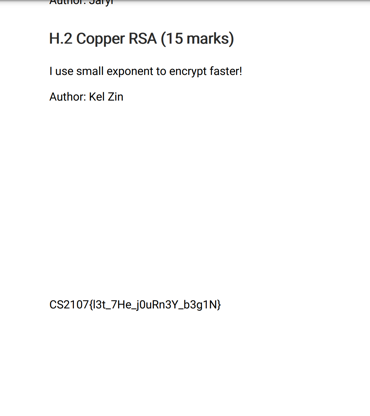
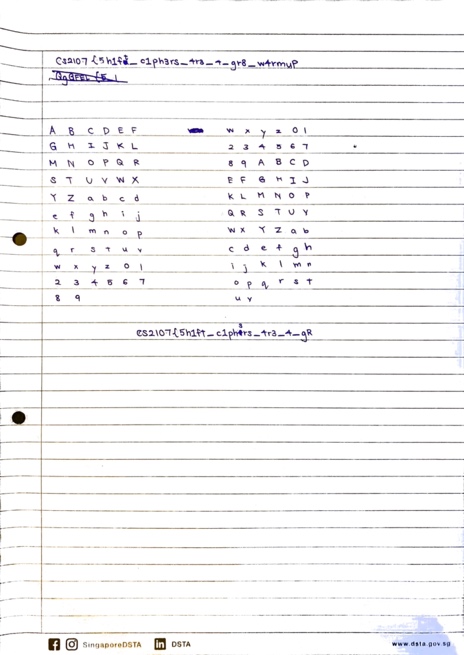
E.1 Sanity Check

I searched through the CS2107 Assignment 1.pdf until I found the flag on Pg. 4



E.2 Something’s Off

I manually calculated the shift, which was -14, created the shifted table from the original table, then did substitution to find the flag.



E.3 MAC

After using the help command to understand different openssl commands. I used the command

*‘openssl dgst -hmac CS21072022 text.txt’* to generate the MAC.

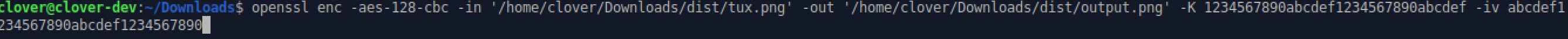


E.4 Secret Penguin

After using the help command to understand the different openssl commands. I used the following command format

*‘openssl enc -aes-138-cbc -in {input\_filename} -out {output\_filename} -K 1234567890abcdef -iv abcdef1234567890abcdef1234567890’*

to output the AES-encrypted file.



Then I used the command format

*‘openssl dgst -r {output\_filename}*

to read the SHA-256 digest of the outputted AES-encrypted file.

