Blockchain and Smart Contracts

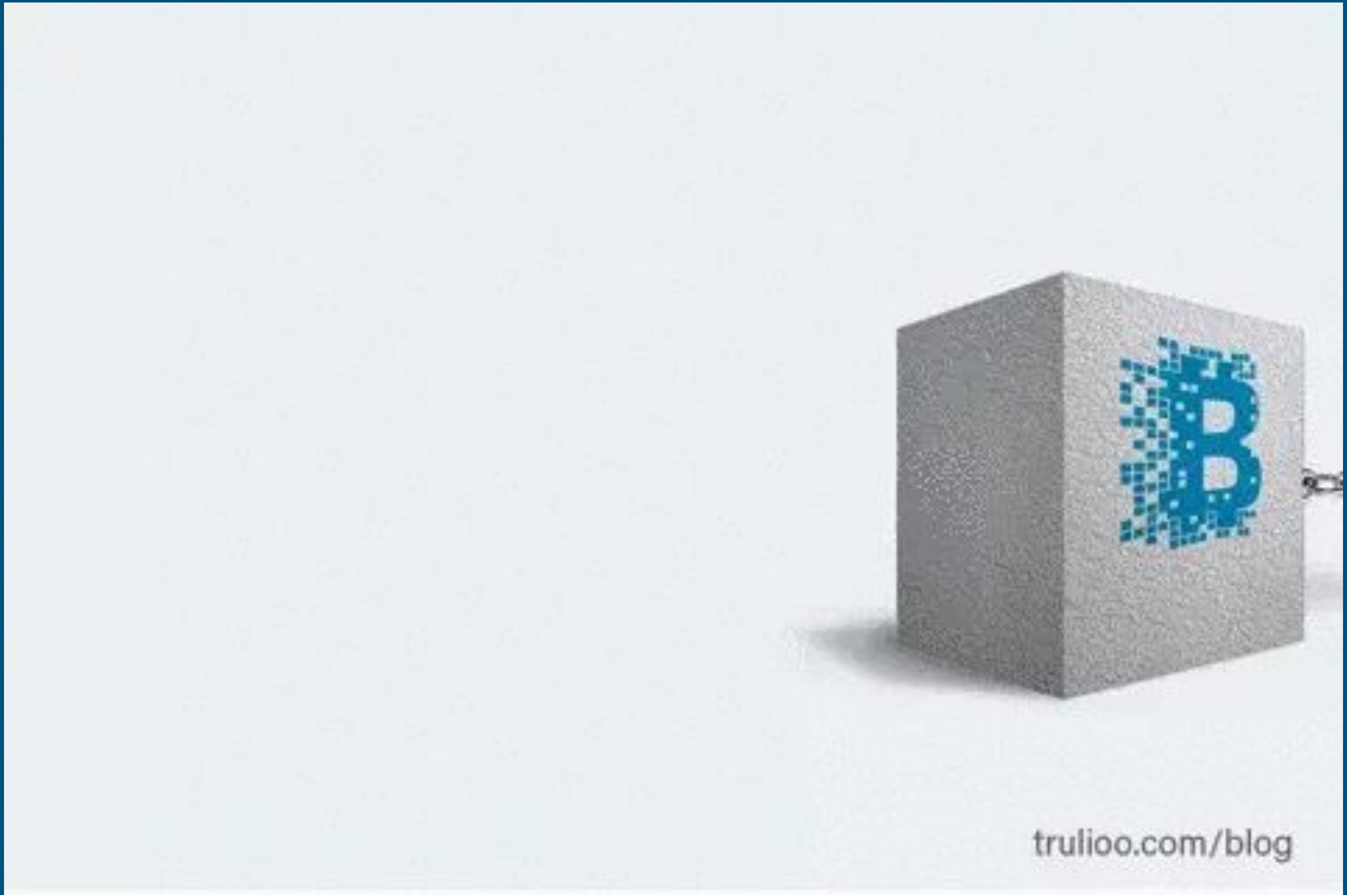
You know… cryptocurrency and stuff

Jack McDowell

University of Virginia

What is a Blockchain?

