Decentralized Applications - Blockchain and Smart Contracts

Nikita Tribhuvan
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
nst2785@cs.rit.edu

I. Introduction

A smart contract is digital contract without the involvement of third parties, it results in transparent and irreversible transactions. Ethereum is an open source project used to build decentralized applications using the next generation blockchain platform which is also used to code and process smart contracts. It is also used as a platform to maintain Ether, Ethereum Classic, Expanse, etc which is a cryptocurrency emerged from ethereum. It enables the user to generate and code a set of operations of any given complexity instead of giving the user a restricted number of operations. There exists a Ethereum Virtual machine (EVM) which is implemented on every node in the network and each node executes the same instruction set. Smart contracts are compiled using the EVM to a byte code format and are executed on the Ethereum blockchain. Another concept is using Ethereum to construct a Decentralized Autonomous Organization (DAO) which are governed by a set of smart contracts on the Ethereum blockchain. The Ethereum network is a set of clients which use different libraries to establish a connection among them and secure the blockchain.

The report is organized as follows, Section II describes the concept of a digital currency. Section III describes the concept of public key cryptography and hashing. Section IV is a detailed explanation of the concept of blockchain followed by Section V which describes Byzantine General Attack problem and this problem is tackled by blockchain network. Section VI and Section VII explain Proof of Work (PoW) and Proof of Stake (PoS) which are mechanisms used by the blockchain system. Section VIII describes the concept and the main features of Ethereum. Finally Section IX is a detailed description of the decentralized music application that was built as a part of the independent study.

II. Digital currency

Imagine a scenario where you go buy goods or an item from a vendor. This implies a transaction wherein physical currency (token) is exchanged for an item. One property of physical currency is that the physical currency is transfered to the vendor and can prevent the fraudulent usage of the currency. In case of a digital currency, the token is scanned and stored in a digital form. This causes a two primary problems, firstly there can be several copies of the same token which makes it hard to determine who the owner of the token. And the user can use the copies of the same token repeatedly, this is known as double spending. The problem of ownership can be solved using Public key cryptography and the problem of double spending is solved by using the concept of a blockchain coupled with hashing.

III. Public key cryptography

Hashing is a primary component used in public key cryptography as well as i blockchain technology. Hashing is a method to calculate an output or a message digest for given input. The message digest is of a relatively fixed sized for an input of any given size. Secure Hash Algorithm (SHA) is commonly used in blockchain which generates an output of length 256 bits. A infinitesimal change in the input will cause result in a completely different message digest. The functions are designed in a way such that it is computationally infeasible to find the input for a given output. Also, hash functions are collision resistant, which implies it is computationally infeasible to find two or more inputs which generate the same output.

Public key cryptography is a technique to cipher the original text so that it can only be read by an individual with the appropriate key. Digital signatures is one of the most commonly used techniques of public key cryptography. It consists of a combination of a public and a private key. the public key is publicly available and the user is addressable using this key. Private key, as the name implies is kept private and is only known by the user and cannot be accessed by other individuals. The original text is encrypted by the sender using their private key and is decrypted by the receiver using the sender's public key. The following are the steps involved in a transaction using public key cryptography:

1) The sender will issue the transaction and sign it using their private key and send it to the receivers public address.
2) The receiver takes the details of the transaction along with the public key of the sender as well as his own and generates a hash value.

3) Next, the receiver takes the digital signature received from the sender and decrypts it using the sender's public key and generates another hash value.
4) If the two hash values match it indicates that the transaction is valid.

IV. Blockchain

Blockchain is the backbone of the cryptocurrency system which is a decentralized network which together abides to validate new blocks. It consists of a distributed ledger wherein all the records of the transactions are validated and recorded. This ledger is distributed across all of the nodes. This, in turn, eliminates the need for a trusted third party to facilitate a transaction. The blockchain structure consists of blocks linked to each other similar to a linked list structure. Each block has a set of transactions associated with it. The blocks are chained using the hash of the header of a previous block. The only exception to this is the first block know as Genesis, which has no parent block. Each block in a blockchain consists of the following:

- Block Header
  - Hash of the previous block
  - Timestamp
  - Nonce
  - Merkle root hash
- Hash of their own block
- List of Transactions

The hash of each block is determined by hashing the header of the block, which consists of hash of the previous block, timestamp, nonce and the merkle root hash. Which implies that if any of the above information is even slightly modified the hash value of the block will be changed completely. Merkle tree is used to combine the hash of the individual transactions until a single root hash is acquired. It avoids storing the hash of each of the individual transactions in the header. This ensures the immutability of transactions, as changing one transaction in one block will cause all the subsequent blocks to have different hashes as the hash of the previous block is included in the header. Once a transaction is recorded it tedious to alter it as it requires a change in all of the subsequent blocks after it. New transactions are submitted to a node, and added to the block only when a given consensus method is completed by a node. After which the block is added to the chain and the ledger must be updated with the new transaction.

