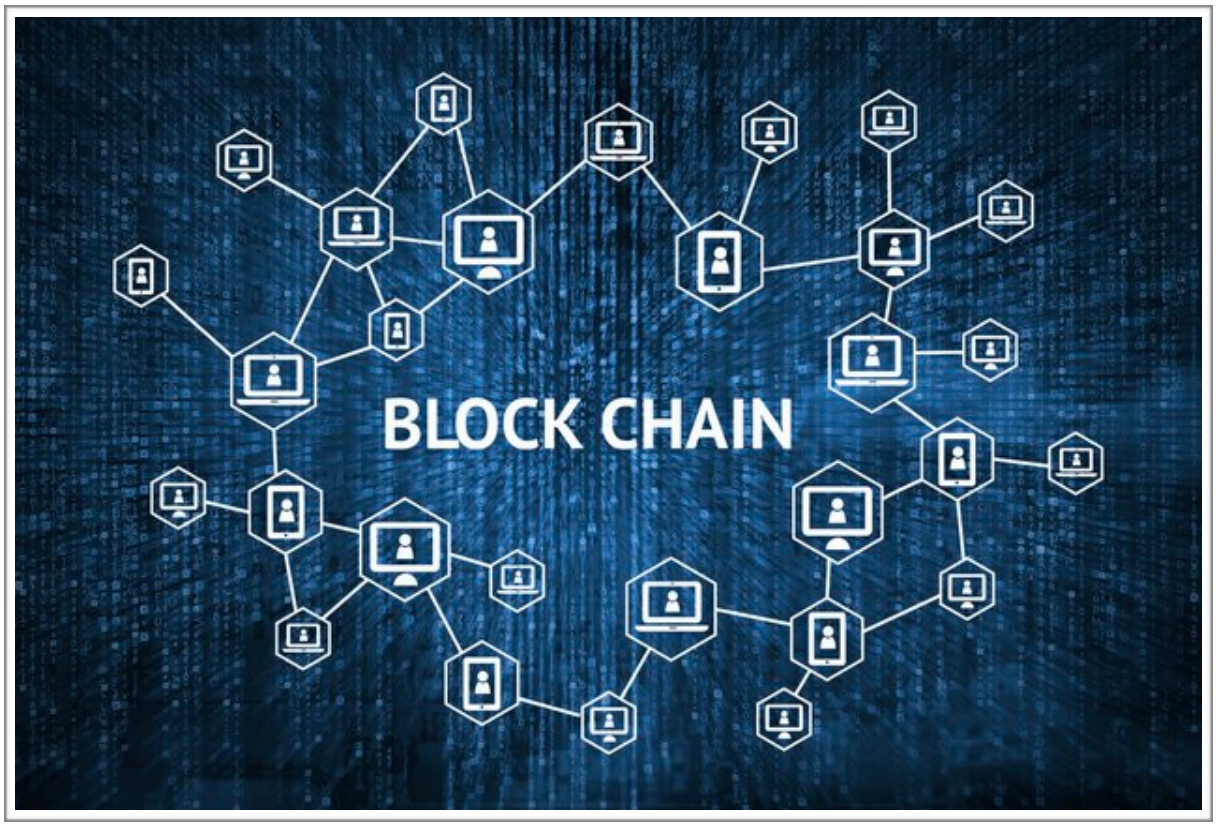
Blockchain

mathematical model in python 3



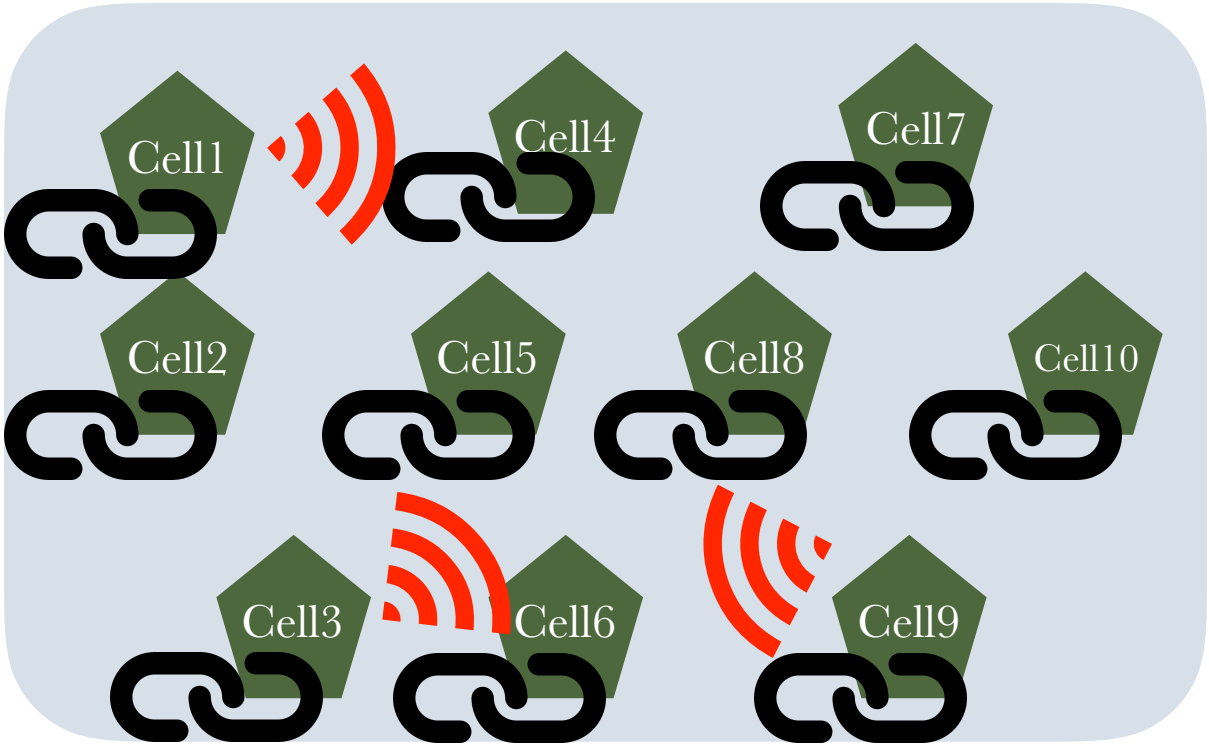
Liu, Yen Fu

11JAN2020

Model Environment Settings

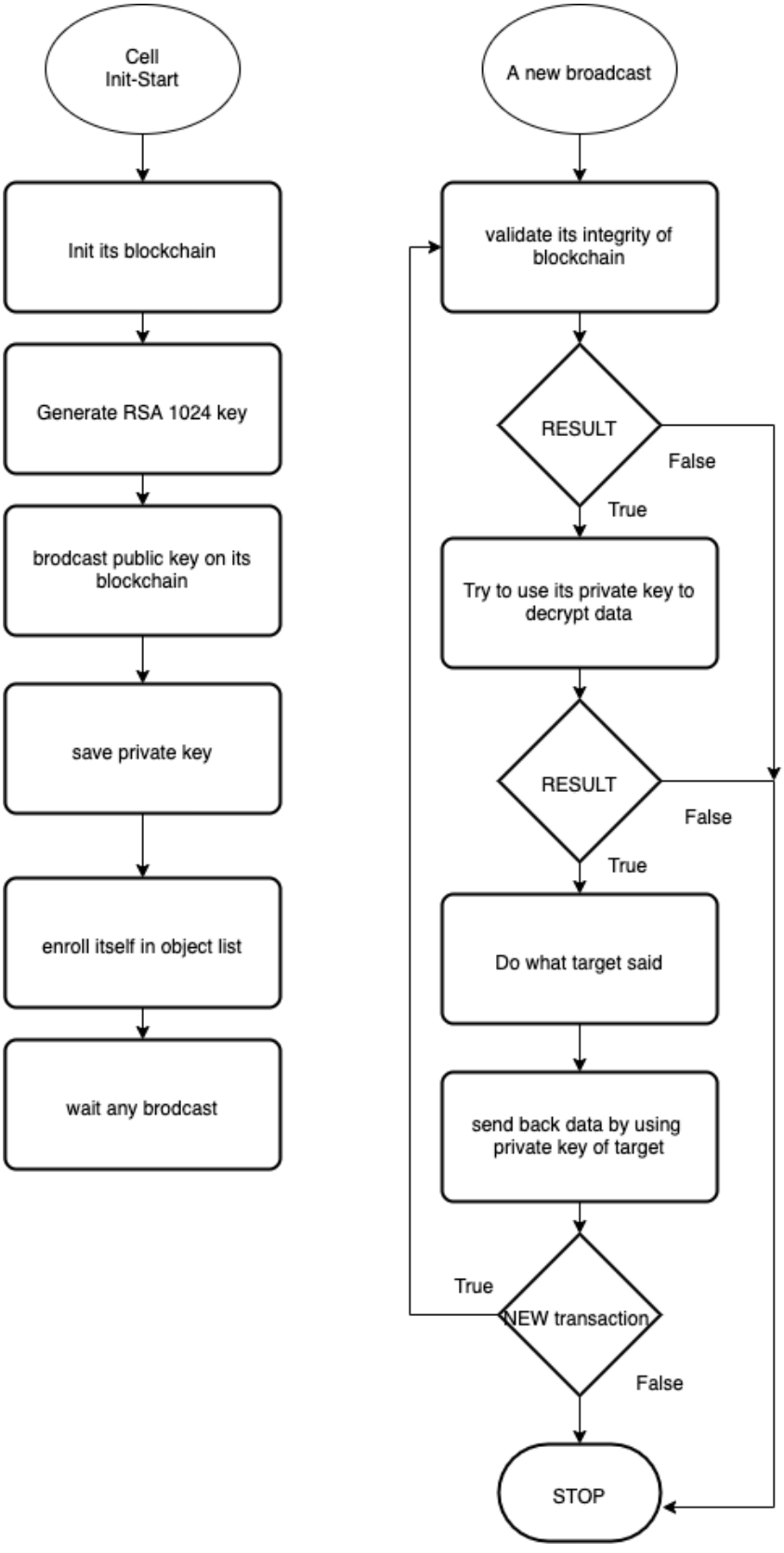
I set our number of objects= 4. They save their blockchain locally and broadcast every transaction to every cell.

*4 objects is for easy visual use. In real case, you can get as much as you want.



Every cell works independently.

Cell program diagram



Cell initialize

class MinimalChain():

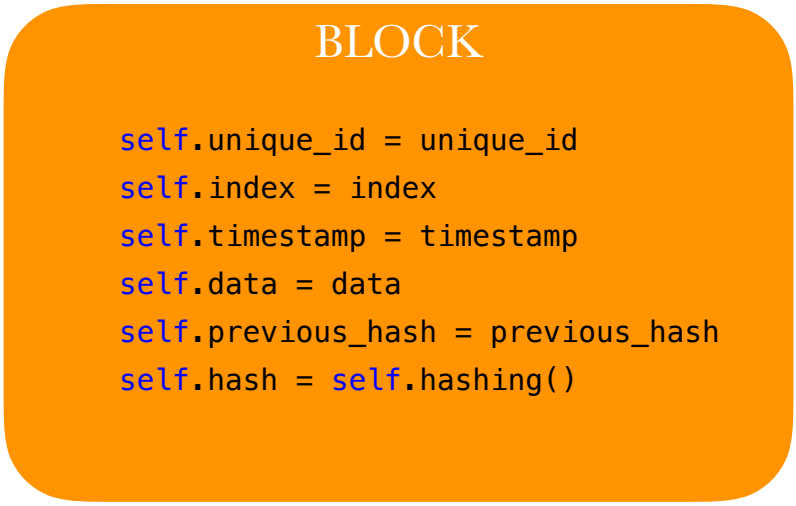
```
    def __init__(self): # initialize when creating a chain
        self.unique_id = id(self)
        self.random_generator = Random.new().read
        self.key = RSA.generate(1024, self.random_generator)
#generate public and private keys
        self.publickey = self.key.publickey()
#generate first block
        self.blocks = [self.get_genesis_block()]

    def get_genesis_block(self):
        return MinimalBlock(0,
                            self.unique_id,
                            datetime.datetime.utcnow(),
                            'This is my public key',
                            self.publickey)
```

…

Block in blockchain

```
class MinimalBlock():
    def __init__(self, unique_id,index, timestamp, data,
previous_hash):
        self.unique_id = unique_id
        self.index = index
        self.timestamp = timestamp
        self.data = data
        self.previous_hash = previous_hash
        self.hash = self.hashing()
```

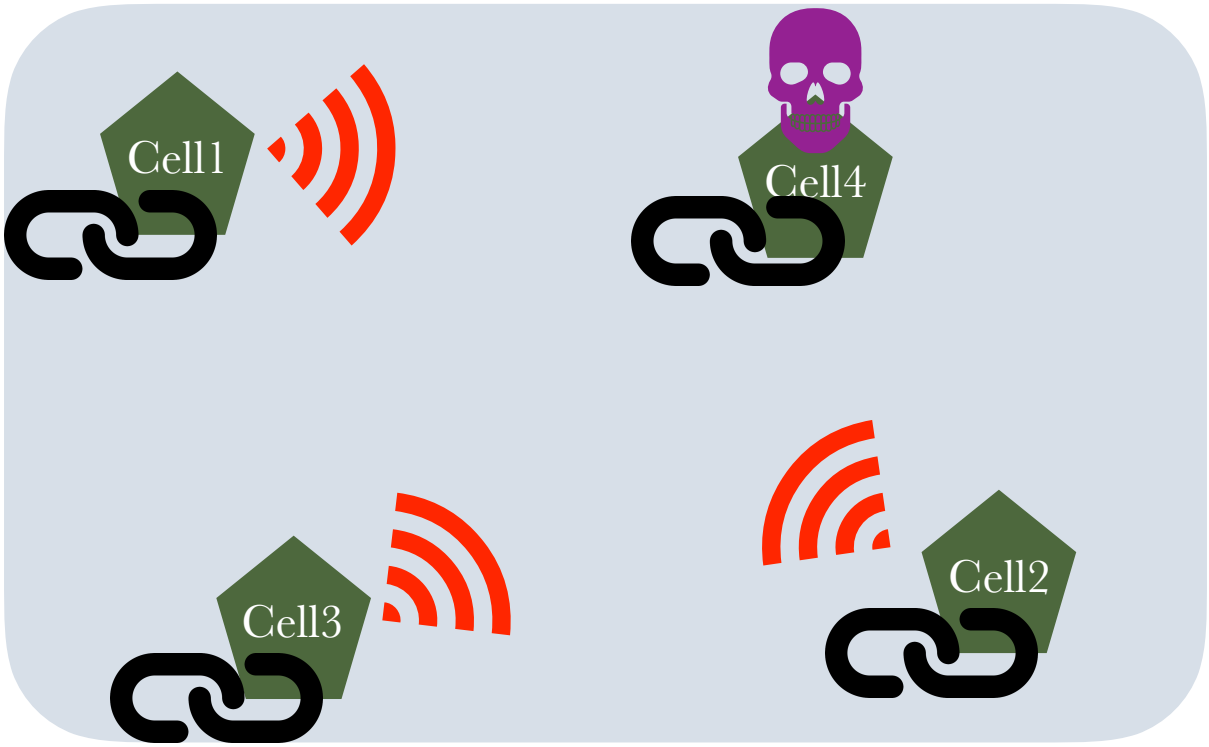
It saves python object id and index of transaction and timestamp and data and previous_hash and hash itself.



