CS641A: Modern Cryptology
Assignment 6

objectStrongly

Dewansh Singh(**170241**)

Nikhil Naidu(**170758**)

Parv jain (**170464**)

Problem statement -

c(x) =
58851190819355714547275899558441715663746139847246075619270745338657007055698378740637742775361768899700888858087050662614318305443064448898026503556757610342938490741361643696285051867260278567896991927351964557374977619644763633229896668511752432222528159214013173319855645351619393871433455550581741643299

n =
84364443735725034864402554533826279174703893439763343343863260342756678609216895093779263028809246505955647572176682669445270008816481771701417554768871285020442403001649254405058303439906229201909599348669565697534331652019516409514800265887388539283381053937433496994442146419682027649079704982600857517093

e = 5

c(x) = x^e(mod n)

find x ?

Solution Proposal

- We first noticed that the value of e is small.
- We were not completely aware of the secret message i.e. we had a certain clue with a good probability of what might have been the secret padding.
- Thus the RSA mode we had was a bit relaxed in terms that we knew with certain probability its first few starting bits.
- So we tried breaking the cipher using Coppersmith attack.

Coppersmith’s Theorem[1]

Given an integer N and a monic polynomial F of degree d over integers , set X = N ^(1/d ϵ) for 1/d > ϵ > 0 then we can find all the x < X such that f(x) ≡ 0(mod N).

- By knowing the padding, our RSA model can be resolved into “f(x) = $(m+x)^5$ - c” In which the known part of the message is ‘m’ and the unknown part is ‘x’.
- If x< $N^{1/5}$, we will find the required password as the root of the polynomial $(m+x)^5$ ≡ c (mod N).
- To solve this above equation we made us of the code available on this Github Repository[2]

Working with the code[3]

- Coppersmith’s LLL attack states that the length of unknown part of the message has to be maximum of $N^{1/e}$. As the length of N is 308 decimal length (equivalent to 1024 bits in binary) and e=5, length of ‘x’ should be less than one-fifth of 1024 i.e; around 200 bits. So we tried the attack for each possible length of ‘x’.
- At first we guessed a padding text which we converted into binary using the ascii_to_bin utility and now we solve for the value of ‘x’ using as6_RSA.py provided in the assignment package.
- We tried with several paddings but none of them returned a value for x. After a few more tries, we found a padding that returned some value of x and that padding was :

  “This door has RSA encryption with exponent 5 and the password is ”

- This padding returned the length of unknown message ‘x’ to be 72; and gave the value of ‘x’ to be: 2147562143725930046825

  Binary form of x is: 1110100 01101011 01101001 01100111 01110010 01100100 01110010 01100101 01101001

  ASCII form of x is: “tkigrdrei”

  Password to the next round : **tkigrdrei**

[1] Source https://en.wikipedia.org/wiki/Coppersmith%27s_attack
[2] Source (https://github.com/mimoo/RSA-and-LLL-attacks/blob/master/coppersmith.sage)
[3] To run the code install sageMath and run the as6_RSA.py on it.