Blockchains are categorized depending on the access rights as permissioned and permissionless. Permissioned blockchain ensure that only trusted users are added to the network. Only these trusted users have read and write permissions and can view the distributed ledger, they are not publicly available. This category of blockchains are used in a scenario where the users yearn to work together but do not completely trust each other. Permissioned blockchain can be set up such that everyone on the network can read them but only a selected few can record transactions. The consensus algorithm is determined by the users based on the level of trust between the users of the network. Permissionless blockchains are more flexible and are open to the users wherein the users don't require permission to be added to the network or to even read or write transactions. The consensus algorithm used is more tedious in type of blockchain to prevent a malicious user from destabilizing the system, as they can be accessed by anyone. These type of blockchains use Proof-of-Work or Proof-of-Stake as their consensus mechanism.

V. Byzantine General Attack

In this problem, multiple army factions surround a castle that they want to attack and thereafter capture. Each faction is led by an army general, with one main lead general. The factions are spread across the castle it, hence it becomes challenging to establish a centralized chain of command among them. In order to capture the castle, all of the factions must attack the castle simultaneously. The generals communicate the time of attack between them to plan a coordinated attack. But there may be a case where some generals are traitors and refrain from communicating the time to the general, or even worse they communicate a wrong time of attack. In such a case it become difficult to determine the loyalty of the generals. This problem can be solved by using a blockchain system and it ensures that the messages communicated can be trusted by maintaining a distributed ledger which will lead to a successful coordinated attack. A consensus algorithm is used to ensure that valid information is recorded in the ledger. Every blockchain system has agreed upon consensus method in place, and each user is if there exists a condition wherein two chains are present then the the longer chain is considered to be valid. Hence a user is motivated to add the new valid block to their chain because if they refuse to accept the block they will be building off a shorter chain which will be rejected eventually.

VI. Proof of Work

Proof of Work is used to add a new block the blockchain. The goal is to compute the solution to a complex mathematical puzzle. It involves in finding a nonce for the given hash value that when hashed with the block header will the required result. As we know, hashing is a one way process, finding the hash input from the message digest is a computationally infeasible task, this is what makes PoW challenging. In order to solve the puzzle, the miner has to hash the candidate block with countess possible values to get the desired result. This task is difficult to compute but easy to verify. Once a miner computes the nonce for the required value, this nonce is validated by the nodes in the network. In order to verify the value only a single hash needs to be computed with the given nonce value. And the verified block is added to the user's blockchain and the nonce is propagated to the other users. One important advantage of this model is that is unbiased. Any user of the blockhain can have their block as the next valid block given they find the solution to the Proof-of-Work. Past attempts of miners solving the challenge does not affect the likelihood of solving it. Each of puzzle is independent and the amount

of work that needs to be done to solve the puzzle is constant. The number of leading zeros determines the complexity of the puzzle, the more the number of leading zeros the more complex the puzzle. It is used in a system where there is little or no trust between the users of the blockchain system. This consensus algorithms ensures that no block can be altered effortlessly, in order to modify one block all of the blocks succeeding it have to be modified. The major disadvantage of this method is the computational power used by the miners to compute the solution.

VII. Proof of Stake

The main idea for Proof of Stake is the ability to add a new block to the blockchain depends on the stake the users holds in the blockchain. Since a user having more stake will not want to want to destabilize it for his own personal gain. No computationally intensive calculations are required to e performed by the users unlike proof of stake. Three different approaches are used to in Proof of Stake namely - random selection of staked users, multi-round voting, coin age system. In random selection proof of stake the creator of the new block is selected randomly depending on the stake owned to the overall stake of the system. A user owning 54% of stake in the system will have a probability of being selected 54% of the time. Multi-round voting proof of stake method, a group of staked users are selected to create the new block and these users vote on the next block that should be added to the chain. The new block is selected after multiple rounds of voting, so that all of the users having a significant stake get an opinion on the new block. The coin age proof of stake enable users to add a new block by spending aged cryptocurrency. The users currency has an additional age property associated with it, and after a given amount of time this aged currency can be spent in order to create a new block. Once spent the age property is set back to 0 and can be spent again after a required amount of time has elapsed. This method prevents users with high stake from dominating a system.