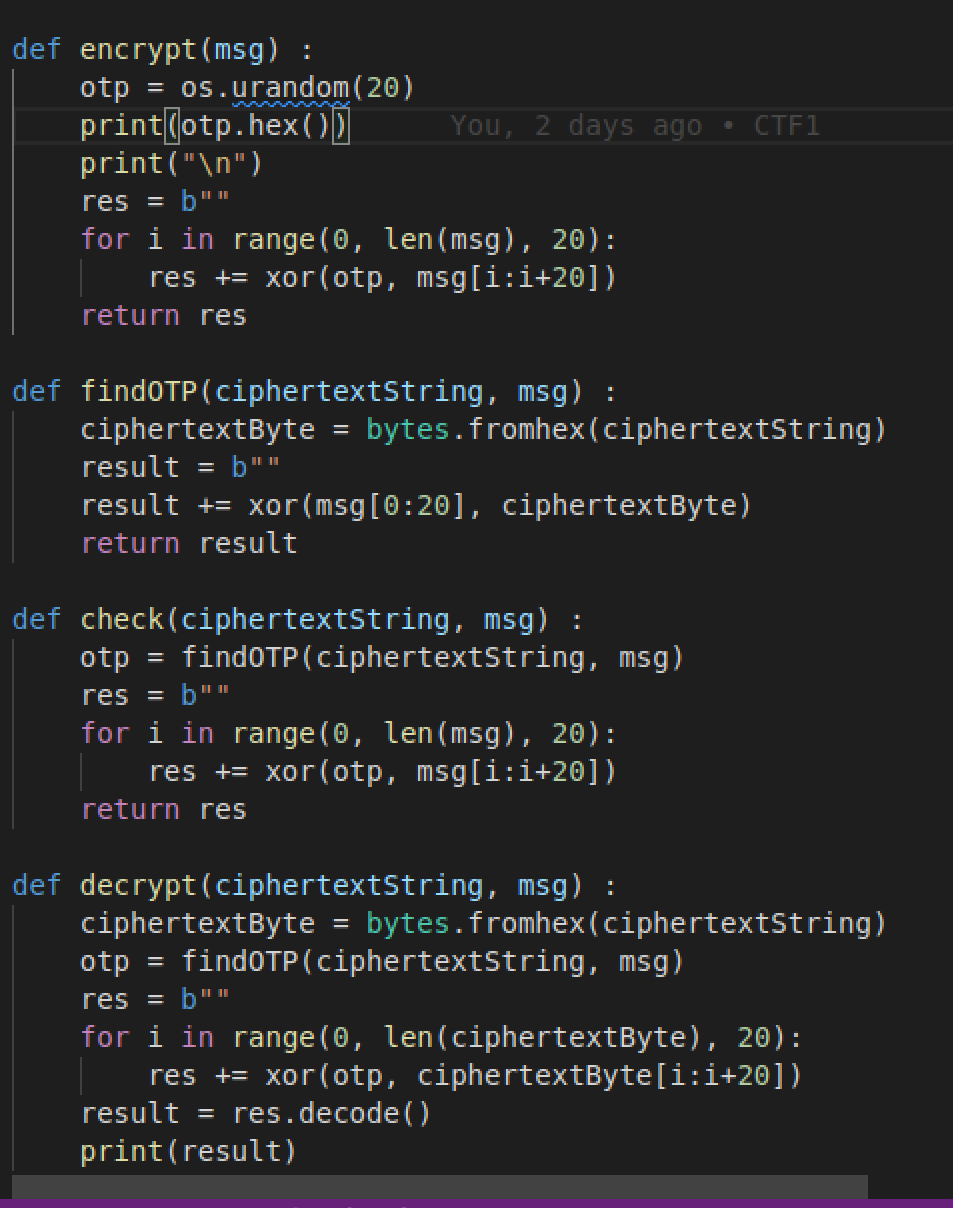


E.5 Prime Time

I used the RSA online calculator tool found at *decode.fr* to solve for N and subsequently decode the RSA.

M.1 Insecure OTP

1. I used the knowledge that the OTP is sufficiently short, being only 20 bytes long while the original plaintext is > 20 bytes long, to determine the OTP is insecure.
2. I re-generated the OTP by xor-ing the first 20 bytes of the message with the ciphertext.
3. I then xor-ed the ciphertext with the 20-byte OTP repeatedly, 20 bytes at a time, until I obtained the original plaintext.

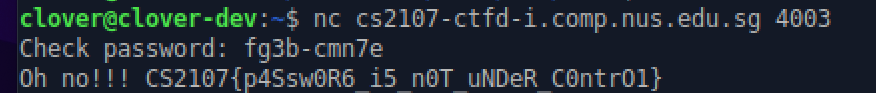


M.2 Public Password

I followed the hints in the question to find Grandma Susan’oo’s password on a picture posted on her twitter

