Homework #4

Due date: **11 December 2023**

Notes:

- Note that there are five attached files: "RSA_Oracle_client.py" for Question 1, "RSA_OAEP.py" for Question 2, "ElGamal.py" for Questions 3 & 4 and "DSA.py" for Question 5.
- 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. (**20 pts**) Consider a deterministic RSA Oracle that is implemented at the server "http://harpoon1.sabanciuniv.edu:9999". Connect to the server using the *RSA_Oracle_Get()* function, and it will send a ciphertext **"**C" **,** modulus **"**N", and public key **"e".**

- You are expected to find out the corresponding plaintext "m". You can query the RSA Oracle with any ciphertext $\bar{c} \neq c$ using the Python function *RSA_Oracle_Query()*, and it will send the corresponding plaintext $\bar{m}$. You can send as many queries as you want as long as $\bar{C} \neq C$**.**
- You should decode your answer into a Unicode string and check it using *RSA_Oracle_Checker().*
- You can use the Python code RSA_Oracle_client.py to communicate with the server.

Important Note: You have to find a mathematical way to find the message "m". Once you find it, code it then check your answer. Querying the server blindly won't get you the right answer.

2. (**20 pts**) Consider the RSA OAEP implementation given in the file "RSA_OAEP.py", in which the random number R is an 8-bit unsigned integer. I used the following parameters for encryption:

ciphertext (**c**) =
155633174361451963459660128709513554675182231102646675371810749734360653505 66
public key (**e**) = 65537
modulus (**N**) =
7342003289123690169505044765550086134382471360514182286685089621205131680183

I selected a random four-decimal digit PIN and encrypted it using RSA. Your mission is to find the randomly chosen PIN.

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

q = 2022967828283532245360658374440322019407596246123901055008730902181 1
p =
125150439093948037534504110226498542737215372510111077488268111684596806283513911544870413205950067362393321924922369439665230537444761277287979638081511425065953301206216633715182811812047978317073494365584431393556723478252677288793762896775172686099596712350592249947854636083306694944571632503735813800362476520309694810467720137992712687101044870221648650048028640760669741530121255510609060541129204698690452233295770159358248644286124467239420404653001859179233050420333063198097126188720637969041327882855184979993274859297309212027459359369138345776102542988092055751620050251708782007865907518500068579214191

g =
225648314374143316341300767506793454289302296833743731228338196494234436544971962825563075239732537645200239878439400850785702538694364543769655824087447134544253239858840674990793000248162416095913219379884242682219391010496213884587342559094634175433414429288600296290155016057848245213807533929482624179964576165532098373538197417763520720847182466751695667991397464334215955003732037881444580229687947056150451168946091620041790261232303967125056750384617599065451291587814320123305097804626955112617815506015878164506218195578196913643590557078745785553000398788704911869952503312081179073959056468431655049313 2
public key (h) =
126512613893337799434879319347734222473695660035496471394558220529065182767460500374129040082614616575245270159422600221303665020886334032132979848926416072893065331590752192613664292834754982514402626203574735018249379555938507013095955249981388520233457599364293512813245854552349848949058688318784839631416487405675769615498951163392762086955722255687685599907930883941741601274620604045561100209252025573612167329896305069363991636796828080702897561459611402223052436015058134488421983451902561977785843043115946156287153700452347216167218285105225846661076288457031089402762830390116167478378832047974721900027 6

And the resulting ciphertext is
r =
381367743944483799038128162476926548407198988349483376536315521407172757362759021303882301805465361404083330653373659378952363671608875160959151785286821705290541575145796194230921380378266117404213106755599686009429631548308737544436245409289196049209879623462439218611265912491587 25

466407231397628744530505921102720369170392930205397248724068560662527794194826516723201320924219398673926687959591553126348048882153006077255843305317202103552015505297649368817612108108831029864641114090965723641855027224775871787101371758286960006830288069206718597979821573839438661113202278301051784216903036276279433371287954460
t =
687908587253288349667963728982775804438849359219227648501842046712717569244767622532757045084519131240982973460873073263678618135172379149975873467997825997443956427111643147855992040692428669811529143452987980194786401248492832411641056271385999898765960731705057532914294883261626409610533297765790556798759071224126102563471954232290324519380269047449932300962686285307035975532477619822290316123173732108318081948470453314136040218587292086840163578492330008036606560162651155033858571644673285437221914195401948090314681929552710521968577436734923476208111651472890746872124105564946175171141006612821878624160
t =
687908587253288349667963728982775804438849359219227648501842046712717569244767622532757045084519131240982973460873073263678618135172379149975873467997825997443956427111643147855992040692428669811529143452987980194786401248492832411641056271385999898765960731705057532914294883261626409610533297765790556798759071224126102563471954232290324519380269047449932300962686285307035975532477619822290316123173732108318081948470453314136040218587292086840163578492330008036606560162651155033858571644673285437221914195401948090314681929552710521968577436734923476208111651472890746872124105564946175171141006612821878624160
q = 144543125469417438164937125914379131119873669003
p =
137248121434045436247980738953059412416367251619167172965225060439638326312552007992983578734870080149141102688002009860722627928048376753275218309927198296531391131491381377746970705292972549293385978940242862964757496679733959578043293370426396437630135799843979374589693726945392682404824784160383287430661
g =
127223641921850109909544249881449009944648689040286349526712184078921702602665543540563817762837809423359475544561229778960073396175252439333049143438367080170746166373310913545533812707513022571241268299810387846306038162098727078834162806032355796383642287190219288720676739470587659262303423658215573377024

$\left(message_1,\ r_1,\ t_1\right)$ = (b'Believe in the heart of the cards.',
981126369090898234738868042307346087836651513598202853843851849265867793188323428404467568452706851518435205925210307780631074614791855841297248385000267419660097063751812009739442913777532935355998701963457948398283879115798092238301956748210799021237004599484194930009559746053404002746439347954181179534311,
765062002788709806228321620877063971849427311758810730722796538791253740267842312430822498385702091977887034189945986637702227749585904843662974734645479761571015367390566383404017099739109229529873329612584145068

77745248599494701005790194262083540626575172771336888597402032923407057219028984697739294234494)

$\left(message_2,\ r_2,\ t_2\right)$ = (b'??????????????????????????????????????',
98112636909089823473886804230734608783665151359820285384385184926586779318832342840446756845270685151843520592521030778063107461479185584129724838500026741966009706375181200973944291377753293535599870196345794839828387911579809223830195674821079902123700459948419493000955974605340400274643934795418117953431,
95801086901355834240081662719865802187550109851113545620170852280638597493801662857576200633666749663318260607079963837967122188013554434395565196430708343554452720734056250267521097855586180792722796772893530089500987302933561979841152407078582329739116130182358926512269862531407749668332924957717479984854)

Can you recover m_2 using the given settings? If yes, demonstrate your work. (**Note:** m_2 is a byte object that contains a meaningful English text.)

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

q = 1805500313882185460993621335578803659943388101853615025430346358319 3
p =
176952242452260222622155504361468152593939623702717492883211963469589133550637571222164000386991258971373382456456546231809074457753974769143264541823312008430398287532100519638386733995377507645193811240740220035330483629535797476949974219326280501747680370084190238919556383336839107832963200683135024679535498456293643286851680553313303784394601072626722079113840299167310404286007959522483856834483390513263738796230245863814849170485308679983008394521850450277431826459960688459152875139747370943110714852798301788023328843229534850329540556982632682916838056115475798531967524712596242424256873326579953494100 9

g =
478907394177723266392594611654851223645400719593071654584425551567192190208845464756292055958640255481925160753302638656844317701259596543265151649487309428467188058704308016870979272958086439952207044001358870142710077078552732171778406853125348901531317163844603480584784572056769141276030722060393916563487443459594857058394895156778390264353963227451031700867667564432415210708332548490156210485764462112134840941155765304182497306321559953952088287144985151338727061340046431487965283635236363783322535096379436227526180189495737251803103189366815162351752394021099534222962803011419041939620734317407037997103 5

public key (beta):
1831408160533218510686903726138665932536518466931856989835941853268730468186911958415037229987343935227988816813155415974234360530276380966386586121747340348158553225363319918657949382937198455018294836381584550181800201868806694527418279797492758151769276850910944244395645572497766748854242598561659704665374023326770662512666613356092618904914953512155804252127648818534285831773370510453137952688543495010103660892413395901461238209725480737625047159275781922088076720717434062444236969393756880954396658965471745598003472511293882525516878617801436300794663357187223445935638034452125753926695866508095018852433

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

(message1, r_1, s_1) = (b'The grass is greener where you water it.', 164729156993233172945115909955723620797521053648980278342384095478 51, 95920542676357017526087813590289547683451743851878312055040026 0096)

(message2, r_2, s_2) = (b'Sometimes you win, sometimes you learn.' 143337088913933182832859305604303579663665718699866932617499244586 61, 99688373390521303397939119290293263537643850410057515778544953 98266)

Also, you discovered that $k_2 = 3k_1 \ mod \ q$. Show how you can find the secret key a.