CS641: THE GREAT CAVES

DEDSEC

Guntas Singh Brar 180274

Divyanshu Bhardwaj 180253

Devanshu Gupta 180233

May 17, 2020

Chapter 6

The Cipher Text

- In this assignment we have to break RSA encryption to obtain password from the ciphertext posted on moodle.
- The RSA scheme has
- **n = 84364443735725034864402554533826279174703893439763343343863260342756678609216895093779263028809246505955647572176682669445270008816481771701417554768871285020442403001649254405058303439906229201909599348669565697534331652019516409514800265887388539283381053937433496994442146419682027649079704982600857517093**
- This door has RSA encryption with exponent 5 and the password is **5885119081935571454727589955844171566374613984724607561927074533865700705569837874063774277536176889970088885808705066261431830544306444889802650355675761034293849074136164369628505186726027856789699192735196455737497761964476363322989666851175243222252815921401317331985564535161939387143345555058174164329 9**

Cracking The Cipher Text

- If M is the plain-text, C is ciphertext, e is public exponent and d is private exponent then we have $M^e = CmodN$ and $C^d = MmodN$.
- As the public exponent is 5, we use coppersmith attack which is one of the low public exponent attack.
  Coppersmith Algorithm
  The Coppersmith method, proposed by Don Coppersmith, is a method to find small integer zeroes of univariate or bivariate polynomials modulo a given integer.
  The coppersmith theorem basically gives us a method to find efficiently all roots $r < N^{1/\delta}$ of polynomial equation $f(x) = 0modN$.
- Now we will formulate the RSA problem as $f(x) = (M + x)^e \bmod N$. If x is smaller than $N^{1/e}$ then we will find the root which will be our password.
- We used a code available on GitHub(see references) and modified the code in following way:
  - N and e are known to us. We get rid of the second part of code because that is irrelevant to us.
  - Now we start with different paddings M and converted them to M_binary.
  - The length of password will be multiple of 8 as it is converted from ASCII to binary and less than $N^{1/e}$ as required by the algorithm which will be less than 200 bits in this case.
  - Hence our polynomial becomes $f(x) = ((M\_binary << length\_x) + x)^e - C \bmod N$.

*

  - The solution of this polynomial which will be our required password can be obtained by the coppersmith code we used from the github.
  - To be able to try different paddings, we modified the code so that coppersmith could be called as a function with parameters.
  - We also added the code to convert the password obtained in binary back to ascii characters.
- We tried some different possible paddings that may be used in the password. One of the most obvious choice appeared the text used in the assignment :
  This door has RSA encryption with exponent 5 and the password is
- We also tried different combination of whitespace characters at the end of padding and got correct hit(verified from tutor) with a single space.
- The password we got was:

"tkigrdrei"

Attachments

The following files are attached:

- **coppersmith.sage -** Runs the LLL to find the required password.
- **coppersmith.sage.py -** The above Sage file compiled into python code.

References

[1] Lecture 4: Coppersmith, Crytography
`https://web.eecs.umich.edu/~cpeikert/lic13/lec04.pdf`

[2] Implemented algorithm on github
`https://github.com/mimoo/RSA-and-LLL-attacks/blob/master/coppersmith.sage`