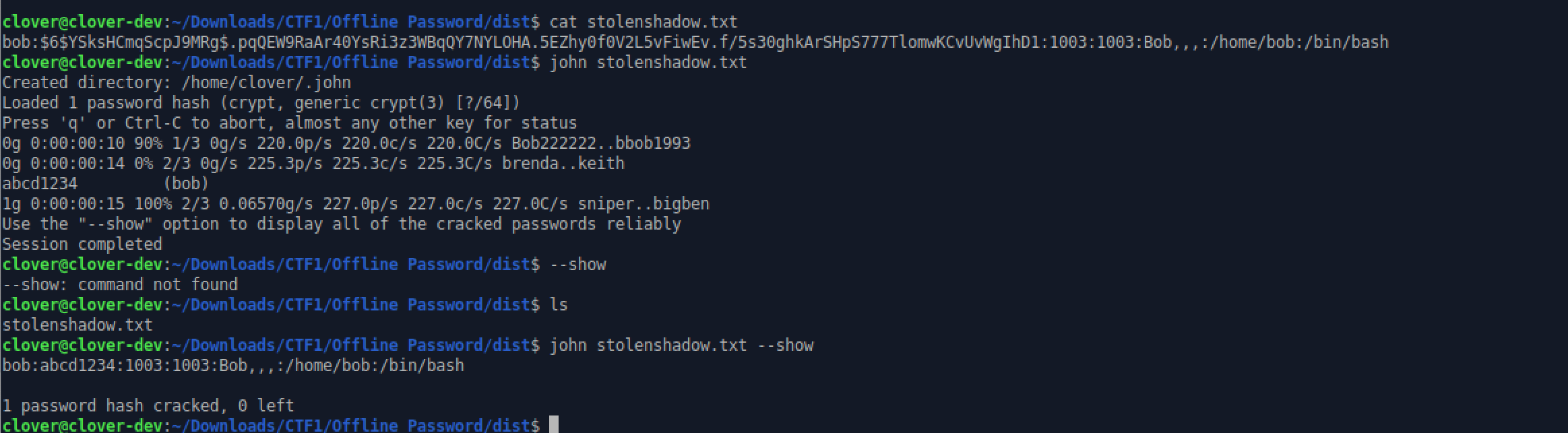


then I checked her password against the server to find the flag.



M.3 Offline Password Cracking

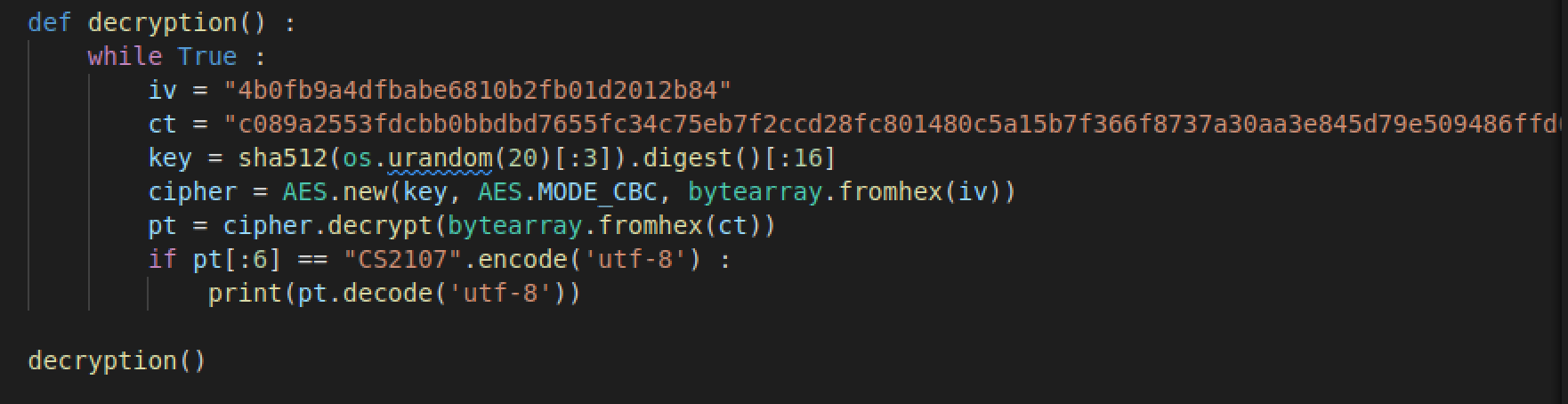
I downloaded the John the Ripper software, followed its instructions on the usage and cracked the password.



