



**Univerzitet u Beogradu
Elektrotehnički fakultet**

Vizuelni prikaz rada Hill-ove šifre i njene kriptoanalize

DIPLOMSKI RAD

Kandidat:
Kulezić Ivan 0166/2013

Profesor:
Dr Stanisavljević Žarko, docent

Beograd
Jul 2018.

Sadržaj

1. UVOD.....	3
2. OPIS ALGORITMA.....	5
2.1. ISTORIJAT RAZVOJA KRIPTOGRAFIJE.....	5
2.2. HILL-OVA ŠIFRA.....	9
2.2.1. Enkripcija.....	10
2.2.2. Dekripcija	11
2.2.3. Kriptoanaliza	14
2.2.3.1. Napad samo šifrovanim tekstom	14
2.2.3.2. Napad poznatim originalnim tekstom	15
2.2.3.3. Napad odabranim originalnim tekstom	16
3. IMPLEMENTACIJA	17
3.1. PREGLED PAKETA U SISTEMU.....	17
3.2. PREGLED KLASA U SISTEMU.....	18
3.3. EKSTERNE BIBLIOTEKE	21
3.4. FORMAT TEST FAJLOVA	22
3.5. POKRETANJE SIMULATORA	22
4. NAČINI KORIŠĆENJA.....	23
4.1. SIMULACIJA ENKRIPCije	24
4.2. SIMULACIJA DEKRIPCije	26
4.3. MOD ZA TESTIRANJE.....	28
4.3.1. Test procesa enkripcije	29
4.3.2. Test procesa dekripcije	30
4.4. SIMULACIJA NAPADA	31
5. ZAKLJUČAK	34
LITERATURA	35

1. UVOD

Iako se koristi već hiljadama godina da sakrije tajne poruke, kriptografija je mlada nauka. Sistematsko proučavanje kriptografije kao nauke (i možda umetnosti) je počelo pre oko stotinak godina. Kriptografija se bavi izučavanjem tehnika bezbedne komunikacije u prisustvu trećeg lica koje je obično označeno kao neprijatelj. Pre modernog doba kriptografija je bila sinonim za enkripciju, konverziju informacija u oblik razumljiv samo primaocu poruke, dok moderna kriptografija podrazumeva mnogo više od samog šifrovanja poruka. Ona se bavi konstrukcijom, analizom i dizajnom protokola koji sprečavaju neprijatelja ili javnost da čitaju privatne poruke. To izučavanje obuhvata i različite aspekte bezbednosti informacija kao što su: poverljivost i integritet podataka, autentikacija, kao i neporecivost porekla podataka.

Danas, u doba univerzalne računarske povezanosti, hakera, elektronskog prisluškivanja i elektronskih prevara, ne postoji trenutak u kome zaštita podataka nije bitna. Eksplozivni rast računarskih sistema i njihove povezanosti putem interneta je povećao zavisnost korisnika, kako organizacija tako i pojedinaca, i informacija koje se prenose i skladište putem navedenih sistema. Od velikog je značaja da se podaci i drugi resursi zaštite od otkrivanja, da se garantuje autentičnost podataka i poruka i da se sistemi zaštite od mrežnih napada. Glavni cilj je obezbediti da se poverenje koje postoji u fizičkom svetu prenese u elektronski svet. Napredak kriptografije i sigurnosti do danas je doveo do razvoja praktičnih i lako dostupnih aplikacija za ostvarivanje tog cilja.

Hill-ova šifra^[1] nastala je u nastojanju da se prevaziđe zajednička mana svih prethodnih metoda šifrovanja. Naime sve, do tada poznate, šifre zasnivale su se na principu supstitucije znakova drugim znakovima i mešanjem ili premeštanjem redosleda znakova u poruci. To je omogućilo da se sve šifre mogu probiti jednostavnom analizom učestanosti pojavljivanja znakova u tekstu. Šifrovanje Hill-ovom šifrom^[1] se zasniva na množenju matrica. To je prva poznata upotreba matematike u kombinaciji sa kriptografijom. Originalnost ove ideje se zasniva u tome što se šifruju blokovi originalnog teksta što onemogućava merenje učestanosti pojavljivanja znakova kao i ostale tradicionalne metode kriptanalize. Pored toga ovo je prva šifra koja je mogla da šifruje blokove veličine 3 ili više znakova i predstavlja preteču savremenih blok šifri. Korišćenje matematike da se šifruje tekst je jedna od velikih prekretnica u istoriji kriptografije. Kombinovanjem matematike i kriptografije otvoren je novi opseg mogućnosti koji je vremenom doveo do kriptografije kakvu poznajemo danas.

Cilj ovog rada je konstrukcija simulatora za vizuelni prikaz rada Hill-ove šifre^[1] i njene kriptanalize. Zadatak autora je da projektuje i implementira edukativni softver, u vidu simulatora, koji će omogućiti korisnicima da razumeju i ovladaju algoritmom za kriptovanje i dekriptovanje Hill-ove šifre^[1], kao i njenom kript analizom. Pored edukativne strane, simulator treba da obezbedi i način da se testira stečeno znanje.

U drugoj glavi je dat istorijat razvoja kriptografije kao i detaljan opis algoritma za kriptovanje i dekriptovanje Hill-ovom^[1] šifrom sa primerima. Detaljnije su opisane neke interesantne matematičke operacije, kao i skup podataka nad kojima se radi. Opis obuhvata sledeće faze: izbor ključa, šifrovanje poruke odabranim ključem, računanje inverznog ključa i dešifrovanje poruke inverznim ključem. Svaka od ovih faza je prvo objašnjena, a zatim i prikazana na primeru. Pored opisa algoritma dat je i opis mogućih napada na ovu šifru sa stanovišta kriptanalize.

U trećoj glavi se opisuje kompletna implementacija sistema. Navedene su korišćene tehnologije, kao i razlozi za njihov izbor. Dati su UML dijagrami organizacije koda po paketima, kao i klasni dijagrami.

U četvrtoj glavi je dat način korišćenja sistema. Predstavljen je početni prozor koji se dobija pokretanjem aplikacije, a zatim i svi ostali kroz koje se korisnik kreće prilikom procesa enkripcije i dekripcije. Pored toga prikazana je i simulacija jednog od napada na zadati algoritam, kao i mod za testiranje znanja korisnika.

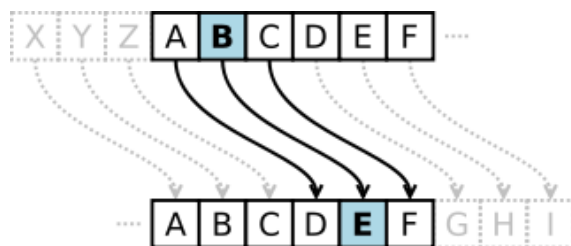
U petoj glavi daje se zaključak, kao kritički osvrt na ispunjenje ciljeva postavljenih na početku ovog rada, kao i rezime svega urađenog. Takođe se daje i procena korišćenog alata i iznose se uočene prednosti i nedostaci.

2. OPIS ALGORITMA

2.1. ISTORIJAT RAZVOJA KRIPTOGRAFIJE

Prvi poznati slučaj korišćenja jednog vida kriptografije je pronađen u natpisu uklesanom oko 1900. godine pre nove ere, u grobnici Hnumhotepa II, u Egiptu. Natpis je, na pojedinim mestima, umesto uobičajenih sadržao neobične hijeroglife. Svrha nije bila da se sakrije neka poruka već da se promeni forma teksta kako bi izgledao uzvišenije. Iako natpis nije bio vrsta tajne poruke, sadržao je elemente transformacije originalnog teksta. Dokazi upotrebe neke vrste kriptografije pronađeni su u svim većim ranim civilizacijama. Arthašastra, klasičan primer zakonika starih civilizacija, opisuje špijunsku službu u Indiji i pominje prosleđivanje zadataka špijunima šifrovanim porukama.

Poznato je da je, oko 100. godine pre nove ere, Julije Cezar koristio vrstu enkripcije da prenese tajne poruke svojim generalima na bojnem polju. Ova supstituciona šifra, verovatno najpoznatija u akademskoj literaturi, je poznata kao Cezarova šifra^[1]. Zasniva se na kružnom pomeranju slova za određeni broj mesta. Svako slovo originalne poruke bilo je zamenjivano drugim slovom koje bi učestvovalo u formiranju šifrovane poruke. Slika 2.1 prikazuje mapiranje slova originalne poruke u slova šifrovane poruke. Iako je primetiti da ovakva šifra ne zavisi od tajnosti korišćenog ključa već od sistema korišćenog za šifrovanje. Jednom kada je sistem poznat, sve šifrovane poruke se mogu dešifrovati sa lakoćom probajući svih 25 ključeva.



Slika 2.1. Mapiranje slova u Cezarovoj šifri¹

Jedno poboljšanje Cezarove šifre^[1] je monoalfabetska šifra^[1]. Umesto jednostavnog pomeranja slova za određen broj mesta, alfabet se promeša (permutuju se sva slova alfabeta) i svako slovo originalne poruke se preslikava u različito slovo šifrovane poruke. Slika 2.2 prikazuje jedno moguće mapiranje slova. Ključ je dužine 26 slova što znači da postoji 26! mogućih ključeva. Veliki broj ključeva sugerise da je metod šifrovanja siguran, ali to nije tačno.

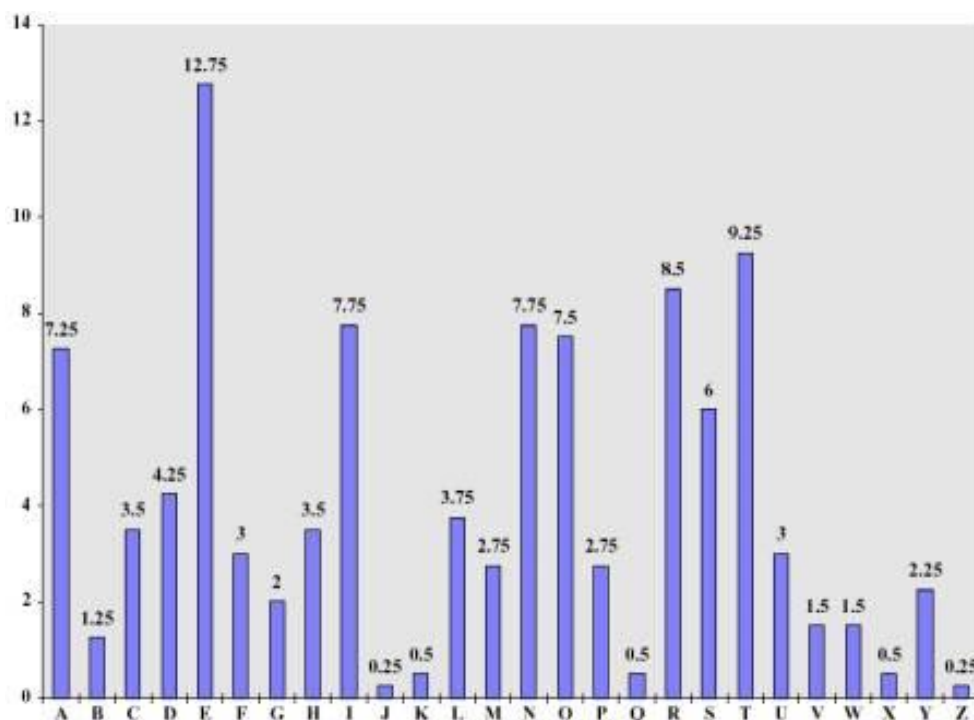
¹ Preuzeto 03.07.2018. sa:

https://sh.wikipedia.org/wiki/Cezarova_%C5%A1ifra#/media/File:Caesar3.svg

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

Slika 2.2. Mapiranje slova u monoalfabetskoj šifri

Osnovni problem supstitucionih šifara je što ne menjaju relativnu učestalost slova. Jezici su redundantni i slova se ne upotrebljavaju sa istom učestanošću. Supstitucione šifre se lako dešifruju analizom šifrovanog teksta, pri čemu se odredi učestanost slova i uporedi sa poznatom statistikom jezika. Slika 2.3 prikazuje učestanost slova u engleskom jeziku.



Slika 2.3. Učestanost slova u engleskom jeziku²

Kasnije su se javili pokušaji zaravnjivanja učestalosti slova korišćenjem više alfabeti. Prvu takvu šifru dao je Vigenère^[1] u 16. veku gde svako slovo ključa određuje alfabet koji se koristi. Slika 2.4 prikazuje pomoćnu tablicu za supstituciju slova originalnog teksta slovima šifrovanog teksta. Zasniva se na više ponovljenih Cezarovih šifara^[1]. Alfabeti se ponavljaju posle određenog slova u poruci. Iako je učestanost slova narušena, nije i potpuno izgubljena. Iako poboljšana ova šifra i dalje nije bila dovoljno dobra. Iz ponavljanja u šifrovanom tekstu može se izvući podatak o dužini perioda. Dalje se korišćenjem frekvencije slova ova šifra razbija kao niz monoalfabetskih šifara^[1].

