Week: 10 Date: 11/4/2021

15-110 Recitation Week 10

Reminders

- HW5 due Monday 11/8 @ Noon EDT

Overview

- Security Review
- Meme Cipher Encryption
- Reading from Text Files: Code Writing
- Working with Helper Functions: Code Writing

Problems

Security Review

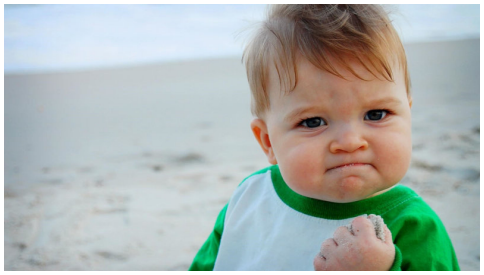
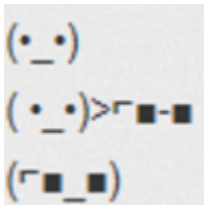
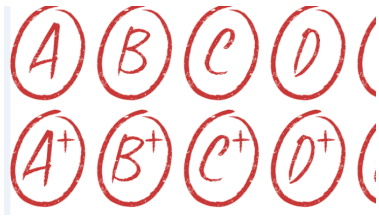
Describe the two main types of authentication.

What makes RSA nearly impossible to break?

Match the descriptions below to the corresponding types of security attack:

Every student in 15-110 goes to gradescope at the same time to check their Test 4 grades Answer: One malicious student connects to a class wifi access point and looks at the packets for their roommate's andrew ID and password to send prank emails from their accounts Answer:

Meme Cipher Encryption

Carnegie Mellon	
surprised	
I	
good	
cool	
grade	

Encrypt:

Carnegie Mellon is cool.

Decrypt:

What is the plaintext?

What is the ciphertext?

Is this a symmetric or asymmetric encryption algorithm?

How many keys are used?

What is the key?

What is the runtime to break this cipher? Keep in mind that an adversary knows each meme corresponds to a word, but they don't know which words are being used in the message. This means they would have to check each possibility in the dictionary. For this question, assume there are N words in the dictionary and 6 memes that are used.

Reading from Text Files: Code Writing

You have been provided with a text file called tas.txt that contains all of the names of TAs for 15-110. Using this text file and the starter file provided, first practice reading in the file as demonstrated in lecture. You can do this at the top level in your editor.

Then, write the function countTwoANames that takes in a text file as a parameter and returns the number of TAs in 15-110 who have **at least 2 a's** (upper or lowercase) in their name.

Working with Helper Functions: Code Writing

1) Write a function avg(L) that takes in a list and returns an average of the elements in the list.
2) Write a function cleanUpValues(L) that takes in a list and converts all strings to integers and all None values to 0
3) Write a function createAvgDict(d) that takes in a dictionary and returns a **new** dictionary that maps the student name to their average score. You should be calling the helper functions from steps 1 and 2 above.
4) Write the function getGradesAboveB(d) that takes in a dictionary and returns a list of students who have average grades of 80 or above. You should be using the cleaned dictionary by calling the helper function from step 3.