

SVEUČILIŠTE U ZAGREBU  
FILOZOFSKI FAKULTET  
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI  
Ak. god. 2020./2021.

Nikola Klobučar

**Izrada konzolne aplikacije za pomoć pri učenju  
kritopisnih sustava**

Završni rad

Mentor: dr. sc. Vjera Lopina

Zagreb, lipanj 2021.

## **Izjava o akademskoj čestitosti**

Izjavljujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenom i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

*Ovdje možete napisati kratku zahvalu ili stranicu možete ostaviti praznom.*

# Sadržaj

Sadržaj .....	ii
1. Uvod .....	1
2. GLAVNI DIO – O KRIPTOLOGIJ I .....	2
2.1. Zamjenski kritopisni sustav(i)? .....	2
2.1.1. Postupak zakrivanja koristeći Zamjenski kritopisni sustav .....	3
2.1.2. Postupak raskrivanja koristeći Zamjenski kritopisni sustav .....	4
2.1.3. Programska implementacija .....	4
2.1.4. Problemi .....	4
2.2. Premještajni kritopisni sustavi .....	4
2.2.1. Postupak zakrivanja koristeći jednostruki stupačni premještaj vs. dvostruki stupačni premještaj .....	5
2.2.2. Postupak raskrivanja koristeći jednostruki stupačni premještaj vs. dvostruki stupačni premještaj .....	6
2.2.3. Programska implementacija .....	7
2.2.4. Problemi .....	8
2.3. Složeni kritopisni sustav .....	8
2.3.1. Postupak zakrivanja koristeći Složeni kritopisni sustav .....	8
2.3.2. Postupak raskrivanja koristeći Složeni kritopisni sustav .....	9
2.3.3. Programska implementacija .....	9
2.3.4. Problemi .....	9
2.4. Polialfabetski kritopisni sustav (Vigenereova šifra) .....	9
2.4.1. Postupak zakrivanja koristeći Polialfabetski kritopisni sustav .....	10
2.4.2. Postupak raskrivanja koristeći Polialfabetski kritopisni sustav .....	11
2.4.3. Programska implementacija .....	12
2.4.4. Problemi .....	12

2.5. DES – Data Encryption Standard.....	12
2.5.1. Postupak zakrivanja i raskrivanja koristeći DES .....	16
2.5.2. Programska implementacija i problemi .....	17
2.6. Asimetrični kritopisni sustav (RSA – Rivest, Shamir, Adleman) .....	17
2.6.1. Postupak zakrivanja koristeći Asimetrični kritopisni sustav.....	18
2.6.2. Postupak raskrivanja koristeći Asimetrični kritopisni sustav .....	18
2.6.3. Programska implementacija .....	19
2.6.4. Problemi .....	19
3. Zaključak .....	20
4. Literatura .....	21
5. Popis slika .....	22
6. Prilog .....	23
Sažetak .....	24
Summary .....	25

# 1. Uvod

Predmet bavljenja kolegija Kriptologije jest upoznavanje studenata s tradicionalnim i suvremenim kriptopisnim sustavima, uporabom raznih vrsta zamjenskih, premještajnih i složenih sustava te s razumijevanjem načela modernih kriptopisnih sustava. U tom pogledu bi rad, opisan u ovom djelu, bio koristan jer omogućuje prisjećanje postupka zakrivanja, odnosno raskrivanja poruka te čak omogućuje i direktno zakrivanje, odnosno raskrivanje koristeći prave algoritme i objašnjavajući postupno što program radi. U ovom radu će biti ukratko opisan svaki kriptopisni sustav s kojim se studenti upoznaju na kolegiju, tijekom izrade konzolnog programa koji omogućuje zakrivanje i raskrivanje poruka tim sustavima te problemi koji su nastali tijekom izrade.

Cilj rada je omogućiti studentima lakše upoznavanje s kriptopisnim sustavima tako što imaju mogućnost individualnog pristupa svakom sustavu, što nije moguće u ovako velikoj ustanovi kao što je Filozofski fakultet. U svrhu rasprostranjivanja programa i olakšavanja korištenja, odabran je programski jezik C# (čitaj: „see sharp“) koji prilikom kompajliranja programa kreira binarnu izvršnu datoteku koju je moguće pokrenuti na bilo kojem Windows računalu bez potrebe ikakve instalacije.

## 2. GLAVNI DIO – O KRIPTOLOGIJI

Malo o povijesti kriptologije, zašto je čak i bitnija danas, format poruke, podjela kriptopisnih sustava

### 2.1. Zamjenski kriptopisni sustav(i)?

Zamjenski kriptopisni sustav ne mijenja redoslijed ni raspored slova, nego samo njihovu vrijednost, odnosno preslikavaju se u neka druga slova. Postoji nekoliko vrsta zamjenskih kriptopisnih sustava. Monoalfabetски sustav koristi fiksnu zamjenu tokom cijele poruke. Homofoni sustavi dodaju nasumičnost u monoalfabetски sustav. Poligramski sustav rade zamjenu nad većim grupama slova u jasnopisu. Polialfabetски sustav koristi nekoliko mogućnosti zamjena ovisno o položaju pojedinih slova.

Primjer monoalfabetskog zamjenskog kriptopisnog sustava je Cezarova šifra. Cezarova šifra je koristila zakritni slovored dobiven pomakom slova abecede za 3 udesno, odnosno zakritni slovored je počinjao slovom "D". Prvi red u tablici na slici 1 je jasnopisni slovored, a drugi kriptopisni.

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
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

Slika 1. Cezarov jasnopisni i zakritni slovored

Algoritam za kriptiranje/enkriptiranje Cezarovog kriptopisnog sustava:

- $z_i = (j_i + K) \bmod 26$
- $j_i = (z_i + (26 - K)) \bmod 26$
- $z_i$  – pojedino slovo zakritka
- $j_i$  – pojedino slovo jasnopisa
- $K$  – ključ, u ovom slučaju 3

Prilikom zakrivanja i raskrivanja ključevi su isti, ali su algoritmi različiti. Kod zakrivanja imamo pomak udesno za 3 pozicije, a kod raskrivanja pomak ulijevo za 3 pozicije. Algoritmi su nepromjenjivi dio ovog kriptopisnog sustava, a ključ je

promjenjivi dio. Odnosno, uvijek je prilikom zakrivanja pomak udesno. Samo je pitanje za koliko mjesta.

