Homework #4

Due date: 18 December 2020

Notes:

- Note that there are five attached files: "RSA_Oracle_client.py" for Question 1, "RSA_OAEP.py" and "RSA_OAEP_client.py" for Question 2, "ElGamal.py" for Question 3 and "DSA.py" for Question 4 and the bonus question.
- You are expected to submit your answer document as well as a separate Python code for each question. Do not modify the source codes that are given to you and do not submit them. You must import them to your sources as they are, do not solve the questions in those files.
    - Source files to submit: Q1.py, Q2.py, Q3.py, Q4.py, Qbonus.py
- Print out your numerical results in integer format, without "-e". (We do not want to see results like 1.2312312341324523e+24).
- Winzip your programs and add a readme.txt document (if necessary) to explain the programs and how to use them.
- Name your winzip file as "cs411_507_hw04_yourname.zip"
- Create a PDF document explaining your solutions briefly (a couple of sentences/equations for each question). Also include your numerical answers (numbers that you are expected to find). Explanations must match source files. Please also add the same explanations as comments and explanatory output.

1. (**30 pts**) Consider a deterministic RSA Oracle that is implemented at the server "cryptlygos.pythonanywhere.com/RSA_Oracle/*<your_id>*". When connected to the server, you will be sent a ciphertext $\boldsymbol{c}$ and expected find out the corresponding plaintext $\boldsymbol{m}$. You can query the RSA Oracle with any ciphertext $\bar{\boldsymbol{c}} \neq \boldsymbol{c}$, and you will be given the corresponding plaintext $\bar{\boldsymbol{m}}$. You can send as many queries as you want as long as $\bar{\boldsymbol{c}} \neq \boldsymbol{c}$**.** You can use the Python code RSA_Oracle_client.py to communicate with the server.

2. (**30 pts**) Consider the RSA OAEP implemented at the server http://cryptlygos.pythonanywhere.com/RSA_OAEP/ <your_id>. The server will send you a ciphertext (c), public key (e) and modulus (N). The RSA-OAEP implementation at the server is given in the file "RSA_OAEP.py", in which the random number R is an 8-bit unsigned integer.

I select a random four decimal digit PIN and encrypt it using RSA. You need to connect to the RSA encryption server that challenges you with $(\boldsymbol{N}, \boldsymbol{e}, \boldsymbol{c})$. Your mission is to find the randomly chosen PIN. You can use the Python code RSA_OAPE_client.py to communicate with the server.

3. (**20 pts**) Consider the ElGamal encryption algorithm implemented in the file "ElGamal.py", which contains a flaw. I used this implementation to encrypt a message using the following parameters:

q = 2022967828283532245360658374440322019407596246123901055008730902181 1

p =
12515043909394803753450411022649854273721537251011107748826811168459680628351391154487041320595006736239332192492236943966523053744476127728797963808151142506595330120621663371518281181204797831707349436558443139355672347825267728879376289677517268609959671235059224994785463608330669494457163250373581380036247652030969481046772013799271268710104487022164865004802864076066974153012125551060906054112920469869045223329577015935824864428612446723942040465300185917923305042033306319809712618872063796904132788285518497999327485929730921202745935936913834577610254298809205575162005025170878200786590751850006857921419

g =
22564831437414331634130076750679345428930229683374373122833819649423443654497196282556307523973253764520023987843940085078570253869436454376965582408744713454425323985884067499079300024816241609591321937988424268221939101049621388458734255909463417543341442928860029629015501605784824521380753392948262417996457616553209837353819741776352072084718246675169566799139746433421595500373203788144458022968794705615045116894609162004179026123230396712505675038461759906545129158781432012330509780462695511261781550601587816450621819557819691364359055707874578555300039878870491186995250331208117907395905646843165504931321

public key (h) =
12651261389333779943487931934773422473695660035496471394558220529065182767460500374129040082614616575245270159422600221303665020886334032132979848926416072893065331590752192613664292834754982514402626203574735018249379555938507013095955249981388520233457599364293512813245854552349848949058688318784839631416487405675769615498951163392762086955722255687685599907930883941741601274620604045561100209252025573612167329896305069363991636796828080702897561459611402223052436015058134488421983451902561977785843043115946156287153700452347216167218285105225846661076288457031089402762830390116167478378832047974721900027 6

And the resulting ciphertext is
r =
38136774394448379903812816247692654840719898834948337653631552140717275736275902130388230180546536140408333065337365937895236367160887516095915178528682170529054157514579619423092138037826611740421310675559968600942963154830873754443624540928919604920987962346243921861126591249158725466407231397628744530505921102720369170392930205397248724068560662527794194826516723201320924219398673926687959591553126348048882153006077255843305317202103552015505297649368817612108108831029864641114090965723641855027224775871787101371758286960006830288069206718597979821573839438661113202278301051784216903036276279433371287954 46

t=
10192033240113377640860169195054315981727514327329008790444130729107056930047299547755150775636237252367979032815426685448329207318153807026080454908281501010744250818034651670583477952735248499512182344163892706808295058861406156976805817052235913385764008189983049947270530103937035152 1

420218366455339131411357551140764438126771947195782053945008617757152744
217016924020123490958494912868839202572900622972687511545403791087788818
666974197010060741654181636018567265909959822441880913688112140583853563
039674753932740550977819372969408746190277245110503970803036214270054032
007366960967640136372910067377537941198l4

Can you find my message?

