JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Lucky Draw D-APP System Using Blockchain to Avoid Third Party Interference

[1] Pratik Deshmukh, [2] Anurag Sharma, [3] Aryaman Jagtap, [4] Vedanti Deshmukh, [5] Sanchi Gandodhar

[1]Assistant Professor,, [2,3,4,5] Student,
Dept. of Computer Science and Engineering
Prof Ram Meghe Institute Of Research And Technology

Abstract : Lucky draws have been a classical form of entertainment and charity for centuries. However, the traditional lucky draw industry is often plagued by transparency and fraud issues.The traditional lucky draw industry is often plagued by transparency and fraud issues. For example, it is difficult to verify the fairness of the lucky draw process, and there is a risk of fraud or manipulation by the organizers. This paper proposes a decentralized lucky draw application (DApp) on the Ethereum/Polygon network blockchain. The DApp uses smart contracts to ensure transparency and fairness, and it is immune to fraud or manipulation. The proposed DApp has several benefits over the traditional lucky draw industry. First, it is more transparent and fair. Second, it is immune to fraud or manipulation. Third, it is more secure. Fourth, it is more efficient. The proposed DApp is a promising solution to the transparency and fraud issues in the lucky draw industry. It is more transparent, fair, secure, and efficient than the traditional lucky draw industry.

***Keywords* - Blockchain, Decentralized, Lucky Draw, Fraud, Security, Efficiency**

I. INTRODUCTION

Blockchain is developing as one of the most important technologies that affect our day-to-day life. It is structured in a way that allows you to create records in a safe and transparent way. Each block in the blockchain contains the data transaction, timestamp and hash of the previous block. It is not feasible to forge the data which enables the data to be reliable in the blockchain. The turning characteristic of the blockchain is the significant use of encryption used as a link element between blocks.

Accommodation of a Lucky Draw system in a blockchain can result in increased trans-parency. There is an urgent need to replace traditional systems with fully computerized systems that ensure fairness. Transparency and fair distribution of funds are all the most common issues that most people doubt, distrust, or complain about. When playing the Lucky Draw, the winning percentage is determined purely by chance without skill. Lucky Draw players are not classic gamblers. Lucky Draw players rely entirely on opportunities, hoping that they have the potential to thrive and win life-changing awards, and a small amount of money to buy the dream of becoming an instant millionaire. Each participant purchases a ticket in anticipation of the realization of their dreams. Under the current system, there are intermediaries and distributors who sell Lucky Draw tickets to people in each region.

The rest of the process, such as auditing and declaring results, is also manually performed by the intermediary under government obedience. In this system, people are concerned about unpredictability and consistency, as well as random number fairness, testability, and tamper resistance. And based on these facts, we have equipped the blockchain with a Lucky Draw buying and selling process. Evaluate payments, tickets and payments in the distribution environment using blockchain technology. The blockchain network allows all players to participate on an equal footing. There is no central concept of power, but power is distributed to all involved. The Lucky Draw process involves certain steps such as registration, purchase, completion, verification, random selec-tion of winners, announcement of winners, and payment. The security of payment, ticketing and payment in the sales environment is guaranteed by the system using the blockchain.

A. *Overview*

The Lucky Draw dapp's purpose is to create a lucky draw based smart contract 1 Lucky Draw Dapp System using Blockchain on Ethereum blockchain for increasing transparency and reducing frauds in the lucky draw in-dustry. Once the contract has been deployed by the administrator, there will be a minimum contribution amount for players to register in the game and a price pool will be maintained by the smart contract. Lucky draws have been a classical form of entertainment and charity for centuries. We use smart contracts and blockchain in decentralized, intelligent, and secure systems for lucky draw industries. Moreover, we are inspired by the algorithm of RANDAO, an outstanding way of random number generation in the blockchain scenario. The winning process will be structured in a way such that only the administrator's wallet will be authorized to initiate the process to randomly pick an address and the smart contract will by definition transfer the prize to the winner. The contract once deployed on blockchain cannot be changed by the administrator to maintain transparency and fairness. The Amazon Lucky Draw is a thrilling way to win gifts and help your favorite organizations, and this project can idealize the philosophy behind it. The Amazon Lucky Draw makes it simple for you to donate to your preferred charities by utilizing the strength of blockchain. With the help of this project, we can mimic amazon lucky draw with the help of blockchain.

