Homework #4

Due date: **10 December 2021**

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 Questions 3 & 4 and "DSA.py" for Questions 5 and 6.
- 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, Q5.py, Q6.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. (**20 pts**) Consider a deterministic RSA Oracle that is implemented at the server "cryptlygos.pythonanywhere.com/RSA_Oracle/". Connect the server using *RSA_Oracle_Get()* function, and it will send a ciphertext $\boldsymbol{c}$ **,** modulus $\boldsymbol{N}$ and public key **e.** You are expected find out the corresponding plaintext $\boldsymbol{m}$. You can query the RSA Oracle with any ciphertext $\bar{\boldsymbol{c}} \neq \boldsymbol{c}$ using the python function *RSA_Oracle_Query()*, and it will send the corresponding plaintext $\bar{\boldsymbol{m}}$. You can send as many queries as you want as long as $\bar{\boldsymbol{c}} \neq \boldsymbol{c}$**.** Then, check your answer using *RSA_Oracle_Checker()*
You can use the Python code RSA_Oracle_client.py to communicate with the server.

2. (**20 pts**) Consider the RSA OAEP implemented at the server http://cryptlygos.pythonanywhere.com/RSA_OAEP. By using the *RSA_OAEP_Get()* funtion get your 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. Your mission is to find the randomly chosen PIN. You can use the Python code RSA_OAEP_client.py to communicate with the server.

3. (**10 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 = 219513665504930007182618531052019965399406140369458564929892124340 43

p =
190885173558311304099347483787798059053027056419653641599421250901016179999753027018189667983545289186285230294418641605036825240678097397395230404359507071559404389248216599778765719713846458305369800636274222016830546342326148389081626885189560815658702360521026851058633029338654212437141532481678718233170785139259807373498180114308976944784406655175400625539527446266874803273512004508308289383817366261385711471144678517608879306099468067802238512104280975543754563520312324843005014163082784867067493063831332232307885072030133009137003833145105538537740573073700482783653386269696771584777005458361676730195269

g =
110874950916035308201558971318404732618584098845349297851209369703503765983404428968835746417446964796284726243423592721516375391368959336179390305203914071844167089070134876151918097100789097413593600008572563950448928630448151253847163474358130661958154856242917194434666888790494240079118444463388062441859223302103127079438253921155844702677052734051802239534092327598870055016955978239282077553105785546036019197246668095278163698881088115108784785384722750921541832394798817363426032256504191127746655167662401180025327253725932059230276993516464111941585907639160650510685760640117324612159323994472488373144 10

public key (h) =
987320377917824491735857382210871486622556457509380838121352205509962690504857471837707984399183951038894244713384696768128945681994737068382628603050350281850071815481158793767153887039702039469691684647739659826174304777422240197064874592125287633405202443442595608289511372737002747289000918490890348427274974048191519920465764390375545135758012114725748919054090278301411984618992540495713579651863596237507088245986545812744303577221229325542826403404963549061226157338788752488880206716490499348493732185647347936375766054825175543257545331077010497328719450203901452547531512637567611587435903881933071324129 1

And the resulting ciphertext is

r =
132391654622962474731982860849733838649963782531365577404295217191440638318217291777993993059508484290101645915425785626840239462410963011397468731251853378852497076591684939427893138197446673356190457539005062403691360866238595002360183450872862597694471899480203805714337374549191512025712202384019026853161731606526647288148027302868627165018561304748615978345573944100642206602629510864733812138720010018120892451772953239990876222010981246839010647002364348987632103680665656200499849037733793890465089178057067570745991739129755372613419062137634094792764218316189702023122560914855495553609909376159512328382 68

t=
292804521838291345744368414401211925215870260693565040608272290904375022206353595256844539924654933868653590583751336136243235231042791561700988951279870088942537729008874317926727314888524372408731094089634092103263182629330181819613100772281181361875213936491276626431504569387444413295751608457690371799121512592173469502709116752877243676613486304301791824830771168126440524765508203739226544881266746681692174908432802975076549111297801617207794471157538456044898580916779988297691216582487002579072845977500868596507107914364394458854975733936699279861653817076680172669373971736606326598544835667466271193204

Can you find the message?

4. (**10 pts**) We encrypted two messages m_1 and m_2 using ElGamal encryption algorithm given in "ElGamal.py", however, it contains a flaw and we lost m_2.

q = 136717678766217461388598745958821987937222095350 7
p =
123673323891242867177514096819580236465497274063458984325241055844011153224145054858864521371774649105018822971116982650715941117881035552815726337827743610567394319322651354827890393410346287094867949471330649611353144350280202803994699252851189649371736656279618955988650904092455232405972031238033017555179
g =
404621221188721819453789213524870729395169520186627356860554846147313773002412967932741940153306778745619138622570323493191953104542885799318977831726993684766176439355920665365929642039967646173934977630301024546455538895898585081558636590082441384507341030343306537048864152036452181985020914107487403516 45

