

A NOVEL FRAMEWORK TO TEACH HANDS-ON LABORATORY EXERCISES IN BLOCKCHAINS

Bertony Bornelus, Hongmei Chi and Hossain Shahriar

Outline

Background

- In recent years financial institution, social media giants such as Facebook, and medical institutions have all faced substantial amount of scrutiny for exposure of customers data
- With the emergent of Bitcoin, the first popular blockchain decentralized application; **blockchain has become a viable solution for recording transactions** in a growing list called blocks which are linked and protected using cryptography.

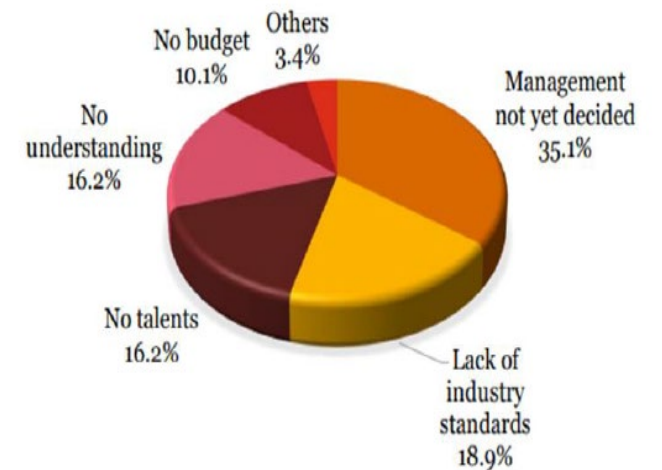


Basic Components in Blockchain

Reasons not to adopt blockchain

- Application of blockchain showed promises to various areas
 - Security traceability
 - Distributed data storage
 - Identity authentication
- Respondent of a survey believed that core features of blockchain technology are “tamper – resistance and distributed system”.
- Feedback from companies - why not applied blockchain technology refer to challenges
 - management has yet to decide to make layout in the field of blockchain
 - lack of industry standards and third, no talents

Survey conducted by PwC and VeChain



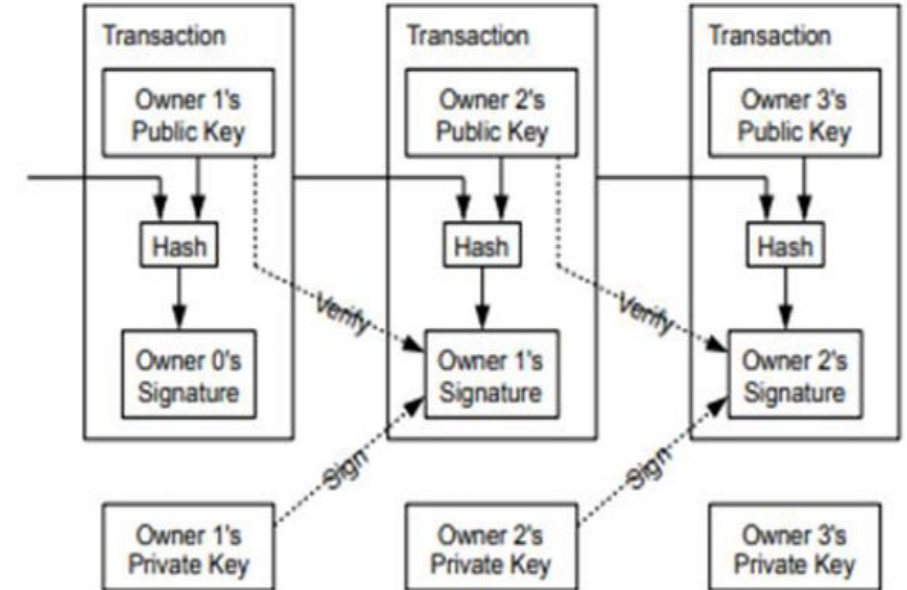
Reasons not applying blockchain technology

BLOCKCHAIN & APP

- Blockchain is an emerging technology originated from the distributed cryptocurrency Bitcoin - Satoshi Nakamoto in 2008
 - Peer to peer Electronic Cash System
- Bitcoin as an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.
 - Create transactions that are impossible to reverse - protect sellers and buyers from fraud