- Public, shared ledger of transactions
    - Everyone has a copy that gets written to
    - Append-only log
- Each entry references a previous one
    - Verified through math, crypto, and hashing so previous entries retain their integrity
- As a new transaction is processed, thousands of “miners” validate it, and upon validation, add it to the blockchain



trulioo.com/blog

Most common examples

1. Bitcoin
    a. The first “cryptocurrency,” or currency to run on a blockchain
    b. Market cap higher than many countries’ GDPs
2. Ethereum
    a. First programmable blockchain
        i. Smart contracts
3. LiteCoin
    a. Another cryptocurrency, designed to be faster and cheaper than bitcoin
4. Use cases: Supply Chain and Voting

Ethereum

- Runs off blockchain
- Uses a cryptocurrency called “ether”
- Transactions use “gas”
    - Pays miners to validate the transactions
    - Each transaction has a gas price and a gas amount
    - Gas price is set by transactor, in Wei
        - 10^18 wei per ether
        - If gas price isn’t high enough, miners won’t process the transaction
- Blocks can contain “smart contracts”

Smart Contracts

- Marketed as “programmable blockchain”
- Basically
  - Pieces of code lying on blockchain
  - Cannot be modified once deployed
  - Completely public
- In effect
  - Guarantees a payment will go through once conditions are met

Smart Contracts (continued)

- Written in a language called “Solidity”
- Object oriented language
- All fields and functions can be accessed from anywhere
    - Privacy modifiers can ensure only certain contracts or addresses can call certain functions
- Code runs on the EVM (ethereum virtual machine)
    - Completely sandboxed
        - No access to internet, file systems, or anything

Accounts on EVM

- Accounts
  - External (human)
    - Address comes from public key
  - Contract (smart contracts)
    - Address is created when contract is deployed
  - All accounts are treated equal by EVM
  - All accounts have “storage”
    - Mapping of 256 bit address to blocks of 256 bits
  - All accounts have a balance, kept in Wei

Transactions

- On the most basic level, a transaction is a message sent from one account to another
    - Message may contain data and value
- If destination account exists, the code at the destination is executed and given the data from the message
- If the destination is 0
    - Basically, creates a new address and runs code in the data to create a new contract

Storage, Memory, and Stack

- All forms of storage and memory come in blocks of 256 bits, or words
- Storage
    - Persistent memory
    - Mapping from word to word
- Memory
    - Temporary memory
    - Dynamically allocated, paid for in gas as it expands
    - Reads limited to 256 bits
    - Writes limited to 256 bits or 8 bits
- Stack
    - EVM runs on a stack, not registers
    - Stack is limited to 1024 elements, each 256 bits
    - Top 16 elements can be accessed and swapped around

Events

- Contracts can emit events
- Events contain data
- Scripts off the chain can listen for events
- All past events are recorded and always available

Sample Contract

- We’ll just use NSA Codebreaker challenge contract for this
- Three separate contracts
  - Ransom
  - Registry
  - Escrow
- Here’s what we’ll do
  - Deploy a new contract
  - Call some of its functions
  - Show how to see its storage
- Need MetaMask to do much
  - Chrome extension that lets javascript interact with blockchain

NSA Codebreaker Challenge

- Cybersecurity and Cryptography Challenge by NSA
- Walk through incident response to a hack
  - This year, respond to a ransomware outbreak
- Later challenges require knowledge of Ethereum and Smart Contracts

Scenario

A new strain of ransomware has managed to penetrate several critical government networks and NSA has been called upon to assist in remediating the infection to prevent massive data losses. For each infected machine, an encrypted copy of the key needed to decrypt the ransomed files has been stored in a smart contract on the Ethereum blockchain* and is set to only be unlocked upon receipt of the ransom payment. Your mission is to ultimately (1) find a way to unlock the ransomware without giving in to the attacker's demands and (2) figure out a way to recover all of the funds already paid by other victims. Are YOU up to the challenge?

Codebreaker Challenge Sign-up

- Sign up at: https://codebreaker.ltsnet.net
- Hints: Wireshark and IDA are helpful :)

- We'll be walking around to help give you hints