2014-01-29.sagews

January 31, 2014

Contents

1 January 29, 2014: Public-key Cryptography, part 2 (RSA)

- whiteboard: `http://youtu.be/oLda6PVMpxw`
- projector: `http://youtu.be/KkojthaCo5c`

Approach: we will focus on the core number-theoretic ideas behind public-key crypto for now, completely ignoring the nice infrastructure that must be added on top in order to make it pleasant to use and more secure.

1.1 Whiteboard

(explain the theoretical idea behind RSA and how it works)

```
rsa={}

@interact
def f(bits=128, go=button('Make another RSA cryptosystem', classes="btn-\
   large btn-default fa fa-refresh")):
    b = bits//2
    print "Creating %s-bit RSA cryptosystem"%(2*b)
    p = ZZ(next_probable_prime(randrange(2^b,2^(b+1)-1)))
    q = ZZ(next_probable_prime(randrange(2^b,2^(b+1)-1)))
    n = p*q
    while True:
        try:
            e = ZZ(randrange(0,n-1))
            d = Mod(e,(p-1)*(q-1))^(-1)
            break
```

```
        except:
            pass
    print "\nPublic key:\n(e,n) =",(e,n)
    print "\nFull key:          = stored in a the global variable 'rsa'"
    global rsa
    rsa = {'e':e,'n':n,'d':d}
```

```
my_rsa = dict(rsa)  # make a copy
```

```
my_rsa['e']
```

72330484215228990241330972840062122571271501977939996743994003855274295694604713315918131764490744949217940669683114885812798041107565486303594958745768475818278981025977437554876774701449630190717573708098578237531035873263229298105115724397081857311972480435661171323015414227686214012491894182072690790959909636427702211785827007260313370725161147033797043392873088469816114263115248646842187325796021210957070606131631274456032237675252996445220006841835412028409030741761313531167972455658817768528582707269904182377113035362511693190971572161612940268572171337514708685550714804842082492209841661521073675660803

```
my_rsa['n']
```

80587966154926772272218789712892350007614334669220796549905512753560571491138113672111968712776887847804550876973926832754164506531422013795934649064602484821930582521218528208863133176333819428002680561853960344609844317368667642425539942190563999754683870536014990267740249212184401932107887288182146166940513484689169921040024593749621060120178308392181203645586305267327481438807074429287808753487856631867518449138274480597005589545704408598434563572959793628711405174270093651053898676386178154571570610896096742980616443018306119071726041386335583413982253655806283301574614270146436794151933476669329675016417

```
Mod(480,my_rsa['n'])^my_rsa['e']
```

13135518089016428773403319392223914075106173445405777581995153281519995058824368139997549698328520249570472463731009199502134688275193526087634969103980511438365617071121855399947507913858235981462630722467224723231199301959217346949044765590935758556174977423304309764819406577676599207499534675196436306198798723178176397728679165103770112206848574385097258190389727437570234773101851800085081176798272618112032663785760053851462153095808842818521752271809595284668824319894763004605204380176356664988309328073519219184322377913363463764638409289028648444276783089590184672365262955638888657277723253496075936665787

```
Mod\
   (1313551808901642877340331939222391407510617344540577758199515328151999505882436813
   my_rsa['n'])^my_rsa['d']
```