BLOCKCHAIN & APP - Transactions

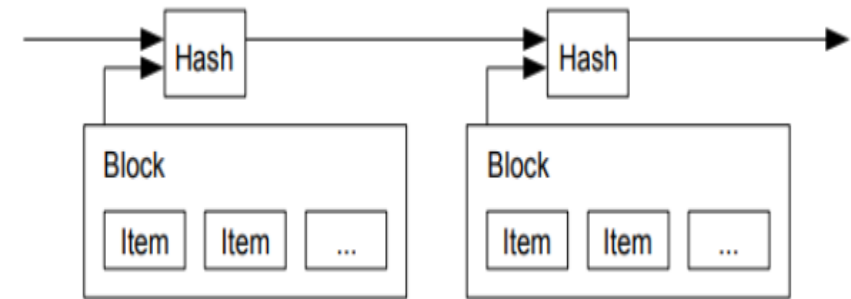
- Each bitcoin is an electronic chain of digital signatures, where **each owner transfers the cryptocurrency to the next by a digital signature of hash** for the pervious transaction and the public key of the next owner and added the information at the end of the currency.
- A payee can easily verify the signatures to verify the validity of the chain owner.
- To avoid the process of over spending or duplicate transaction, all transactions are included in the chain of the pervious block and are publicly announced to all participants to agree on a single history of the order of the block chain, therefore, a **time stamp** is needed to verify the order of the chain



Bitcoin Transactions

Timestamp Server

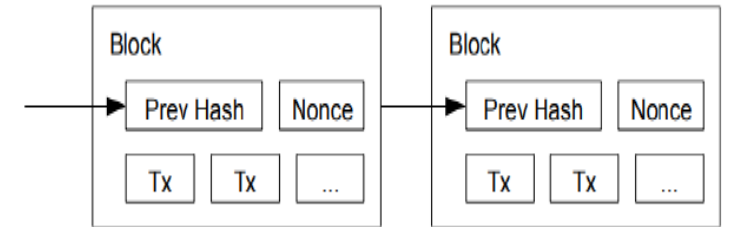
- The timestamp server proposed by Satoshi, to work by taking a hash block of items to be timestamped then **broadcast the hash block**
- Each timestamp is included in the previous timestamp hash, creating a chain, with each timestamp strengthening the block before it seen



Bitcoin timestamp

Proof -of -Work

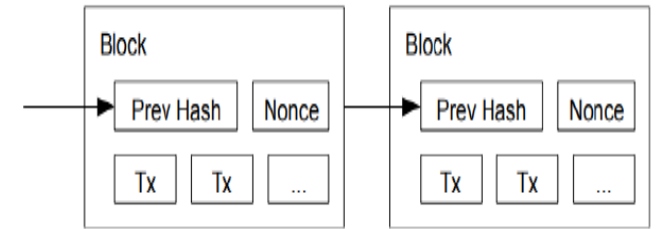
- The proof-of-work algorithm is the cornerstone of bitcoin technology, to create a tampered proof and fraud resistance block.
- Proof-of-work involves searching for a value that when hashed, such as with SHA-256 - the average work required is exponential in the number of zero bits required and can be verified by executing a single hash.
- For the timestamp network, bitcoin implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits.
- Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work.
- As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



Proof of Work

Proof -of -Work

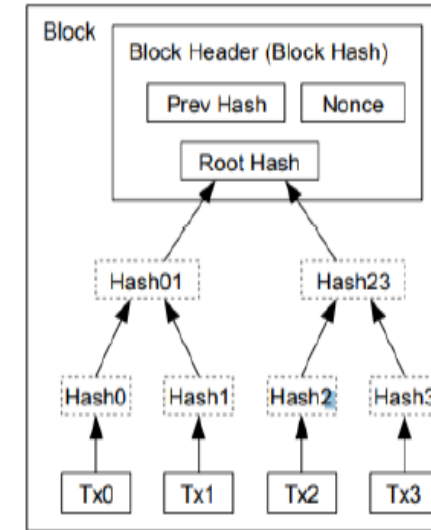
- For instance, given that two blocks exist for the same transaction block A - genuine and Block B- a tampered block, until the next the block is created both blocks are retained, once the next transactions have taken place it will be added to the growing genuine block and shortest block B will be disregard.
- Authenticating the vitality of the transaction (proof-of-work).
- Not every block makes it to every node on the peer to peer network, however, when the block reaches the nodes, the node will accept the longest block as the genuine.