² Preuzeto 05.06.2018. sa: <http://sjsu.rudyruicker.com/~haile.eyob/paper/>

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

Slika 2.4. Tablica za šifrovanje Vigenère-ovom šifrom³

Kako veliki broj ključeva u monoalfabetskoj šifri^[1] nije obezbedio sigurnost, pristupilo se šifrovanju većeg broja slova odjednom. Primer ovakve šifre je Playfair^[1] šifra iz 1854. godine. Kao tabela za zamenu koristi se matrica dimenzija 5×5 , čiji se redovi ispune odabranim ključem, uz izostavljanje duplikata, i dopuni neiskorišćenim slovima jezika. Originalni tekst se šifrjuje uzimajući dva po dva slova, zamenom posmatranog slova slovom iz tabele preslikavanja. Odgovarajuće slovo se bira iz tablice na osnovu para slova gde red određuje posmatrano slovo, a kolonu par posmatranog slova. Slika 2.5 prikazuje ključ matricu i prikaz zamene slova. Postoji 26×26 parova slova što znači da je potrebno isto toliko tabela za analizu učestanosti. Iako je široko korišćena u prvom svetskom ratu ova šifra nije zaživela jer je i dalje sadržala mnoge elemente originalnog teksta. Zbog statistike na dovoljno velikom uzorku šifrovanog teksta, od nekoliko stotina karaktera, ova šifra je uspešno probijena.

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

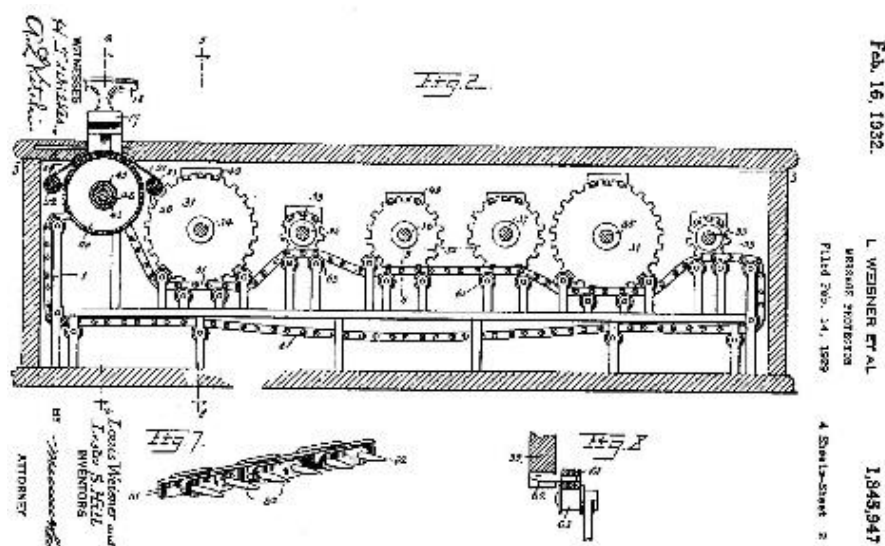
Slika 2.5. Tablica za šifrovanje Plejfer šifrom⁴

Šifrovanje bloka dužine veće do 2 slova bi još više smanjilo učestalost pojavljivanja slova u šifrovanom tekstu. Postojali su neki pokušaji proširenja Playfair^[1] šifre, ali nijedan nije bio uspešan. 1929. godine Lester Hill dao je opis kriptografskog sistema koji je mogao da šifrjuje blokove originalnog teksta proizvoljne dužine. Sistem je zasnovan na linearnoj algebri. Kao ključ, koristi se matrica dimenzija $n \times n$, gde je n dužina bloka koji se šifrjuje. Originalni tekst

³ Preuzeto 05.06.2018. sa: <https://www.egress.com/blog/encryption-101-the-vigenere-cipher>

⁴ Preuzeto 05.06.2018. sa: <https://playfaircipher101.weebly.com/encryption.html>

se podeli u blokove koji odgovaraju dužini ključa, a zatim se blok množi matricom ključa što daje blok šifrovanog teksta kao rezultat. Šifrovani tekst se dekriptuje množenjem blokova šifrovane poruke matricom inverznom matrici ključa. Računanje inverznog ključa je veoma teško kako zbog dimenzija matrice tako i zbog toga što nema svaka matrica inverznu matricu. Hill je ovaj problem pokušao da prevaziđe predlogom da se kao ključ koriste involutivne matrice, što bi značajno smanjilo skup mogućih ključeva, ali izbacilo potrebu za računanjem inverznog ključa.



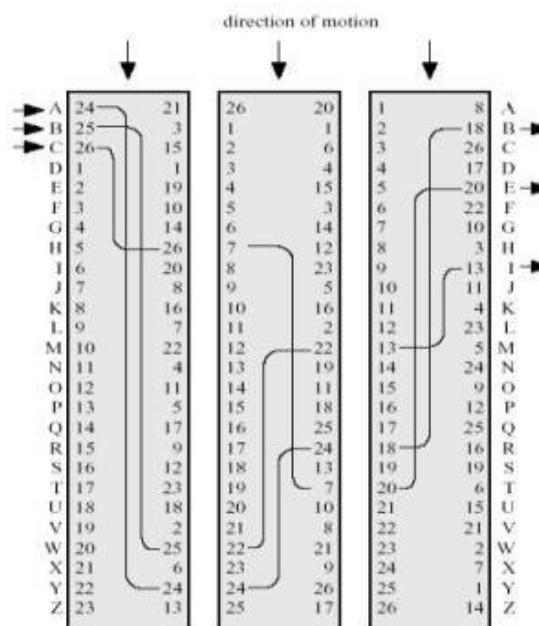
Slika 2.6. Šema Hill-ove mašine⁵

U praksi se pokazalo da je šifrovanje Hill-ovom šifrom^[1] previše komplikovano za svakodnevnu upotrebu. Zahteva veliki broj operacija koje bi čovek morao ručno da izvede. Iz tog razloga, Hill i njegov kolega su dali mehaničku implementaciju koja je radila operaciju šifrovanja koristeći matricu 6×6 . Slika 2.6 predstavlja šemu mehaničke implementacije Hill-ove šifre^[1]. Glavni problem mašine je bilo to što je položaj zupčanika bio fiksiran za mašinu a samim tim i ključ kojim se šifrjuje poruka, zbog čega se nikada nije ni prodavala. Hill-ova šifra^[1] je potpuno linearna što znači da neprijatelj koji presretne n^2 parova slova običnog i šifrovanog teksta može da konstruiše i reši linearni sistem jednačina odakle dobija vrednost ključa. Ukoliko sistem nije određen dovoljno je da neprijatelj sazna još nekoliko parova slova originalnog i šifrovanog teksta i lako dobija ključ. Radi dodatne sigurnosti, pre i posle samog šifrovanja, predloženo je da se primeni nelinearna transformacija na sam tekst kako bi se u kombinaciji sa šifrovanjem koje je davala mašina postigao dodatni stepen konfuzije. Iako njegova šifra nikada nije zaživela, Hill je zaslužan za veliki doprinos razvoju i upotrebi matematike u dizajnu i analizi kriptosistema, posebno u oblasti teorije brojeva.

Algoritmi bazirani isključivo na supstituciji nisu dovoljno sigurni zbog jezičkih karakteristika.

⁵ Preuzeto 05.06.2018. sa: <http://www.cs.jhu.edu/~cgarman/Cryptography2.php>

Zato se koristi više algoritama uzastopno, algoritam zamene praćen algoritmom transpozicije, što daje novu, mnogo komplikovaniju, šifru. Ovakve kombinacije se nazivaju produkcionim sistemima^[1] i predstavljaju most između klasičnih i modernih šifri. Klasičan primer produkcionog sistema je rotor mašina^[1], poznatija kao Enigma, koja je korišćena u II Svetskom ratu. Ovde se koristi sistem međusobno povezanih cilindara, gde je izlaz jednog cilindra povezan na ulaz drugog cilindra. Svako ulaznoj i izlaznoj tački se dodeli po jedno slovo engleskog alfabeta. Svaki cilindar rotira kada prethodni napravi pun ciklus. Slika 2.7 prikazuje princip rada rotor mašine^[1] sa 3 cilindra.



Slika 2.7. Princip rada rotor mašine⁶

Moderna kriptografija se u velikoj meri zasniva na matematičkoj teoriji i informatičkoj praksi. Kriptografski algoritmi se dizajniraju na pretpostavci da je teorijski moguće, ali praktično teško probiti takav algoritam sa trenutno dostupnim praktičnim sredstvima. Ovakvi algoritmi se konstantno moraju poboljšavati u skladu sa razvojem teorijske nauke, kao na primer ubrzavanje algoritma za faktORIZACIJU celih brojeva, kao i razvoj brzine računarskih sistema.

2.2. HILL-OVA ŠIFRA

Hill-ova šifra^[1] je jedna od najpoznatijih poligrafskih šifara. Ovo je prva (sistematična ali jednostavna) šifra te vrste koja je mogla da šifruje grupe veće od 2 karaktera. Hill-ova šifra^[1] predstavlja prvi uopšteni metod primene linearne algebre na polialfabetske šifre, na način koji je praktičan.

⁶ Preuzeto i obrađeno 05.06.2018. sa: http://www.brainkart.com/article/Rotor-Machines_8389/

Alfabet Hill-ove šifre^[1] sastoji se od m znakova označenih celim brojevima od 0 do $m - 1$. U ovom radu autor se odlučio da alfabet čine velika slova engleskog jezika kojih ima 26. Na slici 2.8 dat je prikaz konkretnog alfabeta i brojeva u koje se elementi alfabeta preslikavaju. Ne postoji poseban razlog za numerisanje slova alfabeta počevši od 0 u rastućem redosledu. U praksi bi slova bila označena u proizvoljnom poretku, poznatom samo pošiljaocu i primaocu poruke. Zbog jednostavnosti autor je odlučio da zadrži mapiranje prikazano na slici 2.8.

Brojevi u koje se elementi alfabeta Hill-ove šifre^[1] mapiraju su iz skupa $\{0, 1, \dots, m - 1\}$. Ako saberemo ili pomnožimo dva broja iz tog skupa i uzmemo ostatak pri deljenju sa m , dobijamo broj koji pripada istom tom skupu. Navedeni skup, zajedno sa operacijama sabiranja i množenja čini brojni sistem celih brojeva po modulu m , u oznaci Z_m . U konkretnom slučaju brojni sistem korišćen u ovom radu je Z_{26} . Zbog osobina ekvivalencije po modulu m da očuva operacije sabiranja i množenja, možemo rezultat bilo koje od ovih operacija zameniti ostatkom po modulu m . Ukoliko imamo složeniji izraz, koji predstavlja kombinaciju više operacija sabiranja i množenja, možemo konačan rezultat zameniti ostatkom po modulu m . To je potpuno isto kao da smo nakon svake operacije zamenili rezultat ostatkom po modulu m , kada on izađe iz opsega $[0, m - 1]$, i tako u svakom međukoraku do konačnog rezultata.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Slika 2.8. Alfabet Hill-ove šifre

2.2.1. ENKRIPCIJA

Neka je dat alfabet veličine $m, m > 1$ i ceo broj $n, n > 1$. Tada je Hill-ova šifra^[1] reda n data matricom K , dimenzija $n \times n$, čiji elementi pripadaju Z_m . Matrica K predstavlja ključ za ovako definisanu šifru. Za takvu matricu K , Hill-ov algoritam za kriptovanje poruke je sledeći:

1. Podeliti originalnu poruku u i podgrupa dužine n karaktera. Ukoliko i -ta grupa nema n elemenata dopuniti je do n elemenata proizvoljnim karakterom iz alfabeta.
2. Svako slovo zameniti odgovarajućim brojem iz tablice preslikavanja alfabeta.
3. Svaku od i grupa predstaviti u vidu matrice, dimenzija $1 \times n$. Redom pomnožiti svaku od dobijenih matrica matricom K po modulu m .
4. Elemente svake od i matrica, dobijenih iz prethodnog koraka, dimenzija $1 \times n$, zameniti odgovarajućim karakterima iz tabele preslikavanja alfabeta. Spajanjem redom ovako dobijenih i grupa karaktera dobija se šifrovani tekst.

Primer 2.2.1. Za alfabet dat na slici 2.1. i dužinu ključa 3, šifrovati tekst „ZASTITA“ Hill-ovom šifrom^[1]. Kao ključ koristiti matricu $K = \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix}$. Kao karakter za ispunu koristiti „X“.

Rešenje:

Podelimo originalnu poruku na grupe dužine 3. Imamo:

Z A S | T I T | A

Kako poslednja grupa nema 3 karaktera, dodajemo karakter „X“ dva puta. Sada imamo:

Z A S | T I T | A X X

Zamenom svakog slova originalnog teksta odgovarajućim brojem iz tabele mapiranja dobijamo:

25 0 18 | 19 8 19 | 0 23 23

Pretvorimo grupe u matrice i pomnožimo ih matricom ključem po modulu 26. Dobijamo:

$$\begin{bmatrix} 25 & 0 & 18 \end{bmatrix} \times \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} = \begin{bmatrix} 23 & 21 & 10 \end{bmatrix} \pmod{26}$$

$$\begin{bmatrix} 19 & 8 & 19 \end{bmatrix} \times \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} = \begin{bmatrix} 20 & 14 & 23 \end{bmatrix} \pmod{26}$$

$$\begin{bmatrix} 0 & 23 & 23 \end{bmatrix} \times \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} = \begin{bmatrix} 22 & 9 & 22 \end{bmatrix} \pmod{26}$$

Spojimo elemente u redosledu. Imamo:

23 21 10 | 20 14 23 | 22 9 22

Zamenom brojeva odgovarajućim karakterima iz tabele preslikavanja alfabeta dobijamo:

X V K | U O X | W J W

Šifrovani tekst koji odgovara originalnom tekstu „ZASTITA“ je „XVKUOXWJW“.

2.2.2. DEKRIPCIIJA

Za Hill-ovu šifru^[1], transformacija iz šifrovanog u originalni tekst je inverzna operacija operaciji šifrovanja originalnog teksta. Drugim rečima Hill-ovoj šifri^[1] sa ključ matricom K , inverzna operacija je Hill-ova šifra^[1] sa ključ matricom K^{-1} . Matrica inverzna matrici K , u oznaci K^{-1} se računa na sledeći način:

$$K^{-1} = \frac{1}{\det(K)} \text{adj}(K).$$

Kako smo već definisali operacije sabiranja i množenja u Z_m , odreživanje $\text{adj}(K)$ nije

problematično. Postavlja se pitanje kako odrediti $\frac{1}{\det(K)}$. Izraz $\frac{1}{\det(K)}$ se u stvari može zameniti multiplikativnim inverzom broja $\det(K)$. U opštem slučaju Z_m nije polje pa osobina da svaki element iz Z_m ima multiplikativni inverz ne važi. Ta osobina važi za svaki element iz Z_m samo kada je m prost broj. Konkretno za $m = 26$ neće uvek postojati matrica inverzna matrici ključa. Zbog toga treba biti pažljiv pri izboru matrice ključa jer ukoliko ona nema inverznu matricu, šifrovana poruka nikada ne može biti dešifrovana. Često se u praksi alfabet od 26 slova engleskog jezika dopunjuje sa još nekoliko simbola („“, „_“ i „?“) da bi m bio prost broj. Time se prevazilazi ovaj problem jer u skupu Z_{29} svaki element ima multiplikativni inverz. U nastavku su dati primeri koji ilustruju računanje inverznog ključa i dešifrovanje Hill-ovom šifrom.

Primer 2.2.2. Za ključ $K = \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix}$ odrediti K^{-1} .

Rešenje:

Najpre odredimo determinantu matrice K :

$$\begin{aligned} \det(K) &= 3 \cdot \begin{vmatrix} 6 & 15 \\ 17 & 21 \end{vmatrix} - 25 \cdot \begin{vmatrix} 23 & 15 \\ 13 & 21 \end{vmatrix} + 4 \cdot \begin{vmatrix} 23 & 6 \\ 13 & 17 \end{vmatrix} \\ &= 3 \cdot (-129) - 25 \cdot 288 + 4 \cdot 313 \\ &= -387 - 7200 + 1252 \\ &= -6335 \\ &= 9 \pmod{26} \end{aligned}$$

Sada treba pronaći multiplikativni inverz broja 9. Najpre proverimo da li on postoji. Računamo $\gcd(9, 26) = 1$ što znači da postoji multiplikativni inverz. Sada ga računamo koristeći prošireni Euklidov algoritam^[2]. Imamo:

$$26 = 9 \cdot (2) + 8 \quad (1)$$

$$9 = 8 \cdot (1) + 1 \quad (2)$$

$$8 = 1 \cdot (8) + 0 \quad (3)$$

Polazeći od (2) i smenjujući (1) u (2) dobijamo:

$$\begin{aligned} 1 &= 9 + 8 \cdot (-1) = 9 + (26 + 9 \cdot (-2)) \cdot (-1) = \\ &= 9 + 26 \cdot (-1) + 9 \cdot (2) = 9 \cdot (3) + 26 \cdot (-1) \end{aligned}$$

Ako primenimo operaciju $\text{mod } 26$ dobijamo:

$$1 = 9 \cdot 3 \pmod{26}$$

Konačno, 3 je multiplikativni inverz broja 9. Sada je $\frac{1}{\det(K)} = 3$.

Sledeći korak je računanje $\text{adj}(K)$. Adjungovana matrica se dobija transponovanjem matrice kofaktora, u oznaci C^T . Najpre odredimo matricu kofaktora. To je matrica dimenzija

3×3 čiji je svaki element $(i, j) = (-1)^{i+j} \cdot M_{ij}$, $i \in \{0, 1, 2\}$, $j \in \{0, 1, 2\}$. M_{ij} je determinanta matrice dimenzija 2×2 koja se dobija kada se iz matrice K izostave red i i kolona j . Konkretno imamo:

$$adj(K) = C^T = \begin{bmatrix} (-1)^{0+0}M_{00} & (-1)^{0+1}M_{01} & (-1)^{0+2}M_{02} \\ (-1)^{1+0}M_{10} & (-1)^{1+1}M_{11} & (-1)^{1+2}M_{12} \\ (-1)^{2+0}M_{20} & (-1)^{2+1}M_{21} & (-1)^{2+2}M_{22} \end{bmatrix}^T$$

$$C = \begin{bmatrix} + \begin{vmatrix} 6 & 15 \\ 17 & 21 \end{vmatrix} & - \begin{vmatrix} 23 & 15 \\ 13 & 21 \end{vmatrix} & + \begin{vmatrix} 23 & 6 \\ 13 & 17 \end{vmatrix} \\ - \begin{vmatrix} 25 & 4 \\ 17 & 21 \end{vmatrix} & + \begin{vmatrix} 3 & 4 \\ 13 & 21 \end{vmatrix} & - \begin{vmatrix} 3 & 25 \\ 13 & 17 \end{vmatrix} \\ + \begin{vmatrix} 25 & 4 \\ 6 & 15 \end{vmatrix} & - \begin{vmatrix} 3 & 4 \\ 23 & 15 \end{vmatrix} & + \begin{vmatrix} 3 & 25 \\ 23 & 6 \end{vmatrix} \end{bmatrix}$$

$$C = \begin{bmatrix} -129 & -288 & 313 \\ -457 & 11 & 274 \\ 351 & 47 & -557 \end{bmatrix}$$

$$adj(K) = C^T = \begin{bmatrix} -129 & -457 & 351 \\ -288 & 11 & 47 \\ 313 & 274 & -557 \end{bmatrix} = \begin{bmatrix} 1 & 11 & 13 \\ 24 & 11 & 21 \\ 1 & 14 & 15 \end{bmatrix} (mod\ 26)$$

Konačno:

$$K^{-1} = 3 \cdot \begin{bmatrix} 1 & 11 & 13 \\ 24 & 11 & 21 \\ 1 & 14 & 15 \end{bmatrix} = \begin{bmatrix} 3 & 33 & 39 \\ 72 & 33 & 63 \\ 3 & 42 & 45 \end{bmatrix} = \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} (mod\ 26)$$

Primer 2.2.3. Koristeći inverzni ključ iz primera 2.2.1. dešifrovati poruku „XVKUOXWJW”.

Rešenje:

Podelimo šifrovanu poruku na grupe dužine 3. Imamo:

$$X\ V\ K\ |\ U\ O\ X\ |\ W\ J\ W$$

Zamenimo svako slovo šifre odgovarajućim brojem iz tabele mapiranja. Dobijamo:

$$23\ 21\ 10\ |\ 20\ 14\ 23\ |\ 22\ 9\ 22$$

Pretvorimo grupe u matrice i pomnožimo ih matricom ključem po modulu 26. Dobijamo:

$$\begin{bmatrix} 23 & 21 & 10 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} = \begin{bmatrix} 25 & 0 & 18 \end{bmatrix} (mod\ 26)$$

$$\begin{bmatrix} 20 & 14 & 23 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} = \begin{bmatrix} 19 & 8 & 19 \end{bmatrix} (mod\ 26)$$

$$\begin{bmatrix} 22 & 9 & 22 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} = \begin{bmatrix} 0 & 23 & 23 \end{bmatrix} (mod\ 26)$$

Spojimo elemente u redosledu. Imamo:

25 0 18 | 19 8 19 | 0 23 23

Zamenom brojeva odgovarajućim karakterima iz tabele preslikavanja alfabeta dobijamo:

Z A S | T I T | A X X

Originalni tekst koji odgovara šifrovanom tekstu „XVKUOXWJW” je „ZASTITAXX”.

2.2.3. KRIPTOANALIZA

Kriptoanaliza^[1] je proces izučavanja šifara sa ciljem da se neautorizovane osobe spreče u čitanju enkriptovanih poruka. Iako se smatralo da su prve izmišljene šifre (mahom su to bile supstitucione šifre) bezbedne od napada neprijatelja, izmišljene su razne tehnike za pronalaženje ključa, što omogućava dešifrovanje svih enkriptovanih poruka. U nastavku je dat opis sledećih napada na Hill-ovu šifru^[1]:

- napad samo šifrovanim tekstom^[3]
- napad poznatim originalnim tekstom^[4]
- napad odabranim originalnim tekstom^[5]

2.2.3.1. NAPAD SAMO ŠIFROVANIM TEKSTOM

Iscrpna pretraga^[3] je najjednostavniji metod za probijanje šifara. Ovo je ujedno i jedini mogući napad na Hill-ovu šifru^[1] kada je poznat samo šifrovani tekst. Tehnika iscrpne pretrage je moguća u većini kriptografskih sistema jer imaju konačan prostor ključeva. To omogućava da se probaju svi mogući ključevi dok se ne pronađe traženi.

U slučaju Hill-ove šifre^[1] postoji $m^{n \times n}$ ključeva gde je m veličina alfabeta a n dužina ključa, u našem slučaju $26^{n \times n}$. Naravno treba istaći da nisu svi ključevi kandidati jer nemaju svi inverznu matricu. To svakako ne isključuje isprobavanje istih jer je potrebno proveriti da li su oni invertibilni pa tek ako jesu pronaći inverz i isprobati ključ. Iako, na prvi pogled, deluje kao dobra ideja obično se ispostavi da je pronalazak ključa na ovaj način nedostižan jer je broj ključeva koje treba isprobati veliki. Zbog toga se dešava da se ovakav postupak u praksi ne može obaviti u razumnom vremenu.

Trivijalna iscrpna pretraga prostora ključeva za Hill-ovu šifru^[1] reda n sa alfabetom od 26 karaktera zahteva 26^{n^2} množenja matrica. Primetimo da, uzimajući u obzir invertibilnost matrica, ne dolazimo do znatnog poboljšanja napada. Broj invertibilnih matrica dimenzija $n \times n$ nad Z_{26} , odnosno veličina prostora ključa iznosi^[4]:

$$|KS| = 26^{n^2} \prod_{i=1}^n (1 - 2^{-i}) \cdot (1 - 13^{-i}) > 0.229 \cdot 26^{n^2}$$

Ovo pokazuje da se dolazi do marginalnog poboljšanja napada, kao i da se asimptotska

kompleksnost napada ne menja. Pored toga određivanje invertibilnosti ključa zahteva dodatan napor.

Ovaj napad zahteva $O(26^{n^2})$ operacija koje se sastoje od $O(1)$ množenja matrica dimenzija $n \times n$. Ukoliko se ne koriste algoritmi za brzo množenje matrica, kao npr. Strassen-ov metod^[6], kompleksnost trivijalnog algoritma iscrpne pretrage iznosi $O(n^3 26^{n^2})$ ^[4]. Koristeći Strassen-ov metod^[6] kompleksnost postaje $O(n^{\log_2 7} 26^{n^2})$ ^[4]. Danas postoje i bolji pristupi^[4] koji uključuju eliminaciju ponavljanja računanja i dostižu kompleksnost $O(n 26^{n^2})$ ^[4]. Detalji ovih napada prevazilaze opseg ovog rada.