Cezarov kritopisni sustav je dosta slab i jednostavan za kriptanalitičara. Ima samo 26 mogućih ključeva te je jednostavno isprobati sve opcije. Da bi se povećao broj mogućih ključeva potrebno je promijeniti ključ. Ključ ne mora biti pomak za određeni broj, nego ključ može biti bilo koja druga permutacija slova.

### 2.1.1. Postupak zakrivanja koristeći Zamjenski kritopisni sustav

Zamjenskim kritopisnim sustavom se može zakrivati na dva načina, pomakom i ključem. Pomak je cijeli broj u rasponu 0-(n-1), gdje je n broj slova abecede. Dakle, za hrvatski jezik bi taj raspon bio 0-29 jer se hrvatska abeceda sastoji od trideset slova. Pomak nam govori za koliko treba pomaknuti početno slovo abecede, odnosno, ukoliko je pomak 3, onda zakritni slovored izgleda ovako:

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

Slika 2. Zakritni slovored kreiran koristeći pomak "3"

Ključ je neka riječ hrvatskog jezika koju ispišemo tako da maknemo slova koja se ponavljaju te ispod tih slova upisujemo abecednim redom ostala slova abecede. Na kraju iščitamo ta slova redom po stupcima. Kao primjer je dan ključ „ključ“:

```
privremeni slovored:  
K Lj U Č  
A B C Ć  
D DŽ Đ E  
F G H I  
J L M N  
Nj O P R  
S Š T V  
Z Ž  
jasnopisni slovored:  
A B C Č Ć D DŽ Đ E F G H I J K L Lj M N Nj O P R S Š T U V Z Ž  
zakritni slovored:  
K A D F J Nj S Z Lj B DŽ G L O Š Ž U C Đ H M P T Č Ć E I N R V
```

Slika 3. Kreiranje zakritnog slovoreda ključem



Nakon kreiranog zakritnog slovoreda, traži se pojedino slovo poruke u jasnopisnom slovoredu te se umjesto tog slova upisuje slovo koje se nalazi na tom mjestu u zakritnom slovoredu, a ostali znakovi i interpunkcije se samo prepisu.

### **2.1.2. Postupak raskrivanja koristeći Zamjenski kritopisni sustav**

Prilikom raskrivanja zamjenskim kritopisnim sustavom, potrebno je kreirati zakritni slovored kao i kod zakrivanja (vidi: 2.1.1.), ali se sada slova zakritka traže u zakritnom slovoredu, a ispisuju se ona iz jasnopisnog slovoreda.

### **2.1.3. Programska implementacija**

Nakon pokretanja programa, zamjenski kritopisni sustav se odabire upisivanjem slova „z“ te pritiskom na tipku enter. Zatim je potrebno odabrati između zakrivanja i raskrivanja poruke, pripadajućim unosima „z“ i „r“. Kada je odabrano zakrivanje ili raskrivanje, potrebno je upisati poruku, pritom pazeći na format poruke, i ključ ili pomak. Program početno ispisuje cijeli postupak te postepeno ispisuje što je potrebno napraviti u kojoj fazi.

### **2.1.4. Problemi**

Prvi problem koji se pojavio, a koji se odnosi i na sve druge kritopisne sustave, je problem dvoglasa. Dogovoreno je da se dvoglas pišu prvi znak velikim, a drugi malim slovom. Sva ostala slova pišu se velikim slovom. Do tog problema dovela je pretpostavka da u hrvatskom jeziku nije moguće imati slovo „L“ pa slovo „J“, a da oni ne čine slovo „LJ“. Međutim, upisivanjem valjanih zakritaka, moguće je upisati dva susjedna slova koja inače u hrvatskom tvore dvoglas, npr. „L“ i „J“, a da to ne vrijedi za zakritak. Dakle u hrvatskom bi to bilo jedno slovo, a u zakritku nije. Zbog toga je bilo potrebno uvesti provjeru formata poruke i time ne dozvoliti unos niti malih niti velikih slova.

## **2.2. Premještajni kritopisni sustavi**

Premještajni kritopisni sustav radi na principu promjene redoslijeda slova jasnopisa, a ne njihovih vrijednosti. U ovom sustavu ključ predstavlja primjenjenu transpoziciju. Prva šifra ovog tipa je Skitala koju su koristili Spartanci oko 500.p.ne. Princip rada šifre je jednostavan. Obavija se kožna traka oko štapa, zatim se poruka piše na tako dobijenom omotaču duž štapa. Nakon što je poruka napisana, kožna traka se

razmota te se na njoj dobije zakritak. U tom slučaju, ključ je bila debljina štapa jer bez štapa pravilne debljine nije bilo moguće pročitati poruku.

U današnje doba bi se Skitala mogla rekreirati dvodimenzionalnom matricom. Broj stupaca matrice određen je duljinom ključa, a broj redaka duljinom poruke. U retke se upisuje jasnopis, a zakritak se iščitava stupac po stupac i to uzlaznim redoslijedom indeksa pojedinog slova ključa.

Kriptoanalitičar, u ovom slučaju, može pomoću duljine zakritka pokušati odrediti o kojoj je matrici riječ. Ali čak i da to uspije, onda mora ispisivati sve moguće kombinacije stupaca jer ne znajući ključ, ne zna ni redoslijed iščitanih stupaca koji tvore zakritak. Ovaj kritopisni sustav je puno sigurniji od zamjenskog jer je algoritam kompliciraniji, ali i domena ključeva je puno veća. Postoji i dvostruki stupačni premještaj koji je još sigurniji od jednostrukog jer zahtjeva ponavljanje postupka zakrivanja poruke.

### **2.2.1. Postupak zakrivanja koristeći jednostruki stupačni premještaj vs. dvostruki stupačni premještaj**

Za zakrivanje premještajnim sustavom potreban je ključ koji je jedna hrvatska riječ, što duža naravno. Ispod svakog slova se ispišu brojčane oznake abecednim redoslijedom. Ukoliko, riječ sadrži nekoliko istih slova, brojevi se upisuju slijeva nadesno.