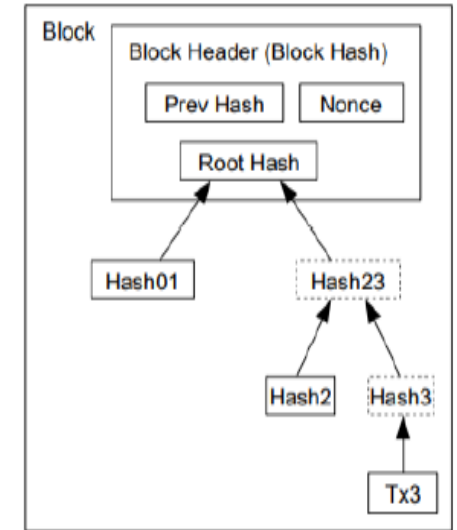
Proof of Work

Merkle tree

- Transactions are hashed in a Merkle tree.
- Once the genuine block grown enough block, the Merkle tree compresses disk space and not break the block chain.
- A node can create and propose a transaction, validate transactions, and undertake mining to support consensus and establish the integrity of the data.
- When nodes create transactions, these are signed by nodes using their private key to validate that these nodes are the true owners of the asset that they are transferring to someone else in the blockchain secured network.



Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

Merkle Tree

Smart Contracts

- Self- automated computer programs that can carry out the terms of any contract.
- Mostly based on objective conditions precedent. (if, then criteria).

```
if HAS_EVENT_X_HAPPENED() is true:
```

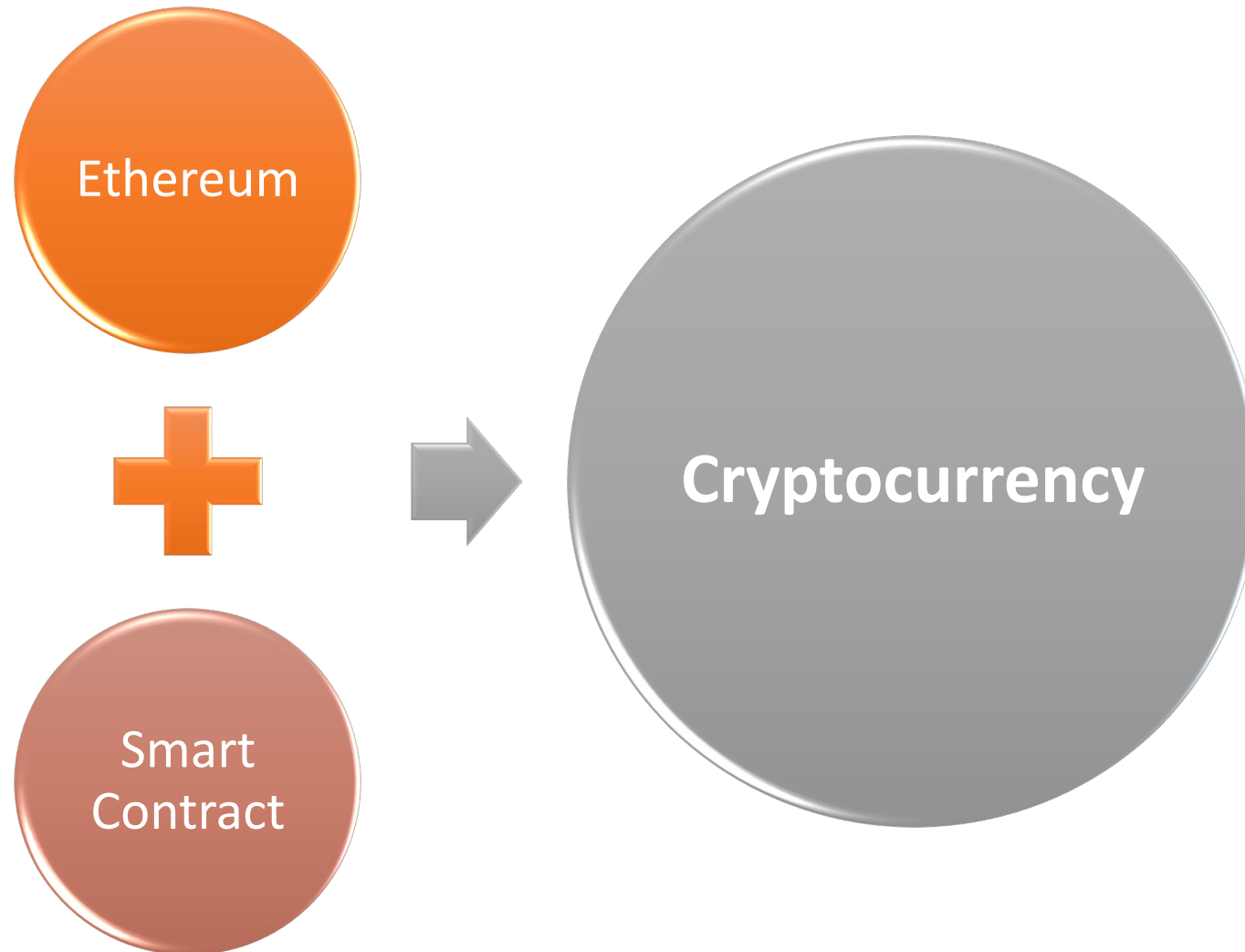
```
    send(party_A, 1000)
```

```
else:
```

```
    send(party_B, 1000)
```