480

```
my_rsa['d']
```

21023521139220306871129829278676797772749012626544374685718569190642640284400201955744284942706342421462509468496141871619401842334612484830510038325657457712872276028516705015151280703978366837997358651137060054097782398137077965963499387622878196336263302976358785065432395892524551446850177363792734581396135883389319607376335173376873977262109760080632669596859149625150898958170395728658383892586161274868071846751115991001919864613597111390217284101370612287325393892602350573054736611381090947452798065014349696750181678058661450141594308143652931552727288684624360281371007401898363995819184045032893325475047

```
t = my_rsa['e'] * my_rsa['d'] - 1
t
```

```
0
```

```
n = my_rsa['n']
```

```
parent(my_rsa['d'])
```

```
Ring of integers modulo 805879661549267722722187897128923500076143346692207965499055127535
605714911381136721119687127768878478045508769739268327541645065314220137959346490646024848
219305825212185282088631331763338194280026805618539603446098443173686676424255399421905639
997546838705360149902677402492121844019321078872881821461669399348041104003379442887946961
340339841240992751890210900554937517700332379368801791921614335689995692681993439504791085
594252437847020994272527588676235126311805802670244827979300453788412588034125185010339704
860174873538337517017804716942383930242735889092025653978855970396774922086511641373179507
75717662996
```

```
phin = \
   80587966154926772272218789712892350007614334669220796549905512753560571491138113672
```

```
p_plus_q = ZZ(n) + 1 - ZZ(phin)
p_times_q = ZZ(n)

R.<x> = ZZ[]
f = x^2 - (p_plus_q)*x + p_times_q
```

```
p_plus_q
```

```
578680578769583095735799053487026136054209116992182555530811515557448200870194250095647319
918857062599319105187795372037580345761002309171181804705336280997530824907245610853123853
297544919351159052109862126256963129089184554417291254347147942559140393344453240885415977
574592654228142987796158718553957353423
```

```
f
```

```
x^2 - 578680578769583095735799053487026136054209116992182555530811515557448200870194250095
647319918857062599319105187795372037580345761002309171181804705336280997530824907245610853
123853297544919351159052109862126256963129089184554417291254347147942559140393344453240885
415977574592654228142987796158718553957353422*x + 8058796615492677227221878971289235000761
433466922079654990551275356057149113811367211196871277688784780455087697392683275416450653
142201379593464906460248482193058252121852820886313317633381942800268056185396034460984431
736866764242553994219056399975468387053601499026774024921218440193210788728818214616694051
348468916992104002459374962106012017830839218120364558630526732748143880707442928780875348
785663186751844913827448059700558954570440859843456357295979362871140517427009365105389867
638617815457157061089609674298061644301830611907172604138633558341398225365580628330157461
4270146436794151933476669329675016417
```

```
v = f.roots();
p = v[0][0]; q = v[1][0]
```

```
p*q == n
```

```
True
x^2 + 805879661549267722722187897128923500076143346692207965499055127535605714911381136721
119687127768878478045508769739268327541645065314220137959346490646024848219305825212185282
088631331763338194280026805618539603446098443173686676424255399421905639997546838705360149
902677402492121844019321078872881821461669393561235316307548485529956426470078480698901581
```

```
968385345246822362125850370666859290965141136501425066688802387626837365218448980236997902
560709541622872316336497553597788719448061920812963394522534663911718442290543582646491972
844892173470904504651331955647493245124696194650848379805081763411592322217603095 73*x + 80
587966154926772272218789712892350007614334669220796549905512753560571491138113672111968712
776887847804550876973926832754164506531422013795934649064602484821930582521218528208863133
176333819428002680561853960344609844317368667642425539942190563999754683870536014990267740
249212184401932107887288182146166940513484689169921040024593749621060120178308392181203645
586305267327481438807074429287808753487856631867518449138274480597005589545704408598434563
572959793628711405174270093651053898676386178154571570610896096742980616443018306119071726
04138633558341398225365580628330157461427014643679415193347666932967501 6417
```

```
v = f.roots(x, ring=RealField(50000)); v[1]
```

```
(-1.000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000014361463786220773483326968464106138564607766071 30
125720693675732223235349944850296380392478407693802959434360006415200480257388897561485810
120372271239583102643818338495848739696582603153977876923734854480568872788543769223590720
789550701316582838964636540735355295867437222496895138725059303743911025351318083777840325
946600510010753950524333467293971950408115883463011410653639295341333728400892013700324920
564057686907272795377700495813896827307383287265656087872821034022285685621421348896536634
475287210675138141613335433105759234005069540664108844341443944625436672182689257074620538
572093857699391110392965037295103436479816234081157995166474719385281692089903950667184972
342641589115415357170253569726865719794178784719086402108103828803009152849652494630741148
483021375249713456038098363906680095309393180150661751264533747475468915436157342095764846
305721634978159328159205511511089877371671916611904574750874418938761842448559387169829101
027515392937292894248328283746395588270032272421819874377127747436948066047085506622085819
587531733132236864595535861798381668501768776691933169821871394590245498489885724429487034
411339904344786150176617140914722142569515841491325931760199383012661030408756355704581899
855457420691098524024317389214953877950006822654280436985251342291265222087990874400736417
856831318483014522555311898511798945066761754446749450390685475474013331833142519283615537
376817798367670107932952733667125689992618723155471429393139173553062752256059484662766903
435010553434955017724427713446820134235310679090934218269681982267599258276214406353403842
287077965212350292829391811355672442554889396580317288256907132474043765789503849041529706
677885032023585447434808555039303808978036348551269111494578375489152723641437229243117143
591383774481433113018385598489614017842332897311280682520222938709133019270480004446250302
435627385035489872103967133842534409565595229998372605857379148539001543356437710142968295
383184960316085801508235723101638993728228346343825807933088573993251890818618611039292248
885524608850185848815674404561697873508970346747043842417236512873874671271250106228018547
281675918754568013938369257914591013740970490624459356441243977820744056592585952810687688
771706828567100486717736138713566552215558745894209990146843519968005857688549726657263997
899950455266605572168352281061681929857150036574791740062172918035748541497330058234638785
769156588623991808707112638142214115870678640765752954360629603555478314812558939300413281
708215223547820832754468839090181771872355339510998289839103925141931490361696581897077313
769115948577934298982412309470170941049980813134847844789638199286074536793454076460285061
104779334249489362902124842217653990171934525611529226845719327123167482381217918719427403
954912829128905882014349574524396445866379679075998814054042244876976470641374305407619903
350339728724411761254636803421854442019398284154778341037860334278712175553213774774445407
434793612454999757749807939692712231587830725937462563016253877599055201571440823530092107
944321163222751930853121278523054548151030367861177310896434265707939763902689624444726957
897630114130622508720083534354471666867814984501676893615000486829042931420688724549537736
```