A	S	O	N	A	N	C	A
1	8	7	5	2	6	4	3

Slika 4. Primjer ključa s indeksom za pojedino slovo

Ta slova ključa čine stupce tablice u koju se upisuje poruka. Nakon upisane poruke, ispisuje se zakritak. Zakritak se sastoji od slova iz stupaca ispisanih po redu brojeva iznad stupca. Dakle, prvo se ispisuju sva slova iz stupca ispod broja „1“, zatim ispod broja „2“ itd. Dakako zakritak se zaisuje u blokovima od pet znakova. Time je završeno jednostruko stupačno zakrivanje.

A	S	O	N	A	N	C	A
1	8	7	5	2	6	4	3
T	K	O	M	E	Z	O	V
E	K	R	O	Z	S	N	O
V	E	M	O	J	E		

Ispis zakritka:  
TEVEZ JVOON MOOZS EORMK KE

Slika 5. Primjer jednostrukog stupačnog premještaja

Ukoliko se radi o dvostrukom stupačnom premještaju, onda je potrebno zakritak dobiven jednostrukim stupačnim premještajem ponovno upisati u tablicu te ponoviti postupak. Novodobiveni zakritak čini konačni zakritak poruke koji se onda može poslati primatelju poruke.

```
A S O N A N C A
1 8 7 5 2 6 4 3
T K O M E Z O V
E K R O Z S N O
V E M O J E

A S O N A N C A
1 8 7 5 2 6 4 3
T E V E Z J V O
O N M O O Z S E
O R M K K E

Ispis zakritka:
TOOZO KOEVS EOKJZ EVMME NR
```

Slika 6. Primjer dvostrukog stupačnog premještaja

### 2.2.2. Postupak raskrivanja koristeći jednostruki stupačni premještaj vs. dvostruki stupačni premještaj

Da bi se raskrio zakritak koji je zakriven metodom jednostrukog stupačnog premještaja potrebno je prvo odrediti dimenzije tablice. Dimenzije tablice sastoje se od broja redaka i broja stupaca. Broj stupaca je određen duljinom ključa, odnosno brojem znakova od kojih se ključ sastoji. Broj redaka se izračuna matematičkom operacijom dijeljenja i to tako da se podijeli broj znakova zakritka s brojem znakova ključa, odnosno stupaca. Broj koji se dobije je broj punih redaka, a ostatak dijeljenja govori o broju znakova u posljednjem, dodatnom, retku.

```
Broj stupaca je: 8  
Broj redaka je duljina poruke/duljina ključa i iznosi: 3
```

Slika 7. Ispis broja stupaca i redaka

Nakon što su izračunate dimenzije tablice, vrijeme je da se upiše zakritak. Zakritak se upisuje tako da se broj znakova, određen temeljem matematičke operacije dijeljenja, upiše u stupac ispod broja „1“, zatim ispod broja „2“ itd. Pri završetku upisa zakritka u tablicu, ukoliko je sve odrađeno pravilno, poruku je moguće iščitati iz tablice čitajući znakove u retcima slijeva nadesno.

No, međutim, ukoliko se radi o dvostrukom stupačnom premještanju, poruka ni tada neće biti raskrivena, nego je potrebno ispisati taj prvi raskritak po retcima te ponoviti upis u tablicu po stupcima određenim rednim brojevima. To će, naposljetku, prikazati originalan jasnopis zakriven dvostrukim stupačnim premještanjem.

### 2.2.3. Programska implementacija

Nakon pokretanja programa, upisuje se slovo „p“ za odabir premještajnog kritopisnog sustava. Zatim se odabire jednostruki stupačni premještaj ili dvostruki stupačni premještaj, upisivanjem „j“ ili „d“. Nakon odabira vrste premještajnog kritopisnog sustava, odabire se postupak zakrivanja ili raskrivanja unosom slova „z“ za zakrivanje ili slova „r“ za raskrivanje. Nakon odabira svih tih opcija, došlo je vrijeme za upis poruke i ključa.

Program ispisuje postupak zakrivanja premještajnim kritopisnim sustavom te ispisuje jednu ili obje tablice, ovisno o odabranoj vrsti premještaja. Konačno, ispisuje zakritak u blokovima od po pet znakova, odvojenih razmakom.

Naravno, prilikom odabira raskrivanja, program ispisuje postupak raskrivanja te tablicu s jasnopisom koji je potrebno iščitati iz tablice te ga zapisati sa svim nedostajućim interpunkcijama.

Ako se radi o dvostrukom stupačnom premještanju, onda se kao nova poruka sprema novodobiveni zakritak te se ponavlja postupak. To je bilo jednostavno implementirati u program jer nije potrebno sve ponovo zapisivati, nego se može ponovno pozvati ista funkcija s tim novim izmijenjenim argumentima.

#### **2.2.4. Problemi**

Općeniti problem ovog sustava je nemogućnost prijenosa interpunkcijskih znakova pa ovisno o duljini poruke to može dovesti do nepreglednosti ili pomutnje pri iščitavanju poruke nakon raskrivanja.

Javio se i problem pri ponavljanju postupaka zakrivanja, odnosno raskrivanja, u dvostrukom stupaćnom premještaju jer bi program dva puta ispisao postupak zakrivanja te time učinio ispis nepreglednim. To je bilo jednostavno otkloniti koristeći `if()` grananje koje je osiguravalo ispis postupka samo pri prvom pozivu funkcije premještajnog kritopisnog sustava.

#### **2.3. Složeni kritopisni sustav**

Složeni kritopisni sustav koristi oba prijašnja sustava za kreiranje zakritka. Time se povećava sigurnost sustava jer se dodatno komplicira algoritam. Ovaj sustav mijenja prvo redoslijed, a zatim i vrijednost pojedinog slova jasnopisa.

Kriptoanalitičar ima puno problema s razbijanjem ovog sustava. On mora pronaći ključ koji je korišten u oba dijela zakrivanja, a to nije tako jednostavno. Da bi otkrio ključ mora prvo raskriti premještajni sustav. Lako se otkrije prava matrica, ali sada nije moguće „pogađati“ redoslijed stupaca jer su slova zamijenjena pa ne daju smislene riječi. Iz tog razloga nije moguće pronaći ključ, a bez ključa nije moguće raskriti zamjenski sustav. Zamjenski kritopisni sustav je inače relativno jednostavan za kriptoanalizu jer je moguće isprobati sve opcije - broj opcija ovisi o broju slova abecede pojedinog jezika - te pronaći pravu, ali u ovom slučaju korišten je i premještajni sustav te je uz svaku opciju zamjene potrebno pokušati pronaći smislenu riječ premještajući redoslijed slova u riječima – ne znajući duljinu pojedine riječi. Korisni napad na ovaj sustav je frekvencijska analiza pojedinih znakova u jeziku čije se zakrivene poruke žele čitati bez poznavanja ključa.

##### **2.3.1. Postupak zakrivanja koristeći Složeni kritopisni sustav**

Postupak zakrivanja jasnopisa složenim kritopisnim sustavom sastoji se od kreiranja kritopisnog slovoreda u zamjenskom kritopisnom sustavu koristeći ključ. Zatim se tim ključem zakrije poruka koristeći premještajni kritopisni sustav. Konačan zakritak se dobije zakrivanjem privremenog zakritka - dobivenog premještajnim kritopisnim

sustavom – koristeći novi kritopisni slovored – dobiven zamjenskim kritopisnim sustavom.

### **2.3.2. Postupak raskrivanja koristeći Složeni kritopisni sustav**

Postupak raskrivanja kritopisa složenim kritopisnim sustavom sastoji se od kreiranja kritopisnog slovoreda koristeći ključ u zamjenskom kritopisnom sustavu. Zatim se tim slovoredom raskrije zakritak i dobije privremena poruka. Završno se ta privremena poruka raskrije u premještajnom sustavu koristeći isti ključ. Poruka se iščita iz tablice koju kreira premještajni sustav.

### **2.3.3. Programska implementacija**

Nakon što je program pokrenut, upiše se slovo „s“ te se tako odabere složeni kritopisni sustav. Zatim se odabire raskrivanje ili zakrivanje te se upisuju poruka i ključ. Program će tim ključem ući u klasu zamjenskog kritopisnog sustava te kreirati kritopisni slovored. Nakon toga će program istim ključem ući u klasu premještajnog kritopisnog sustava te njime kreirati privremeni zakritak. Prilikom ulaska u druge klase, odnosno pozivanja funkcija drugih klasa, program će ispisivati poruke što obavlja tako da korisnik može pratiti što se događa.

### **2.3.4. Problemi**

Najveći problemi su bili kako iskoristiti funkcije drugih klasa. Bilo je potrebno korigirati vidljivosti pojedinih objekata, pozivati ih s drugog mjesta u kodu te je neke stvari čak trebalo i kopirati u novu klasu.

Idući problem je bio kontroliranje ispisa jer je drugačiji ispis prilikom pokretanja zakrivanja samo zamjenskim ili samo premještajnim kritopisnim sustavom.

Zakrivanje koristeći jedan jednostavni sustav ispisuje cijeli postupak zakrivanja koji nije potreban prilikom zakrivanja složenim kritopisnim sustavom jer taj ima svoj ispis postupka.

## **2.4. Polialfabetски kritopisni sustav (Vigenereova šifra)**

U zamjenskom kritopisnom sustavu svakom slovu jasnopisa odgovara jedinstveno slovo zakritka. To je vrsta monoalfabetskog kritopisnog sustava. Vigenereova šifra je primjer polialfabetskog zamjenskog kritopisnog sustava. Kao što ime kaže ovo je zamjenski sustav koji umjesto jedne opcije zamjene za pojedino slovo, ima ih više. Metodu sustava je prvobitno opisao Giovan Battista Bellaso, ali je kasnije shema

algoritma bila pripisana Blaisu de Vigenereu te je tako sustav dobio i ime. Ovaj sustav se može opisati i kao multi-Cezarov sustav jer svako slovo ima svoju zamjenjeni kritopisni slovored, odnosno pripadajući parnjak kojim ga se mijenja u zakritku.

Vigenereova šifra je jedna od najpopularnijih kritopisnih sustava u povijesti. Bila je u širokoj uporabi tijekom Američke revolucije, krajem 18. st., a korištena je i u Američkom građanskom ratu.

Postupak kriptanalize je relativno kompliciran, ali zahtjevi su jednostavni. Potrebno je prvo odrediti duljinu ključa. To je moguće na dva načina. Prvi način je pomoću Kasiskijevog testa, a drugi pomoću indeksa koincidencije. Nakon određene duljine ključa potrebno je odrediti sam ključ. To je moguće postići koristeći međusobni indeks koincidencije dvaju nizova.

	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 8. Izgled Vigenereove tablice

#### 2.4.1. Postupak zakrivanja koristeći Polialfabetski kritopisni sustav

Prvi korak u zakrivanju poruke koristeći Polialfabetski kritopisni sustav je dati indeks, odnosno brojčanu vrijednost, svakom slovu abecede. Za hrvatski jezik su to brojevi u rasponu 0-29. Zatim je potrebno zapisati jasnopis te ispod njega ispisati ključ i to tako da ide slovo ispod slova te ukoliko je potrebno, ponavljati slova ključa dok se ne stigne do kraja jasnopisa. Na taj način se omogućuje zakrivanje koristeći nekoliko različitih alfabeti. Slovo zakritka se dobije korištenjem uređenog para brojčanih vrijednosti za slovo jasnopisa i slovo ključa. Postoje dva načina za pronalaženje

slova zakritka. Prvi način je da se slovo pronađe po koordinatama uređenog para u tablici zvanj „Vigenereov kvadrat“, a drugi da se brojčana vrijednost slova zakritka izračuna po formuli iz modularne aritmetike koja glasi:  $(J+K) \bmod(30)=Z$ , gdje je J slovo jasnopisa, K slovo ključa, 30 je broj slova u abecedi, a Z slovo zakritka.

```
Ispis jasnopisa i ključa:
POLIALFABETSKI KRITOPISNI SUSTAV
VIGENEREVIGENE REVIGENERE VIGENE

Primjer izračuna nekoliko znakova zakritka:
P->21 + V->27 mod(30) = 18->N
O->20 + I->12 mod(30) = 2->C
L->15 + G->10 mod(30) = 25->T

Ispis zakritka:
NCTONSBEZODBCO DŽAFĐAŽABGO OEČČND
```