- Computer program executed in a secure environment that directly controls digital assets

Process



Ethereum

- Blockchain with expressive programming language
 - Programming language makes it ideal for smart contracts
- Why?
 - Most public blockchains are cryptocurrencies
- Can only transfer coins between users
 - Smart contracts enable much more applications

How Ethereum Works

- Two types of account:
 - **Normal account** like in Bitcoin
 - has balance and address
 - **Smart Contract account**
 - like an object: containing (i) code, and (ii) private storage (key-value storage)
 - Code can
 - Send ETH to other accounts
 - Read/write storage
 - Call (ie. start execution in) other contracts

The “Hello World” of Ethereum

```
data domains[](owner, ip)
```

Private
Storage

```
def register(addr):
```

```
    if not self.domains[addr].owner:
```

```
        self.domains[addr].owner = msg.sender
```

Can be invoked by
other accounts

```
def set_ip(addr, ip):
```

```
    if self.domains[addr].owner == msg.sender:
```

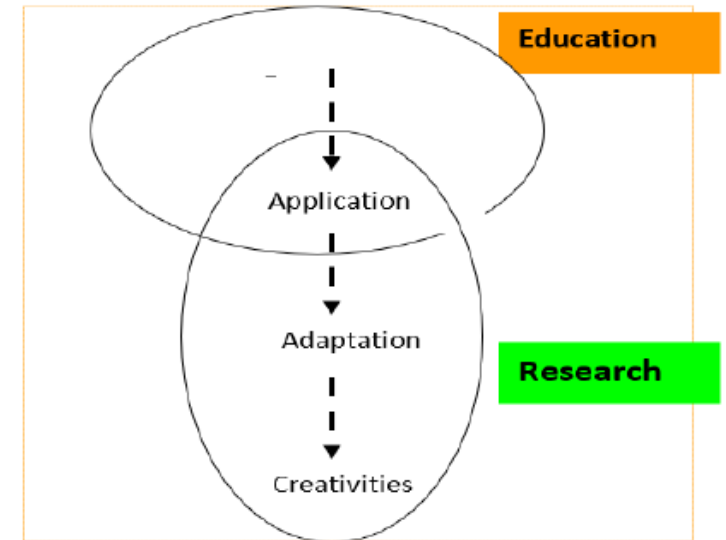
```
        self.domains[addr].ip = ip
```

LITERATURE REVIEW

- Currently, there are few works about designing hands-on labs that addresses blockchain development and implementation
- Mainly due to the infancy of blockchain application students are highly unaware of the use for blockchain application
- Delmonlino [8] in the Fall 2015, at the University of Maryland, conducted a series of lessons using Smart Contract to create their cryptocurrency labs presented to their undergraduate security -level students.
- Throughout the series they identified pitfalls in designing a safe and secure contract and advocating the best practices for implementing smart contracts.
- Smart contracts are user – defined programs that specify the rules governing transaction, and that are enforced by peer network which aim to the lower legal and transaction in comparison to traditional financial contracts.
- However, there several unique challenges when programming smart contracts “play for keeps”, if a smart contract is buggy the functionality of the contracts could lead to economic loss and or worst the program malfunction.
- Therefore, smart contract requires “economic thinking” and must be written to ensure that all parties are safeguard when using the program.
- During the research conducted at University of Maryland, they used the Ethereum’s Serpent language, however, focus was not language-specific but the broad model.

HANDS-ON LAB DESIGN


- Our goal is to develop series of hands-on labs that would address every main application of blockchain and thus provide practical tools to educate Cybersecurity professionals and equip them to address the cyber security problem blockchain.
- This approach will help students to systematically learn and comprehend the fundamental concepts in blockchain.
- We are following active learning model, which starts from hands-on to creativities educational approach.
- From education to research given a comprehensive stage of the security development of blockchain arming them with required information to educate the management in blockchain.



Learning Tree Model

HANDS-ON LAB DESIGN

- Solidity is a programming language that is smart contract oriented
- Ethereum is a decentralized platform for applications that run exactly as programmed without any chance of fraud, censorship or third-party interference
- Truffle is a development environment, testing framework and asset pipeline
- Ethereum Ganache quickly fire up a personal Ethereum blockchain which students can use to run tests, execute commands, and inspect state while controlling how the chain operate
- Meta Mask allows students to run Ethereum decentralized Apps(d-Apps) right in their browser without running a full Ethereum
- Node. JS node is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a browser.




Understanding the security behind Blockchain	<ul style="list-style-type: none">•Topics covered:•SHA256•Merkle Tree•elliptic curve•Public-Private Key
Hands on lab: Build your own crypto – system	<ul style="list-style-type: none">•Topics Covered:•Creating your own Crypto –system using solidityRemix on the Ethereum platform•Various article and current events on blockchain development
Past, Present, and Future of Blockchain development	<ul style="list-style-type: none">•Topics Covered:•Bitcoin and Other Cryptocurrencies•Ethereum development application•Block- Lattice•Various current event article on blockchain development.
Hands on Lab: dApps crypto – system	<ul style="list-style-type: none">•Topics Covered:•Part II of Creating your own Crypto-system using Ethereum open source: to create your local development environment with Truffle and Ganache to launch dApps.

Contents in Hands-on lab

CASE STUDY

- Design of specific hands-on lab intended to help students understanding the fundamental concepts of blockchain technology and implementation of smart contract using the Ethereum platform, power point presentation and article written by blockchain developers and enthusiast
- We layout a few hands-on labs based on blockchains applications
- The labs would be built based on real-life scenarios, to enhance their ability to understand and solve real-life cybersecurity problems.
- This integrated approach would expose the students to the cost to risk involved at each step



Understanding the security behind Blockchain	<ul style="list-style-type: none">•Topics covered:•SHA256•Merkle Tree•elliptic curve•Public-Private Key
Hands on lab: Build your own crypto – system	<ul style="list-style-type: none">•Topics Covered:•Creating your own Crypto –system using solidity Remix on the Ethereum platform•Various article and current events on blockchain development
Past, Present, and Future of Blockchain development	<ul style="list-style-type: none">•Topics Covered:•Bitcoin and Other Cryptocurrencies•Ethereum development application•Block- Lattice•Various current event article on blockchain development.
Hands on Lab: dApps crypto – system	<ul style="list-style-type: none">•Topics Covered:•Part II of Creating your own Crypto-system using Ethereum open source to create your local development environment with Truffle and Ganache to launch dApps.

Contents in Hands-on lab

CASE STUDY

- **Understanding Theory behind Blockchain**

- The purpose of this topic is to introduce students to Blockchain and the encryption computation behind this technology
 - Merkle trees are a fundamental part of blockchain technology
 - SHA - blockchain such as bitcoin uses a SHA256 hash function and Elliptic Curve Cryptography to improve security and privacy.
- Therefore, we will define and explain the computational properties of these functions.

- **Developing cryptocurrency token lab**

- The purpose of this lab is to introduce students to the leading Blockchain platform Ethereum, via writing smart contract using Solidity a contract-oriented programming language
- It is used for implementing smart contracts on various blockchain platforms and Remix a powerful, open source tool that help students write Solidity contracts straight from the browser all running on the Ethereum platform.

