rsa_zgled

December 16, 2022

0.1 RSA kriptosistem

RSA kriptosistem bomo predstavili v Pythonovskem zvezku. Seveda potrebujemo nekaj dodatnih orodij.

[1]:
```
import math
from sympy import prime
from sympy.ntheory import ecm
from sympy.ntheory import totient
from random import randint
```

prime? praštevilo

ecm? praštevilski razcep

totient? Eulerjeva funkcija, alternativno ime

randint? slučajno število

0.1.1 Dve veliki praštevili

Radi bi poiskati dve primerljivo veliki praštevili. Primerljivo veliki? Skoraj isto število binarnih/decimalnih mest.

[2]:
```
p = prime(1000000000)
q = prime(800000000)
print(p)
print(q)
```

```
22801763489
18054236957
```

Produkt praštevil p in q označimo z n.

[3]:
```
n = p*q
print(n)
```

```
411668441067877062973
```

0.1.2 Eulerjeva funkcija

S *phi* označimo Eulerjevo funkcijo števila $n = p\ q$. Ker poznamo praštevilski razcep števila n je naloga otročje lahka.

[4]:
```
phi = (p-1)*(q-1)
print(phi)
```

```
411668441027021062528
```

[5]:
```
ecm(n)
```

[5]:
```
{18054236957, 22801763489}
```

[6]:
```
totient(n)
```

[6]:
```
411668441027021062528
```

Naše število je dovolj majhno, da zna njegov razcep oziroma Eulerjevo funkcijo izračunati tudi Python. To pomeni, da naši ključi v tem zgledu nikakor niso varni!

0.1.3 Konstrukcija javnega in privatnega ključa

Izberimo si poljubno število d, manjše od *phi*, ki je **tuje** *phi*. Poskusimo s slučajnim številom.

[10]:
```
d = randint(1,n)
print(d)
print(math.gcd(d, phi))
```

```
88859890694783175795
1
```

Par *(n,d)* je Borutov privatni ključ.

[11]:
```
(n,d)
```

[11]:
```
(411668441067877062973, 88859890694783175795)
```

Določimo naravno število e, ki reši diofansko enačbo: $e\ d = 1 + k\ phi$. Ker sta d in *phi* tuji števili, je ta LDE rešljiva.

Z drugimi besedami, produkt $e\ d$ je po modulu *phi* kongruenten *1*.

[12]:
```
e = pow(d, -1, phi)
print(e)
```

```
275253687879854124347
```

Tole zgoraj je trik. Iskali smo *inverz* k d za množenje po modulu *phi*.

Preverimo.

[13]:
```
e * d % phi
```

[13]:
```
1
```

Par (n,e) je Borutov javni ključ.

[14]:
```
(n,e)
```

[14]:
```
(411668441067877062973, 275253687879854124347)
```

Zelo pomembno se je znebiti števila *phi*. Ravno tako moramo paziti, da nihče ne more do privatnega ključa.

0.1.4 Prenos kriptiranih podatkov

Ančka bi rada Borutu poslala sporočilo.

Sporočilo, ki ga Ančka počilja Borutu mora biti kratko. Manjše od n. To ne predstavlja nobenega problema. Če je sporočilo daljše, ga Ančka lahko razseka na ustrezno kratke dele.

[15]:
```
sporocilo = 12345678987654321
```

Sporočilo Ančka zaklene z Borutovim javnim ključem. Potencira ga na potenco e in določi ostanek pri deljenju z n.

[16]:
```
zaklenjenosporocilo = pow(sporocilo,e,n)
print(zaklenjenosporocilo)
```

```
362266224501015323438
```

Zaklenjeno sporočilo lahko pošlje Borutu po nezavarovanem kanalu. Odklene ga lahko samo Borut.

Borut sporočilo odklene s svojim privatnim ključem. Potencira ga na potenco d in določi ostanek pri deljenju z n.

[17]:
```
odklenjenosporocilo = pow(zaklenjenosporocilo,d,n)
print(odklenjenosporocilo)
```

```
12345678987654321
```

[18]:
```
sporocilo == odklenjenosporocilo
```

[18]:
```
True
```

[ ]:
```

```