2.2.3.2. NAPAD POZNATIM ORIGINALNIM TEKSTOM

Napad poznatim originalnim tekstom^[4] podrazumeva da napadač zna n linearno nezavisnih blokova originalnog teksta i odgovarajuće blokove šifrovanog teksta, gde je n dužina ključa. Neka je originalni tekst dat kao $P = (p_1, p_2, \dots, p_{n \cdot n})$, gde p_l , $l = 1, 2, \dots, n \cdot n$ predstavlja pojedinačno slovo originalnog teksta. Neka je jedan blok originalnog teksta dužine n dat kao $P_i = (p_{(i-1)n+1}, p_{(i-1)n+2}, \dots, p_{(i-1)n+n})$, $i = 1, 2, \dots, n$, gde P_1 blok od prvih n slova originalnog teksta, P_2 blok od narednih n slova originalnog teksta, itd. Neka je na sličan način dat šifrovani tekst C kao i blok šifrovanog teksta C_i , $i = 1, 2, \dots, n$ koji odgovara bloku originalnog teksta P_i i dobijen je kao $C_i = P_i K$. Svi parovi blokova (P_i, C_i) mogu biti prikupljeni iz jednog ili više različitih parova originalnog i šifrovanog teksta, u oznaci (P, C) . Na osnovu prikupljenih informacija napadač može da konstruiše matrice $U = (P_1^T, P_2^T, \dots, P_n^T)^T$ čiji redovi predstavljaju blokove originalnog teksta i $W = (C_1^T, C_2^T, \dots, C_n^T)^T$ čiji redovi predstavljaju blokove šifrovanog teksta. Odatle se lako izračunava odgovarajuća matricu ključa kao $K = U^{-1}W$. Kako su redovi matrice U linearno nezavisni, invertibilnost matrice U je zagarantovana.

Primer 2.2.4. Pronaći ključ matricu ako je poznat originalni tekst „ZASTITAXX” i njemu odgovarajući šifrovani tekst „XVKUOXWJW”.

Rešenje:

Podelimo originalni i šifrovani tekst na blokove dužine 3:

Z A S | T I T | A X X

X V K | U O X | W J W

Konstruišimo matrice U i W :

$$U = \begin{bmatrix} Z & A & S \\ T & I & T \\ A & X & X \end{bmatrix} = \begin{bmatrix} 25 & 0 & 18 \\ 19 & 8 & 19 \\ 0 & 23 & 23 \end{bmatrix}$$

$$W = \begin{bmatrix} X & V & K \\ U & O & X \\ W & J & W \end{bmatrix} = \begin{bmatrix} 23 & 21 & 10 \\ 20 & 14 & 23 \\ 22 & 9 & 22 \end{bmatrix}$$

Računamo U^{-1} :

$$\det(U) = 1541 = 7 \pmod{26}$$

$$\frac{1}{\det(U)} = 15$$

$$\text{adj}(U) = \begin{bmatrix} -253 & 414 & -144 \\ -437 & 575 & -133 \\ 437 & -575 & 200 \end{bmatrix} = \begin{bmatrix} 7 & 24 & 12 \\ 5 & 3 & 23 \\ 21 & 23 & 18 \end{bmatrix} \pmod{26}$$

$$U^{-1} = \frac{1}{\det(U)} \text{adj}(U) = 15 \cdot \begin{bmatrix} 7 & 24 & 12 \\ 5 & 3 & 23 \\ 21 & 23 & 18 \end{bmatrix} = \begin{bmatrix} 1 & 22 & 24 \\ 23 & 19 & 7 \\ 3 & 7 & 10 \end{bmatrix} \pmod{26}$$

Konačno dobijamo ključ kao:

$$K = U^{-1}W = \begin{bmatrix} 1 & 22 & 24 \\ 23 & 19 & 7 \\ 3 & 7 & 10 \end{bmatrix} \times \begin{bmatrix} 23 & 21 & 10 \\ 20 & 14 & 23 \\ 22 & 9 & 22 \end{bmatrix} = \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} \pmod{26}$$

2.2.3.3. NAPAD ODABRANIM ORIGINALNIM TEKSTOM

Ideja iza napada odabranim originalnim^[5] tekstom je da, šifrujući pažljivo odabrani originalni tekst, lako izvučemo informacije o ključu iz dobijenog šifrovanog teksta. Kako Hill-ova šifra^[1] u osnovi algortima ima množenje matrica, možemo iskoristiti osobinu matrica:

$$I \times K = K$$

da konstruišemo originalni tekst koji bi, kada se šifrjuje, dao tačno vrednost ključa. Elementi jedinične matrice su iz skupa $\{0,1\}$ tako da je jedini potreban podatak za ovaj napad informacija o karakteristikama alfabeta koji se mapiraju u vrednosti 0 i 1.

Primer 2.2.5. Ako je poznato da se slova A i B alfabeta mapiraju u brojeve 0 i 1, respektivno, odrediti originalni tekst koji bi šifrovanjem dao šifrovani tekst koji ima vrednost ključa korišćenog pri šifrovanju. Pretpostaviti da je dužina korišćenog ključa 3.

Rešenje:

Kako je dužina originalnog ključa 3 potrebno je konstruisati jediničnu matricu dimenzija 3×3 . Ona izgleda:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Dalje zamenom brojeva odgovarajućim slovima dobijamo blokove originalnog teksta:

$$P = \begin{bmatrix} B & A & A \\ A & B & A \\ A & A & B \end{bmatrix}$$

Odakle imamo da je originalni tekst koji treba šifrovati da bi se dobila vrednost ključa kao šifrovani tekst „BAAABAAAB”.

3. IMPLEMENTACIJA

Simulator rada Hill-ove šifre^[1] realizovan je kao desktop aplikacija. Sistem omogućava korisniku interakciju putem grafičkog korisničkog interfejsa. Aplikacija je implementirana koristeći programski jezik *Java*^[7] dok je za grafički korisnički interfejs korišćena *JavaFX*^[8] tehnologija. Kako su korišćene najnovije opcije programskog jezika *Java*^[7], minimalna verzija *JRE*^[7] potrebna za pokretanje simulatora je 1.8. *Java*^[7] je odabrana zbog portabilnosti tako da se kod ne mora rekompajlirati za različite arhitekture. Pored toga, dodatni motiv je bio to što su drugi simulatori, iz oblasti zaštite podataka, na Elektrotehničkom fakultetu u Beogradu, takođe realizovani u *Java*-i^[7] pa će potencijalno integrisanje sa postojećim simulatorima biti jednostavnije.

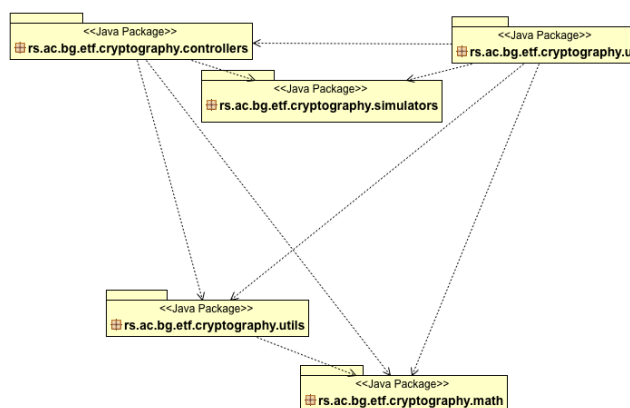
Aplikacija je razvijena koristeći razvojno okruženje *Eclipse*^[9]. Bitno je napomenuti da dalji razvoj aplikacije nikako ne zavisi od konkretnog razvojnog okruženja. To je obezbeđeno korišćenjem *ant*^[10] bild sistema za kompajliranje i generisanje izvršnog fajla. Koristeći *ant*^[10], bilo ko, ko ima izvorni kod, može sa lakoćom da ga kompajlira bez brige o podešavanju promenljivih okruženja(kao što je CLASSPATH) i povezivanju sa korišćenim eksternim bibliotekama. Za verzionisanje koda korišćena je *git*^[11] tehnologija dok je za menadžment projekta korišćena platforma *GitHub*^[12].

Simulator se distribuira u dva formata. Prvi format je klasičan java izvršni fajl, *jar* file. Drugi format je *jnlp* format koji omogućava da aplikacija stoji na serveru a da je korisnici mogu pokrenuti iz svog brauzera koristeći *Java Web Start*^[13].

U nastavku je dat detaljniji opis organizacije koda po paketima, klasni dijagrami od interesa, eksterne biblioteke koje su korišćene u realizaciji simulatora, opis formata fajlova za testiranje znanja korisnika i uputstvo za manipulisanje izvornim kodom.

3.1. PREGLED PAKETA U SISTEMU

U nastavku je dat opis sadržaja svakog od paketa u sistemu. Slika 3.1 prikazuje UML dijagram paketa realizovanog sistema.



Slika 3.1. UML dijagram paketa

Paket *rs.ac.bg.etf.cryptography.controllers* sadrži klase sa biznis logikom simulatora. One predstavljaju interfejs za klase grafičkog korisničkog interfejsa čime se postiže kompletno odvajanje prikaza podataka od njihovog generisanja.

Paket *rs.ac.bg.etf.cryptography.simulators* sadrži glavne klase za svaki simulator. To obuhvata sve klase koje predstavljaju ulaznu tačku u pojedini simulator.

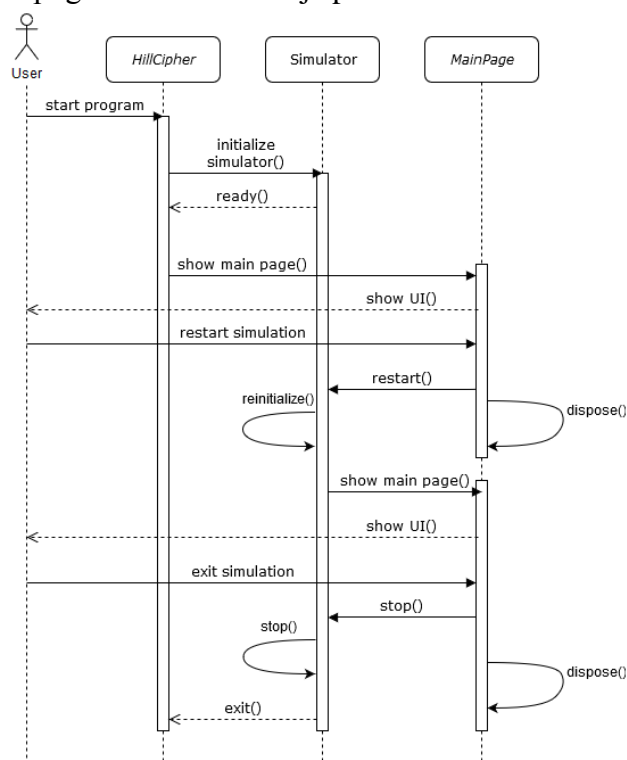
Paket *rs.ac.bg.etf.cryptography.ui* sadrži klase koje predstavljaju forme i skelotone formi za grafički korisnički interfejs sistema.

Paket *rs.ac.bg.etf.cryptography.utils* sadrži pomoćne klase. Ovde su izvučene sve statičke funkcije čije se funkcionalnosti koriste u velikom broju drugih paketa. To obuhvata funkcije za generisanje delova korisničkog interfejsa koji su slični na većini formi, kao i razne funkcije koje konvertuju podatke iz jednog u drugi format, pogodniji za prikaz.

Paket *rs.ac.bg.etf.cryptography.math* sadrži klase sa matematičkim operacijama neophodnim za realizaciju logike simulatora. Takođe, sadrži interfejse prilagođene za korišćenje klasa iz korišćenih eksternih biblioteka za rad sa matricama.

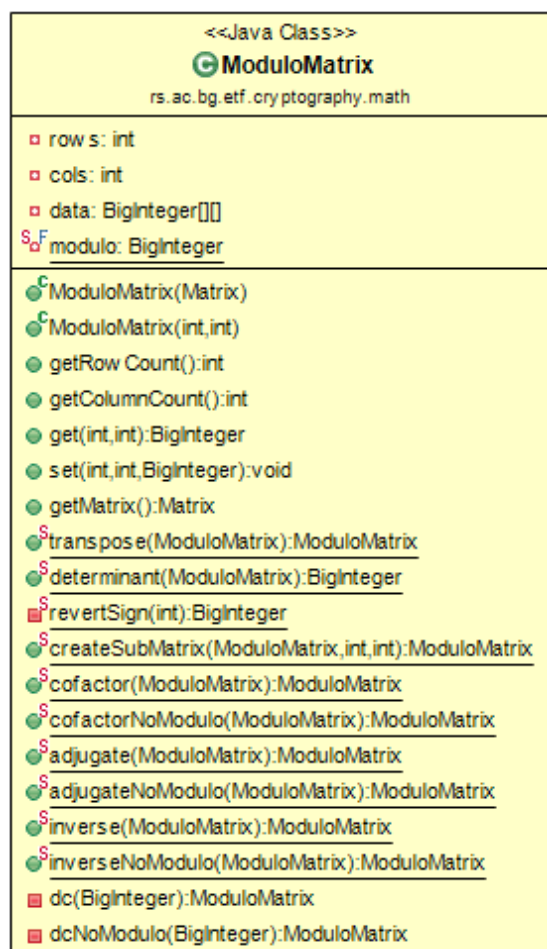
3.2. PREGLED KLASA U SISTEMU

Ulazna tačka u sistem je klasa *HillCipher*. Ona komunicira sa klasom *Simulator* kojoj zadaje komandu za početak simulacije. Kada simulacija bude spreman za pokretanje, glavna klasa poziva početnu stranicu grafičkog korisničkog interfejsa *MainPage*. Na istom principu funkcioniše ceo simulator. Cela logika i stanje simulacije se nalaze u statičkoj klasi *Simulator*, a ostale UI klase komuniciraju sa njom, odakle dobijaju stanje i u zavisnosti od toga prikazuju podatke na pogodan način. Ovo je prikazano na slici 3.2.