$(message_1, r_1, t_1) =$ (b'I am gonna make him an offer he cannot refuse',
358136611273315274693584000616023624685848840559475235423036227215724990866631249621977597325701282995385041655594717994202301561513613556576877688425582079130001730700815455797827567591161809324984483902594026320025069040427553697239132503463193974619246951654470863274166006598766699823647495403008436129 70,
436272242181157972282892494759210329070342204604923561118955039360828544181682504164970729413474719947437279732546423255854535840817407799063602998440418026542203284314119516166793469955598432642546280685016132929594545821185568056663476917337149220315896141651073798837755969447986536317058117950892417711 23)

$(message_2, r_2, t_2) =$ (b'??????????????????????????????????????',
358136611273315274693584000616023624685848840559475235423036227215724990866631249621977597325701282995385041655594717994202301561513613556576877688425582079130001730700815455797827567591161809324984483902594026320

025069040427553697239132503463193974619246951654470863274166006598766699823647495403008436129 70,
595681361927820114080092427206809055456490929597996245063065651974043047750494240739299340959992508520364723319385101864709165711242868753806360664717325176286420842837300958204263948269554791374473141738781633562102557234485444228380751512557992129821766632830152987864931995770891520063152940834400761128 8)

Can you recover m_2 using the given settings? If yes, demonstrate your work.

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

q = 21050461915163064005698472752818467960484664222419461240422905587329
p =
166350012684247702483624960208789827948559730427271231102762182213631152485321216569269984953244228842019005469082472994225298670202577953059156033991197841238179775514759387498436307470908055513031878231189178701070612920135277680943020983237055039783199518335674190378326362190994721425403906793160843412530207970726743867508316581615914136438986707880504091973526325187625121048274055764701312958323181996487958617151547860175051687922070906846393742742561234022198080875417223026819462689334848242864117748000000448732499418289657445948173579373881036361075512135200183361431071999478635811912302431943039717281 93
g =
409316209796607490968734702020982808779338767387886312652237001531251351938414278090418335087913649957799748693058456436137378789213031021601841984495312049017317816796121062649530217902210296634640163356870134181395034601787974591435423590515009773320665843668567792787586385542473707813385608228801244487451564165725038895882068989096564645832341913994604332904195872831552155924295687686237830596535191981309628250139994665077358186000006346169433722976057429006410973538262567497727108655658411548500539156450645011181472707759988651178689172865670421964261741363906444092172729294764685946664316791030772843827 6
public key - beta =
333000142450304432593219767613612726668939199509191301567259738108836319544363643621410059050899531964348842712542468652394804413725920233890764670393459854801740832318401792488664896286474431696492411302218619895035696390789148524777716364767139044719785911738347323622486681245728141874710965952911008281827763121075208786284121607574116788990548808966942318570402085239628131043375095357374570562932388371209766986764420837217747742747867891059071710943307144176311820220665601335449187640557818334013047927239416642628563428978520210195537616338295809154700900641908071622503662814730863927266990367936210491765 1

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

(message$_1$, r_1, s_1) = (b"Asking questions during the lectures helps you understand Crypto",
26044485576050631880584159036418931121126749840345760793824044079 5,
15045429964567421250403275656320025283600046882519690784113588548158)
(message$_2$, r_2, s_2) = (b"Keep your friends close, but your enemies closer",
26044485576050631880584159036418931121126749840345760793824044079 5,
14016151436550334193141059702675072658308100333231844563375725796770)

Can you find the private key?

6. (**20 pts**) Consider the DSA scheme implemented in the file "DSA.py". The public parameters and public key are:

q = 127492866524845675045925547614226832022201099194 3
p =
943990828777386403563448350936338517422268109465480581675941066095993041014833761986016286446455789786658677433715162135495590175092700137852628251248881697386920885609199950755091463798028663470213532995799528071257894680233195234170310305952701353038911199408595154445665408603348158204290113449877398812 7
g =
747576130488870932097416342282284259029485722229656838929667828296542988007917890848613567043463712449212019388188808996473489749254519534502793005145946428963437513890858384665833844529025644779811271175052595853039388714362413277142446891539715423985000585155992329222006061717884272148739864644415164232 73
public key - beta =
939107882201222226424848385395795545007452184709686653345968136954694488623502385773843818710242429818437743578915453942050048457634393242225073275980083797933646389625186320359798816290641392473648855423990861417005712739958850161542890723995498494698202457193803447680684163348805080276741437359544426199 7

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

(message$_1$, r_1, s_1) = (message1 = b'Erkay hoca wish that you did learn a lot in the Cryptography course',
780456265196245442017019073827244628033034896446,
21415418947154624496513920216012504530287434837 7)
(message$_2$, r_2, s_2) = (b'Who will win the 2021 F1 championship, Max or Lewis?',
92729414271524120562335078065987936862296521576 7,
15111064221429655851794373090156142679228091058 9)

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