3469756671770094143096109855971544720035406823620866232216459526521760249259567688040869l9

6541469875679749316797041171957379534247365827557750990273596501292158336871009304061757087931468611996032803820661230096205247027049646812483255311013563004110707330518680725059961498450010869638384347945651065467974738101467695164526511864525835430417432475210163274295132170388291791367790928195210662296856946841684297827977934678374603717374443621882463335022759241427087036495300694378870925177454341202802197881554549007160210769218459569429638904520441298647264711539708223866728765186811613181718082493586289376074587858640654123106783986690854026026760972742774726206730544673641003967156595316479031730575831855866375248370808177709701838629578283806629013861572650544131053995492211708056966773286371136465287010891024308556427506941491746616794509267381004666829875977429137083284239725792614647432556970754392775716043097290629626985774601106172224185693760652891223286930091876580718584675486342361020698499453936110473521918786807493500830362620146691123563784156231806540980165920024850825542344267381161896519840817530399645501482036738216549753707579010362819818769035812065996363146372950675723032368924692499408755670032464812479453711531180002872622301311026975555613516628219486635697107065402659970338921132050617101447842912126265905047755723733045899991893267555681576199098217235685435374970578047114489420042135365242408248128137865936569533386598516763570882091891373354182210119986604182885823558736788056665767796543993704207608414944882220040845105328232686326707485086552839127248462872075686857279346772157990076625965035376683488426986787461199772391524108511312325874715456983719888409100542297703387663397827369246654238708317638071469491200506770455187531796723153352044164734651294879228133869122453551926780811885238501441996374911272924488739028079340521420062475717806764132632988079786858551358305321221171105701019441711754188232518344428884532482173030007821875982219331858965441998334907137131016850852363911241507550415590326258566218245783744943502098973592797090654906810770883616422328206480892207752218611914625471621922793614168528974934553242664047950285660588233188243914287721205605525627447866798865763680353179573388864856884851756757770492284134570163929691999458933734927530454376508905187629981775196423641119105215324897268669410045123693003340955706518667781501871858109952748615374937528778987548184779882431901719025753548923119902277000796746233136137939686786945847972800235238926620656874804484346076900909071069894613937402149597259000614898330004514396373311997535823276511658085459405948645003280474884943063572552621364609718934722265246464975202922330410775228748637573092420550611622469430240455978559231140968029576612053390737301030500803929774887411716031741792758279826023615644645168497269286401077770219459160213557465321708181981588894199072758251361424000733515088854202106802152619536006120522279327293362567223246888349586496669969965039006926057784347614249942222671469321099527240352296112554385378213859877504972690211480980183624718800697746304765672119550087417101538360921778124308631069113724076767520815509506810304194421305315433217687633346665708041199872290984420991698837677743408170846801382610791371106021780247092861022448260044024121372381291744922182505178515346625890471096676024182445214517981151773905017964835614338578501317622756518563778890534570590988890628440603498988907654787632420562383253587414417681711652349699927221247616760904226558726972287826033532341430783431866141646745807482378395002701603593354085567085106994525050920344706016852724308485434494181196304999120777798320396358753149997937533902048158704574576592806888693080102958279123081256541744939565099177044129887072883661902340982840068922300542062069945403008060900053810623818542599243681635573046664110437730217977424733951637289809522831195045472311783075488620173438119050275056043613519955908040677608113804761775417292722686440059472846534161857144823093323247891511494522736058672033666382161704023560402544488505965877800007776733294618961626973065404880912462248515357599642620796589640015894628579414725198660868930569010933110217544264769017847925619471860896067170546801239916738844532104195085853822280591172170604070661673588896003796993833762079328630456079028261891530271052123397393492347805863960456876465633906818187628266978255708102002640133038889804430115684934795069132233141558204953517660050574043381572058603485840542282181416147177136891136443187614371554348648788501342835334077607167156694514105994003748985333066388356775648117407434820602542747641644973150151273736391545714 7

```
9959913307842852604270104248449569787071531838600467204954422425223256215777053857810859576
9999488658423111304518913239372842879640985231264499253500110855603385877867650613008765877
8050916963339366380133265138109119394771028449649034815272422457253258713485365173597232982
92619061361262491187287383082923701643691275190209051418582022082090071623446404838629878
48553687365597392794863787956875095743045190736685348747883551218562177915875837184835251
5626447458495081520391406122485670486377108835064709839892610955206787378190086942144843118
7410376912147768580825051964987171339446614085059079009538225067054820494281854945036441118
06407848984629916632386509110393597375425396228644802774852229263669228142006809673342472
78921690409922245987516685193343832307121349205039469295915700416619237019891534631842212
85365065521527868081301812261509826941246760892343755540194286317964274959016919775895835595
649685374170316189005999534463774235939248092544225247189572238813553266607400068034064399
37538800029885397126334855289740521230772638896135955967121867934109185099752070932871604
9411701957724646807687131880502299287155938009851342119912218061076775883659468307159611166
945886263062468711829970469175551644305062057499310031017915856857950044651329879689929349
551085778639501037537116060276439926502275636296733871839405788921917690768954182061239732
47807464283151168778012746179138172572968773593660608921316160084637185580385140003596695
939395380885984688287204905810281157688780846277036727839355017566056322317204426104155177
11084956313575966305579845667735134209069711666559931165699722108422910205742542684677197
745984217499068715939293633712701808423522320557718308372131314769859289722746037927376403
618200138047305424398426942682980824955114975447812224490769090101979593587631545834670587
998072005257157599808653916535586957391315190999450293954231611932585726807151242676213478
717376475106906825502930581196643564969827612303124232566937723641402275185325743719023345
297865351565448141458154997104693537315273269229235328567722557857808602866119578893546928
848632700198690344497089131960574560220261086179665184048248614361061797513818849021863170
911813997818951216263526949017081758411268772833613444277215269690057153142862869701690269
218338939720321572104521231551892145973396528417983216815239003333939036727501842999060404
246825768900242656368744, 1)
```

```
f(-1)
```

```
115736115753916619147159810697405227210841823398436511106162303111489640174038850019129463
983771412519863821037559074407516069152200461834236360941067256199506164981449122170624770
659508983870231810421972425251392625817836910883458250869429588511828078668890648177083195
5149185308456285975592317437107914706845
/projects/d6df9d1b-2462-4aa2-91e0-995610ea1726/.sagemathcloud/sage_server.py:680:
DeprecationWarning: Substitution using function-call syntax and unnamed arguments is
deprecated and will be removed from a future release of Sage; you can use named arguments
instead, like EXPR(x=..., y=...)
See http://trac.sagemath.org/5930 for details.
  exec compile(block+'\n', '', 'single') in namespace, locals
```