VIII. Ethereum

Ethereum is an open source platform using the concept of the blockchain framework which is used to implement smart contracts. Smart contracts enable programmers to create their own set of protocols which can be triggered when a transaction is addressed to it. Smart contracts are stored within the blockchain. They are analogous to procedures stored on data management systems. They are executed on each and every node in the network in a prescribed manner as implemented in the smart contract. A smart contract is deterministic in nature as every node will execute independently and must return the same results on execution.

The primary data structure used the Ethereum network is the Merkle Patricia radix tree which is optimized for key-value mapping. Every ethereum node in the network runs Ethereum Virtual Machine (EVM) to maintain consensus across the blockchain. EVM is a Turing-complete compiler which enables the programs to run irrespective of the programming language. EVM compiles smart contract into byte code and then executes them. Every computational step executed by the EVM is priced at a unit of gas. Since the EVM cannot predict the quota of resources that will be required to validate a given transaction. The nodes can vote upon the maximum amount of gas required for a given block.

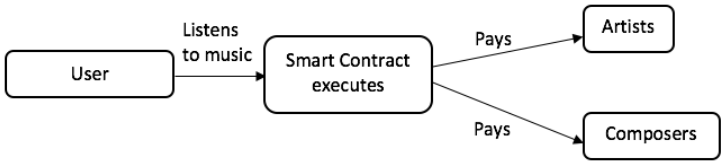

Fig. 1. Application flow of smart contracts in the music industry

IX. Decentralized Application

Figure 1 shows the general flow of for the application of smart contracts in the music industry. Every time a user listens to an album or a record, a smart contract is executed.The smart contract is executed and user pays the record company and the artist in the form of cryptocurrencies. The features offered by the web application are listed as follows:

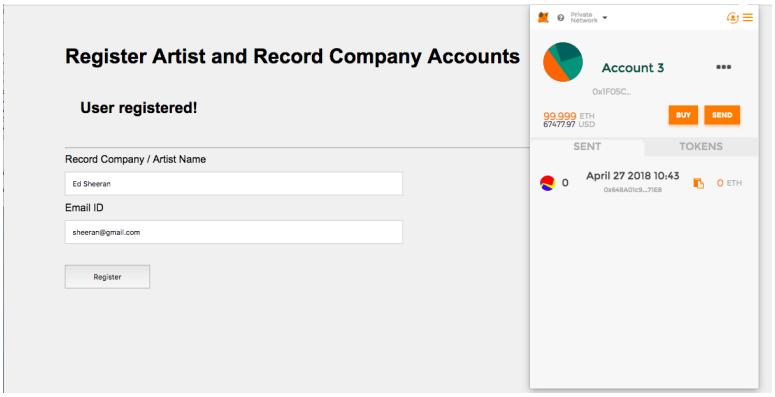

Fig. 2. Register Artist

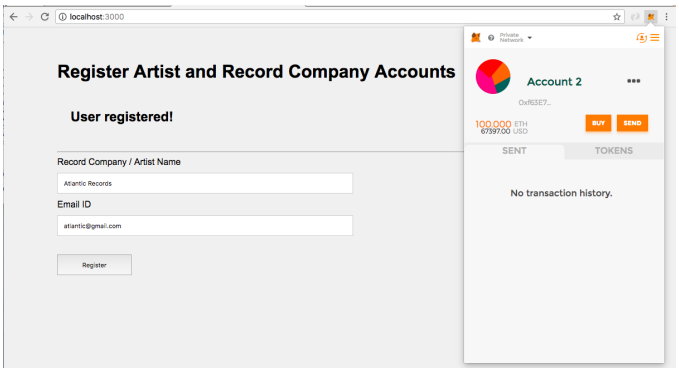

Fig. 3. Register Record Company

1) Creating accounts: The artist and the record company need to create accounts before publishing a any song.
2) Publish a song: Publishing the song, the song name and the name of the registered record company and artist has to be provided to successfully publish the song on the website.