B. *Problem Statement*

Lucky draws have been a popular form of entertainment and fundraising for centuries. However, the traditional lucky draw system is often plagued by problems of fairness, security, transparency, and traceability.

One of the biggest problems with the traditional lucky draw system is that it is often difficult to verify the fairness of the draw. This is because the draw process is typically controlled by a single entity, such as a company or organization. This means that there is a risk that the entity could manipulate the draw in order to favor certain participants or to increase their own chances of winning. Another problem with the traditional lucky draw system is that it is often difficult to ensure the security of the participants' personal information and the prize money. This is because the data is typically stored on a centralized server, which makes it vulnerable to hacking and theft. Finally, the traditional lucky draw system is often lacking in transparency. This is because the draw process is typically not open to the public, and it can be difficult to track the movement of funds. This lack of transparency can make it difficult to trust the system and to ensure that the draw is conducted fairly.

Blockchain technology can be used to address these problems and to create a more fair, secure, transparent, and traceable lucky draw system. Blockchain is a decentralized, distributed ledger that is secure and transparent. It can be used to create a system where the draw process is controlled by the participants themselves, rather than by a single entity. This helps to ensure that the draw is fair and that there is no risk of manipulation. Blockchain is also a secure platform. The data on the blockchain is encrypted and distributed across multiple nodes, which makes it very difficult to hack or tamper with. This helps to protect the participants' personal information and the prize money. Finally, blockchain is a transparent platform. The entire draw process is recorded on the blockchain, which means that it is open to public scrutiny. This helps to build trust and confidence in the system. In conclusion, blockchain technology is a promising solution to the problems of fairness, security, transparency, and traceability in lucky draws. By using blockchain, it is possible to create a fair and transparent system that is immune to fraud and manipulation.

C. *Traditional Problems and Solutions*

Problem

The traditional lucky draw system has several problems, including:

1. It is time-consuming and inconvenient for individuals to hold a lucky draw event in a traditional way.
2. There is a risk of fraud and manipulation, as the third party who is responsible for drawing the winners may not be completely fair.
3. The third party may pay the winners a smaller amount of money or even nothing, so that they can keep more of the prize money for themselves.
4. The traditional lucky draw system is not transparent, as it is difficult to track the movement of funds and to verify the fairness of the draw.

Solutions

This project presents the design and implementation of a secure online lucky draw system that addresses the problems of the traditional system. The proposed system has the following features:

1. It is accurate, as the winners are drawn using a secure and transparent process.
2. It is private, as the participants' personal information is not shared with third parties.
3. It is transparent, as the entire draw process is recorded on the blockchain.
4. It is verifiable, as the results of the draw can be verified by anyone.

II. LITERATURE SURVEY

A. *Review:*

Blockchain technology can provide transparency and fairness in a lucky draw system by utilizing smart contracts to define and enforce the rules of the draw. The immutable nature of blockchain ensures that the draw process is transparent, and the outcome is tamper-proof, providing participants with trust in the system. Security and fraud prevention: Blockchain can enhance the security of a lucky draw system by using cryptographic techniques to secure transactions and participant information.

The decentralized nature of blockchain reduces the risk of fraud, as the system’s operations are distributed among multiple nodes, making it difficult for any single entity to manipulate the results. Decentralization and trust: Blockchain’s decentralized nature eliminates the need for a central authority or intermediary in a lucky draw system. This reduces the reliance on trust in a single entity and enables trust among.