Cryptocurrency lab - example

1. Open the Remix Solidity IDE link: <https://remix.ethereum.org/> , you will see the layout of Remix IDE, as shown below:



2. Create a new Solidity File, On the far left of the screen press the plus sign within the solid circle



name your cryptocurrency whatever you like.

Example below:

×

File Name
RattlerCoin.sol

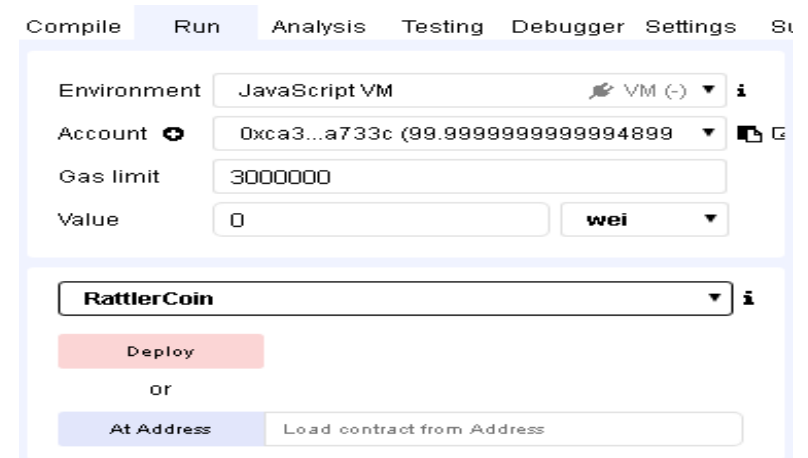
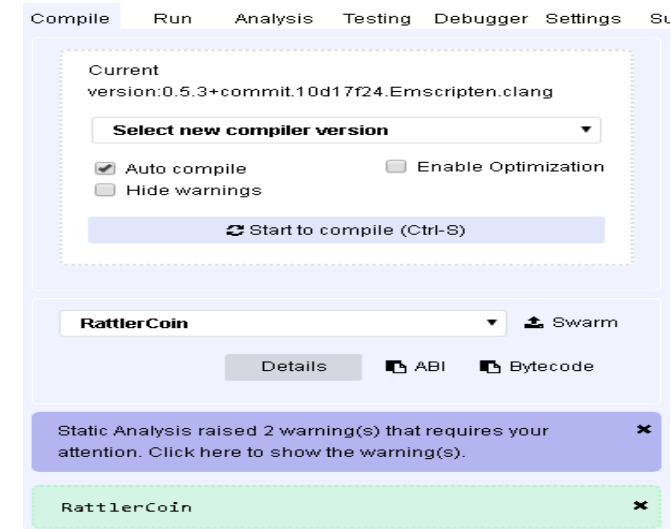
OK Cancel

Cryptocurrency lab - example

3. Modify, copy and paste the following code into the Remix Solidity IDE code writing and editing section:

```
pragma solidity ^0.5.3;
contract RattlerCoin {
  string public name = 'RattlerCoin';
  //currency name. Please feel free to change it
  string public symbol = 'rc';
  //choose a currency symbol. Please feel free to change it
  mapping (address => uint) balances;
  //a key-value pair to store addresses and their account balances
  event Transfer(address _from, address _to, uint256 _value);
  //declaration of an event. Event will not do anything but add a record to the log
  constructor () public {
    //when the contract is created, the constructor will be called automatically
    balances[msg.sender] = 10000;
    //set the balances of creator account to be 10000. Please feel free to change it to any number you want.
  }
  function sendCoin(address _receiver, uint _amount) public returns(bool sufficient) {
    if (balances[msg.sender] < _amount) return false;
    // validate transfer
    balances[msg.sender] -= _amount;
    balances[_receiver] += _amount;
    emit Transfer(msg.sender, _receiver, _amount);
    // complete coin transfer and call event to record the log
    return true;
  }
  function getBalance(address _addr) public view returns(uint) {
    //balance check
    return balances[_addr];
  }
}
```

We just created some variables to record our currency's name and symbol - "RattlerCoin" and "rc". We use mapping to keep track of the balances of all addresses. In the constructor, it gives the initial creator 10,000 value of tokens. It's consistent with the basic definition of currency as I mentioned in the beginning—a database with one operation, that is subtracting X units from A and adding X units to B