1.2 A live demo with somebody in the class

1.2.1 Part 1: receive a message

- I will put an e and an n into a chat.
- Then wait for a number back.
- Then we will decrypt the number.

1.2.2 Part 2: send a message

- Student will make up an e and an n and paste them into a chat.
- Then we will encrypt message to student.
- Student will decrypt it.

1.2.3 Part 3: sign a message

- Student will make up a message
- Student will encrypt message with their decryption key.
- Send it. We will verify the signature.

1.3 Crazy project idea crypto is a pain to use for communication, so make it easy!?

Create a small javascript library that a person can embed in a webpage, which provides a little textbox for encrypting a message to you in it. So, e.g., I could put it at http://wstein.org, and then even my mom could easily encrypt a message to me. She would type the message into a box, then click encrypt (or maybe even it would encrypt as you type), then take the output and paste it into an email to me.

The configuration for the javascript library would only include the public part of the key (n, e) and nothing else, so there are no security risks due to that. The main problem with this is that the person sending the message has to trust the website that contains the public key.

Note that the message never gets sent over the internet before encryption it stays in the browser.

There are many existing libraries written in Javascript that basically solve this problem, e.g., `http://www-cs-students.stanford.edu/~tjw/jsbn/`; so you dont have to worry about actually implementing big integer arithmetic, etc. This would be more a matter of making something look pretty and friendly and EASY.