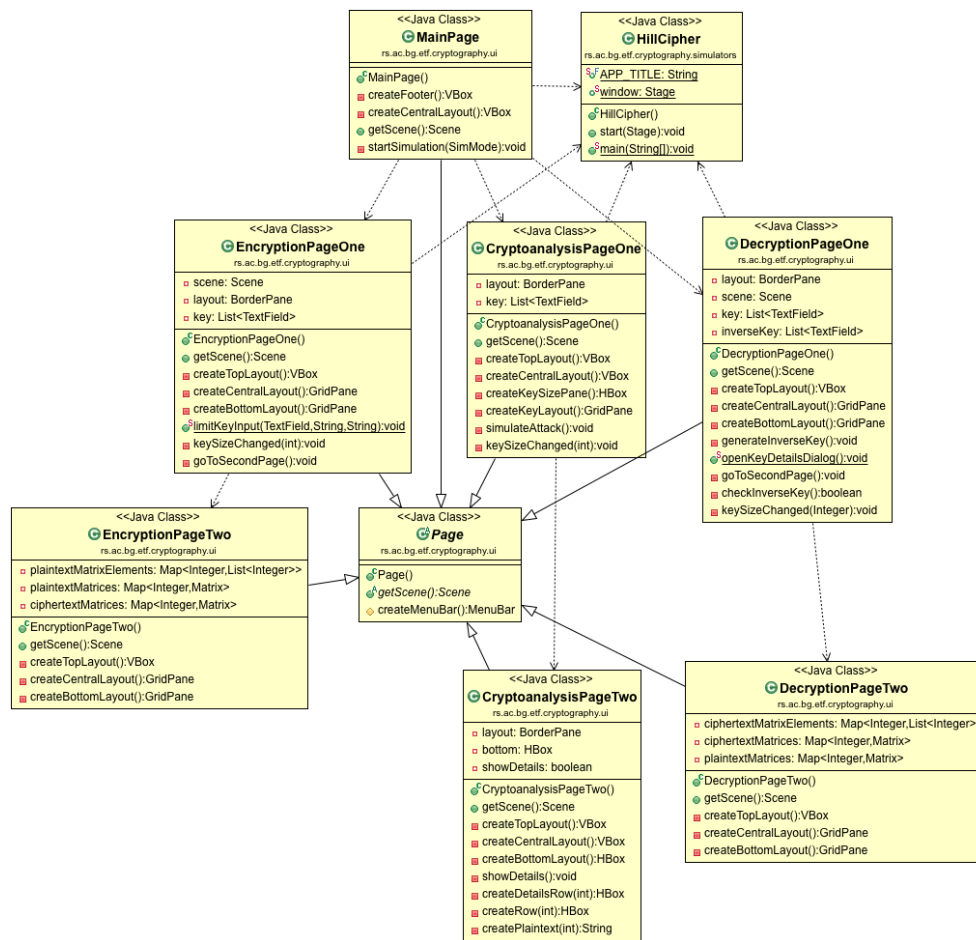
Slika 3.2. Dijagram sekvence pokretanja, restartovanja i gašenja simulatora

U klasi **ModuloMatrix** se nalazi matematički aparat za baratanje operacijama nad matricama po zadanom modulu. UML dijagram ove klase dat je na slici 3.3. Detaljniji opis same implementacije dat je u poglavlju 3.3.



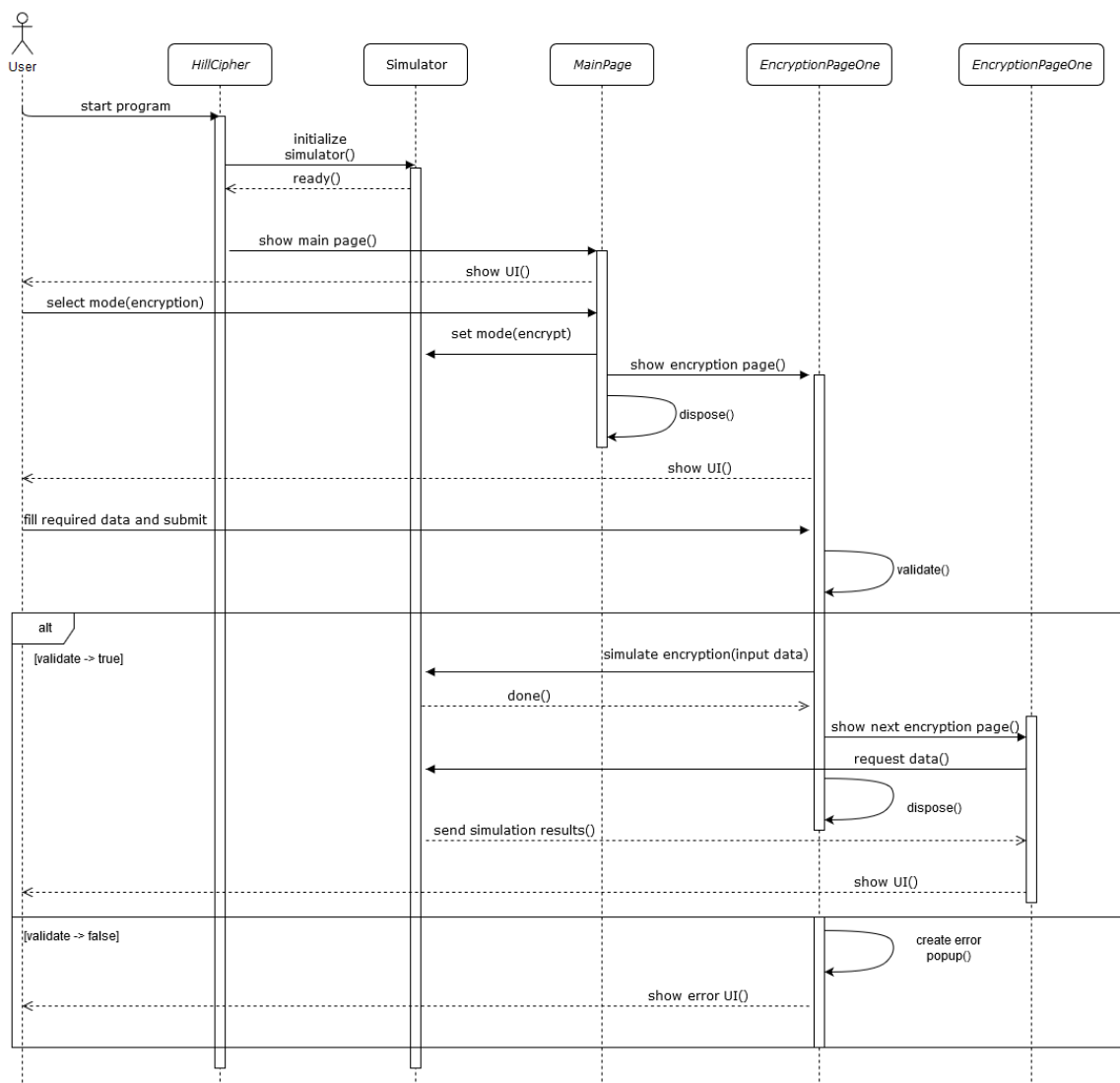
Slika 3.3. Dijagram klase *ModuloMatrix*

Sve UI klase imaju kao osnovnu klasu klasu **Page**. Ova klasa ima podrazumevani *layout* koji se koristi kroz celu aplikaciju i unapred postavlja određene elemente koji se ponavljaju kao što su meni linija i tabela preslikavanja alfabeta. *Layout* je takav da definiše 5 delova: centralni, gornji, donji, levi i desni. Ostale UI klase koje nasleđuju ovu klasu mogu da redefinišu svaki od tih 5 delova pojedinačno, u zavisnosti od njihove uloge i podataka koje prikazuju. Na ovaj način je postignuto da kod koji je isti ne bude ponavljan kroz klase već je dostupan svima iz osnovne klase. Dijagram UI klasa dat je na slici 3.4.



Slika 3.4. Klasni dijagram UI klasa simulatora

Klasa **MainPage** izlistava korisniku sve modove rada i na osnovu selekcije vrši redirekciju na UI klasu zaduženu za određeni mod funkcionisanja. Svaki od modova ima dve strane. Na prvoj strani je uvek korisniku prikazan meni za navigaciju, iz koga može da resetuje simulaciju ili ugasi simulator, kao i tabela preslikavanja alfabeta. Takođe, na svakoj prvoj strani je zahtevano od korisnika da popuni neophodna polja za nastavak simulacije kao što su originalni tekst, šifrovani tekst ili ključ matrica. Sve forme vrše validaciju unetih podataka koristeći pomoćne klase i ukoliko je sve dobro uneto, daju signal klasi **Simulator** da nastavi simulaciju i predaju kontrolu narednoj formi. Ovo je prikazano na slici 3.5. Kako su svi modovi (enkripcija, dekripcija, test i napadi) slični, dat je samo dijagram sekvence za mod enkripcije. Klase **DecryptionPageOne**, **EncryptionPageOne** i **CryptoanalysisPageOne** redom odgovaraju modovima rada dekripcije, enkripcije i kriptanalize.



Slika 3.5. Dijagram sekvence UI klasa za process enkripcije

3.3. EKSTERNE BIBLIOTEKE

Za operacije sa matricama korišćena je biblioteka *JAMA*^[14], koja pruža podršku za osnovne operacije sa matricama kao što je množenje matrica, transponovanje, itd. Kako ova biblioteka ne podržava operacije sa matricama koristeći modulo, neke od ovih klasa su zavijene u interfejs koji je pogodniji za upotrebu u simulatoru i dodate su operacije za rad sa matricama u modulo aritmetici. Konkretno je napravljena klasa *ModuloMatrix* čiji je klasni dijagram dat na slici 3.3. Ona sadrži matricu konfigurabilnih dimenzija čiji su svi elementi tipa *Java-ine*^[7] standardne klase *BigInteger*. Ova klasa je odabrana za tip elementa jer ima uslužnu metodu za određivanje multiplikativnog invezu po zadatom modulu, kako bi se izbegla implementacija pretrage multiplikativnog inverza proširenim Euklidovim algoritmom^[2]. Klasa *ModuloMatrix* sadrži statičko polje *modulo* koje označava po kom modulu se vrše računanja. Ovo polje je konfigurabilno i ima podrazumevanu vrednost 26. To omogućava da se u budućnosti ista klasa koristi za neke druge dužine alfabeta. Svaka od operacija klase *Matrix* iz biblioteke

JAMA^[14], je zamotana u metodu, čime je postignuta reupotreba kompletne logike, koja poziva osnovnu metodu klase *Matrix* a zatim za svaki od elemenata izačuna vrednost po modulu 26 i zameni odgovarajući element matrice. Dodata je i ključna funkcionalnost izračunavanja inverzne matrice po modulu koja sračuna determinantu matrice na standardan način, skalira dobijenu vrednost po modulu 26, odredi njen multiplikativni inverz i njime pomnoži adjungovanu matricu dobijenu pozivom odgovarajuće metode klase *ModuloMatrix*.

Za učitavanje fajlova sa testovima korišćena je biblioteka *Gson*^[15], koja omogućava manipulaciju sa fajlovima *json* formata.

3.4. FORMAT TEST FAJLOVA

Fajlovi za testiranje znanja studenata imaju specifičan format. Zbog lakoće proširivanja i sveopšte podržanosti izabrani format je *json* i razlikuje se za različite vrste testova. U nastavku je dat prikaz formata test fajlova za enkripciju i dekripciju, respektivno.

```
{
  "test": "encryption",
  "plaintext": "<plaintext>",
  "key_size": "<key_size>",
  "fill": "<fill_character>",
  "key": {
    "<row_number>": "<space_separated_row_values>",
    "<row_number>": "<space_separated_row_values>",
    ...
  }
}
```

Slika 3.6. Format test fajla za enkripciju

```
{
  "test": "decryption",
  "ciphertext": "<ciphertext>",
  "key_size": "<key_size>",
  "key": {
    "<row_number>": "<space_separated_row_values>",
    "<row_number>": "<space_separated_row_values>",
    ...
  }
}
```

Slika 3.7. Format test fajla za dekripciju

3.5. POKRETANJE SIMULATORA

Izvorni kod se može kompajlirati iz komandne linije na sledećom komandom:

```
ant -f build/build.xml
```

Na ovaj način izgenerisane su dve vrste izvršnog fajla u direktorijumu *build/dist/*.

