Understanding Blockchain Basics creating a simple **B**lock**C**hain in Python

1.5 Hours session*

S. Joshi
shivgan3@gmail.com

*Last 30 minutes for understanding the code / Audience can open the websites and play with the tools

Game 1

Lets calculate the mean salary of this session without anyone disclosing their salary.
Who will encrypt and who will decrypt?

Game 2

Please compute the hash after writing your name and rating.

Hash = Last hash + character count of your name +rating.

Example: Hash-past = 1; Joshi, 10 then Hash-current =1 + 5 +10 =15

Two Games for Today

Secure average - Sheet will come to you - Add your number, in this case a fictitious salary, that we want to calculate our average with.

Creating a blockchain on paper - Create a new hash using

1) Count of name string
2) Previous Hash
3) Rating

Joshi + 1 +10 = 16

Game Blockchain

Past Hash = 1
Name = Joshi
Rating = 10
HashCurrent = Past Hash + count of string (name) + rating = 1+5+10= 16
Past Hash = 16
Name = XXX
Rating = 1
HashCurrent= 16+3+1 = 19

Four parts of the session

Introduction - 5 minutes

1. Visually Look at the tool (10-15 minutes)*
2. Come back to the term (10-15 minutes)
3. Understanding Bitcoin Blockchain (5-10 minutes)*
4. Understanding a Blockchain using Python (25-35 minutes)*

* (Audience can open the page)

Part 1 - Looking at the online GUI Tools

Practice using the GUI Tool

Blockchain Demo | Hash | Block | Blockchain | Distributed | Tokens | Coinbase

Blockchain

3

37

012fa9b916eb9078f8d98a7864e697ae83

0b9015ce2a08b61216ba5a0778545bf4d

e

Block: # 4

Nonce: 35990

Data:

Prev: 0000b9015ce2a08b61216ba5a0778545bf4d

Hash: 0000ae8bbc96cf89c68be6e10a865cc47c6c48

Mine

Block: # 5

Nonce: 56265

Data:

Prev: 0000ae8bbc96cf89c68be6e10a865cc47c6c48

Hash: 0000e4b9052fd8aae92a8afda42e2ea0f17972

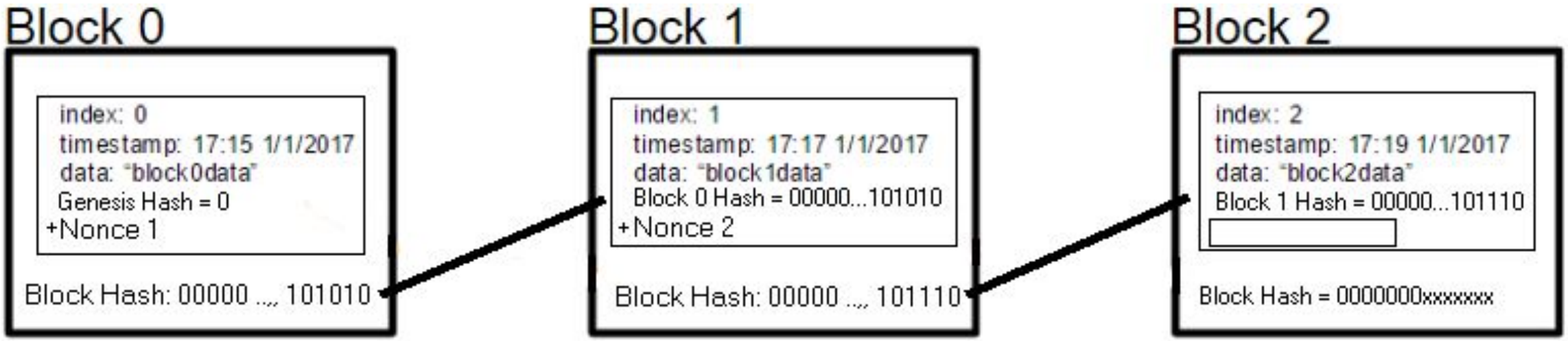
Mine

https://anders.com/blockchain/

Part 2 - Learning more about the terms

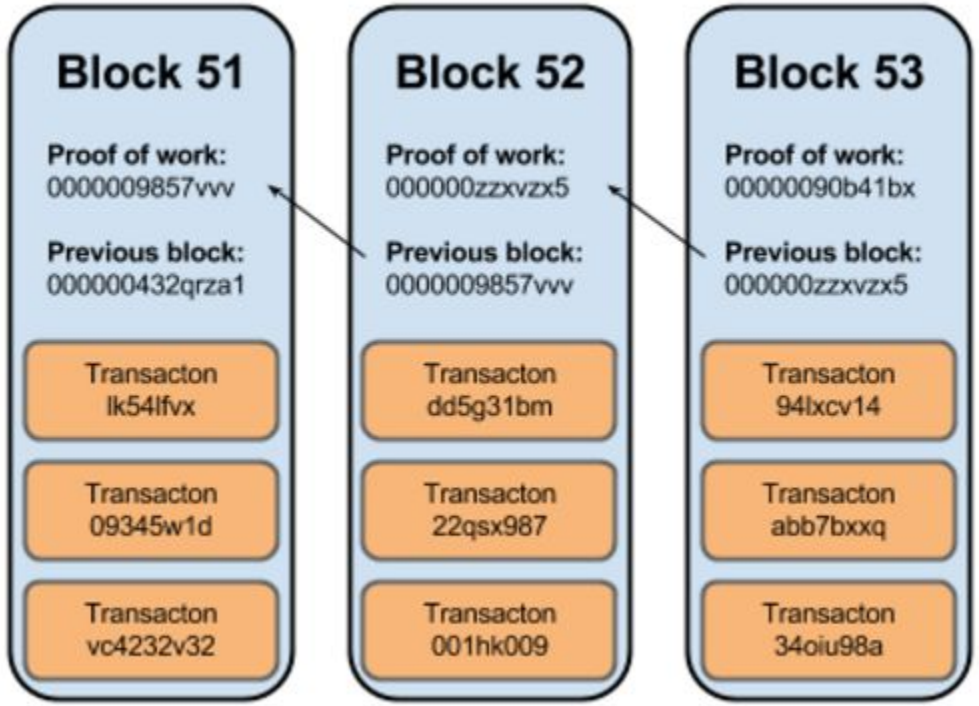
Components of the block in a Blockchain

Index
Timestamp
Data
Referring to last block
Hash



Sample Blockchain - Four things about the Block

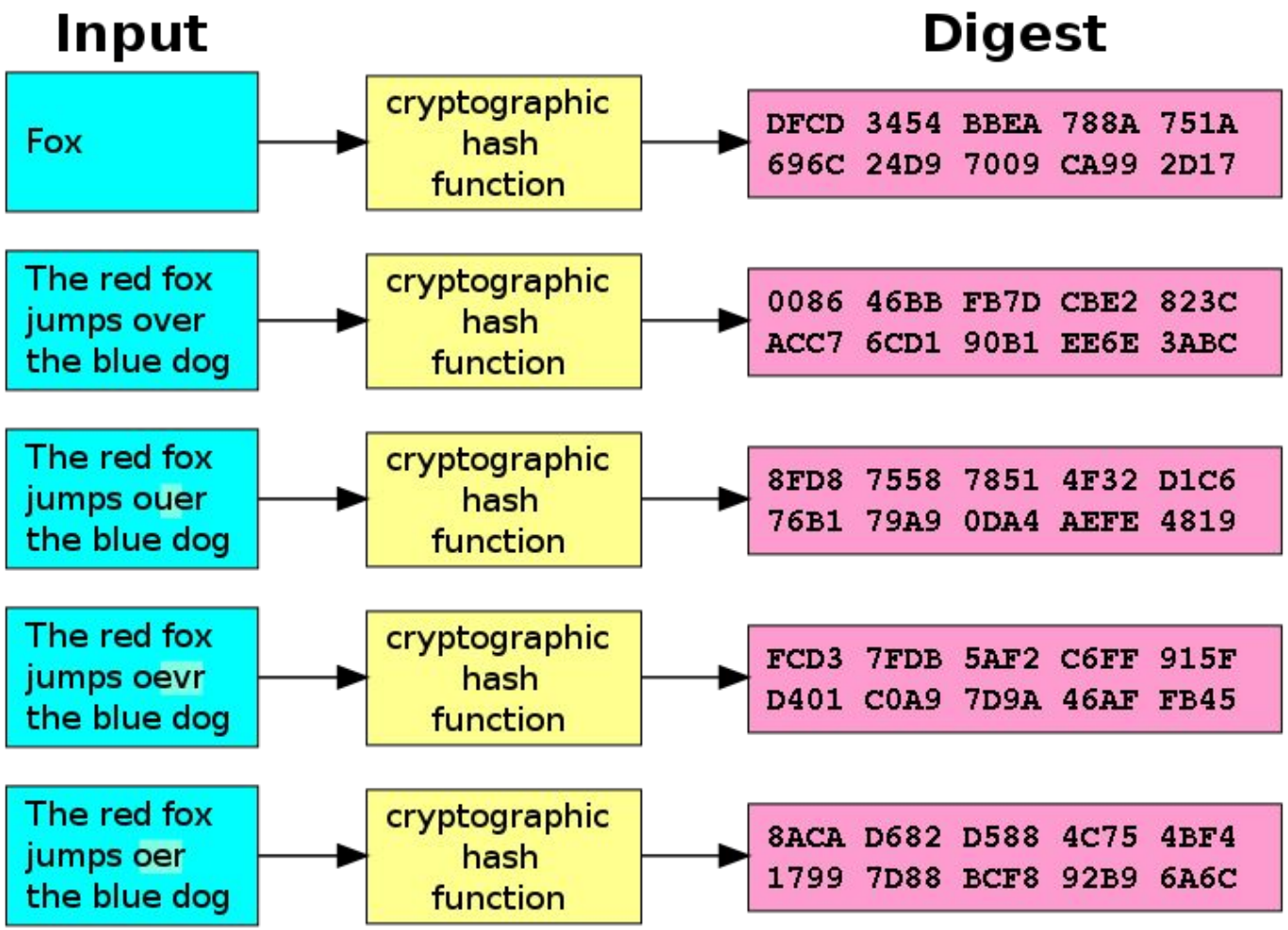
1. A block can have one or many transactions
2. Proof of work is the process to create Hash
3. Each hash is used to link to the previous block
4. Any change in value will require us to rehash it



