Γιώργος Μιχούλης Άσκηση 7 Dai16067 Κρυπτογραφία

3)

$\varphi(n)=\varphi(p)*\varphi(q)=2*10= 20$ ο αντίστροφος του του d στο n =e =3 c $=m^e$ modn=26 , επαλήθευση με την αποκρυπτογράφηση c^d modn=5

1. e = 3, y = 26

n=p*q=55 $\varphi(n)=40$, κρυπτογράφηση c=m^emod n=14 άρα με τον αντίστροφο του d του e βρίσκω d=27

2. d = 27, y = 14

4) p = 31, q = 37, e = 17, y = 2

- $n = 31 \cdot 37 = 1147$

  $d = 17^{-1} = 953 \bmod 1080$
- $d_p = 953 \equiv 23 \bmod 30$

  $dq = 953 \equiv 17 \bmod 36$
- $x_p = y^{dp} = 2^{23} \equiv 8 \bmod 31$

  $x_q = y^{dq} = 2^{17} \equiv 18 \bmod 37$
- $c_p = q^{-1} = 37-1 \equiv 6^{-1} \equiv 26 \bmod 31$

  $c_q = p^{-1} = 31-1 \equiv 6 \bmod 37$
- $x = [qc_p]* x_p + [pc_q]*x_q =$

  $[37 * 26]*8+ [31 * 6]*18 = 8440 = 721 \bmod 1147$

2) m=2, e=79, n=101

M=3, e=197, n=101

$79=1001111_2$

i	6	5	4	3	2	1	0

b_i	1	0	0	1	1	1	1
	m	m^2	m^{2*2}	$m^{4*2}*m$	$m^{9*2}*m$	$m^{19*2}*m$	$m^{39*2}*m$
			m^4	m^9	m^{19}	m^{39}	m^{79}

όπου $m=2$

$2 mod101=2$

$2^2 mod101=4$

$4^2 mod101=16$

$16^2*2 \ mod101=7$

$7^2 *2 \ mod101 = 98$

$98^2 *2 \ mod101= 18$

$18^2 *2 \ mod101= 42$

άρα $2^{79} mod101=42$

$197=11000101_2$

$m=3$ $3mod101=3$

$3^2*3 mod101=27$

$27^2 \ mod101=22$

$22^2 \ mod101=80$

$80^2 \ mod101=37$

$37^2*3 \ mod101=67$

$67^2 \ mod101=45$

$45^2*3 mod101=15$

Άρα $3^{197} mod101=15$

Με βάση τα παραπάνω, δείξτε ότι δε θα ήταν ασφαλές μετά την επιλογή ενός τυχαίου πρώτου p να επιλέξουμε ως q τον επόμενο πρώτο αριθμό.

In [2]:
```
bits=1024
p=next_prime(ZZ.random_element(2^bits))
print "prime p=",p
q = next_prime(p)
print q
n= p*q

# Factoring n
[p1,q1]=crack_when_pq_close(n)
print "The possible first prime is", p1
print "The possible second prime is", q1
print "Verification", n==p1*q1
```

```
prime p= 16504579541543489411060278237962464548284801608261020140022408318937378085558128367487635692451836860478734723524804900
94884418180938090530543605865552980376881618443300870777842705025774561085040217073389918495975715419837994932928996469390647107
5420984362869712856120789195835210411479015531537595642117569 9
```

```
prime p= 1650457954154348941106027823796246454828480160826102014002240831893737808555812836748763569245183686047873472352480490
0948844181809380905305436058655529803768816184433008707778427050257745610850402170733899184959757154198379949329289964693906471
07542098436286971285612078919583521041147901553153759564211756 99
1650457954154348941106027823796246454828480160826102014002240831893737808555812836748763569245183686047873472352480490094884418
1809380905305436058655529803768816184433008707778427050257745610850402170733899184959757154198379949329289964693906471075420984
36286971285612078919583521041147901553153759564211761 27
214
The possible first prime is 1650457954154348941106027823796246454828480160826102014002240831893737808555812836748763569245183686
0478734723524804900948844181809380905305436058655529803768816184433008707778427050257745610850402170733899184959757154198379949
3292899646939064710754209843628697128561207891958352104114790155315375956421176127
The possible second prime is 165045795415434894110602782379624645482848016082610201400224083189373780855581283674876356924518368
6047873472352480490094884418180938090530543605865552980376881618443300870777842705025774561085040217073389918495975715419837994
93292899646939064710754209843628697128561207891958352104114790155315375956421175699
Verification True
```

Άρα από τον παραπάνω αλγόριθμο προκύπτει ότι ο επόμενος πρώτος αριθμός αν επιλεγεί για τον rsa τον κάνει αυτόματα ευκολά μη ασφαλή .

Δοκιμάστε να τρέξετε τη **crack_rsa_factor(n,phi_n)** για n μήκους 1024 bits.

In [9]:
```
def keygen(bits):

    p = next_prime(ZZ.random_element(2**(bits)))
    q= next_prime(ZZ.random_element(2**(bits)))
    n=p*q
    e=next_prime((p-1)*(q-1))
    d=inverse_mod(n,e)

    return n, e, d, p, q
```

In [11]:
```
n, pub_exp, pri_exp, p, q = keygen(1024)


print "Modulo:", n
phi_n=(p-1)*(q-1)
print "phi_n:", phi_n

[p1,q1]=crack_rsa_factor(n, phi_n)
print "The possible first prime is", p1
print "The possible second prime is", q1
print "Verification:", n==p1*q1
```

```
Modulo: 3752812482370888754951224399577562158602524871569572727339952633433064219987124759757058272474853036099599811764737961 2
46908857622764736469848799673915539239137622002151848028407918411706466250348234618962462128818816955442805646021660149233813 48
75815348201742678702094454305641128283932058648367266187578404873843642774921279610935033624594341671989501362018805360353759 73
79171157855233624598907952354443116507079817247743101740473147946601359054833003500970418885970264210295263823974520694176593 53
661175620322183138990772083193291110374005262033924354383630668865717131425884821567843496377725022206937847422413 1
phi_n: 37528124823708887549512243995775621586025248715695727273399526334330642199871247597570582724748530360995998117647379612 4
69088576227647364698487996739155392391376220021518480284079184117064662503482346189624621288188169554428056460216601492338134 87
58153482017426787020944543056411282839320586483672661875784009785847991957352398707468071575506879439907595517299958041109251 95
52690752981161144906694506645945467308254450967494086339350094976407836512597506226202148411091338637018304417938076728504670 61
72735622958382044178700130417261536338075877928355341205061743669722478274055414442089838407686287104037706968103 2
The possible first prime is 21487460104877301979314656695258913333720399446924710686417776661318470772633479227764434613934087 7
00771062120674341444988158959621506518397123786308759180237062911406082038340275992835620501935053735302355442626538915453630 72
873052165740727223275127533941192024960467983250717906912997550484414982782047583 7
The possible second prime is 17465128330914558418087225569411523203559587971178177409144651684105431309923638282343803887755760
86900519225147665224811837917882745405700644946656959102380779821894051139504190472018570562453983066467128256832158180407412 95
67009837636433783457380510349840039696926838461564078467449724882506879173584067263
Verification: True
```