Krypteringsprojekt i MA1123

Kristian Lundkvist, 900831
Kristina Stendahl, 6604267903

February 22, 2013

Contents

1 Uppgift 1

1.1 Kryptogram 1

Vi gissade att kryptogrammet var ett förskjutningskrypto, även känt som Caesarchiffer, då kryptogrammet var för kort för att kunna lösas på något annat tänkbart sätt (för kort för frekvensanalys till exempel).

1.1.1 Lösning

Med hjälp dokumentet "Introduktion till Pari/GP" så skrev vi kod som forcerade kryptogrammet och fick fram både klartexten och nyckeln. Koden var som följer (lite ändrad för läsbarhet):

```
kryptotext = Eng(kryptogram1)
for(i=-24, 0, temp = Caesarcipher(kryptotext, i, 26);
        klartext = Eng(temp, -1);
        print(i);
        print(klartext)
)
```

Detta kommer visa att krypteringsnyckeln är 8 och att klartexten (med tillagda mellanslag) är: "all work and no play makes jack a dull boy the next cipher is a subsitution cipher". Detta gav oss även hjälp till nästa uppgift.

1.1.2 Diskussion

Introduktionen till Pari/GP var väldigt lärorik, eftersom man hade möjligheten att testa sig fram lite. Det här gav en inblick i hur kraftfullt och användbart programmet faktiskt är. Dock insåg vi ganska snabbt att det var bättre att läsa in filerna som argument till programmet, istället för att blint lita på \r filnamn, då detta visade sig att inte alltid fungera på ett tillfredställande sätt, då GP inte hittade kommandona.

Det här första projektet var en bra uppvärming inför det som komma skall, eftersom man nu har en liten föraning om hur programvaran fungerar, vilket kan vara en stor fördel när vi sedan ska lösa kommande projekt.

1.2 Kryptogram 2

Enligt ledtråden från förra uppgiften så visste vi att denna uppgiften var ett subsitutionskrypto. Utifrån detta påbörjade vi en frekvensanalys.

1.2.1 Lösning

Vi började med en frekvensanalys av kryptogrammet där vi listade hur många gånger varje bokstav förekom, vilken procentsats denna utgjorde av det hela och jämförde sedan med en baslinje för svenska språket[1]. Detta gjorde att vi

[1] wikipedia

kunde bestämma vad vissa av tecknen i kryptogrammet skulle vara men inte alla. Efter detta började vi leta efter bigram och trigram som vi kunde koppla till bokstäverna vi redan fått fram vilket gav oss ett par nycklar till. Det var nu som av en ren slump vi kände igen delar av Strindbergs Röda Rummet och detta visade sig stämma bra med resten av kryptogrammet. Den slutgiltiga tabellen för kryptogram, klartext, antal instanser och frekvensen i procent blev:

Kryptogram	Antal instanser	Frekvens	Klartext
A	130	5,73	O
B	43	1,89	B
C	44	1,94	Ä
D	33	1,45	C
E	59	2,6	P
F	53	2,33	Å
G	119	5,25	L
H	180	7,94	N
I	112	4,94	D
J	188	8,29	T
K	136	6	S
L	3	0,13	X
M	48	2,11	Ö
N	104	4,58	I
O	68	3	K
P	43	1,89	U
Q	51	2,25	V
R	197	8,69	E
S	24	1,05	J
T	68	3	G
U	184	8,12	R
V	0	0	
W	0	0	
X	11	0,48	Y
Y	0	0	
Z	47	2,07	H
Å	50	2,2	F
Ä	72	3,17	M
Ö	191	8,43	A

Slutgiltiga klartexten blir då (med insatta mellanrum):

