Homework #3

Due date: 23 November 2022

Notes:

- For Question 5, you can use a Python module for arithmetic in $GF(2^8)$.
- You are expected to submit your answer document as well as the Python codes you used.
- Do not submit .ipynb files, **only .py** scripts will be considered. You can work on Colab but please, submit a python file in the end.
- Zip 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_hw03_yourname.zip"**

1. **(20 pts)** You are in a job interview, and you were given the following RSA parameters:

N=
1416037683198508354923469195261580608875448276931386397247261210496042689722352846577135547157937440663064541821004645700622259511456561023077139964207300893906457134116339104019289505366774623919874531383660835786168835297334589671489349344591259966030302377108646344424587724466258847549085213970957665207134704078245202042057177305078113263352853462184264627234758734493076295111590640973006655811091565860290408615235644087016520791178107631915978279465803652469243626727820108772050549172075512900186626664370820817920955813360599193968787528992938952756946761821880802277361809574943299495509900213806177455809264693505392276872986759634456035485402433369306384831556884327401667763423594326565387016840348375482877866033127357137496266936772314585940742043708721187317938666956572295027470668470956482591717684609947017584858771235926201145369128985619340624559880510942526528247192759868317773179509963649942878396210326534638486338064556378395711649055109537193994195358884142190120801820883323198096325277854987035903498660740189507374962110953073927408462645712678690906258741835784287809356201807656048159929559000248143127239879504133444449707805721481828407639971809740271487838163125308653795894538384651553882687678853128876682587974889064511761353358599470464645604774224983230355228056059895044293527504098166480239685717261770946779751194318388543290206086641657571197777229722339185163614341216688042008669622434626766192686457787107435111257323828329771769312021242856650711094552667613878481978803823983929528878835168776879100815551466023144434324843402036377253148159517407651165480612956126181980588235768436395081199323291305539088674790557155643860281233927180922511100770198266540819485972423337385644180616762951461378511440507844682387542080550807661368510806629242816615902930158600419458047547336975109536082993954414O3

C=
696400013480214749035042136565033624961425061086594186450756053097306582936301374258672056455276148537211622174062717212912387652326593555921239814807828333848385588437439080830381566506168724225159624620021697301

54915953964183119114980126908766330985368848309500450339408734729613619
44443087691593107828862821646570032433642983137001852284845829650930647
06223012406708943135942308034577714781652449936804193376038948881234784
23874543403519874324974507113200783002944486585486851600539535628386531
41801302708015723513455630185668900982623299071334073972206896315094654
23370485258248363051965708675838699603428210849598364780048917355790608
65524567162160429678280461454911199062400526031135684121111239955162812
89156205173721081926194326415619996714900950517971362831818945544753717
88753487916467074480831971721615478110660543494213591753799041395668218
40066652595424137210482881658501927325492040686183535597528410306227932
11811546498498416331840835131156082144938676253315415796019426308810557
10772521613539736496538502897482744276266996184051595286258970763465218
55833152050437833779268709820967663507879956765807906582220904644347215
01852708698323447101795557300364436799578803807748130751797841330640652
95040609126587134691678763590715741842451613877119115706937778499202286
16110285482656647558683620463689959080278744496130657075835781539867329
28925323015151853338913936042198087258685083559185030411159644723900086
245429733121433845958021799786535702133974611668238336

e = 2^4+1

You are asked to retrieve the plaintext “M” using only these given parameters, the plaintext is a 289-bit number and M << N (which means that M is too small than N). Show your work.

2. (**25 pts**) Alice encrypts the private factors of the modulus using her public key. In order to increase security, she multiplies them with a random integer *k* (a process called blinding). Namely, she performs the following operations:

$c_p = (kp)^e \bmod n$ and $c_q = (kq)^e \bmod n$,

where:
n =
73512973335060030081468282098336397584185840115543849725566479428982375
03037718695637830609047406561083968328616312232137592331991101538410373
74050058977286914202841273524024666932250131821113376668815406723621016
49097977282533745476584222793770179097317864458142937324148557309714017
33550710145929124769939738763589008485016472082711182204941268594892277
94428441209121779400263427064528800044153594488608641089810344852811969
04768332375221604708731576259326563612846635064979826680730540182601947
65233835743738656549160897983547411689218105035219148455858391611808212
37530439632455643648718893847945602507596343025039916399868955525098511
95516884533439960978043849158219089300897385142338631732809146184196710
17933875694012909890988102141890827802411783985398632856594216125677645
60272257730862326168958364031351936876932865703535175597367552038394861
00365686967751078058559507545057056861806132432140658073493899816919764
70235866303540467163203311155468259406351404248680181614400433190033519

00977331129699978699948137567053016132949764669941675005253460547841482
62517489203107663821626294529576908407126832705381935224215695242840900
51656775595480488299992596031724073766519823034866191167042526714139900
26284868294201399181512043
e = 65537

a. (**10 pts**) Explain why this is not secure as anyone who obtains c_p or c_q can factor n.