B. *Bitcoin And Blockchain*

Bitcoin garnered enormous popularity in recent years and has become a household name. However, most of the people are not aware of the underlying technology blockchain that powers bitcoin. Bitcoin is a decentralized digital currency not regulated by any cen-tral bank or a single administration, it can be sent from a user-to-user on the peer-to-peer bitcoin network without the need for intermediaries. Transactions (movement of bitcoin from one account to another account) are verified by the nodes connected in the network through cryptography and recorded into a public distributed ledger called blockchain. Bitcoin was created by an anonymous person or a group of people going by the name Satoshi Nakamoto and released in the year 2009 as an open-source software. Bitcoins are rewarded to the nodes connected in the network when they perform a process called mining. Bitcoins are now accepted in various places for exchange of local currency, goods or services.

In the bitcoin white paper written by Satoshi Nakamoto and released in year 2008, an electronic coin is defined as a chain of digital signatures. Ownership of a coin is transferred when the owner of a coin initiates a new transaction by digitally signing a hash of both owner's public key and the previous transaction that is added to the end of the coin. Af-ter signing, a new transaction is added to the chain as shown in Bitcoin transaction chain of ownership in Figure 1 bitcoin transaction chain of ownership. Transactions can include multiple inputs, all of them have to be signed individually, as well as multiple outputs. The 6 main reason behind this design is the coins do not have to be handled individually, similar to using the smallest unit of currency such as pennies, instead can be combined and split when needed in transactions.

This transaction chain system makes it possible for the recipient of the coin to validate the chain of coin ownership. However, this raises a new problem where the recipient cannot prove that the coin has been already spent which is commonly known as double spending problem. Previous currency systems similar to bitcoin had run into the same double spend-ing problem. The major breakthrough achieved by bitcoin was the development of a new mechanism that enables it to work without relying on a trusted third party to validate the transactions as most of the crypto currencies that preceded it. The new mechanism devised by bitcoin was Blockchain, a decentralized digital public ledger managed by a peer-to-peer network to maintain consensus on the current state of the system. With the consensus built into the system it makes it impossible to spend the same coin twice because everyone on the network, via the block mining algorithm they run, will agree on the same sequence of transactions that determine the current state of the coin ownership.

C. *Ethereum*

Ethereum is an open-source public distributed computing platform and operating sys-tem based on blockchain technology first used by Bitcoin. Ethereum extends the usefulness of Blockchain well beyond cryptocurrencies by making the blockchain programmable accord-ing to the developer's needs. It was proposed by a cryptocurrency researcher Vitalik Buterin in his whitepaper published in the year 2013, where he states the intention of Ethereum is to provide, "a blockchain with a built-in fully fledged Turing-complete programming language that can be used to create "contracts" that are used to encode arbitrary state transition functions". These features make it the apt choice for building truly decentralized applications similar to this project and many other decentralized applications (dapp). Although Ethereum blockchain is much more advanced and intricate, it is still based on the same principles as Bitcoin's. Ethereum, similar to Bitcoin, also uses a proof-of-work algorithm run by a peer-to-peer distributed network to find consensus on the current state of the system, with the miners being rewarded in Ether (crypto currency used by Ethereum network). Network gets transactions from users distributed across the globe and the proof of work algorithm at regular intervals determines a sequence of those transactions to be in-cluded in the next block in the blockchain. Every new block added to the chain determines the state of the system. The block creation time in the case of Ethereum averages around 14 seconds while that of Bitcoin averages around 10 minutes, both operate on the same set of core principles of blockchain.

The major difference in Ethereum is the complexity of both, the state stored by the blockchain and how the transactions can alter the state of the blockchain. Ethereum's state mainly consists of objects called accounts located by a 20-byte address [7]. There are two types of accounts, externally owned accounts and contract accounts. Externally owned accounts or EOAs can be accessed by private keys similar to Bitcoin and have a field to store the current ether balance of that account. Contract accounts or CAs similar to EOAs have additional fields to store the contract code and storage. As CAs are a part of contracts, their interaction with other accounts and how they access or modify their storage are controlled by the own-ing contract. The contract storage is a key value store of data persistent among transactions.