Slika 9. Primjer zakrivanja korištenjem modularne aritmetike

#### 2.4.2. Postupak raskrivanja koristeći Polialfabetski kritopisni sustav

Prilikom raskrivanja poruke u polialfabetskom kritopisnom sustavu je, također, potrebno zapisati slova zakritka te ispod njih slova ključa. Nakon što svako slovo zakritka ima pripadajuće slovo ključa, odnosno pripadajući alfabet korišten za zakrivanje tog slova, ispisuju se uređeni par slovo zakritka – slovo ključa. Slovo jasnopisa se dobije traženjem po koordinatama uređenog para u tablici ili po formuli modularne aritmetike.  $(Z-K) \bmod(30)=J$

```
Ispis zakritka i ključa:
NCTONSBEZODBCO DŽAFĐAŽABGO OEČČND
VIGENEREVIGENE REVIGENERE VIGENE

Primjer izračuna nekoliko znakova raskritka:
N->18 - V->27 mod(30) = 21->P
C->2 - I->12 mod(30) = 20->O
T->25 - G->10 mod(30) = 15->L

Ispis jasnopisa:
POLIALFABETSKI KRITOPISNI SUSTAV
```

Slika 10. Primjer raskrivanja korištenjem modularne aritmetike



### 2.4.3. Programska implementacija

Nakon pokretanja programa korisnik upiše slovo „q“ te tako izabere kritopisni sustav „Kvadratna šifra“. Nakon toga se odabire između zakrivanja („z“) i raskrivanja („r“) te se upisuju poruka i ključ. Naravno, oboma se provjerava sintaksa. Nakon što je sve pravilno uneseno, program poziva metodu zakrivanja ili raskrivanja koja se nalazi u klasi „KvadratnaŠifra“. Ukoliko je odabrano zakrivanje, ispisuje se postupak zakrivanja, jasnopis, ključ te primjer izračuna indeksa za nekoliko znakova zakritka i konačno, cijeli zakritak. S druge strane, odabirom raskrivanja, ispisuje se postupak raskrivanja, zakritak, ključ te primjer izračuna indeksa za nekoliko znakova jasnopisa i konačno, cijeli jasnopis.

### 2.4.4. Problemi

Ovo je bio jedan od jednostavnijih sustava za izradu. Bilo je potrebno izraditi listu slova abecede da bi se omogućilo indeksiranje. Jedini problematični dio je bila prilagodba znakova ključa znakovima poruke. Potrebno je osigurati razmake na jednakim mjestima te nastavak ispisa znakova ključa, a ne preskakanje onih koji se nalaze na tom mjestu. Taj problem je riješen tako da se u novu varijablu upisuju znakovi ključa sve dok ne naiđe na razmak u poruci. Kad naiđe na razmak, smanji indeks za jedan te prepíše taj znak u ključ. U ovom trenutku, program bi samo prepisao i sve druge posebne znakove, odnosno one koji se ne nalaze u abecedi.

## 2.5. DES – Data Encryption Standard

Data Encryption Standard je razvijen 1970-ih godina. Bazira se na IBM-ovoj fejestel šifri zvanj Lucifer. Fejestel šifra znači da se poruka dijeli na lijevu i na desnu polovinu. DES je također i blokovska šifra jer zakriva blokove jasnopisa veličine 64 bita. NSA (engl. National Security Agency) je kreirala DES iz Lucifera tako što je izmijenila duljina ključa sa starih 128 bitova na 56 bita. Prema tome je DES lakše razbiti od Lucifera koristeći potpunu pretragu ključeva. Sumnjalo se da je NSA namjerno oslabila DES, ali naknadne kriptanalize su pokazale da je DES praktično jak kao i s ključem duljine 128 bita. Zatim je NSA promijenila kutije zamjene, tzv. „S - kutije“. I to je izazivalo sumnju o ostavljanju stražnjeg ulaza (engl. *backdoor*), ali s vremenom je postalo jasno da su te promjene S-kutija zapravo ojačale algoritam nudeći zaštitu protiv kriptanalitičkih metoda koje nisu bile poznate u to vrijeme. Cilj je uvesti konfuziju u zakritak tako da on ovisi od jasnopisa i ključa te difuziju tako da

svaki bit zakritka ovisi o svim bitovima jasnopisa i ključa. Uz to, kod DESa, promjena jednog bita ključa ili jednog bita od 64 bita jasnopisa mijenja 50% bita bloka zakritka.

Sumirajući, DES je Feistel šifra s dužinom bloka jasnopisa od 64 bita, duljinom ključa od 56 bitova (ukupno  $2^{56}$  mogućih ključeva), prođe kroz 16 iteracija prije nego generira jedan blok zakritka te je duljina ključa pri svakoj iteraciji jednaka 48 bita.

Ključne dijelove DES algoritma čine kutije koje mogu proširivati, mijenjati poruku i ključ te generirati podključeve.

E-kutija odrađuje ekspanzivnu permutaciju, odnosno proširuje svoj ulaz od 32 bita na 48 mijenjajući redoslijed bitova te čak i ponavljajući neke. Točno mapiranje prikazano je ekspanzivnom tablicom E (vidi sliku 11). E-kutija je potrebna jer se jedna polovica poruke sastoji od 32 bita, a ključ kojim se zakriva u ovoj iteraciji je 48 bita. E-kutija se brine da su nizovi jednake duljine da bi bilo moguće izvesti XOR operaciju nad ta dva niza.

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

*E*

Slika 11. Grafički prikaz E-kutije

S-kutije su jedna od najvažnijih sigurnosnih funkcija u kompletnom algoritmu. Za razliku od drugih linearnih kutija, S-kutije su nelinearne i zato teške za kriptanalizu. Svaka S-kutija preslikava 6 bitova u 4 bita, a DES ih koristi osam te tako preslikava 48 bitova u 32 bita. S-kutije su zapravo matrice koje se sastoje od četiri retka i šesnaest stupaca te se na svakom sjecištu retka i stupca nalazi neka 4-bitna vrijednost. Koriste se iste kutije kroz svaku iteraciju i služe za ponovno smanjenje duljine polovice poruke s 48 na 32 bita.