EObject class

```
import blockchain
class eobject():
    def __init__(self):
        #init
        self.chain= blockchain.MinimalChain()

    #job method - broadcast data and encrypt data by using
target public key
    def job(self,subject_list,target,data):
        self.subject_public_key =
subject_list[target].chain.get_public_key()
        if(self.subject_public_key !=
subject_list[target].chain.get_block(0)[1]):
            print(target,' is fake!')
        self.data =
self.subject_public_key.encrypt(data.encode('utf-8'),32)
        for i in subject_list:
            subject_list[i].chain.add_block(self.data)
```

A demo

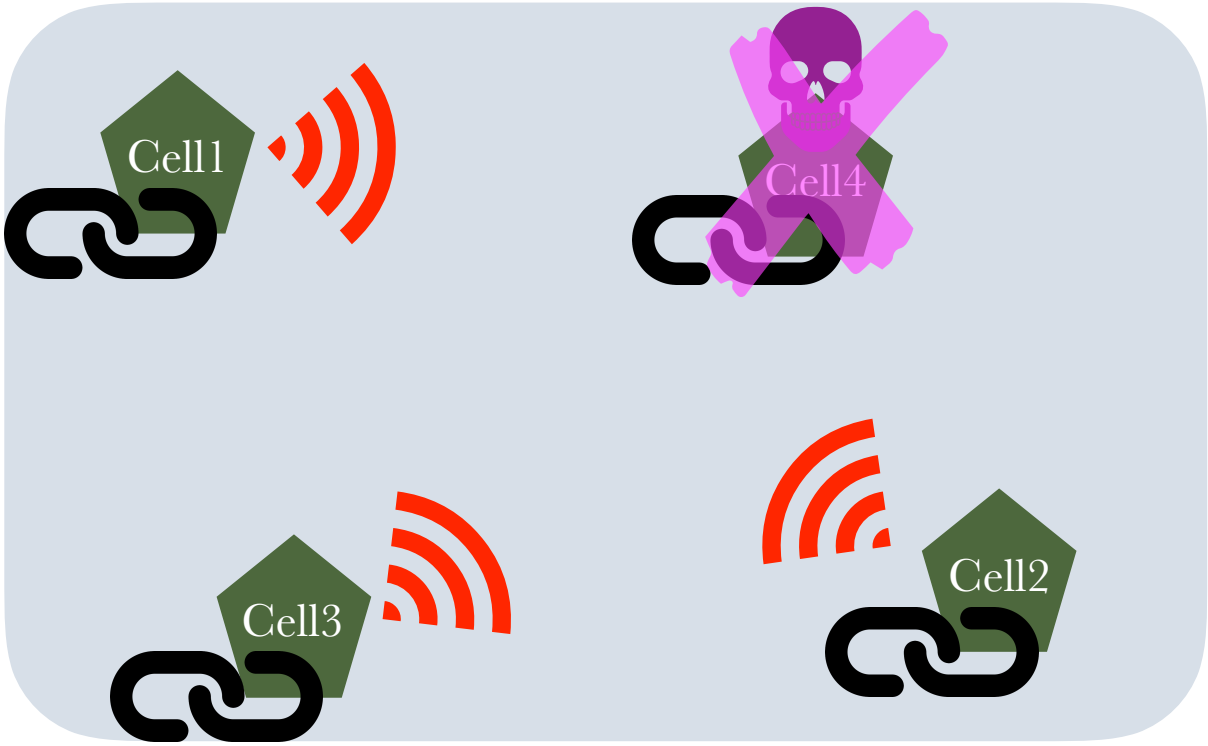



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sat Jan 11 21:02:42 2020

@author: ethan
"""

import hashlib
import copy
import datetime
import Crypto
from Crypto.PublicKey import RSA
from Crypto import Random
import eobject
#create a broadcast dictionary
```

```
my_eobject={}

#assign role
chair = eobject.eobject()
couch = eobject.eobject()
fridge = eobject.eobject()
#store itself in broadcast
my_eobject['chair'] = chair
my_eobject['couch'] = couch
my_eobject['fridge'] = fridge

#chair send a data to couch
sent_data = "I am chair. I am sending data to couch!"
chair.job(my_eobject,'couch',sent_data)
print('This is what chair wants to send: ',sent_data)
print('================================================')

#fridge want to read the data chair sent to couch
middle_attack = fridge.chain.decrypt(fridge.chain.get_block(1)
[0])
print('This is what fridge gets: ',middle_attack)
print('================================================')
#this is what couch get
result = couch.chain.decrypt(couch.chain.get_block(-1)
[0]).decode("utf-8")
print('This is what couch gets: ',result)
print('================================================')