https://qph.ec.quoracdn.net/main-qimg-83c9a9555372d25d2a6be9d0cb3369df

Hashing

Different Methods used in hashing



Mining

- The difficulty of mining a block is astounding. At the current difficulty, the chance of a hash succeeding is a bit less than one in 10^{19}.

Example in GUI Tool & Python

How does a ledger of bitcoin block looks like?

Bitcoin block doesn’t shows the public ip but a public address created by the private key which changes every time.

Which means we cannot find out who did the transaction

https://blockexplorer.com/block/0000000000000000000014b08ef581fc7031b9293698f1fea972295fcd561e25b

Part 3 - Applying what we learned to Bitcoins

Hash in a block of Bitcoin

BLOCKCHAIN info | Home | Charts | Stats | Markets | API | Wallet | Search | English

Transaction View information about a bitcoin transaction

ad41c723258c0f7da48aad01191c8c77015483c4fa700a87395ff041d2509eed

18e55pf66kGYsSMBcfAH4WfwcYfHSfMcYE	1BBxZrr6Qqf6gkrzTWEAgkETPZYMzEhVSq	0.0150049 BTC
16kKavdcbXwWS7WAzkrcg6VLJbVDG4Xzs9	13YcacVVyo5J7HEkiPtcJKa3pAxWAF2xpG	0.03345953 BTC

16 Confirmations | 0.04846443 BTC

Summary	
Size	373 (bytes)
Received Time	2016-11-18 17:00:00
Lock Time	Block: 439497
Included In Blocks	439546 (2016-11-18 17:06:20 + 6 minutes)
Confirmations	16 Confirmations
Relayed by IP	37.187.119.41 (whois)
Visualize	View Tree Chart

Inputs and Outputs	
Total Input	0.04865143 BTC
Total Output	0.04846443 BTC
Fees	0.000187 BTC
Estimated BTC Transacted	0.03345953 BTC
Scripts	Show scripts & coinbase

Part 4 - Python Implementation

```
import hashlib as hasher
class Block:
 def __init__(self, index, timestamp, data, previous_hash):
   self.index = index
   self.timestamp = timestamp
   self.data = data
   self.previous_hash = previous_hash
   self.hash = self.hash_block()
 def hash_block(self):
   sha = hasher.sha256()
   sha.update(str(self.index) +
              str(self.timestamp) +
              str(self.data) +
              str(self.previous_hash))
   return sha.hexdigest()
```

Public and private key

Anyone can decrypt but only you can encrypt

Public key encryption

Secure sum

Pen sign

Definitions

Dapp abbreviate as Decentralized Application used to Developed Applications using Front-end(HTML+CSS+JS) Web page + Back-end(Solidity Smart contract) Programming code + Server(TestRPC) Private Blockchain/Dummy Network in Ethereum platform

Links

https://www.ethereum.org/crowdsale
https://www.coinist.io/how-to-launch-an-ico-on-ethereum/
https://theethereum.wiki/w/index.php/ERC20_Token_Standard
https://remix.ethereum.org/#optimize=false&version=soljson-v0.4.20+commit.3155dd80.js
https://github.com/willitscale/learning-solidity
https://github.com/willitscale/learning-solidity/blob/master/tutorial-02/myfirstcontract.sol
https://medium.com/crypto-currently/lets-build-the-tiniest-blockchain-e70965a248b
https://theethereum.wiki/w/index.php/ERC20_Token_Standard
http://solidity.readthedocs.io/en/develop/introduction-to-smart-contracts.html