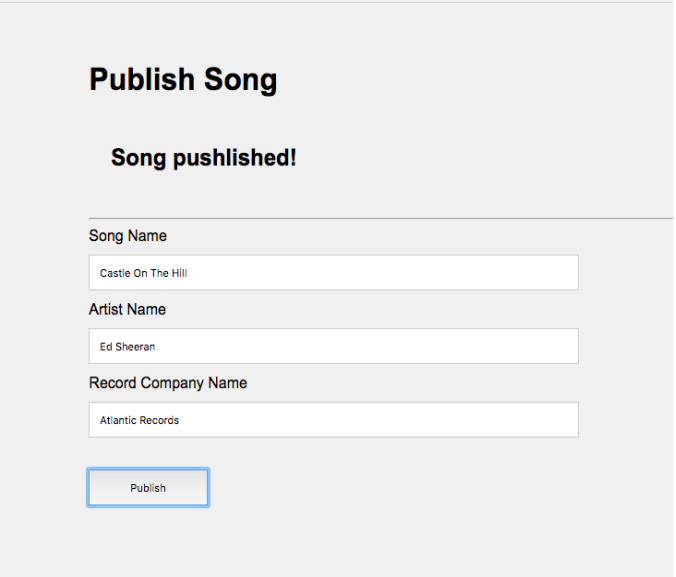

Fig. 4. Publish a song

3) List all of the information about the published songs including the artist name, album name, cover of the album. And enable a user to buy and listen to the song as shown in Figure 5.

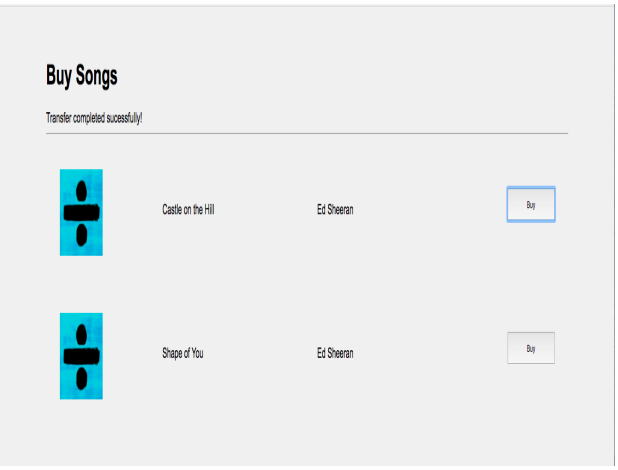

Fig. 5. Interface to buy music

Functions of the smart contract:

1) Once the accounts are created, the account address using which the user was registered is stored in a dictionary using the smart contract.
2) The smart contract creates another dictionary and saves this published song information using the song name as key and the address of the record company and artist as its values.
3) Transfer the money the respective addresses when a song is purchased by the user and distribute the ether between the record company and artist using a predetermined split as shown in Figures 6 and 7.