#now we try to forged a fake block
fake_chair = eobject.eobject()
my_eobject['chair2'] = fake_chair

#we pretend send data to couch
sent_fake_data = 'I am chair. Hey! couch tell me the
passcode!'
fake_chair.job(my_eobject,'couch',sent_fake_data)
print('This is what fake chair wants to send:
',sent_fake_data)
print('================================================')
result_from_fake =
couch.chain.decrypt(couch.chain.get_block(-1)
[0]).decode("utf-8")
```

```
print('This is what couch gets: ',result_from_fake)
print('================================================')
#couch send password back with chair public key
passcode = 'I LOVE PARIS'
couch.job(my_eobject,'chair',passcode)
print('This is what couch sends: ',passcode)
print('================================================')
fake_chair_get_ans =
fake_chair.chain.decrypt(fake_chair.chain.get_block(-1)[0])
print('This is what fake chair gets:',fake_chair_get_ans)
print('================================================')
#only real chair can read passcode using its private key
chair_get_ans = chair.chain.decrypt(chair.chain.get_block(-1)
[0])
print('Only chair can see this:
',chair_get_ans.decode('utf-8'))
print('================================================')

#IF I want to know my blockchain is orignal not forgery
print('Is my data legit?',chair.chain.verify())

#I try to modify a block
chair.chain.add_fake_block('My bank account got 5000NT')
print('After add a customed block, is my data
legit?',chair.chain.verify())
```

Result