The actual state changes are done by the Ethereum virtual machine (EVM) upon receiving a transaction, by running the low-level contract bytecode. A fascinating concept called gas is used by EVM to operate; it can be thought of as fuel purchased to execute a transaction. Bitcoin also allows scripting for complicated transactions but it is not Turing complete like Ethereum. Ethereum solved a major problem that Bitcoin had by finding a solution to transactions that involve loops. In case a transaction consisted of an infinite loop then calculating the state would be impossible, computational time would be massive, that can create problems for the blockchain. Ethereum's fairly simple solution to overcome this problem was to have a cost for each transaction that it executed. For every transaction a user creates the amount of gas required for the transaction to be processed is purchased from the ether balance in the sender's account at an arbitrary price, typically the market price at that moment (the amount of gas sent directly affects the transaction processing time as the miner running the Ethereum blockchain get that gas amount is a incentive for mining so higher amount create higher incentive). Every bytecode operation done in the EVM costs a certain integer amount of gas, operations such as modify or add to contract storage are the most expensive because all those changes are persisted on the blockchain forever. The gas starts depleting when a transaction starts executing and stops if it completes or runs out of gas. If the transaction completes and some gas is left behind then it is refunded back to the transaction sender. However, if it terminates because of the gas amount depleted to zero then no ether is returned and the transaction fails.

D. *Web 3.0*

Web 3.0, or simply Web3, is a novel conception of the architecture that underpins the internet, and it follows from two broad "iterations" of the world wide web (Web 1.0 and 2.0). The term was coined by Gavin Wood (of Polkadot and

Ethereum fame), and has gar-nered 4 considerable currencies among certain futurists in the recent past. For the purposes of comparison, Web 1.0 was characterized by rudimentary read-only passivity in users, who would browse a generally clunky and poorly organized system of limited accessibility at low bandwidths. Web2, which is our current state, and which has dominated for a good 15 years now, is characterized by a more dualist interactivity among the producers and consumers of content. For example, people tweeting, blogging, liking, reviewing, and posting; all rep-resent active forms of content creation over platforms. These activities create value in and of themselves, and users thus absorb content created by others as much as they go about creating content of their own.

Web 3.0, or simply Web3, is a novel conception of the architecture that underpins the internet, and it follows from two broad "iterations" of the world wide web (Web 1.0 and 2.0). The term was coined by Gavin Wood (of Polkadot and Ethereum fame), and has gar-nered 4 considerable currencies among certain futurists in the recent past. For the purposes of comparison, Web 1.0 was characterized by rudimentary read-only passivity in users, who would browse a generally clunky and poorly organized system of limited accessibility at low bandwidths. Web2, which is our current state, and which has dominated for a good 15 years now, is characterized by a more dualist interactivity among the producers and consumers of content. For example, people tweeting, blogging, liking, reviewing, and posting; all rep-resent active forms of content creation over platforms. These activities create value in and of themselves, and users thus absorb content created by others as much as they go about creating content of their own.

III. Proposed Algorithms

A. Randao

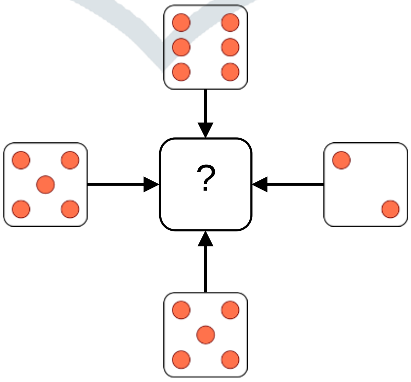
Randao is based on Blockchain technology and provides the service of random number generation that is open source, decentralized, socialized and verifiably fair. Randao not only has the characteristics of uncontrollability and unpredictability which it inherits from the common random number generator, but also is more accessible and provably fair. Ran-dao helps individuals to observe their impact on the generation process of random numbers through providing each stakeholder participation channel. The transparent, irreversible gen-eration process ensures the fairness of the random numbers.