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0yyyy1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
1yyyy0	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
1yyyy1	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S-box 1

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
0yyyy1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
1yyyy0	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
1yyyy1	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S-box 2

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
0yyyy1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
1yyyy0	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1yyyy1	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S-box 3

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
0yyyy1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
1yyyy0	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
1yyyy1	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S-box 4

	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
0yyyy1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
1yyyy0	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
1yyyy1	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S-box 5

Slika 12. Prikaz prve četiri S-kutije (S1-S4)

S <sub>5</sub>																
	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
0yyyy1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
1yyyy0	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
1yyyy1	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S <sub>6</sub>																
	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
0yyyy1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
1yyyy0	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
1yyyy1	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S <sub>7</sub>																
	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
0yyyy1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1yyyy0	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
1yyyy1	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S <sub>8</sub>																
	x0000x	x0001x	x0010x	x0011x	x0100x	x0101x	x0110x	x0111x	x1000x	x1001x	x1010x	x1011x	x1100x	x1101x	x1110x	x1111x
0yyyy0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
0yyyy1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1yyyy0	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1yyyy1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Slika 13. Prikaz druge četiri S-kutije (S5-S8)

P-kutija čini inicijalnu permutaciju nad 64-bitnim blokom jasnopisa. Ne doprinosi sigurnosti sustava i njena prava svrha nije poznata. To je jedina kutija koja utječe na cijeli blok jasnopisa, a kasnije i zakritka.

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

***P***

Slika 14. Grafički prikaz P kutije

K-kutije služe za generiranje šesnaest podključeva duljine 48 bita na osnovu ključa duljina 56 bita. Sastoji se od tri faze. Prvo se odbacuju krajnjih 8 bitova ključa (bitovi parnosti dodani na početku), zatim se preostali bitovi mijenjaju na temelju tablice PC-1 (Permutation Change 1). U drugoj fazi se ključ dijeli na lijevu i desnu polovicu te se kreira šesnaest podključeva s dužinom od 48 bitova. Konačno se u trećoj fazi koristi PC-2 za kreiranje prvog podključa.

<i>C</i>						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
<i>D</i>						
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	13	28	20	12	4

**PC1**

Slika 15. Prikaz Permutacijske tablice 1 (PC1)

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

**PC2**

Slika 16. Prikaz Permutacijske tablice 2 (PC2)

### 2.5.1. Postupak zakrivanja i raskrivanja koristeći DES

Zakrivanje koristeći DES počinje s nasumičnim odabirom svakog bita od ukupnih 56 bita ključa. Taj ključ se proširuje tako da se na svako osmo mjesto dodaje bit parnosti. Ukoliko je u prošlih sedam bitova paran broj „1“, onda se dodaje „1“ kao osmi bit, a inače se dodaje 0. Zatim se kreira 16 podključeva duljine 48 bita koristeći K-kutije. Nakon kreiranja podključeva kreće obrada jasnopisa.

Blok jasnopisa se sastoji od 64 bita, odnosno od osam 8-bitnih ASCII znakova. Prvo se kodira blok jasnopisa koristeći Inicijalnu permutaciju IP. Zatim se taj permutirani 64-bitni niz podijeli na lijevu i desnu polovicu te se proširi s 32 bita na 48 koristeći E-kutiju. Kada su ključ i permutirani jasnopis jednake duljine, 48 bita, vrijeme je da se provede XOR operacija nad njima. Koristeći S-kutije dobiveni niz se smanjuje s 48 bita na 32 bita i to tako da se podijeli niz na osam jednakih dijelova od po 6 bita. Nad svakim dijelom se obavlja jednak postupak. Prvi i zadnji bit govore o retku u S-kutiji, a srednja četiri bita govore o stupcu. Na sjecištu tog retka i stupca, nalazi se 4-bitna vrijednost koja zamjenjuje onu inicijalnu 6-bitnu. Svaki 6-bitni dio ima svoju S-kutiju. Nad novodobivenim nizom se obavlja XOR operacija s lijevom stranom iz prošle iteracije te to postaje nova lijeva strana, i obratno. Na kraju svih šesnaest iteracija dobije se 64 bitni niz  $L_{16}R_{16}$  kojem se zamijeni redoslijed pa postane  $R_{16}L_{16}$  i nad

time se obavi konačna permutacija  $IP^{-1}$  te se dobije zakritak veličine 64 bita, odnosno osam 8-bitnih ASCII znakova.

Raskrivanje je identičan proces zakrivanju, ali se ključevi u iteracijama koriste suprotnim redoslijedom. Dakle, ne kreće se s  $K_0$ , nego s  $K_{16}$ .

### **2.5.2. Programska implementacija i problemi**

Zbog velikog broja iteracija i nemogućnosti kvalitetnog prikaza svih potrebnih tablica, odnosno kutija, DES je u program implementiran koristeći već ugrađenu DES klasu u C# u imenski prostor „System.Security.Cryptography“. Potrebno je bilo dodati imenski prostor te inicijalizirati klasu. U programu se ispisuje i postupak zakrivanja i raskrivanja za potrebe učenja, ali se ne ispisuje pojedinačno svaki korak. No, omogućeno je zakriti i raskriti poruku koristeći DES radi boljeg upoznavanja standarda te opcije korištenja istog na nesigurnim web sjedištima.

## **2.6. Asimetrični kritopisni sustav (RSA – Rivest, Shamir, Adleman)**

Dosad spomenuti kritopisni sustavi su tzv. Simetrični kritopisni sustavi, odnosno koriste samo jedan ključ za zakrivanje poruke. Glavni problem tih sustava je potreba za dogovaranjem, odnosno slanjem ključa što omogućuje kriptanalitičaru da presretne tu informaciju i olakša razbijanje ostalog dijela sustava. Whitfield Diffie i Martin Helman predlažu primjenu asimetričnog kritopisnog sustava. Asimetrična kriptologija koristi dva ključa. Jedan za zakrivanje i jedan za raskrivanje. To omogućuje slanje ključa za zakrivanje kroz komunikacijski kanal jer dok je ključ za raskrivanje tajan, sustav je siguran. Ovakvi sustavi rade sporije od simetričnih i nisu pogodni za zakrivanje velikih količina podataka, ali se mogu koristiti za zakrivanje, digitalni potpis i razmjenu simetričnih ključeva.