Izvršni *jar* file se pokreće na standardan način komandom:

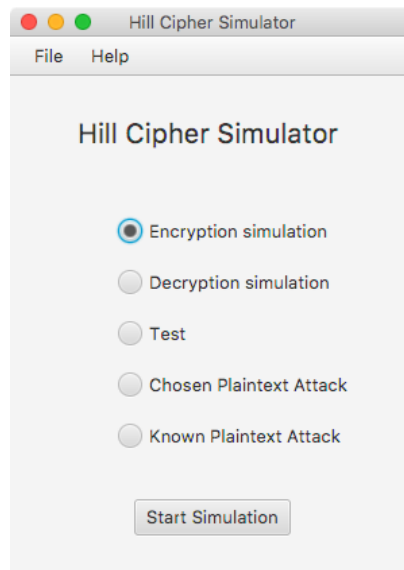
```
java -jar build/dist/HillCipherSimulator.jar
```

Izvršni *jnlp* fajl se pokreće sledećom komandom:

```
javaws build/deploy/HillCipherSimulator.jnlp
```

4. NAČINI KORIŠĆENJA

U ovom poglavlju dato je detaljno korisničko uputstvo za implementirani sistem. Pokretanjem sistema otvara se početna strana prikazana na slici 4.1. Korisniku je ovde na raspolaganju da izabere jedan od četiri moda rada simulatora. Nakon odabira klikom na dugme **Start Simulation** korisnik započinje odabranu simulaciju.



Slika 4.1. Početna strana simulatora

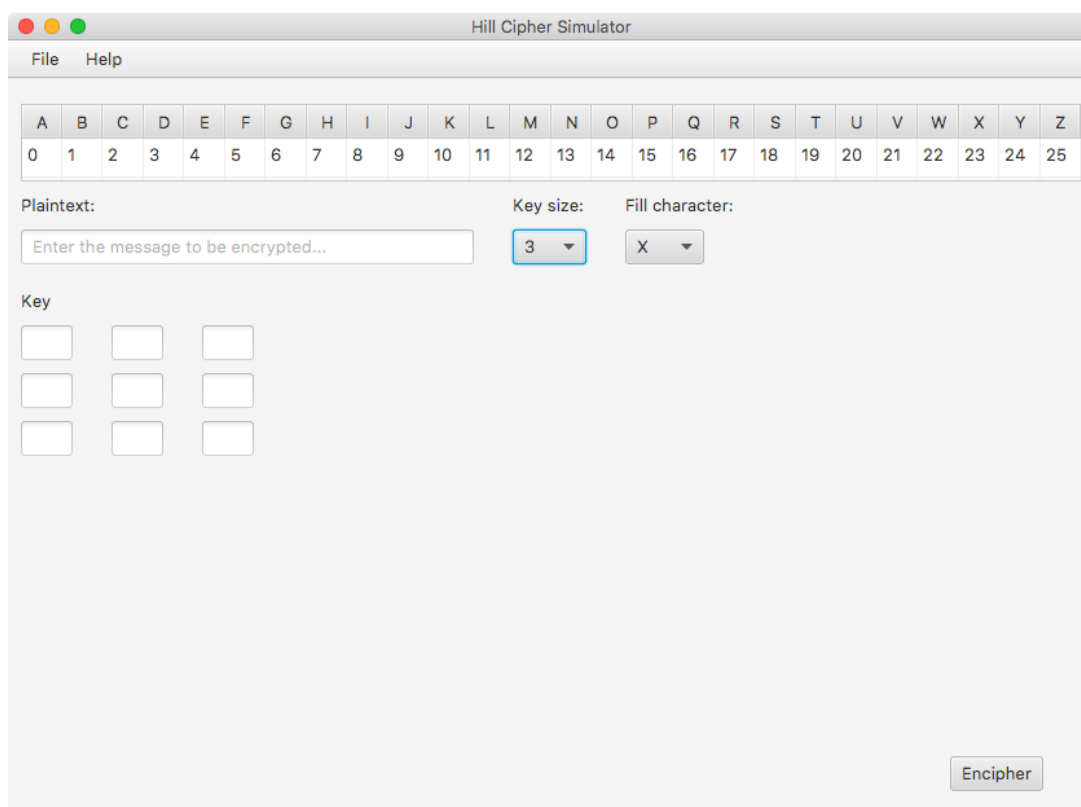
U svakom trenutku tokom simulacije korisniku je dostupan meni sa opcijama **File** i **Help**. Pod opcijom **File** se nalaze dve akcije: **Restart**, koja omogućava korisniku da prekine trenutnu simulaciju i vrati se na početnu stranu, i **Exit** koja korisniku omogućava da ugasi simulator. Pod opcijom **Help** nalazi se akcija **About** čijim odabirom korisnik dobija prozor sa informacijama o simulatoru. Prikaz prozora sa informacijama dat je na slici 4.2.



Slika 4.2. Prozor sa informacijama

4.1. SIMULACIJA ENKRIPCije

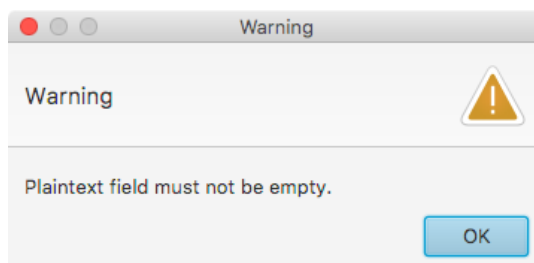
Odabirom moda **Encryption simulation** i klikom na dugme **Start Simulation** korisnik je redirektovan na stranicu prikazanu na slici 4.3. Ovde je korisniku prikazan alfabet koji se koristi kao i brojevi u koje se određena slova mapiraju da bi lakše mogao da prati sve delove simulacije. Od korisnika se ovde očekuje da unese originalni tekst koji želi da šifruje, izabere veličinu ključa za šifrovanje i odabere karakter za ispunu, za slučaj da ne želi da koristi podrazumevanu vrednost. Nakon odabira veličine ključa, potrebno je da korisnik unese i vrednosti elemenata ključa. Te vrednosti moraju biti u opsegu od 0 do 26.



The screenshot shows the 'Hill Cipher Simulator' application window. It features a menu bar with 'File' and 'Help'. Below the menu is a 26x26 grid mapping letters A-Z to numbers 0-25. The main interface includes a 'Plaintext:' label with a text input field containing the placeholder 'Enter the message to be encrypted...'. To the right of the input field are two dropdown menus: 'Key size:' set to '3' and 'Fill character:' set to 'X'. Below these is a 'Key' section with a 3x3 grid of input fields for key elements. An 'Encipher' button is located in the bottom right corner.

Slika 4.3. Prva strana simulacije enkripcije

Ukoliko forma nije popunjena korisnik dobija poruku prikazanu na slici 4.4 a ukoliko ključ nema inverznu matricu prikazuje se poruka sa slike 4.5.



Slika 4.4. Poruka o grešci kada nisu popunjena sva polja forme



Slika 4.5. Poruka o grešci kada uneti ključ nema inverz

Ukoliko je sve popunjeno validnim podacima klikom na dugme **Encipher**, korisnik je redirektovan na stranicu prikazanu na slici 4.6 gde su mu dati svi detalji procesa enkripcije. Korisnik može da vidi sve parametre koje je uneo na prethodnoj stranici, a sada i dodatno korak po korak ceo proces enkripcije.

Hill Cipher Simulator

File Help

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext: Key size: Fill character:

$$\begin{bmatrix} 25 & 0 & 18 \end{bmatrix} \times \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} = \begin{bmatrix} 309 & 931 & 478 \end{bmatrix} = \begin{bmatrix} 23 & 21 & 10 \end{bmatrix} \pmod{26}$$

$$\begin{bmatrix} 19 & 8 & 19 \end{bmatrix} \times \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} = \begin{bmatrix} 488 & 846 & 595 \end{bmatrix} = \begin{bmatrix} 20 & 14 & 23 \end{bmatrix} \pmod{26}$$

$$\begin{bmatrix} 0 & 23 & 23 \end{bmatrix} \times \begin{bmatrix} 3 & 25 & 4 \\ 23 & 6 & 15 \\ 13 & 17 & 21 \end{bmatrix} = \begin{bmatrix} 828 & 529 & 828 \end{bmatrix} = \begin{bmatrix} 22 & 9 & 22 \end{bmatrix} \pmod{26}$$

Ciphertext:

Slika 4.6. Druga stranica u simulaciji enkripcije

U svakom redu su prikazani odgovarajući blok originalnog teksta, ključ kojim se šifruje i rezultat množenja matrica. Prelaskom miša preko polja bloka matrice koja sadrži originalni ili šifrovani tekst iskače balončić sa porukom koja predstavlja slovo u koje se odgovarajući broj mapira što je prikazano na slici 4.7. Takođe prelaskom preko rezultujuće matrice u balončiću su prikazani detalji množenja matrice (kompletan izraz) što se vidi na slici 4.8. Odatve se korisnik vraća na početnu stranicu odabirom opcije **File -> Restart**.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext: Key size: Fill character:

x = = (mod 26)

Slika 4.7. Prikaz slova kome odgovara element matrice

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext: Key size: Fill character:

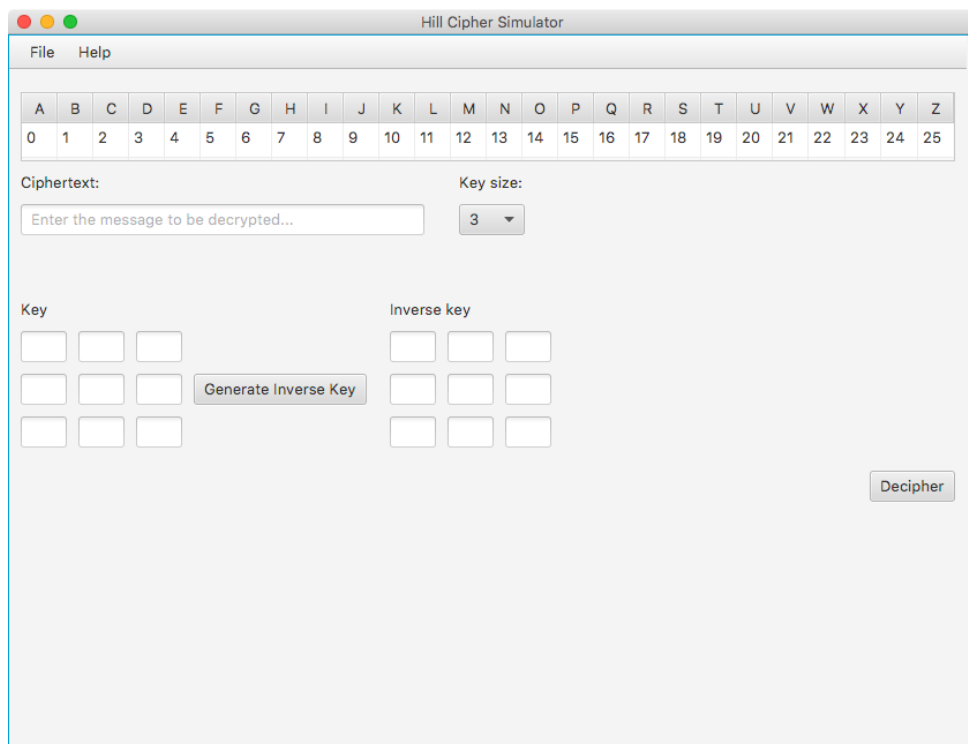
x = = (mod 26)