```
In [180]: runfile('/Users/ethan/.spyder-py3/untitled0.py', wdir='/Users/ethan/.spyder-py3')
Reloaded modules: eobject, blockchain
This is what chair wants to send:  I am chair. I am sending data to couch!
================================================
This is what fridge gets:  b'CVnV\xbc\x0f\x98il\xb4\xf7Z\xf6\xd2\x18<\xa9\x9a~\x19\xc0\xfa\x13*\xc6e
\xd3\x05\xf3\xe9c\x00\x8f\x08\xe1\x80\xb1s@\xc7\x0bB4\x9f1\x13\xa7[\x99\\E\xbf\x98<\xfc#g/,\x80B7\x11\x93a
\x98\xeaBs\x0f\x128\xdd\xa5\x0fi?\xd4^\xa0\xcc\xdf\xd7\xcc\x03D\x91\xe5\xb7\xaf\xce\xf9x\xb6\x07\x8a\xfb
\x8c\x1a\xd1-\x98:\xdc\xa4dn\xa9E\x80\xb6k\xf0d\xc8\xce0\xc8\xb2\x05\xa0\xf0\xabn\xcbC\x7f\x1c'
================================================
This is what couch gets:  I am chair. I am sending data to couch!
================================================
This is what fake chair wants to send:  I am chair. Hey! couch tell me the passcode!
================================================
This is what couch gets:  I am chair. Hey! couch tell me the passcode!
================================================
This is what couch sends:  I LOVE PARIS
================================================
This is what fake chair gets: b'k\xe1\xcaKZ\r\xdd\xbc&\xe9\xba\x90\xcaI\xe2\xc7\xdf\n:\x08-\x84\x93\x87f?
\x12Q\x0fj\\\xbe\r\xd6\xc6+;yoM\xb8\x11\xf3\x02k\xc1\xbdK\xed\x12\x19P\xaf\x06\x8b\xb3\x06U\x03\xbeaj\x8c!
a!X\x03=W\x87~w\x92C\xb8\x11\x9d\x1f\xbb\xeabsh\xb8\x0e\x84\xbbr#3a\xb1\xf2\xdf\xb2\xfa\xad\xab\x1c\xb2\\
\xf2kT\x92\xa6M\n\xfe\x07\xd0\xc4;}\xa2|\x0e\xc1\xf6\xb5/R\x86#\xda\x13\xb9'
================================================
Only chair can see this:  I LOVE PARIS
================================================
Is my data legit? True
Wrong previous hash at block 4.
After add a customed block, is my data legit? False
```

Conclusion

This model represents our internet of object in blockchain. This model not only provides data integrity and data security. Any fake cell cannot modify transaction which has already done and cannot do middle-attack.

The model can apply to not only just object layer in network but also node layer in M2M communication.

Reference

Facebook

https://www.facebook.com/

Untitled24.ipynb - Colaboratory

https://colab.research.google.com/drive/1C8c-zOQZsvRQzjz23c46SPocjfxDOhRi#scrollTo=0Jth4owmMEI3

How to correct wrong informations on a blockchain? : CryptoTechnology

https://www.reddit.com/r/CryptoTechnology/comments/8zkers/how_to_correct_wrong_informations_on_a_blockchain/

What's inside a Block on the Blockchain?

https://learnmeabitcoin.com/beginners/blocks

Building a Minimal Blockchain in Python - Towards Data Science

https://towardsdatascience.com/building-a-minimal-blockchain-in-python-4f2e9934101d

8.17. copy — Shallow and deep copy operations — Python 2.7.17 documentation

https://docs.python.org/2/library/copy.html

blockchain - Google 搜尋

https://www.google.com/search?q=blockchain&sxsrf=ACYBGNTvAL4urBSm0suc8NfpMpt_wPRMnA:1578773466336&source=lnms&tbm=isch&sa=X&ved=2ahUKEwi05pC9rfzmAhWR3YUKHWNJCXQQ_AUoAXoECBMQAw&biw=1680&bih=844#imgrc=2EtlJACwyAx00M:

Creating a list of objects in Python - Stack Overflow
https://stackoverflow.com/questions/3182183/creating-a-list-of-objects-in-python

Blockchain: Everything You Need to Know
https://www.investopedia.com/terms/b/blockchain.asp

How secure is blockchain really? - MIT Technology Review
https://www.technologyreview.com/s/610836/how-secure-is-blockchain-really/

公開金鑰加密 - 維基百科，自由的百科全書
https://zh.wikipedia.org/wiki/%E5%85%AC%E5%BC%80%E5%AF%86%E9%92%A5%E5%8A%A0%E5%AF%86

Untitled22.ipynb - Colaboratory
https://colab.research.google.com/drive/1QO9xzHs4njCNUz2kRLZ9tn8WJExSHgZ4#scrollTo=2GOeL6WyGLIu

Error with encrypt message with RSA python - Stack Overflow
https://stackoverflow.com/questions/30512458/error-with-encrypt-message-with-rsa-python

python - Encrypt & Decrypt using PyCrypto AES 256 - Stack Overflow
https://stackoverflow.com/questions/12524994/encrypt-decrypt-using-pycrypto-aes-256

3. Data model — Python 2.7.17 documentation
https://docs.python.org/2/reference/datamodel.html

RSA Key Formats

https://www.cryptosys.net/pki/rsakeyformats.html

python - How do I use an external .py file? - Stack Overflow

https://stackoverflow.com/questions/3980059/how-do-i-use-an-external-py-file

python - Using pycrypto, how to import a RSA public key and use it to encrypt a string? - Stack Overflow

https://stackoverflow.com/questions/21327491/using-pycrypto-how-to-import-a-rsa-public-key-and-use-it-to-encrypt-a-string

Python Object as Dictionary Value - Stack Overflow

https://stackoverflow.com/questions/13368498/python-object-as-dictionary-value

python - Convert bytes to a string - Stack Overflow

https://stackoverflow.com/questions/606191/convert-bytes-to-a-string

forgery verb - Google 搜尋

https://www.google.com/search?sxsrf=ACYBGNTaqYnlvbq2ccsHpZeVWqbBwK9XOA%3A1578782291706&ei=U04aXoTjKsSKlwSJmYHwCg&q=forgery+verb&oq=forgery&gs_l=psy-ab.1.2.0i71l4.0.0..928...0.2..0.0.0.....0.....gws-wiz.d8VvoIXhTVY

Untitled Diagram.drawio - draw.io

https://www.draw.io/

enroll myself into - Google 搜尋

https://www.google.com/search?q=enroll+myself+into&oq=enroll+mys&aqs=chrome.2.69i57j0l7.3723j0j7&sourceid=chrome&ie=UTF-8