M.5 Perfect AES, Imperfect Key

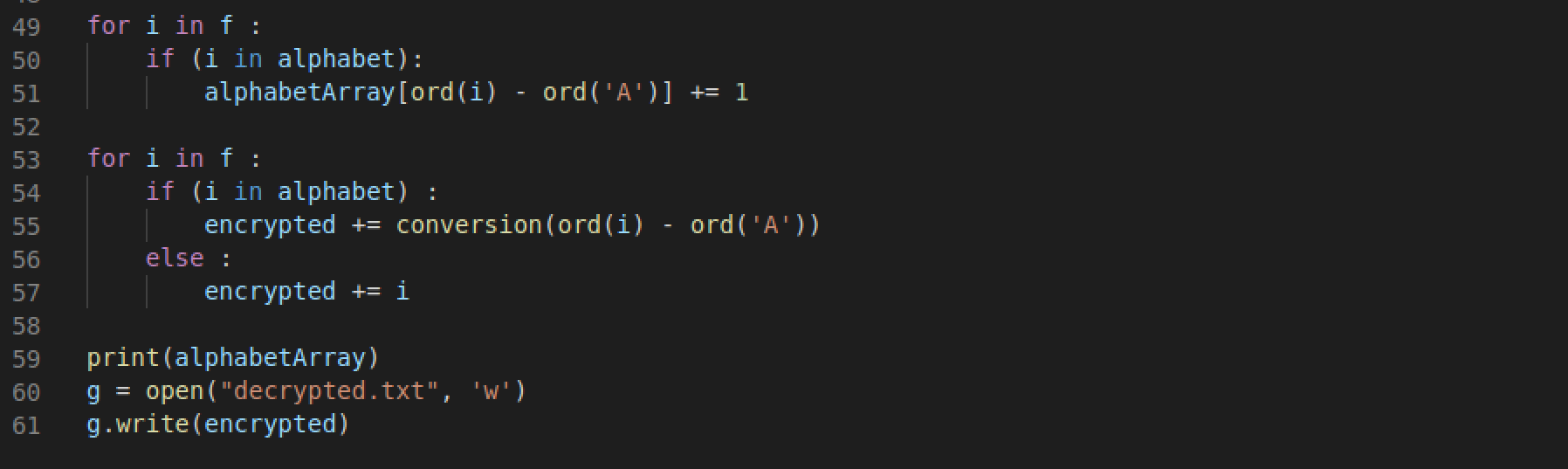
I made use of the fact that the first 6 bytes of the plaintext is definitely “CS2107”.

1. I generated a SHA-512 key and a cipher using the randomly generated key and the known IV
2. I used the cipher to decrypt the ciphertext to see if it matches the plaintext.
3. Using a while loop, I repeat steps 1-2 until the the obtained plaintext has the first 6 bits “CS2107”.



M.6 Substitution Cipher

1. I first obtained iterated through the entire ciphertext to obtain the frequencies of every alphabet in the ciphertext.
2. I then compared it to the frequency of alphabets table on the internet and used it to make my first guess
3. I then made substitutions by using known facts such as the last line must start with “CS2107” and that bullet points are in the format “I, II, III, IV”, and sub-bullet points are in the “A, B, C, D” format to derive the rest of the plaintext.



H.2 Copper RSA

1. I used the fact that the exponent is a low number (3) to implement the Chinese Remainder Theorem, obtaining a value X which is the numerical value of the original plaintext, after going through the quadratic equation, to the power of 3.
2. I found the quadrated value of the original plaintext by taking the cube root of X
3. I then used binary search to solve the quadratic equation and find the original plaintext, in numerical form.
4. I then converted the plaintext to byte form, and then to a string, to find the original plaintext.