$25 \cdot 3 + 0 \cdot 23 + 18 \cdot 13$

Slika 4.8. Prikaz računanja vrednosti elementa matrice

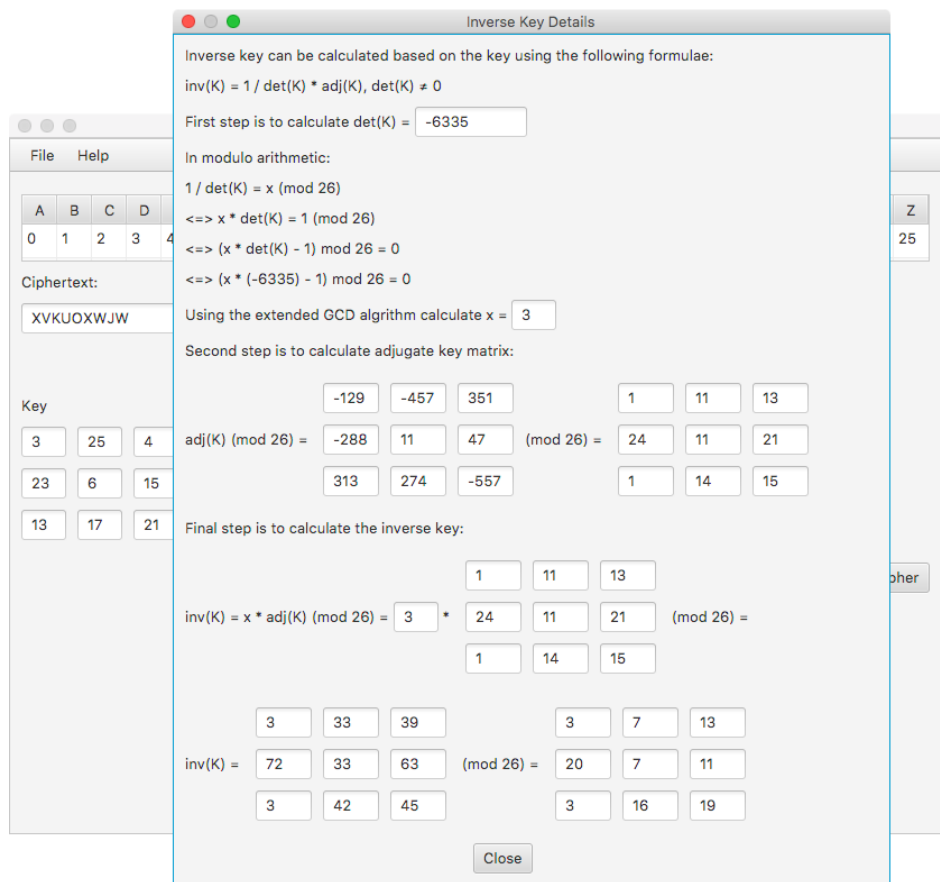
4.2. SIMULACIJA DEKRIPCije

Odabirom moda **Decryption simulation** i klikom na dugme **Start Simulation** korisnik je redirektovan na stranicu prikazanu na slici 4.9. Ovde je korisniku prikazana tabela mapiranja alfabeta, kao i na prethodnim formama. Od korisnika se ovde očekuje da unese šifrovani tekst koji želi da dešifruje i izabere veličinu ključa za dešifrovanje. Nakon odabira veličine ključa, potrebno je da korisnik unese i vrednosti elemenata ključa. Te vrednosti moraju biti u opsegu od 0 do 26. Nakon unosa ključa potrebno je izgenerisati inverzni ključ koji se koristi pri dešifrovanju.



Slika 4.9. Prva strana simulacije dekripcije

Klikom na dugme **Generate Inverse Key** korisnik generiše inverzni ključ i otvara mu se prozor prikazan na slici 4.10 sa detaljima izračunavanja inverznog ključa.



Slika 4.10. Detalji generisanja inverznog ključa

Konačno klikom na dugme **Decipher** korisnik je redirektovan na stranicu sa detaljima procesa dekripcije ukoliko su sva polja dobro popunjena. Izgled stranice prikazan je na slici 4.11. Ukoliko neko od polja nije popunjeno ili podaci u nekom od polja nisu validni, korisniku je prikazan prozor sa odgovarajućom porukom sličan prozoru na slici 4.5.

The screenshot shows the 'Hill Cipher Simulator' window. At the top is an alphabet table (A-Z, 0-25). Below it, the 'Ciphertext' field contains 'XVKUOXWJW' and the 'Key size' is set to 3. The main area displays three rows of matrix multiplication for deciphering:

$$\begin{bmatrix} 23 & 21 & 10 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} = \begin{bmatrix} 519 & 468 & 720 \end{bmatrix} = \begin{bmatrix} 25 & 0 & 18 \end{bmatrix} \pmod{26}$$

$$\begin{bmatrix} 20 & 14 & 23 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} = \begin{bmatrix} 409 & 606 & 851 \end{bmatrix} = \begin{bmatrix} 19 & 8 & 19 \end{bmatrix} \pmod{26}$$

$$\begin{bmatrix} 22 & 9 & 22 \end{bmatrix} \times \begin{bmatrix} 3 & 7 & 13 \\ 20 & 7 & 11 \\ 3 & 16 & 19 \end{bmatrix} = \begin{bmatrix} 312 & 569 & 803 \end{bmatrix} = \begin{bmatrix} 0 & 23 & 23 \end{bmatrix} \pmod{26}$$

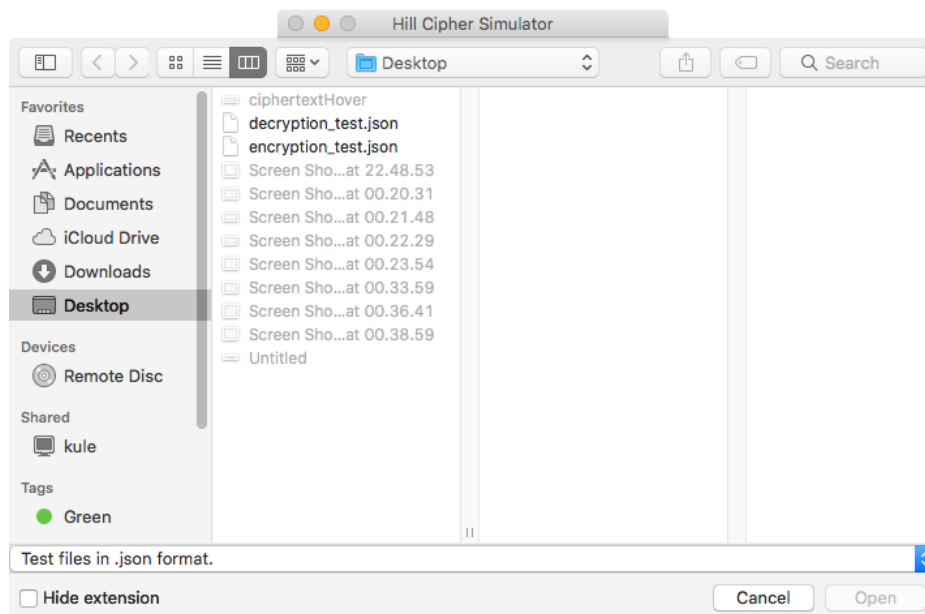
At the bottom, the 'Plaintext' field contains 'ZASTITAXX'.

Slika 4.11. Druga strana simulacije dekripcije

Kako se proces enkripcije od procesa dekripcije razlikuje samo u ključu, ova stranica nije detaljno komentarisana.

4.3. MOD ZA TESTIRANJE

Pored simulacionih modova postoji i mod za testiranje stečenog znanja. Ovom modu se pristupa sa glavne stranice aplikacije odabirom opcije **Test** i klikom na dugme **Start Simulation**. Postoje dve vrste testa, jedan za proces enkripcije i drugi za proces dekripcije. Odabirom ove opcije korisniku se otvara prozor prikazan na slici 4.12 gde korisnik treba da odabere putanju do fajla koji sadrži test. Ukoliko je fajl pogrešnog formata korisniku se ispisuje poruka o grešci slična poruci prikazanoj na slici 4.5. Ukoliko je korisnik odabrao validan test fajl redirektovan je na jednu od dve stranice u zavisnosti od vrste testa.



Slika 4.12. Izbor fajla sa testom

4.3.1. TEST PROCESA ENKRIPCIJE

Prikaz stranice sa testom za proces enkripcije prikazan je na slici 4.13. Ova stranica izgleda slično kao stranica za simulaciju samo su određena polja već popunjena podacima iz test fajla i dodato je polje za upis odgovora. Od korisnika se ovde očekuje da samostalno reši zadatak i rešenje upiše u odgovarajuće polje.

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Plaintext:

Key size: 3 Fill character: X

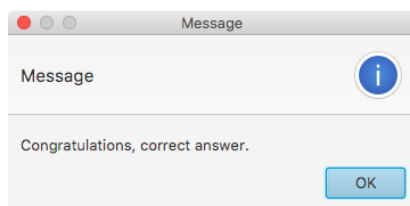
Key

3	25	4
23	6	15
13	17	21

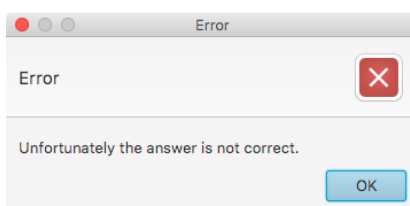
Ciphertext:

Slika 4.13. Test za proces enkripcije

Klikom na dugme **Check Solution** korisnik se redirektuje na stranicu sa detaljnim rešenjem koje izgleda kao na slici 4.6 uz poruku o ispravnosti unetog rešenja. Poruka u slučaju tačnog odgovora data je na slici 4.14 a u slučaju netačnog odgovora na slici 4.15.



Slika 4.14. Poruka kada je odgovor tačan



Slika 4.15. Poruka kada je odgovor netačan

4.3.2. TEST PROCESA DEKRIPTCIJE

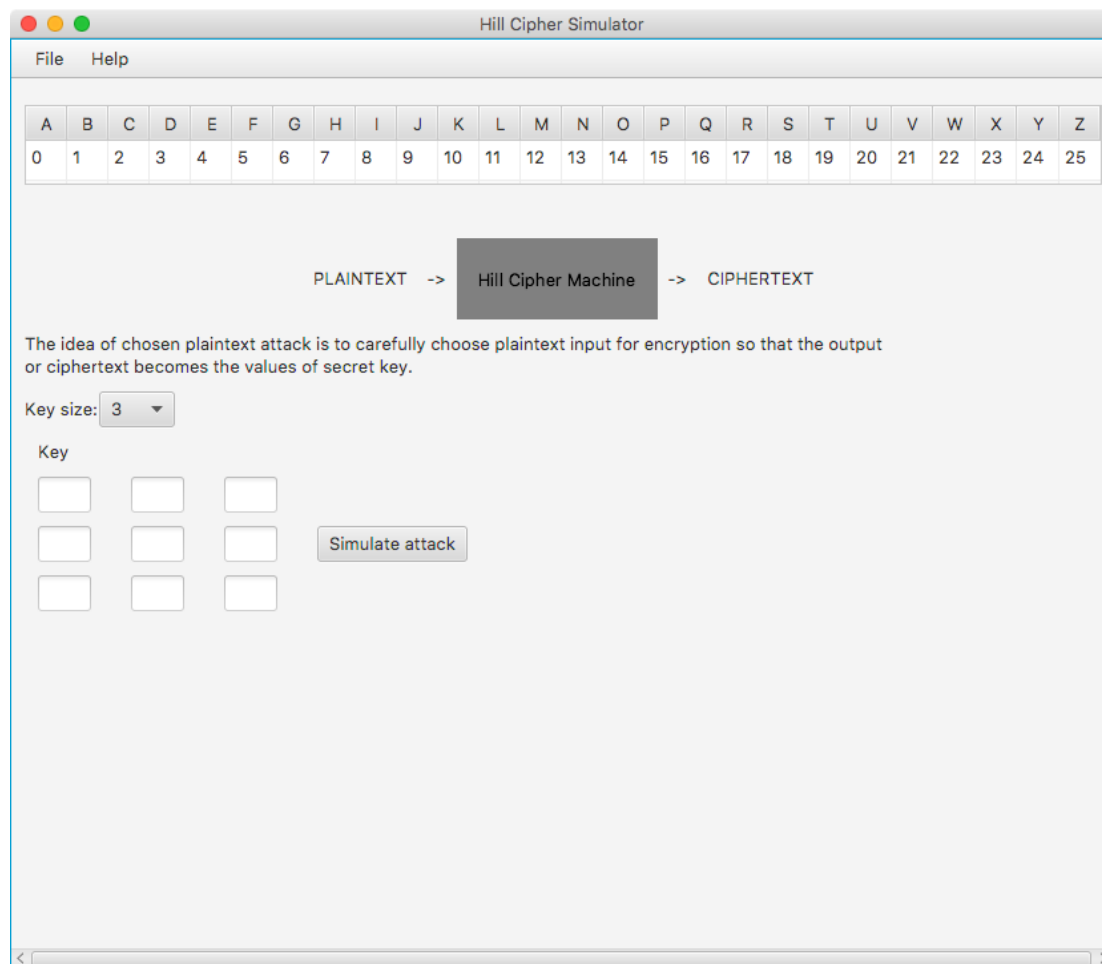
Slično kao kod testa za proces enkripcije i ovde se prvo otvara forma slična formi za proces dekripcije kod koje su popunjena sva polja osim vrednosti inverznog ključa i dešifrovanog teksta. Izgled ove forme dat je na slici 4.16. Korisnik ima zadatak da popuni inverzni ključ i dešifrovani tekst nakon čega klikom na dugme **Check Solution** proverava da li je odgovor tačan. Poruka u slučaju tačnog odgovora prikazana je na slici 4.14 a u slučaju netačnog odgovora na slici 4.15.

 A screenshot of a web application titled "Hill Cipher Simulator". At the top is a menu bar with "File" and "Help". Below it is a 26x26 grid of letters A-Z and numbers 0-25. The "Ciphertext:" field contains "XVKUOXWJW". The "Key size:" dropdown is set to "3". There are two sections for keys: "Key" and "Inverse key", each with a 3x3 grid of input boxes. The "Key" section has values 3, 25, 4 in the first row; 23, 6, 15 in the second row; and 13, 17, 21 in the third row. The "Inverse key" section has empty boxes. At the bottom, there is a "Plaintext:" field and a "Check Solution" button.

