

l9-lattice-attacks-sage

April 23, 2025

0.1 RSA

```
[1]: while True:
    p = random_prime(2^512)
    if gcd(p-1, 3) == 1: break
print(f'p = {p}')
while True:
    q = random_prime(2^512)
    if gcd(q-1, 3) == 1: break
print(f'q = {q}')
n = p*q
e = 3
d = inverse_mod(e, (p-1)*(q-1))
print(d*e % ((p-1)*(q-1)) == 1)
print(f'n = {n} ({n.bit_length()} bitov)')
```

p = 3895727810119476026794112645136053606307045830615600171815372992821334306843
84399330996349536782219490252285663412356794143767493822031511206872848327233
q = 2859131109373079989761023654410548366790196664527350623327040875767528760457
72019916298702905755379134727319311130976411614949910054962640258774615472223
True
n = 1113839657556245700649787422116680616657619090307463995532352807222505794748
27898458323757622498778222321102582567108031086159063768117911148564303764952451
21552420308950153790207091463034483721831514262338167685420549117433685292180958
4338725283905912794235624945373644421301852962202571596230645925948959 (1014
bitov)

0.1.1 Krátky OT, najskôr bez výplne

```
[2]: def str2int(s):
    return int(bytes(s, encoding='utf-8').hex(), 16)

def int2str(i):
    return bytes.fromhex(Integer(i).hex())

heslo = 'm:LJF39#@d2=123n'
m = str2int(heslo)
c = m^e % n
```

```
print(f'm = {m} ({int2str(m)}, {m.bit_length()} bitov)\nc = {c}')
```

```
m = 145188552106090629460942896073756783470 (b'm:LJF39#0d2=123n', 127 bitov)
c = 3060533395870120830879062017662616847001105985534645637381685422852895825081
531870613621757391475393277122687923000
```

```
[3]: m2 = c^(1/3)
print(int2str(m2))
m2 == m
```

```
b'm:LJF39#0d2=123n'
```

```
[3]: True
```

0.1.2 Fixná výplň (padding)

```
[4]: padding = 'TotoJeFixnáVýplňNaKonciJeHeslo:'
m = str2int(padding+heslo)
print(f'm = {m} \n ({int2str(m)}, {m.bit_length()} bitov)')
print('\nBude stačiť "odmocniť" ako predtým?')
c = m^e % n
m2 = c^(1/3)
print(f'm2 = {m2}')
```

```
print(int(m2) == m)

m = 8516922848800316893674633894464662854403489832932260112118010361842402925051
27627687041687043683311126943331000771556206
(b'TotoJeFixn\xc3\xa1V\xc3\xbdpl\xc5\x88NaKonciJeHeslo:m:LJF39#0d2=123n',
399 bitov)
```

```
Bude stačiť "odmocniť" ako predtým?
```

```
m2 = 240610498422984613840637212373860100808212841444733801223938859103886838839
06351413545367694979589193164306724310296155735946586485923301682584987412604240
90520859847687477883738450846356719670138600314582436490287984171591183871261365
0474668648030974443227790294780599750865860174310590825300442162676118^(1/3)
False
```

```
[6]: m = str2int(padding+heslo)
print(f'm = {m}\n({int2str(m)}, {m.bit_length()} bitov)')
c = m^e % n
m2 = c^(1/3)
print(f'm2 = {m2}')
```

```
print(f'celočíselný výsledok? {m2.is_integer()}')

m = 8516922848800316893674633894464662854403489832932260112118010361842402925051
27627687041687043683311126943331000771556206
(b'TotoJeFixn\xc3\xa1V\xc3\xbdpl\xc5\x88NaKonciJeHeslo:m:LJF39#0d2=123n', 399
bitov)
m2 = 240610498422984613840637212373860100808212841444733801223938859103886838839
```

```
06351413545367694979589193164306724310296155735946586485923301682584987412604240
90520859847687477883738450846356719670138600314582436490287984171591183871261365
0474668648030974443227790294780599750865860174310590825300442162676118^(1/3)
celočíselný výsledek? False
```

```
[7]: def print_matrix_bits(A):
    for r in A:
        print([c.bit_length() for c in r], end=' ')
        print(f'... 12 = {log(sqrt(sum([c^2 for c in r])),2).n(40):.2f} (v bitoch)')

a = str2int(padding) * (2^(8*len(heslo)))
R = 2^(8*len(heslo))
M = matrix([[R^3, 3*R^2*a, 3*R*a^2, a^3-c], [0,n*R^2,0,0], [0,0,n*R,0], [0,0,0,n]])
B = M.LLL()
print('Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):')
print_matrix_bits(M)
print('\nPočty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):')
print_matrix_bits(B)
```

Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):

```
[385, 656, 927, 1196] ... 12 = 1195.20 (v bitoch)
[0, 1270, 0, 0] ... 12 = 1269.34 (v bitoch)
[0, 0, 1142, 0] ... 12 = 1141.34 (v bitoch)
[0, 0, 0, 1014] ... 12 = 1013.34 (v bitoch)
```

Počty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):

```
[626, 898, 895, 896] ... 12 = 897.95 (v bitoch)
[625, 897, 900, 899] ... 12 = 899.53 (v bitoch)
[998, 991, 993, 994] ... 12 = 997.36 (v bitoch)
[1010, 1011, 1012, 1014] ... 12 = 1013.20 (v bitoch)
```

```
[8]: # Vyrobitme polynom z prveho riadku B a potom vypočítame celočíselný koreň
Q = B[0][0]*x^3/R^3+B[0][1]*x^2/R^2+B[0][2]*x/R+B[0][3]
print(Q)
roots = Q.roots(ring=ZZ)
print(roots)
print(int2str(roots[0][0]))
```

```
6826408617109790424484753158544341294441990543595641917216219463360009339*x^3 +
17441998657892924379055462063985750105521771556476371031950301520124365273103759
17880391790856000037828652968783248170096941232576648715905319756163085522407164
0530829870565853496187031767744512*x^2 - 474392116390733257514424165201731259172
86298546198304081688628449933684541014943051924321271018620026957484016593719397
09105515656674002064921926673323850847178239271480178698686678323459909082507661
87403535719266762684570101851661*x - 2987960677876178080326217311808599241277701
58151634233300471735433415751808869970921928846211090583182627996281623906126979
19856933301489383005772623385856297489235155992121701082321636600551261972785671
2535705947108434383582567294411235718286194483313196493383808994130
```

```
[(145188552106090629460942896073756783470, 1)]
b'm:LJF39#@d2=123n'
```

0.2 Faktorizácia pri čiastočnej znalosti deliteľa

```
[9]: # nech  $p > q$ 
      if  $p < q$ :
           $p, q = q, p$ 

      # rozdelíme  $p$  na  $a + m$ , kde  $|m| = 160$  bitov (predpokladáme  $|n| > 6 * |m|$ )
       $l = 160$ 
       $m = p \% 2^l$ 
       $a = p - m$ 
      print(f' $p = \{p:x\} \setminus nm = \{m:x\} \setminus na = \{a:x\}'$ )
```

```
p = 770312cac56b6bd2c0cc2cb8e1d3795e26f6c64f23924d81bff1d53a7cd09919c69abe57e3c8  
f3daf148a1f17d1f46562c225a9ad10ffc522be68433eb8e241  
m = 17d1f46562c225a9ad10ffc522be68433eb8e241  
a = 770312cac56b6bd2c0cc2cb8e1d3795e26f6c64f23924d81bff1d53a7cd09919c69abe57e3c8  
f3daf148a1f00000000000000000000000000000000000000000000000000000
```

```
[10]: R = 2^1
M = matrix([[R^2, 2*R*a, a^2],[0, R, a],[0, 0, n]])
B = M.LLL()
print('Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):')
print_matrix_bits(M)
print('\nPočty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):')
print_matrix_bits(B)
```

Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):
 [321, 668, 1014] ... 12 = 1013.79 (v bitoch)
 [0, 161, 507] ... 12 = 506.89 (v bitoch)
 [0, 0, 1014] ... 12 = 1013.34 (v bitoch)

Počty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):
 [494, 490, 484] ... 12 = 493.16 (v bitoch)
 [492, 494, 490] ... 12 = 493.33 (v bitoch)
 [501, 504, 507] ... 12 = 506.89 (v bitoch)

```
[11]: Q = B[0][0]*x^2/R^2+B[0][1]*x/R+B[0][2]
      roots = Q.roots(ring=ZZ)
      print(roots)
      print(p == roots[0][0] + a)
      print(f'{{roots[0][0]:x}}')
```

```
[(135988933373000030878663629819809574550062555713, 1)]
True
17d1f46562c225a9ad10ffc522be68433eb8e241
```