Najpoznatiji algoritam asimetrične kriptografije je RSA. Nazvan po početnim slovima prezimena njegovih autora, Ron Rivest, Shamir Adi i Leonard Adleman. Kao jednosmjernu funkciju koristi množenje jako velikih, čak stoznamenastih prostih brojeva jer smatraju da je rastavljanje velikih brojeva na faktore dovoljno dugotrajan posao da se može smatrati nemogućim. Ukoliko osoba želi poslati nekome zakrivenu poruku, mora je zakriti javnim ključem primatelja poruke. S obzirom na to da je ključ javan, svatko može svakome poslati poruku, ali je ne mogu pročitati jer je za to potrebno poznavati tajni ključ. Jednosmjerna funkcija koja se koristi je ovog oblika:  $f(x)=x^e \bmod n$ . Uz poznavanje vrijednosti tajnog ključa, moguće je pronaći

njenu inverznu funkciju, ali inače je praktično nerješiva. Naravno, da bi sustav funkcionirao, javni i tajni ključ moraju biti inverzno recipročni. To svojstvo se može provjeriti po ovoj formuli:  $e \cdot d \bmod \phi(n) = 1$ , gdje je  $e$  javni ključ,  $d$  tajni ključ,  $\phi(n)$  umnožak prostih brojeva, koji čine javni djelitelj, umanjenih za jedan te to obavezno mora biti jednako „1“ jer tako osigurava inverznu recipročnost. Tajnost ovog sustava zasniva se na složenosti faktORIZACIJE javnog djelitelja  $N$  koji se dobije umnoškom nasumičnih, proizvoljnih prostih brojeva. Valja napomenuti da ne postoji čvrst matematički dokaz je li rastavljanje na faktore jedini način za razbijanje RSA te nepostojanje kraćeg načina faktORIZACIJE. Javni ključ  $e$  također mora biti prost pa se često radi brzine uzima što manji broj, ali se time i narušava sigurnost sustava. „Mnogi korisnici koriste isti eksponent  $e$  unutar javnih ključeva. Obično je to vrijednost  $e=65537$ .“<sup>1</sup> Povećanjem duljine ključa bi se povećala sigurnost sustava, ali se i znatno usporava izvršavanje algoritma. Mjerenjem performansi RSA algoritma, utvrđeno je da je RSA algoritam 1500 puta sporiji od DES algoritma. Zbog tih svojstava, RSA se koristi samo za kriptiranje kriptografskih ključeva simetričnih sustava i digitalni potpis, a simetrični sustavi se koriste za zakrivanje poruka.

### 2.6.1. Postupak zakrivanja koristeći Asimetrični kritopisni sustav

Za zakrivanje poruke koristeći RSA kritopisni sustav, potrebni su javni ključ primatelja poruke te javni djelitelj. Javni djelitelj je umnožak dva ili tri velika prosta broja, otprilike stoznamenastih. Javni djelitelj se kreira prilikom svakog početka komunikacije. Nakon što su prikupljeni javni djelitelj i javni ključ primatelja poruke, potrebno je svako slovo jasnopisa pretvoriti u neki broj, npr. ASCII kod ili, u slučaju hrvatske abecede, u broj ovisno o mjestu u abecedi, dakle A-0 te Ž-29. Zatim se taj broj potencira vrijednošću javnog ključa primatelja poruke, dijeli javnim djeliteljem te ostatak tog dijeljenja daje indeks, odnosno broj koji pokazuje na kojem mjestu u abecedi se nalazi slovo kojim zakrivamo ovo.  $Z_i = j_i^e \bmod(n)$

---

<sup>1</sup> Adamović, S. Ž.; Veinović M. Đ.; Kriptologija 1 – Osnove za analizu i sintezu šifarskih sistema; Beograd: Univerzitet Singidunum, 2013.g. Str. 133

### 2.6.2. Postupak raskrivanja koristeći Asimetrični kriptopisni sustav

Za raskrivanje poruke koristeći RSA kriptopisni sustav, potrebni su tajni ključ primatelja poruke te javni djelitelj. Javni djelitelj je jednak onom korištenom za zakrivanje. Tajni ključ i javni ključ moraju biti recipročno inverzni. To se osigurava ovom formulom:  $d \cdot e \bmod \phi(n) = 1$ . Ta formula omogućava izračun tajnog ključa primatelja poruke. Za formulu nam je potreban Eulerov toličnik, koji se računa umnoškom za jedan umanjenih prostih brojeva kojima se kreirao javni djelitelj.  $\phi(n) = (p-1)(q-1)$ . Nakon izračuna tajnog ključa, potrebno je svako slovo kriptopisa pretvoriti u broj te potencirati tajnim ključem. Zatim se dijeli s javnim djeliteljem i ostatak te operacije čini indeks jasnopisnog slova koje je bilo zakriveno tim slovom.

$$J_i = z_i^d \bmod(n)$$

### 2.6.3. Programska implementacija

Nakon što je program pokrenut, upiše se slovo "r" za korištenje RSA kriptopisnog sustava. Zatim se upisuju vrijednosti javnog ključa te javnog djelitelja, odabire se raskrivanje ili zakrivanje upisivanjem "r" ili "z". Tek se sad upisuje poruka. Prikupljene su svi podaci i sad je moguće započeti postupak. Prvo se ispisuje cijeli postupak određenog postupka te se za svako slovo poruke ispisuje indeks, odnosno mjesto u abecedi, vrijednost indeksa nakon potenciranja, ostatak dijeljenja s javnim djeliteljem te konačno slovo kojim se zamjenjuje originalno slovo poruke. Konačno, ispisuje se cijela poruka.

### 2.6.4. Problemi

Najveći problem se javio pri pokušaju određivanja od kojih prostih brojeva se sastoji javni djelitelj. Trenutno je to riješeno tako da program kreira listu od otprilike tisuću prostih brojeva te prolazi kroz listu i pokušava podijeliti javni djelitelj s dva ili više prostih brojeva te traži kojim brojevima može podijeliti, a da nema ostatka, odnosno da je ostatak nula.

Idući problem koji se javio bio je vezan za funkciju potenciranja koja se nalazi ugrađena u C# programski jezik. To je funkcija naziva "Pow()" koja je dio "Math" datoteke (engl. *Library*). Pow() funkcija prima dva argumenta. Vrijednost baze koja se potencira te vrijednost potencije na koju se potencira, a tip podatka koji vraća kao rezultat je tipa double. Double zauzima više memorije te obuhvaća više brojeva od



int tipa kojeg je indeks. Promjena double tipa u int prilikom pretrage slova u abecedi  
jer riješilo problem.