b. (**10 pts**) Factor *n* assuming

c_p =
591585661179913908292670964497762439966067617632980318594933696993454
439728774695673398116279819100780981035776904828340000286556561821690
365937795767595578460020372779099849512212679341203336712606691026980
784826758612415894186878559418781933556513172143678225888892413399407
357572102187313811291121304928928848821740914239643286098598943861830
177312939795889893467307582185899931290683200944769576636703683519903
565708396786917226998954212438208550716658336081133840312436400158765
503544052687056167181841283624622810770703450396047554977193850903901
459806510612970231465678503924692665502872536500180880428444652182260
677241127037607744405052492742074626162505690986835662409717998630436
892609085891138004339602274790886677062920378513594058185353516270486
167595057731308599477294190461449818277150803142015005367485865449897
543408948648598768730426058899226212982674326317239306732825540115470
519539729209054092010428475270982133241739755670905018636775489391024
836583829979747436647847343118308007690487133027090643538988141634837
671142574166059246803646505989258216228280398544311438792645908159200
659488745926355055587928710053702116250352855728835609501615125001498
450195370742144016227930442188681846248774022591908284395554
c_q =
309411729251034251013395318719711706160811892862394890384299153591046
151499917121473269348428595368341405902181095462992704329677547559777
979434641542836057927150208356368122522229704600457038948359531454800
468151991865398538394737136726928964793436886497125779317949812565735
742165451557954880576255424560841686766527870866834024733826068596435
927601933868356893971916253250737903827232908056159513671751051011444
556858915046438167636879877207093812917634178160773097343436768913865
449127177896443942872076146021042760369486915412609972874901144604578
896888158193243455010603320402839406279239012820373216932302406828263
523610721896771030364051463816982602691512661678520212734151948259164
880771727519256923996762844291396810356605766709626469613406423888478
179326659617558931774201308469110659393420333748745481493594465373505
620910145316327984388627313690258503451934907257473375836215663070727
820311136494801887688477512058462477531334799901621825205074964487431
978969359431368816027376768643419649233036868068405017555516631026700
926970709260767677416518151837286394000000084320497258684909083576218
584531666178186869180867507209104151471590432167256468837661447181958
281610404744566801864938311314315544802024261994226882434417

and decrypt the following ciphertext
c_m =
410549325685366799284798358668331291049211237703447489321746464354090
480951066742200309912524869634695769764148842963115769175527179769353
501136785040366302701836605559269222653839509896223034315989474752525
422032311356077141993653194048634554747064603617789881601934904375748
720866525993835837380575119329831703593942313434602973137707915466599
022298281760026130881260276637144571320608488322550304670024203670226
128476075072338279764675774256431888736850559740563151465682499215963
629918224101757068211315022934434916282345252224165558403932898552995
551505591198395331625457165173901358070740812251352212374898226752191
458989544999183050305396079595170576363831967410596235893058231852520
802407515041647925479992058184613103289709752618729186509661786057001
647083804585301703801272797236111687348907977621232083642431914363895
266519463459112358271172160273751026080398345460929751476512586349360
9055882467516252224080678692961596943056437462453579785432258654674275
520068227719175943507656108554707552928863240951458193510571735005767
226590398204791730466993980158501690130104327829589781448607958503898
902133445000532521723138230966914652700465728532524467483612403840506
027543892612439531633794667013713638856396514285792898762678

3. (**20 pts**) Consider the combining function given in the following table, that is used to combine the outputs of four **maximum-length** LFSR sequences:

   $F(x_1, x_2, x_3, x_4) = x_1 \oplus x_1x_2 \oplus x_2x_3 \oplus x_2x_3x_4 \oplus x_1x_2x_3x_4$.

   a. (**5 pts**) The lengths of LFSRs are 60, 95, and 97, and 75 respectively. Compute the linear complexity and the period of the output sequence.

   b. (**15 pts**) Analyze the function F in terms of three criteria:
   - Nonlinearity degree
   - Balance
   - Correlation

   Is this a good combining function? Explain your answer.

4. (**15 pts**) We challenge you to get the plaintext of a ciphertext C that was calculated using an RSA setting, however, we lost the decryption keys, we only have the following:

   N = 15220196297956469159
   C = 6092243189299681137
   e = 2^16+1

   (RSA Encryption: m^e mod N | Decryption: C^d mod N)
   Can you retrieve the message using only this information? If yes, show how.
   - You are not allowed to use external tools (including online tools).

5. (**20 pts**) Consider GF(2^8) used in AES with the irreducible polynomial $p(x) = x^8+x^4+x^3+x^2+1$. You are expected to query the server using *get_poly()* function which will send you two binary polynomials a(x) and b(x) in GF(2^8). Polynomials are expressed as bit strings of their coefficients. For example, p(x) is expressed as '100011111'. You can use the Python code "**client.py**" given in the assignment package to communicate with the server.

   a. (**10 pts**) You are expected to perform $c(x) = a(x)\times b(x)$ in GF(2^8) and return c(x) as bit string using *check_mult()* function.

   b. (**10 pts**) You are expected to compute the multiplicative inverse of a(x) in GF(2^8) and return $a^{-1}(x)$ using *check_inv()* function.