Cryptocurrent lab - example

1. Once your currency is deployed a transaction summary will be displayed below the code writing and editing section, fill the information below:

status	
transaction hash	
contract address	
from	
to	
gas	
transaction cost	
execution cost	
hash	
input	
decoded input	
decoded output	
logs	
value	

Part II

1. Write the pseudocode for the Cryptocurrency lab above?

2. What is the purpose of a nonce?

3. What powers the Ethereum Virtual machine?

4. What is the purpose of a gas?

CASE STUDY

- **Past, Present & Future**

- The purpose of this topic is to introduce students to the real life blockchain applications: Bitcoin, AWS Quantum Ledger Database, Azure MS Blockchain, IBM Hyperledger
- Future blockchain technology such as Block lattice.
- Students will further understand the wide use of blockchain technology.

- **Case Study: Delivery Drone & Smart Door**

- The purpose of this lab is to further the student's development capability of Blockchain technology by create decentralized application (d-Apps) using the following tools: Solidity, Ethereum- is a decentralized platform for applications that run exactly as programmed without any chance of fraud, censorship or third-party interference.
- Truffle - is a development environment, testing framework and asset pipeline for Ethereum
- Ganache - Quickly fire up a personal Ethereum blockchain which you can use to run tests, execute commands, and inspect state while controlling how the chain operates.
- Meta Mask - allows to run Ethereum decentralized Apps right in your browser without running a full Ethereum node.
- JS node - is an open-source, cross-platform JavaScript runtime environment that executes JavaScript code outside of a browser

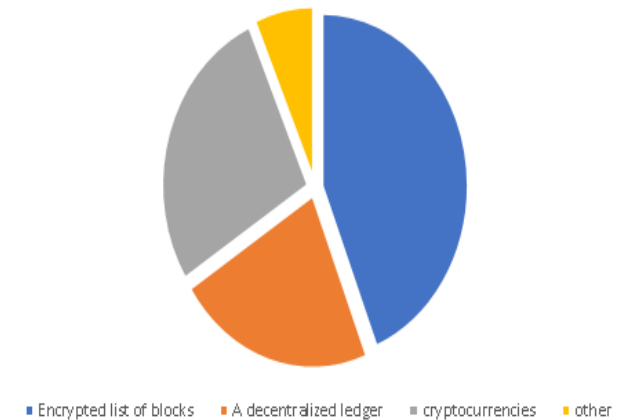
CASE STUDY: Summary

- Lab 1 explains the computation behind blockchain technology we introduce topics such as cryptography, elliptic curves, Merkle tree, and SHA algorithm.
- Lab 2 students will actively develop and implement smart contract to create a their ever-own cryptocurrency and token using the Solidity programming language on the Ethereum Remix IDE.
- Lab 3 will be “The Past, Present and Future of Blockchain”, which is an overview of blockchain use cases, past and future blockchain applications.
- Lab 4 we will challenge our students in creating a Decentralized Application (d-Apps) using, the Ethereum open source using tools such as: Truffle and Ganache.

STUDENTS' FEEDBACK

- During Spring 2018, a joint survey was conducted at our University, Introduction to Computer security.
- During the survey students were asked a series of questions regarding to blockchain, blockchain development platform and interest in blockchain development.
- 32 students responded to the questionnaire.
- We show a few survey results here.
- When students were asked, have they heard of blockchain?
 - 43.7% stated, “No”
 - 56.3 “Yes”. Students who stated, “Yes” describes blockchain.