### 3. Zaključak

U ovom radu je ukratko opisano nekoliko vrsta kriptopisnih sustava, od jednostavnijih prema kompliciranijima i novijima te time i sigurnijima. Uz detaljan opis postupka zakrivanja i raskrivanja, dane su upute korištenja te neki problemi prilikom izrade programa. Izrada programa, uz sve probleme, protekla je uspješno. Svi odabrani sustavi omogućavaju zakrivanje, raskrivanje poruka te ispis postupka i vrijednosti postepeno kako bi korisnici lakše naučili svojstva i algoritme pojedinog sustava.

Konzolna aplikacija je rađena u programu Microsoft Visual Studio, besplatnim računom koji nam je dodijelilo Sveučilište. Za izradu aplikacije korišten je programski jezik C#. Programski jezik Python bi bio jednostavniji za korištenje i izradu aplikacije, ali radi jednostavnosti uporabe, odabran je C#. Programski jezik C# kreira izvršnu datoteku nastavka ".exe" koju je moguće pokrenuti na bilo kojem Windows računalu te ju je moguće poslati korisniku i nije potrebno ništa drugo instalirati za razliku od Pythona koji zahtjeva instaliran interpreter te je to još jedan korak koji može izazivati probleme.

Svaki kriptopisni sustav je svoja klasa koje se mogu, neke i moraju, koristiti unutar drugih klasa. Postoji i jedna glavna klasa nazvana Program koja upravlja upisima i prenosi informacije između korisnika i programa.

Aplikaciju bi se moglo unaprijediti tako da se kreira web aplikacija umjesto ove konzolne te bi se tako omogućilo lakše upravljanje vizualnim i informacijskim dizajnom. To bi program učinilo preglednijim te se ne bi trebala niti slati izvršna datoteka. Korisnik bi bio u mogućnosti samo otvoriti poveznicu web stranice te tako koristiti algoritme i učiti.

Konzolna aplikacija je izrađena kako bi se mogla koristiti za pomoć u nastavi, ali može se koristiti i kao sloj zaštite prilikom slanja e-mail poruka koje inače nisu kriptirane. Pošiljatelj i primatelj bi se mogli dogovoriti oko kriptopisnog sustava te potrebnih podataka i tako omogućiti slanje kriptiranih poruka koristeći ranjive online sustave.

## **4. Literatura**

Adamović, S. Ž.; Veinović M. Đ.; Kriptologija 1 – Osnove za analizu i sintezu šifarskih sistema; Beograd: Univerzitet Singidunum, 2013.g.

Dujella A.; Maretić M.; Kriptografija; Zagreb: Element, 2007.g.

## 5. Popis slika

Slika 1. Cezarov jasnopisni i zakritni slovored.....	2
Slika 2. Zakritni slovored kreiran koristeći pomak "3" .....	3
Slika 3. Kreiranje zakritnog slovoreda ključem .....	3
Slika 4. Primjer ključa s indeksom za pojedino slovo .....	5
Slika 5. Primjer jednostrukog stupačnog premještaja.....	6
Slika 6. Primjer dvostrukog stupačnog premještaja.....	6
Slika 7. Ispis broja stupaca i redaka .....	7
Slika 8. Izgled Vigenereove tablice .....	10
Slika 9. Primjer zakrivanja korištenjem modularne aritmetike .....	11
Slika 10. Primjer raskrivanja korištenjem modularne aritmetike .....	11
Slika 11. Grafički prikaz E-kutije.....	13
Slika 12. Prikaz prve četiri S-kutije (S1-S4).....	14
Slika 13. Prikaz druge četiri S-kutije (S5-S8).....	14
Slika 14. Grafički prikaz P kutije .....	15
Slika 15. Prikaz Permutacijske tablice 1 (PC1) .....	16
Slika 16. Prikaz Permutacijske tablice 2 (PC2) .....	16

## **6. Prilog**

GitHub: <https://github.com/n-klobucar/KriptoLearn>

# **Izrada konzolne aplikacije za pomoć pri učenju kriptopisnih sustava**

## **Sažetak**

Ovaj rad sadrži kratak opis kriptopisnih sustava koji se obrađuju na kolegiju Kriptologije na Filozofskom fakultetu u Zagrebu i kao takav bi trebao služiti samo kao podsjetnik za pomoć u nastavi. Osim opisa zamjenskog kriptopisnog sustava, premještajnih kriptopisnih sustava i složenog kriptopisnog sustava te Vigenereove šifre i DES-a, opisan je i jedan asimetrični sustav pod nazivom RSA. Uz ovaj rad postoji i popratna konzolna aplikacija napisana koristeći C# programski jezik. Nakon pojedinog opisa kriptopisnog sustava napisane su upute za korištenje tog programa te problemi koji su se javljali prilikom njegove izrade. Od obrađenih simetričnih kriptopisnih sustava, samo je DES direktno ugrađen u C# u imenskom prostoru System.Security.Cryptography. U popratnoj aplikaciji se i koristila ta ugrađena implementacija jer je za potrebe kolegija dovoljan opis postupka zakrivanja i raskrivanja, a ovako je jednostavno omogućeno i korištenje DES sustava. Naravno, popratnu konzolnu aplikaciju moguće je koristiti i za razonodu ili kao dodatnu sigurnost na nepovjerljivim internetskim sustavima te ju to čini najjednostavnijim načinom uporabe starih kriptopisnih sustava u suvremenoj tehnologiji.

**Ključne riječi:** pomoć u nastavi, kriptologija, C# konzolna aplikacija, kriptopisni sustavi, završni rad

# Creating console application used for learning encryption systems

## Summary

This paper contains a brief description of cryptographic systems that are processed in the course of Cryptology at the Faculty of Humanities and Social Sciences in Zagreb and as such should serve only as a reminder to help in teaching. In addition to the description of the substitute cryptographic system, the transposition cryptographic systems and the complex cryptographic system, the Vigenere code and DES, an asymmetric system called RSA is also described. In addition to this paper, there is an accompanying console application written using the C# programming language. After each description of the cryptographic system, instructions for using the program and the problems that arose during its creation were written. Of all the symmetric cryptographic systems processed, only DES is directly embedded in C # in the System.Security.Cryptography namespace. This built-in implementation was used in the accompanying application, because for the needs of the course a description of the concealment and disclosure procedure is sufficient, and in this way the use of the DES system is easily enabled. Of course, the accompanying console application can also be used for leisure or as additional security on untrusted Internet systems, making it the simplest way to use old cryptographic systems in modern technology.

**Key words:** help in learning, cryptology, C# console application, cryptography systems, BA thesis