"röda rummet första kapitlet stockholm i fågelperspektiv det var en afton i början av maj den lilla trädgården på mosebacke hade ännu icke blivit öppnad för allmänheten och rabatterna voro ej uppgrävda snödropparne hade arbetat sig upp genom fjolårets lövsamlingar och höllo just på att slutas inkorta verksamhet för att lämna plats åt de ömtåligare saffransblommorna vilka tagit skydd under ett ofruktsamt päronträd syrenerna väntade på sydligvind för att få gå i blommen lindarne bjödo ännu kärleksfilter i sina obrustna knoppar åt bofinkarne som

börjat bygga sina lavklädda bon mellan stam och gren ännu hade ingen mänsko-fot trampat sandgångarne sedan sista vinterns snö gått bort och därför levdes ett obesvärat liv där inne av både djur och bl ommor gråsparvarne höllo på att samla upp skräp som de sedan gömde under takpannorna på navigationsskolans hus de drogo som spillror av rakethylsor från sista höstfyrverkeriet de plockade halmen från unga trädsomåret för utsluppit ur skolan på rosendal och allting sågo de de hittade baregelappar i bersåer och kunde mellan stickorna på en bänk fotdraga fram hårtappar efter hundar som icke slagits där sedan josefinadagen i fjor där var ett liv och ett kiv men solen stod över liljeholmen och sköt hela kvastar av strålar mot öster de gingo genom rökarne från bergsund de ilade framöver riddarfjärden klättrade upp till korset på riddarholmskyrkan kastade sig över till tyskans brantatak lekte med vimplarne på skeppsbrobåtarne illumin-erade i fönstren på stora sjötullen eklärerade lidingöskogarne och tonade bort i ett rosenfärgat moln långt långt ut i fjärran där havet ligger och därifrån kom vinden och hon gjorde sammafärd tillbaka genom vaxholm förbi fästningen förbi sjötulln ut med siklaön gick in bakom hästholmen och tittade på sommarnöjena ut igen fortsatte och kom in i danviken blev skrämd och rusade av utmed södra stranden kände lukten av kol tjära och tran törnade mot stadsgården for upp för mosebacke in i trädgården och slog emot en vägg i det samma öppnades väggen av en piga som just rivit bort klistringen på innan fönstren ett förfärligt os av stekflott ölskvättar granris och sågspån störtade ut och fördes långt bort av vin-den som nu medan köksan drog in den friska luften genom näsan passade på att gripa fönstervadden som var beströdd med paljetter och berberis bär och törn-rosblad och började en ringdans ut efter gångarne i vilken snart grå sparvarne och bofinkarne deltogo då de sålunda sågo sina bosättningsbekymmer till stor del undanröjda nästa krypto är ett rsa publik nyckeln lika med tre ett två sex noll ett fem tre tre ett ett noll åtta åtta noll noll två nio sex åtta nio sju och e likamed fem noll sju noll åtta noll nio"

Intressant här är slutet som berättar att nästa kryptogram är ett RSA krypto med publik nyckel 3126015331108800296897 och där e är 5070809.

1.2.2 Diskussion

Antingen missade vi något när vi skulle utföra frekvensanalysen med hjälp av Pari/GP eller så blev något annat fel, vi fick i alla fall utföra frekvensanalysen genom att kopiera in texten i emacs och använda det för att räkna tecknen. Annars var det inget anmärkningsvärt med uppgiften.

1.3 Kryptogram 3

Enligt ledtråden i förra uppgiften så vet vi att kryptogrammet är krypterat med RSA, att den publika nyckeln (n) är 3126015331108800296897 och att exponenten (e) är 5070809.

