**Blockchain Technology**

**19BCE248**

**Practical 2**

**AIM:** To create a blockchain and implement replay attacks on blockchain.

**Code:**

import hashlib

import time

import json

import threading

import random

class Block:

    def \_\_init\_\_(self):

        self.chain=[]

    def addNewBlock(self,msg):

        data={}

        data["time"]=time.time()

        data["id"]=len(self.chain)

        data["message"]=msg

        if len(self.chain)==0:

            data["prevHash"]=0

        else:

            data["prevHash"]=self.chain[-1]["currHash"]

        str=json.dumps(data).encode()

        data["currHash"]=hashlib.sha256(str).hexdigest()

        self.chain.append(data)

    def validate\_blocks(self):

        while True:

            ind=-1

            no\_blocks=len(self.chain)

            for i in range(1,no\_blocks):

                if(self.chain[i]["prevHash"]!=self.chain[i-1]["currHash"]):

                    print("Tampering Found!!")

                    ind=i

                    break

            if ind !=-1:

                for i in range(ind,no\_blocks):

                    self.chain[i]["prevHash"]=self.chain[i-1]["currHash"]

def validate():

    validate\_thread = threading.Thread(target=block.validate\_blocks, name="Validate Blocks")

    validate\_thread.start()

if \_\_name\_\_=="\_\_main\_\_":

    block=Block()

    for i in range(5):

        block.addNewBlock("My current count is"+str(i))

    # block.chain[2]["currHash"]='1'

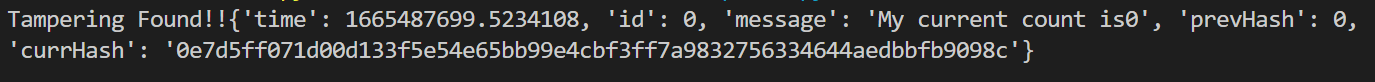
    validate()

# print(block.chain[2])

**Output:**

****

When a particular hash is tampered the value would be



**Explanation of Implementation:**

* Created a class named Block which will acts as a blueprint for each blocks in blockchain.
* It will contain fields such as:

1. Time: time at which transaction was performed
2. Id: Unique key for each block
3. Message: data of each node
4. Prevhash: hash of previous node
5. Currhash: hash of current node

* For validating hash, we will create a thread which will check whether the previous hash of current block and current hash of previous block are same or not.
* For the same I have created a separate function for the same.

**Learning:**

These practical taught me how to build a blockchain from the ground up. Also, how can we validate the blocks if they have been intentionally altered.