0.3 Krátky súkromný exponent

```
[12]: # vytvoríme krátky súkromný exponent
d = random_prime(2^(200))
e = inverse_mod(d, (p-1)*(q-1))
print(f'd = {d}')
d*e % ((p-1)*(q-1)) == 1
```

d = 431914847505162772450704837326985187908449771866261567886243

[12]: True

```
[13]: X = 2^(3*n.bit_length()//4)
Y = 2^(n.bit_length()//4)
M = matrix([[X, e*Y, -1],[0, Y*(n+1), 0], [0, 0, n+1]])
B = M.LLL()
print('Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):')
print_matrix_bits(M)
print('\nPočty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):')
print_matrix_bits(B)
```

Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):

[761, 1266, 1] ... 12 = 1265.70 (v bitoch)

[0, 1267, 0] ... 12 = 1266.34 (v bitoch)

[0, 0, 1014] ... 12 = 1013.34 (v bitoch)

Počty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):

[959, 959, 199] ... 12 = 958.63 (v bitoch)

[0, 0, 1014] ... 12 = 1013.34 (v bitoch)

[1068, 1068, 308] ... 12 = 1067.72 (v bitoch)

```
[14]: print(B[0])
print(B[1])
print(B[2])
```

(-

26193578714260426621207413884689029824090876088865909695132086973095756836771599
84919609302240336183782101552887564927396025021167879778110282763786647875686055
00527042226303710268584700840353440947882875522390973901097035563204133503393436
4296086105663431843070632372192181477998925447168, 27065540021912611519970065611
13866379990812451960152458068384550898090366702564189779940138192760695818843041
58886819973584932442441321276671114953367806773611366790262146791583808185910572
07921449436695613845208222296689843965787945196876653473118642579434803566366766
80258241329176248320,
431914847505162772450704837326985187908449771866261567886243)
(0, 0, 1113839657556245700649787422116680616657619090307463995532352807222505794
7482789845832375762249877822321102582567108031086159063768117911148564303764952
45121552420308950153790207091463034483721831514262338167685420549117433685292180
9584338725283905912794235624945373644421301852962202571596230645925948960)

```
(1865308331567021550947828949182168477676455297828451007772548118704328578835061
95450102869535205567223407251016220396682168007936779348491616227480654465102941
31052570016122847743462777506459695199767976303223171102916581566261634048492562
62110903877708203442161095237466235637571833065858467021979491699502991176930164
736, 180521432676789769037813823348160664124097919615249797387913933734732904317
79822569835375162353830493349982916628510371685477216709894829518979846995611127
85417961150549443503810291360145180252413831218564250925615172485886865617565631
98120309282595938423942208512421583679548813283934583567734141392019969679856249
0777600, -3075770486986803824326694067961810026677850638320702639149210821164022
59990171743156035781011)
```

```
[15]: print(abs(B[0][0])/X == d)
      print(abs(B[0][2]) == d)
      B[0], n
```

True

True

```
[15]: ((-
26193578714260426621207413884689029824090876088865909695132086973095756836771599
84919609302240336183782101552887564927396025021167879778110282763786647875686055
00527042226303710268584700840353440947882875522390973901097035563204133503393436
4296086105663431843070632372192181477998925447168, 27065540021912611519970065611
13866379990812451960152458068384550898090366702564189779940138192760695818843041
58886819973584932442441321276671114953367806773611366790262146791583808185910572
07921449436695613845208222296689843965787945196876653473118642579434803566366766
80258241329176248320,
431914847505162772450704837326985187908449771866261567886243),
1113839657556245700649787422116680616657619090307463995532352807222505794748278
98458323757622498778222321102582567108031086159063768117911148564303764952451215
52420308950153790207091463034483721831514262338167685420549117433685292180958433
8725283905912794235624945373644421301852962202571596230645925948959)
```

```
[16]: M = matrix([[X, e*Y],[0, Y*(n+1)]])
      B = M.LLL()
      print('Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):')
      print_matrix_bits(M)
      print('\nPočty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):')
      print_matrix_bits(B)
```

Počty bitov prvkov matice M (a 12 dĺžka vektora v danom riadku):

[761, 1266] ... 12 = 1265.70 (v bitoch)

[0, 1267] ... 12 = 1266.34 (v bitoch)

Počty bitov prvkov matice B (a 12 dĺžka vektora v danom riadku):

[959, 959] ... 12 = 958.63 (v bitoch)

[1068, 1068] ... 12 = 1067.72 (v bitoch)

```
[17]: abs(B[0][0])/X == d
```

[17]: True

[]: