Kratki opis

OTP je algoritam za kriptovanje podataka koji funkcioniše na sledeći način: izvorni tekst koji želimo da šifrujemo se xor-uje sa ključem (odn. one-time pad-om)i tako se dobija kodiran tekst. Ključ predstavlja tablicu koja sadrži preslikavanja za sva slova azbuke koju koristimo u njihove kodove. Dekriptovanje se vrši tako što se xor-uju kriptovan tekst i ključ, pa se tako dobija izvorni tekst.

Zadatak 1.

Data je azbuka za „one-time-pad" kriptovanje: a = 000, t = 001, n = 010, o = 011, s = 100, v = 101, d = 110 i e = 111. Istim ključem (pad-om) kriptovane su četitri naše reči (2 sa 5 i 2 sa 6 slova) i dobijeni rezultati su ananv, stovv, vodeea i ttovvs. Ako znamo da je jedna od te četiri reči „tavan", odrediti preostale tri, kao i korišćeni ključ (pad).

Rešenje:

```
azbuka          kodirane reči:
a 000           ananv stovv vodeea ttovvs
t 001
n 010           poznata reč:
o 011           tavan
s 100
v 101
d 110
e 111
```

Kako je reč „tavan" dužine 5 slova, to znači da će jedna od reči „ananv" ili „stovv" biti baš ta reč jer su obe dužine 5 slova.
Da bismo dobili ključ, moramo primeniti xor operaciju nad rečima „tavan" i „ananv" ili „tavan" i „stovv" (u ovom slučaju dobićemo dva ključa od kojih će jedan moći da dešifruje ostale reči, ali da bismo otkrili koji je to ključ moramo isprobavati kombinacije).

prvo dobijamo ključ na osnovu reči „tavan" i „ananv":

```
 a   n   a   n   v
000 010 000 010 101
001 000 101 000 010
 t   a   v   a   n
------------------ xor
001 010 101 010 111 ključ1
```

onda ključ primenjujemo na ostale reči:

```
 s   t   o   v   v
100 001 011 101 101
001 010 101 010 111 ključ1
------------------- xor
101 011 110 111 010
 v   o   d   e   n
```

```
 v   o   d   e   e   a
101 011 110 111 111 000
001 010 101 010 111 001  ključ1
------------------- xor
100 001 011 101 000 001
 s   t   o   v   a   t
```

(kako ova reč nema smisla, vratimo se na početak da probamo sada sa rečima „tavan“ i „stovv“ i ključem koji se dobija iz njih)

```
 s   t   o   v   v
100 001 011 101 101
001 000 101 000 010
 t   a   v   a   n
------------------- xor
101 001 110 101 111 ključ2
```

```
 a   n   a   n   v
000 010 000 010 101
101 001 110 101 111 ključ2
------------------- xor
101 011 110 111 010
 v   o   d   e   n
```

```
 v   o   d   e   e   a
101 011 110 111 111 000
101 001 110 101 111 101 ključ2
---------------------- xor
000 010 000 010 000 101
 a   n   a   n   a   v
```

```
 t   t   o   v   v   s
001 001 011 101 101 100
101 001 110 101 111 101 ključ2
---------------------- xor
100 000 101 000 010 001
 s   a   v   a   n   t
```

Po nekom generalnom pravilu, ako je ključ kraći od reči, samo se prekopira od početka do dužine reči, međutim, kako ovde može da se primeti da reči „ananav" i „savant" podsećaju na naše reči „ananas" i „savana" i to u slučaju da produženi deo ključa nije 101 već 100, onda umesto tog generalnog pravila može da se praktično namesti produženi deo ključa kako bi se dobile smislene reči.

```
 v   o   d   e   e   a
101 011 110 111 111 000
101 001 110 101 111 100 ključ2
----------------------- xor
000 010 000 010 000 100
 a   n   a   n   a   s

 t   t   o   v   v   s
001 001 011 101 101 100
101 001 110 101 111 100 ključ2
----------------------- xor
100 000 101 000 010 000
 s   a   v   a   n   a
```

ananv - voden
stovv - tavan
vodeea - ananas
ttovvs - savana

Zadatak 2.

Kriptovanjem četiri reči po one-time-pad algoritmu dobijene su sledeće vrednosti: stnatc, nictsa, rnctgg, atiari. Ako se zna da je jedna od 4 polazne reči string i da je korišćena sledeća azbuka [a, n, i, c, g, r, s, t] (kodirana sa tri bita), odrediti ostale tri reči.

Rešenje:

```
azbuka          kodirane reči:
a 000           stnatc nictsa rnctgg atiari
n 001
i 010           poznata reč:
c 011           string
g 100
r 101
s 110
t 111
```

Reč „string“ je dužine 6 slova, kao i sve kodirane reči, tako da je svejedno koju ćemo reč prvo da xor-ujemo sa „string“ i da pokušamo da tim ključem dešifrujemo ostale. Možemo npr. da krenemo redom kako su reči zadate.

prvo dobijamo ključ na osnovu reči „string“ i „stnatc“:

```
 s   t   n   a   t   c
110 111 001 000 111 011
110 111 101 010 001 100
 s   t   r   i   n   g
---------------------- xor
000 000 100 010 110 111 ključ1
```

```
 n   i   c   t   s   a
001 010 011 111 110 000
000 000 100 010 110 111 ključ1
---------------------- xor
001 010 111 101 000 111
 n   i   t   r   a   t
```

```
 r   n   c   t   g   g
101 001 011 111 100 100
000 000 100 010 110 111 ključ1
---------------------- xor
101 001 111 101 010 011
 r   n   t   r   i   c
```

(kako ova reč nema smisla, vratimo se na početak da probamo sada sa rečima „string“ i „nictsa“ i ključem koji se dobija iz njih)

```
 n   i   c   t   s   a
001 010 011 111 110 000
110 111 101 010 001 100
 s   t   r   i   n   g
---------------------- xor
111 101 110 101 111 100 ključ2
```

```
 s   t   n   a   t   c
110 111 001 000 111 011
111 101 110 101 111 100 ključ2
---------------------- xor
001 010 111 101 000 111
 n   i   t   r   a   t
```

```
 r   n   c   t   g   g
101 001 011 111 100 100
111 101 110 101 111 100 ključ2
---------------------- xor
010 100 101 010 011 000
 i   g   r   i   c   a

 a   t   i   a   r   i
000 111 010 000 101 010
111 101 110 101 111 100 ključ2
---------------------- xor
111 010 100 101 010 110
 t   i   g   r   i   s

stnatc - nitrat
nictsa - string
rnctgg - igrica
atiari – tigris (reka)
```

Zadatak 3.

Kriptovanjem četiri reči po one-time-pad algoritmu dobijene su sledeće vrednosti: acetse, aeikee, ciiee, eeesk. Ako se zna da je jedna od 4 polazne reči krasta i da je korišćena sledeća azbuka [e, c, i, k, a, r, s, t] (kodirana sa tri bita), odrediti ostale tri reči.

Rešenje:

```
azbuka    kodirane reči:
e 000          acetse aeikee ciiee eeesk
c 001
i 010          poznata reč:
k 011          krasta
a 100
r 101
s 110
t 111
```

Kako je reč „krasta" dužine 6 slova, to znači da će jedna od reči „acetse" ili „aeikee" biti baš ta reč jer su obe dužine 6 slova.
Da bismo dobili ključ, moramo primeniti xor operaciju nad rečima „krasta" i „acetse" ili „krasta" i „aeikee" (u ovom slučaju dobićemo dva ključa od kojih će jedan moći da dešifruje ostale reči, ali da bismo otkrili koji je to ključ moramo isprobavati kombinacije).

prvo dobijamo ključ na osnovu reči „krasta“ i „acetse“:

```
 a   c   e   t   s   e
100 001 000 111 110 000
011 101 100 110 111 100
 k   r   a   s   t   a
---------------------- xor
111 100 100 001 001 100 ključ1

 a   e   i   k   e   e
100 000 010 011 000 000
111 100 100 001 001 100 ključ1
---------------------- xor
011 100 110 010 001 100
 k   a   s   i   c   a

 c   i   i   e   e
001 010 010 000 000
111 100 100 001 001 100 ključ1
------------------- xor
110 110 110 001 001
 s   s   s   c   c
```

(kako ova reč nema smisla, vratimo se na početak da probamo sada sa rečima „krasta“ i „aeikee“ i ključem koji se dobija iz njih)

```
 a   e   i   k   e   e
100 000 010 011 000 000
011 101 100 110 111 100
 k   r   a   s   t   a
---------------------- xor
111 101 110 101 111 100 ključ2

 a   c   e   t   s   e
100 001 000 111 110 000
111 101 110 101 111 100 ključ2
---------------------- xor
011 100 110 010 001 100
 k   a   s   i   c   a

 c   i   i   e   e
001 010 010 000 000
111 101 110 101 111 100 ključ2
------------------- xor
110 111 100 101 111
 s   t   a   r   t
```

```
e   e   e   s   k
000 000 000 110 011
111 101 110 101 111 100 ključ2
------------------- xor
111 101 110 011 100
 t   r   s   k   a
```

```
acetse - kasica
aeikee - krasta
ciiee - start
eeesk – trska
```

Zadatak 4.

Kriptovanjem četiri reči po one-time-pad algoritmu dobijene su sledeće vrednosti: dalce, ecsci, pcpci, elpde. Ako se zna da je jedna od 4 polazne reči sesil i da je korišćena sledeća azbuka [p, c, i, s, a, l, e, d] (kodirana sa tri bita), odrediti ostale tri reči.