1.3.1 Lösning

Lösningen görs helt i Pari/GP. Vi börjar med att definiera det vi vet:

```
n=312601533110880029689 7
e=5070809
```

Vi vet sedan att $n = pq$ där p och q är primtal och vi faktoriserar därför n och kollar om faktorerna är primtal:

```
factorint(n)
>20709784709
>150943883533
20709784709*150943883533
>312601533110880029689 7
isprime(20709784709)
>1
isprime(150943883533)
>1
```

Vi har alltså fått ut både p och q och vet att båda är primtal. Vi räknar därefter ut $m = (p-1)(q-1)$:

```
p=20709784709
q=150943883533
m=(p-1)*(q-1)
>312601533093714662865 6
```

Nu kan vi se om allt stämmer, har vi fått rätt resultat än så länge så ska $GCD(e, m) = 1$, är det inte det får vi går tillbaka och leta efter fel. Vi testar:

```
gcd(e,m)
>1
```

Våra uträkningar stämmer alltså. Vi ska nu hitta d som vi får genom $1 = de \bmod m$. Man kan även använda funktionen bezout:

```
d=bezout(e, eulerphi(n))[1]
>-466844297992307580 83
(d*e)%m
>1
```

Vi har alltså hittat allt vi behöver för att kunna dekryptera kryptogrammet. Lösningen blir då:

```
RSA(kryptogram3, n, d)
```

Detta ger oss en vektor med siffror som man sedan kunde koda om till ASCII. Resultatet blir: *Here she comes walking down the street She's got something you would love to meet It's her heart and her heart is black Think of ice cream sliding into a crack The heat sticks to summer's heavy sweat Hang around it'll get hotter yet You got the shakes and it's gonne get worse Don't you know it's all a part of*

the curse She's got the hit that takes you into space Suck mud and make a deal for that taste You got nothing but you're riding on a star You couldn't guess that she could take you that far Some things are so hard to say Even though you'd say them every day Don't let your life be the butt of a joke Get your lips round a cool black Pepsi Coke Here she comes ÖÖÖÖÖ.

1.3.2 Diskussion

Det var rätt lätt att lösa kryptot medans det svåraste var att räkna ut hur man skulle konvertera resultatet till något läsbart. Kryptot visade också hur viktigt det var att ha långa nycklar då vårt första steg hade tagit timmar eller dagar att göra om vi haft en ordentligt lång nyckel.

2 Uppgift 2

2.1 Steg 1, hitta diagonalerna

Det första vi måste göra är att hitta vilka bokstäver som kodar till dubbletter, till exempel AA, DD, FF etc. Detta gör man genom att skicka in ensamma bokstäver i algoritmen och kolla på resultatet. Då dubbletterna inte påverkas av transpositionerning så kan man lätt räkna ut vilka bokstäver som ligger på diagonalen i Polybius boxen.

2.2 Steg 2, ta reda på hur transpositioneringen sker om man skickar in två bokstäver

Vi genom steg 1 så har vi kommit fram till att 'h' kodar till 'AA' och 'x' kodar till 'FF', vi skickar sedan in 'hx' i algoritmen. Om det ger resultatet 'AAFF' så vet vi att ingen transpositionering sker, om det kommer ut 'AFAF' så har kolumn 2 och 3 bytat plats eller att nyckeln är två tecken lång utan att transpositionera. Oavsett så kan vi ta reda på hur transpositioneringen funkar för de första kolumnerna och vi kan därför börja avkoda resten av tecknen i Polybius boxen. Vi vet nu hur de olika kolumnerna flyttar sig så vi kan skicka in ett diagonaltecken och ett okänt och se vad vi får för resultat, därefter bryter vi ut det okända och vet då hur det tecknet kodar.

2.3 Steg 3, forcera kodordet

Vi vet nu hur de olika tecknen kodar och kan därför forcera kodordet. Detta gör vi i två steg.

Först matar vi in en sträng i algoritmen. Vi antar sedan att kodordet har 1 tecken som behåller sin ordning efter transponeringen, kodar av bokstäverna och kollar om vi kan läsa meddelandet, kan vi inte det så testar vi en annan transponering och vi fortsätter så tills vi testat alla transponeringar för det den längden. När vi kört alla transponeringar så ökar vi längden på kodordet med

ett och börjar om. Vi fortsätter så tills vi vet vilken transponering och längd kodordet har. Efter detta så har vi både Polybius boxen och kodordet.