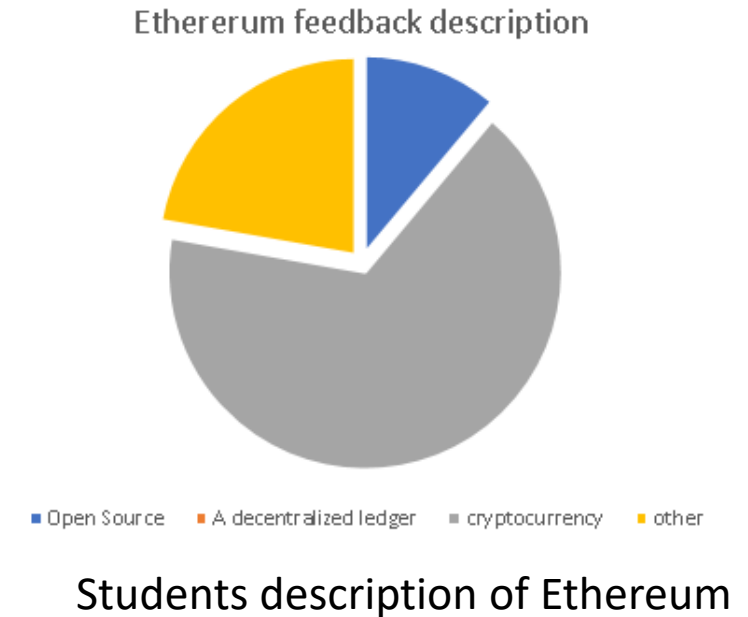
Blockchain feedback description



Students description of blockchain

STUDENTS' FEEDBACK

- An overwhelming number of students describe blockchain as encrypted list of blocks, followed by cryptocurrencies, lastly a decentralized ledger.
- Next, we asked students' have they heard of Ethereum open source blockchain platform and cryptocurrency.
 - 65.5% of students stated, "No" they have not heard of Ethereum
 - 34.3% students have heard of Ethereum, those have heard of Ethereum describes in Fig 13: Students **description of Ethereum** as:

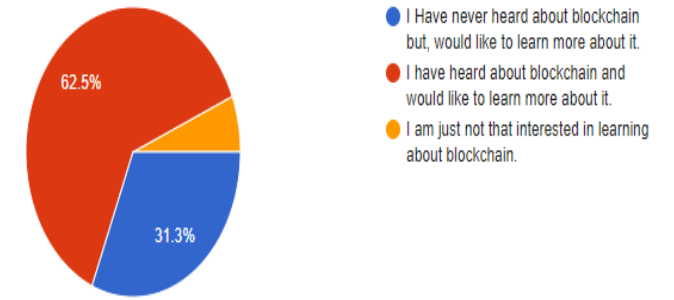


STUDENTS' FEEDBACK

- 31% responded, Yes, to knowing about Ethereum
 - Many describe Ethereum as a cryptocurrency, followed by other, and lastly a few said, “open source”.
- Students have stated, overwhelmingly, that they have heard about blockchain and would like to learn more about blockchain technology.

4. How would you describe your interest in learning about blockchain technology?

32 responses



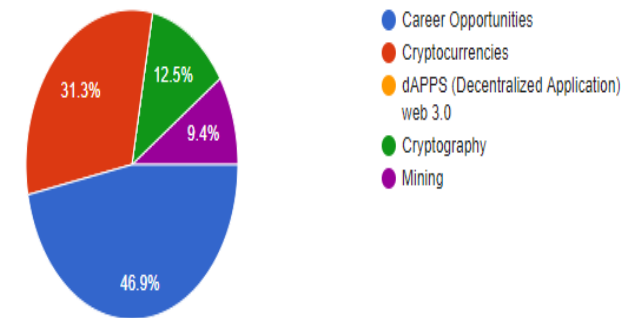
Interest in blockchain technology

STUDENTS' FEEDBACK

- Leading is 31.3%, Have not heard of blockchain and would like to learn more about blockchain technology and 6.3% were not interested in learning about blockchain technology.
- Lastly, students describe prodigiously at 46.9% - career opportunities as a reason for increase interest in blockchain technology, followed by 31.3% - cryptocurrencies, 12.5% - cryptography, 9.4% - mining and 0% for d-Apps (decentralized Application web 3.0).

5.What aspect of Blockchain would make you more interest in this topic?

32 responses



Blockchain area of interest

Conclusion

- Currently, blockchain technology is safe, reliable and secure technology. With many enthusiasts about the implementation for blockchain technology, however, large number of students' and future IT professional are unaware of blockchain and its use cases.
- There are not many blockchain hands-on labs. Thus, we introduce a comprehensive hands-on topics and labs that will enhance student's knowledge of blockchain and blockchain development.
- We described a series of hands-on labs, power point presentation and articles.
- According to students feedback they are enthusiastic about blockchain and the career opportunities that it offers.
- Students' feedbacks are positive, and they are interested in learning more and love to work as blockchain developers.