Slika 4.16. Test za proces dekripcije

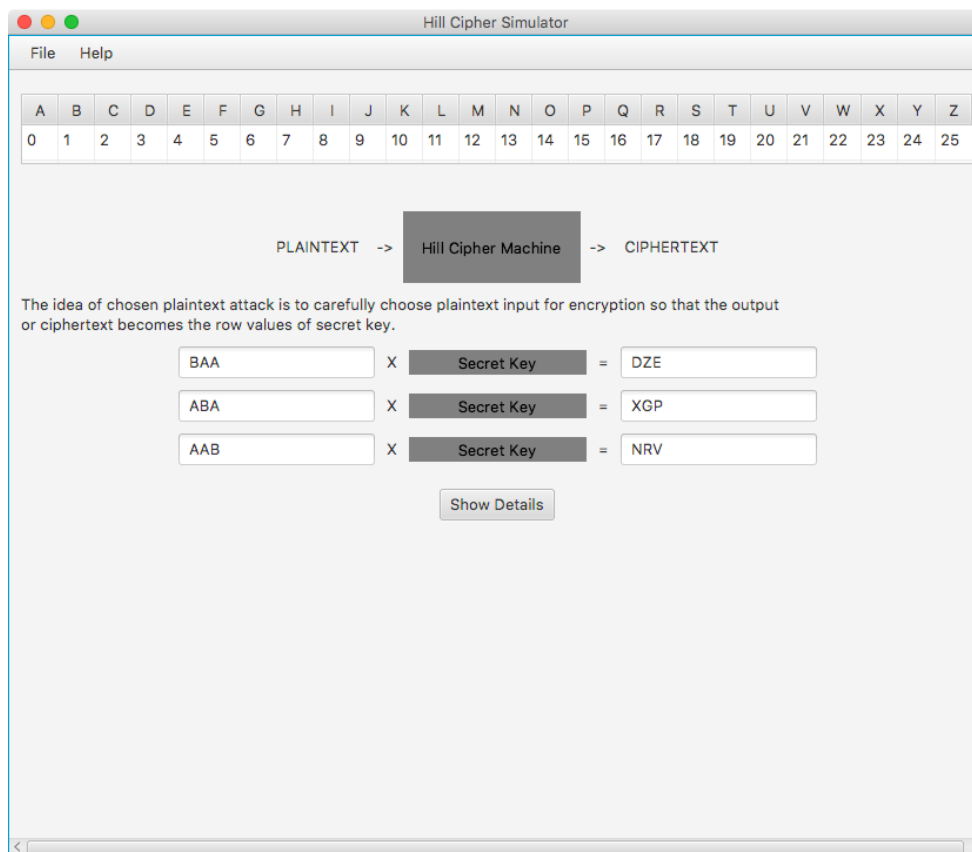
4.4. SIMULACIJA NAPADA

Simulator daje mogućnost prikaza napada na Hill-ovu šifru^[1] odabranim originalnim tekstom. Na slici 4.17 prikazan je prozor u kome se od korisnika očekuje da unese ključ za šifrovanje.

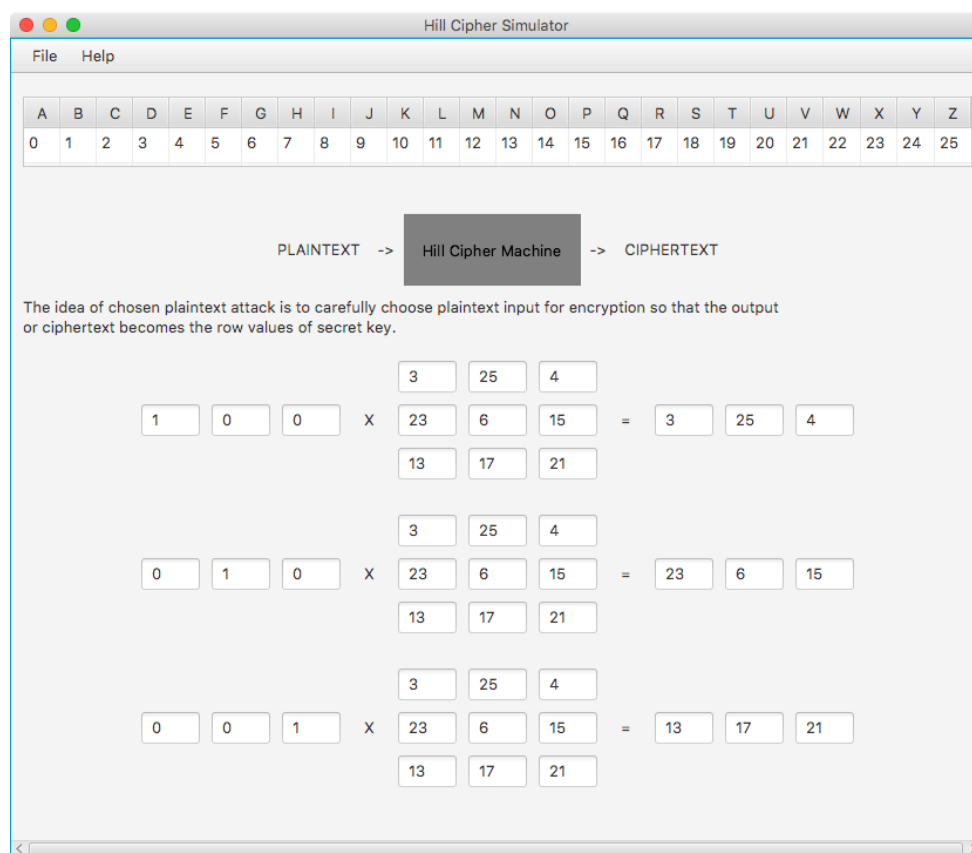


Slika 4.17. Simulacija napada odabranim originalnim tekstom

Ukoliko je uneti ključ invertibilan klikom na dugme **Simulate attack** korisnik se redirektuje na prozor prikazan na slici 4.18 gde je prikazan odabir originalnog teksta kojim se na izlazu direktno dobija uneti ključ. Polja ključa su sakrivena na ovom prozoru kako bi se pokazalo da napadač ne zna te vrednosti. Klikom na dugme **Show Details** otkrivaju se skrivene vrednosti i korisnik može da proveriti da li zaista dobijeni ključ odgovara unetom ključu. Ovaj prozor prikazan je na slici 4.19.



Slika 4.18. Simulacija napada odabranim originalnim tekstom



Slika 4.19. Simulacija napada odabranim originalnim tekstom

Pored napada odabranim originalnim tekstom simulator nudi i opciju napada poznatim originalnim tekstom. Slika 4.20 predstavlja prvu stranicu u simulaciji ove vrste napada. Ovde se od korisnika očekuje da unese odgovarajuće parove blokova poznatog originalnog i šifrovanog teksta. Ti parovi moraju biti linearno nezavisni. Ukoliko nisu simulator prikazuje poruku sličnu poruci sa slike 4.5.

Hill Cipher Simulator

File Help

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Key size: 3

PLAINTEXT BLOCKS

ZAS

TIT

AXX

CIPHERTEXT BLOCKS

XVK

UOX

WJW

Simulate Attack

Slika 4.20. Simulacija napada poznatim originalnim tekstom

Ukoliko su svi podaci validni klikom na dugme **Simulate Attack** prelazi se na stranicu sa simulacijom izračunavanja ključa na osnovu unetih podataka prikazanu na slici 4.21.

Hill Cipher Simulator

File Help

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Key size: 3

U =

25	0	18
19	8	19
0	23	23

W =

23	21	10
20	14	23
22	9	22

inv(U) =

1	22	24
23	19	7
3	7	10

U x K = W

K = inv(U) x W

K =

1	22	24
23	19	7
3	7	10

x

23	21	10
20	14	23
22	9	22

=

991	545	1044
1063	812	821
429	251	411

=

3	25	4
23	6	15
13	17	21

Slika 4.21. Simulacija napada poznatim originalnim tekstom

5. ZAKLJUČAK

U ovom radu dat je kratak istorijat kriptografije kao i klasičnih tehnika šifrovanja. Navedene su sve poznatije klasične šifre i istaknute su njihove mane kako bi čitaoci shvatili kako je tekao razvoj i zašto je Hill-ova šifra bila značajna za doba u kome se pojavila. Zatim je dat detaljan opis algoritma za šifrovanje Hill-ovom šifrom. Objašnjeni su svi neophodni matematički pojmovi i definisani svi uslovi pod kojima se obavlja šifrovanje. Dati su i detaljno rešeni primeri za svaki od ovih pojmova. Dati su implementacioni detalji. Konkretno su navedene tehnologije kojima je simulator realizovan i razlozi za njihov izbor. Data je i objašnjena kompletna organizacija koda koja je u skladu sa dobrom praksom objektno orijentisanog programiranja. Dati su *UML* dijagrami paketa i značajnijih grupa klasa i objašnjeni su njihovi odnosi. Navedene su sve eksterne biblioteke korišćene pri razvoju aplikacije, kao i razlozi za njihovim uvođenjem. Prikazane su šeme *json* fajlova kojima se zadaju testovi kako bi korisnici mogli sami da osmisle svoje testove. Za potencijalne programere koji bi želeli da unaprede ovaj softver ili ga prošire ili integrišu sa svojim softverom date su komande za kompajliranje koda i pokretanje iz komandne linije. Na kraju je dato detaljno korisničko uputstvo koje obuhvata izgled svih korišćenih formi, kao i objašnjenja prikazanih vrednosti na tim formama.

Autor se trudio da, kako simulator tako i rad, budu prilagođeni korisnicima sa različitim nivoima znanja. Zbog toga su forme jednostavne i ne sadrže mnogo detalja dok se detalji mogu dobiti prelaskom miša preko vrednosti u simulatoru ili klikom na dugmad označena sa *more details*. Takođe, autor je dao detaljno rešene primere za sve matematičke operacije koje su neophodne da bi se razumela pozadina algoritma koje napredniji korisnici mogu da preskoče. Autor smatra da se realizovani simulator može koristiti kao edukacioni alat, ali i u svrhe testiranja znanja.

Jasno je da nije moguće napraviti savršen sistem, naročito ako ga koriste ljudi različitog nivoa znanja i interesovanja. Stoga i ovaj sistem ima nedostataka koji se lako mogu nadograditi:

- mapiranje alfabeta u brojeve bi trebalo učiniti konfigurabilnim,
- ograničiti veličinu ključa na 2 do 4 što je dovoljno da se uoči poenta algoritma,
- realizovati simulator u vidu web aplikacije.

Kada govorimo o dobrim stranama, to je raspodela klasa po paketima i funkcionalnosti po klasama što olakšava reupotrebu postojećih formi i dodavanje novih, jer je logika razdvojena od prezentacije.

LITERATURA

1. W. Stallings, Cryptography and Network Security Fourth Edition, Prentice Hall, 2005.
2. The Extended Euclidean Algorithm, Available from: <http://www-math.ucdenver.edu/~wcherowi/courses/m5410/exeucalg.html> (accessed: June 2018).
3. B. Carter and T. Magoc, Classical Ciphers and Cryptanalysis Available from: <https://pdfs.semanticscholar.org/4ef5/260b4709a8f8427e5af24724878780b5e39a.pdf> (accessed: June 2018).
4. S. Khazaei, S. Ahmadi, Ciphertext-only attack on $d \times d$ Hill in $O(d^{13^d})$, Available from: <https://eprint.iacr.org/2015/802.pdf> (accessed: June 2018).
5. Figuring out key in hill cipher (chosen-plaintext attack), Available from: <https://crypto.stackexchange.com/questions/3882/figuring-out-key-in-hill-cipher-chosen-plaintext-attack> (accessed: June 2018).
6. V. Strassen, Gaussian elimination is not optimal, Numerische Mathematik, 13(4):354–356, 1969.
7. Java Documentation, Available from: <https://docs.oracle.com/javase/8/docs/> (accessed: June 2018).
8. JavaFX Documentation, Available from: <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784> (accessed: June 2018).
9. Eclipse Documentation, Available from: <https://www.eclipse.org/> (accessed: June 2018).
10. Ant Documentation, Available from: <https://ant.apache.org/manual/index.html> (accessed: June 2018).
11. Git Documentation, Available from: <https://git-scm.com/doc> (accessed: June 2018).
12. GitHub Documentation, Available from: <https://github.com/> (accessed: June 2018).
13. Java Web Start Documentation, Available from: https://www.java.com/en/download/faq/java_webstart.xml (accessed: June 2018).
14. JAMA User Guide, Available from <https://math.nist.gov/javanumerics/jama/> (accessed: June 2018).
15. Gson Library Documentation, Available from: <https://github.com/google/gson> (accessed: June 2018).