Rešenje:

```
azbuka          kodirane reči:
p 000           dalce ecsci pcpci elpde
c 001
i 010           poznata reč:
s 011           sesil
a 100
l 101
e 110
d 111
```

Reč „sesil" je dužine 5 slova, kao i sve kodirane reči, tako da je svejedno koju ćemo reč prvo da xor-ujemo sa „sesil" i da pokušamo da tim ključem dešifrujemo ostale. Možemo npr. da krenemo redom kako su reči zadate.

prvo dobijamo ključ na osnovu reči „sesil" i „dalce":

```
 d   a   l   c   e
111 100 101 001 110
011 110 011 010 101
 s   e   s   i   l
------------------- xor
100 010 110 011 011 ključ1
```

```
 e   c   s   c   i
110 001 011 001 010
100 010 110 011 011 ključ1
------------------- xor
010 011 101 010 001
 i   s   l   i   c
```

Kako ova reč nema smisla, vratimo se na početak da probamo sada sa rečima „sesil" i nekom drugom i ključem koji se dobija iz njih. Sledeća po redu bi bila „ecsci" međutim kako smo ključem koji se dobija iz reči „dalce" neuspešno pokušali da dekodiramo reč „ecsci", važi i da ćemo ključem koji se dobija iz reči „ecsci" neuspešno dekodirati reč „dalce" tako da nema potrebe nalaziti njen ključ i možemo preći na sledeću reč, što je u ovom slučaju „pcpci":

nalazimo ključ na osnovu reči „pcpci" i „sesil"

```
 p   c   p   c   i
000 001 000 001 010
011 110 011 010 101
 s   e   s   i   l
------------------- xor
011 111 011 011 111 ključ2
```

```
 d   a   l   c   e
111 100 101 001 110
011 111 011 011 111 ključ2
------------------- xor
100 011 110 010 001
 a   s   e   i   c
```

(kako ova reč nema smisla, vratimo se na početak da probamo sada sa rečima „sesil" i „elpde" i ključem koji se dobija iz njih)

nalazimo ključ na osnovu reči „elpde" i „sesil"

```
 e   l   p   d   e
110 101 000 111 110
011 110 011 010 101
 s   e   s   i   l
------------------- xor
101 011 011 101 011 ključ3
```

```
 d   a   l   c   e
111 100 101 001 110
101 011 011 101 011 ključ3
------------------- xor
010 111 110 100 101
 i   d   e   a   l

 e   c   s   c   i
110 001 011 001 010
101 011 011 101 011 ključ3
------------------- xor
011 010 000 100 001
 s   i   p   a   c

 p   c   p   c   i
000 001 000 001 010
101 011 011 101 011 ključ3
------------------- xor
101 010 011 100 001
 l   i   s   a   c

dalce - ideal
ecsci - sipac
pcpci - lisac
elpde - sesil
```

Zadatak 5.

Kriptovanjem četiri reči po one-time-pad algoritmu dobijene su sledeće vrednosti: nnmmtm, lktmam, mssalt, msmmmm. Ako se zna da je jedna od 4 polazne reči ananas i da je korišćena sledeća azbuka [n, a, k, s, t, l, m, i] (kodirana sa tri bita), odrediti ostale tri reči.

Rešenje:

```
azbuka          kodirane reči:
n 000           nnmmtm lktmam mssalt msmmmm
a 001
k 010           poznata reč:
s 011           ananas
t 100
l 101
m 110
i 111
```

Reč „ananas" je dužine 6 slova, kao i sve kodirane reči, tako da je svejedno koju ćemo reč prvo da xor-ujemo sa „ananas" i da pokušamo da tim ključem dešifrujemo ostale. Možemo npr. da krenemo redom kako su reči zadate.

prvo dobijamo ključ na osnovu reči „ananas" i „nnmmtm":

```
 n   n   m   m   t   m
000 000 110 110 100 110
001 000 001 000 001 011
 a   n   a   n   a   s
---------------------- xor
001 000 111 110 101 101 ključ1

 l   k   t   m   a   m
101 010 100 110 001 110
001 000 111 110 101 101 ključ1
---------------------- xor
100 010 011 000 100 011
 t   k   s   n   t   s
```

Kako ova reč nema smisla, vratimo se na početak da probamo sada sa rečima „ananas" i „mssalt" (ne „ltkmam iako je sledeća po redu jer će njen generisani ključ primenjen na „nnmmtm" da daje isto besmislenu reč)i ključem koji se dobija iz njih.

dobijamo ključ na osnovu reči „ananas" i „mssalt":

```
 m   s   s   a   l   t
110 011 011 001 101 100
001 000 001 000 001 011
 a   n   a   n   a   s
---------------------- xor
111 011 010 001 100 111 ključ2

 n   n   m   m   t   m
000 000 110 110 100 110
111 011 010 001 100 111 ključ2
---------------------- xor
111 011 100 111 000 001
 i   s   t   i   n   a
```

```
 l   k   t   m   a   m
101 010 100 110 001 110
111 011 010 001 100 111 ključ2
----------------------- xor
010 001 110 111 101 001
 k   a   m   i   l   a

 m   s   m   m   m   m
110 011 110 110 110 110
111 011 010 001 100 111 ključ2
----------------------- xor
001 000 100 111 010 001
 a   n   t   i   k   a

nnmmtm - istina
lktmam - kamila
mssalt - ananas
msmmmm – antika
```