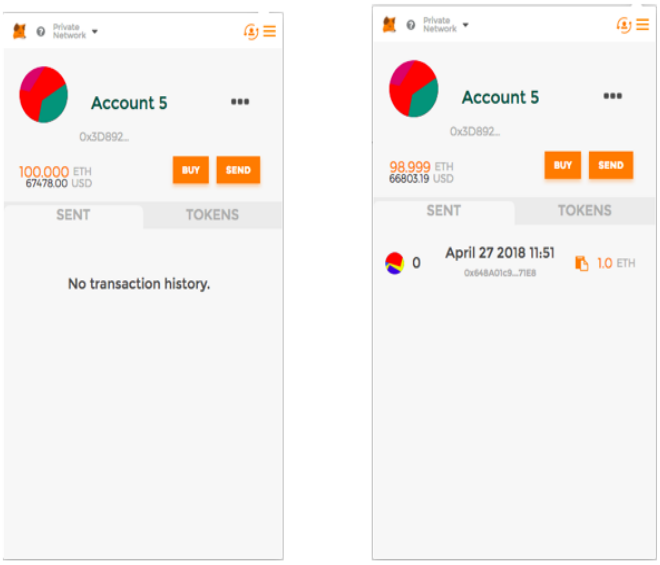

Fig. 6. Balance of the user before and after the transfer

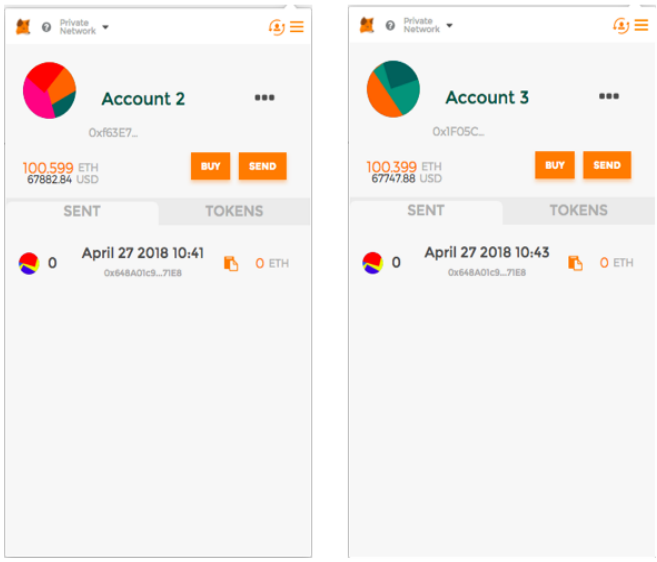

Fig. 7. Balance of the Record Company and Artist accounts after the transfer

All of the strings are converted to a 32-bit byte code when the smart contract is executed. The smart contract has two mappings called accounts and songs. Mapping named accounts keeps track of the names of the registered users which is converted to bytes as the key and the address as value. A structure is created when a song is published, which

retrieves the addresses of the artist and record company from the accounts mapping. this strut is used a the value in the music mapping which is referenced by the song name as key. Each button to buy a song has an associated song name as its value. It uses this song name to fetch the details from the music mapping. One the addresses are fetched, the money is transfered from the user account to the record company's and artist's accounts. The amount is split between the record company and artist as 60:40 respectively.

X. Conclusion

This independent study helped my understand the fundamentals of the blockchain technology. The application developed further aided me to clear the concepts. There are many more things that could be done to improvise the application. As future work, the application can be modified to dynamically display the song details for the user to buy as soon as the song is published. The front-end of the application can be further enhanced and cal accommodate user login and accounts and store list of songs purchased by a user.

References

[1] Developing ethereum smart contracts for beginners, coursetro.com/courses/20/developing-ethereum-smart-contracts-for-beginners.

[2] ethereum project. ethereum project, www.ethereum.org/.

[3] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.

[4] N. R. K. S. Dylan Yaga, Peter Mell. Blockchain technology overview. *National Institute of Standards and Technology*, 2018.

[5] H. Kim and M. Laskowski. A perspective on blockchain smart contracts: Reducing uncertainty and complexity in value exchange. In *Computer Communication and Networks (ICCCN), 2017 26th International Conference on*, pages 1–6. IEEE, 2017.

[6] S. King and S. Nadal. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August*, 19, 2012.

[7] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[8] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, and S. Chen. The blockchain as a software connector. In *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on*, pages 182–191. IEEE, 2016.

Appendix

A. Progress update

- Week 1: Fundamentals of Blockchain, Ethereum Smart contracts, Ethereum Virtual Machine, Proof-of-Work, Proof-of-Stake, Transaction and gas concept: 8 hours
  Creating a presentation to describe the concepts learned and setting up the report for the final submission: 3 hours
- Week 2: Reading the original blockchain whitepaper and reading more research papers and presentations on blockchain provided by Professor Hu. Using the above material to improvise the presentation created in Week 1. Understand the concepts learned in Week 1 in more depth. Report and documentation. Spent approximately 7 hours in Week 2.
- Week 3: Created a presentation on the concepts learned, read more about concepts like hashing, public key cryptography, downloaded Ethereum Wallet and started follow the tutorial to understand the working of the code and smart contract in depth. Spent approximately 8 hours in for all of the tasks mentioned above.
- Week 4: Improvise the current presentation to add more technical details. Finish the tutorial on smart contracts, creating my own token using the Ethereum Wallet. Researching on what can be done with blockchian with data management systems. Documenting and adding more details to the report. Spent approximately 8 hours in for all of the tasks mentioned above.
- Week 5: Implement the tutorial on crowdsale and Decentralized Autonomous Organization (DAOs) using the Ethereum Wallet. Studying in depth about the concept of Crowdsale and DAOs. Spent approximately 7 hours in for all of the tasks mentioned above.
- Week 6: Ideas about how ethereum can be used for a real life applications that can be implemented as a part of the independent study. Read in depth about smart contracts and completing the tutorials on the ethereum website. Spent approximately 8 hours in for all of the tasks mentioned above.
- Week 7: Studying about decentralized applications and going though tutorials on how to develop an application using meteor. Spent approximately 9 hours in for all of the tasks mentioned above.
- Week 8-13: Working on a decentralized application wherein a user buys music and a smart contract distributes the payment to the artist and record company for that song. Creating a smart contract using remix IDE, learning and installing Metamask, using testrpc accounts, using web3js to call the functions in the smart contract. Spent approximately 9 hours every week to complete all of the tasks mentioned above.
- Week 14: Designing a UI for the application and creating the front end for the application. Spent approximately 6 hours in for all of the tasks mentioned above.
- Week 15: Working on the report for the final submission and documentation of the application created. Spent approximately 7 hours in for all of the tasks mentioned above.