4. (**20 pts**) Consider the DSA scheme implemented in the file "DSA.py". The public parameters and public key are:

q = 1846287079795873435846054031580231196374499995450680798150849863509l
p =
218441021121222374840584849902232225278169817028282791714981430365827l6
271485474028380542696862193720852272618397503658771128114568430034544311
183684813255659132427311783911547834305153842743766472298083077116193911
392229647076952769574329680333653523020803663154157355321113027108578071
281798249043320899027800135122873123243743524724602070457967657285884561
385896818773268072336990622221420125028844382472226168282897015873158761
635851740328877679882191439967173809239980967940600640232645849491153541
715211375168860544716843940259887168163262505413440632980952366656691931
523253872172645003708726385493517979869499934551 7
g =
138430796393513409202731847145908844004328470930587709707751330796280153
434746389859495142244692313165093017861918372397347435248047071568376l
531935541921594509486532039975603749073427519750724397889015823137921001
993677556902092176523269334257581700088350846572416755455713241462027141
002127571892258435472678396358353938476569410849475658691697420643000081
672415616727585528670819194152121399807440412629523055909019685252549851
681260299061791687895851524383306222527536435538058772576234339746393791
577436808678860489830511416186993204671106346196262903362008285485594741
704795097110981484264361110301667084125319435624 3
public key - beta =
618748121365817649878712412360168409178004669098522738667412703425403931
658506466553105422417249375141125191924854976697381051441736079923476261
869972509174309127140941080651743898030456747633487761927322752193676171
631421188466276887178326057235498959215675535243710175803133084606449251
307793484772983947165014008497883808476800397448079531920062330698504281
367974025006391433578254859633968702925514987402010031888483663325943691
261887057689382602101878354358031849345625112734143769110252248291974381
728550982145394264479609346268901387983454182509458854320842674999915341
859914868405673669793055732755540914971556038 26

You are given two signatures for two different message as follows:

(message$_1$, r_1, s_1) = (b"He who laugh last didn't get the joke",
61645729931482682785443152461587949660612434566030814273897926987 84,
24128748367753682301949576594052584495795795683405012176181776297 80)
(message$_2$, r_2, s_2) = (b"Ask me no questions, and I'll tell you no lies",
61645729931482682785443152461587949660612434566030814273897926987 84,
34337936512827072053959736709548530112897017827410484618959879516 1)

Can you find my private key?

Bonus Question

5. (**20 pts**) Consider the DSA scheme implemented in the file "DSA.py". The public parameters and public key are:

q = 151413390842115377807984028214686682532338552932502824707074865237 29
p =
154593526781701949990597979538359437037692997985226404859492510212300 61
239872933286596281671875036444766767260825161156339142374953144264667 17
566309353221001697700029628142818005296251209693003462670724094307390 94
299485686471754896419239470555236906623972754998140116596159333130012 20
733558180164993086472379325887209418439076036830595968948122463542565 48
845828555926915281484693046167880615571771594791617514000333739836058 3
671917023018170958737158107689503925766013454346510422824962588987982 93
897916341315693731763534513871295870117294672305447940132333142894162 79
075919670424097289941201659300622308787135740496 9
g =
380056962500864876604954553780747863915825666645383754315686520515734 24
531751953382939145183183899325124191970224921932670724667545946204615 34
567362497841710002599111953091344930343994503431071692400525354528547 91
807541053879027578190026731264198897307542646808702242785595428885829 94
589278088951898431749014172940178634272504225094118257474033479390191 2
974170222604015177323368814264989835679407076289974855552414398779625 52
183725791602255298002762705747306264487965963268120410780612014499890 79
913389132663343211603246514840127524416341402434657309396192425152807 14
356873699965985363402010686851443396200018800199
public key - beta =
138117181949128877312599736875316590172212330726937583393206775560859 61
091741512534312991319990988012320895125273138799484930424656328618986 33
823365079955513189685758600149059560436536808568274327571242813794322 51
197156284058923573060291505745841197858323256056748388011546418957453 11
161271889436502899846458131900988387777254676157672199525938326470244 36
388122781455708218778804666095243363155351706809573436502487691070902 94
168501148540640433388799405429019366249693032485952081087957512253872 03
405395739941042570698164719973037261394764330314120509607344408485820 13
330738888269995501032018331844706567548786132 2141

You are given two signatures for two different message as follows:

(message$_1$, r_1, s_1) = (b"He who laugh last didn't get the joke", 780720772592321367005945670607735754560466840092435474685060772631 0, 10137413521818981860558295844142463248736280669671376607939774420169) (message$_2$, r_2, s_2) = (b"Ask me no questions, and I'll tell you no lies", 136015176629902532449193926230063681738045241396803161473308458516 41, 53546380277079056260451560573610968903778113872483945224190692363 40)

Can you find my private key? (**Hint**: I ran out of random numbers for the signature of the second message)