2.4 Diskussion

Det finns antagligen bättre varianter som man kan använda och sätt att opimera denna metoden men detta är det bästa vi kunde komma på.

3 Uppgift 3

3.1 Antalet kvadratiska matriser

m^{k^2}

3.2 Mängden giltiga nycklar

Enligt svaret i 3.1 så vet vi hur många kvadratiska matriser som finns, vi måste nu räkna ut hur många giltiga nycklar som existserar. En nyckel är giltig om determinanten för matrisen är relativt prima med m. Vi skrev ett program i C++ med Eigen[2] som gjorde detta. Programmet och Makefile finns här.

Resultatet blir:

```
Modulu: 2
Tested keys: 16
Functional keys: 6
Percentage: 37.5

Modulu: 3
Tested keys: 81
Functional keys: 48
Percentage: 59.2593

Modulu: 4
Tested keys: 256
Functional keys: 96
Percentage: 37.5

Modulu: 5
Tested keys: 625
Functional keys: 480
Percentage: 76.8

Modulu: 6
Tested keys: 1296
Functional keys: 288
```

[2] http://eigen.tuxfamily.org/index.php?title=Main_Page

```
Percentage: 22.2222

Modulu: 7
Tested keys: 2401
Functional keys: 2016
Percentage: 83.965

Modulu: 8
Tested keys: 4096
Functional keys: 1536
Percentage: 37.5

Modulu: 9
Tested keys: 6561
Functional keys: 3888
Percentage: 59.2593

Modulu: 10
Tested keys: 10000
Functional keys: 2880
Percentage: 28.8

Modulu: 11
Tested keys: 14641
Functional keys: 13200
Percentage: 90.1578

Modulu: 12
Tested keys: 20736
Functional keys: 4608
Percentage: 22.2222

Modulu: 13
Tested keys: 28561
Functional keys: 26208
Percentage: 91.7615

Modulu: 14
Tested keys: 38416
Functional keys: 12096
Percentage: 31.4869

Modulu: 15
Tested keys: 50625
Functional keys: 23040
Percentage: 45.5111
```

```
Modulu: 16
Tested keys: 65536
Functional keys: 24576
Percentage: 37.5

Modulu: 17
Tested keys: 83521
Functional keys: 78336
Percentage: 93.792

Modulu: 18
Tested keys: 104976
Functional keys: 23328
Percentage: 22.2222

Modulu: 19
Tested keys: 130321
Functional keys: 123120
Percentage: 94.4744

Modulu: 20
Tested keys: 160000
Functional keys: 46080
Percentage: 28.8

Modulu: 21
Tested keys: 194481
Functional keys: 96768
Percentage: 49.757

Modulu: 22
Tested keys: 234256
Functional keys: 79200
Percentage: 33.8092

Modulu: 23
Tested keys: 279841
Functional keys: 267168
Percentage: 95.4714

Modulu: 24
Tested keys: 331776
Functional keys: 73728
Percentage: 22.2222
```

```
Modulu: 25
Tested keys: 390625
Functional keys: 300000
Percentage: 76.8

Modulu: 26
Tested keys: 456976
Functional keys: 157248
Percentage: 34.4106

Modulu: 27
Tested keys: 531441
Functional keys: 314928
Percentage: 59.2593

Modulu: 28
Tested keys: 614656
Functional keys: 193536
Percentage: 31.4869

Modulu: 29
Tested keys: 707281
Functional keys: 682080
Percentage: 96.4369

Modulu: 30
Tested keys: 810000
Functional keys: 138240
Percentage: 17.0667

Tested matrixes: 5273998
Functional keys: 2688726
Percentage: 50.9808
```

3.3 Diskussion

Det verkar som att desto fler tecken det finns i alfabetet desto fler inverterbara matriser vilket gör att det är svårare att hitta rätt nyckel. Det verkar även som att matriser med jämna modulu ger färre giltiga nycklar.