifru, zasnovanog na Fridmanovoj metodi indeksa koincidencije prirodnih jezika.

Implementacija algoritma sadrži korisnički interfejs za šifrovanje, dešifrovanje i kriptoanalizu šifre pomeranja i Vižnerove šifre. Pored toga, program daje i mogućnost interakcije korisnika tokom rada, izborom dužine Vižnerovog ključa ili pojedinačnih slova ključa. Razvijene su definicije i primeri za dva prirodna jezika, a program omogućava rad na proizvoljnom prirodnom jeziku.

Za potrebe rada, eksperimentalnim izračunavanjima dobijene su i prikazane vrednosti koje opisuju statističke osobine primenjenih prirodnih jezika – raspodelu učestanosti pojavljivanja slova jezika i indeks koincidencije jezika. Korišćeni su uzorci književnih tekstova na primenjenim prirodnim jezicima.

Predstavljene su tri metode za određivanje dužine Vižnerovog ključa. Eksperimentalnim izračunavanjima za unapred fiksirane dužine ključeva, upoređeni su rezultati uspešnosti algoritma u zavisnosti od upotrebljene metode.

Izvršena je analiza uspešnosti algoritma kriptoanalize, u zavisnosti od odnosa dužine poruke i dužine ključa, eksperimentalnim izračunavanjima nad slučajno generisanim porukama i ključevima. Zaključak je da uspešnost algoritma zavisi od broja ponavljanja ključa u tekstu poruke. Za uspešnost algoritma preko 50% procenjuje se da je potrebno bar trideset ponavljanja ključa u tekstu poruke.

U cilju dobijanja merodavnijih podataka o statistici jezika, mogu se nabaviti reprezentativniji tekstovi za izabrani jezik, sa uravnoteženim udelima raznih žanrova, autora, stilova.

Razvijeni program moguće je poboljšati na nekoliko načina. Izbor jedne od tri pomenute metode za određivanje dužine ključa, tokom postupka kriptoanalize može se vršiti interaktivno. Mogu se nabaviti rezultati statističkih karakteristika jezika drugih autora i ugraditi u program.

7 Literatura

[1] Schneier, Bruce, Primenjena kriptografija, Mikro knjiga, Beograd, 2007
naslov originala: Applied Cryptography

[2] Singh, Simon, Knjiga o šiframa, Plato, Beograd 2003
naslov originala: The Code Book

[3] Stinson, Douglas, Cryptography: Theory and Practice, CRC Press, 1995

[4] Živković, Miodrag, Algoritmi, Matematički fakultet, 2000

[5] Saverchenko, Ilya, Classical Cryptography, JASS 05, 2005

[6] Merkle, Milan; Vasić, Petar, Verovatnoća i statistika, Akademska misao, Beograd, 2001