With the Randao service, users can quickly build a verifiably fair application for different scenarios, for example, public management, entertainment, sports, finance, and corporate management.The simplest version of RANDAO is to have each member of a group come up with a random number on their own. We can then take these random numbers and ”mix” them together in a way that each person’s number has an equal impact on the final result.

RANDAO systems need some sort of ”mixing” function that input values from each partic-ipant and produces some output value. In a very high-level sense, mixing functions always take the form:

mix(input1, input2, ..., input n) > output

Where each input value corresponds to a number contributed by a RANDAO participant. When picking a mixing function, we need to be careful to preserve the key property that each input value has equal ”influence” on the output value. We can get an intuitive feel for this property by looking for ”bad” mixing functions. We clearly want to avoid, for example, mixing functions that completely throw out every second input value or simply always return the same constant output value.



B. An Aside-Certifiable Function:

It’s possible to improve upon the ”commit-reveal” version of RANDAO using some more fancy math. Although not currently included in the official Eth2 specification, Verifiable Delay Functions, or VDFs, were developed by researchers at Stanford in 2018 to this end. VDFs are, effectively, algorithms that take a long time to execute and can’t be sped up by running the algorithm on multiple computers at the same time.

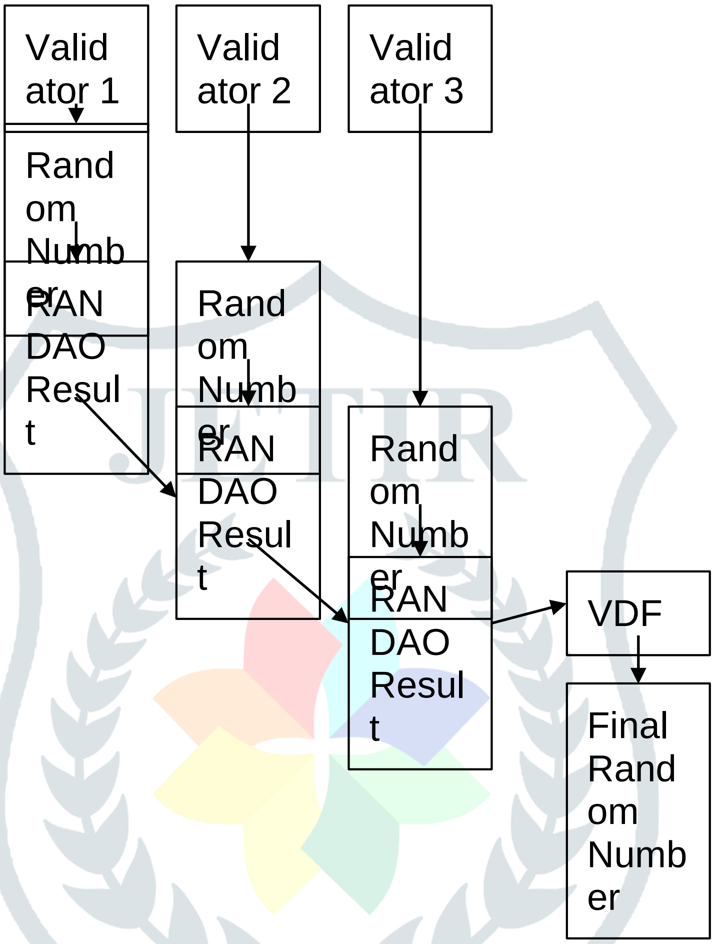
VDFs have the following function signature:

input > (output, proof)

VDFs have a corresponding verification algorithm that, given proof tells us whether output is correct for the given input:

(input, output, proof) > *true, false*

Although it takes a long time to compute the output, it only takes a short period of time to use the proof to verify that the output was, indeed, correct. We can tune a parameter in the VDF to change the amount of time that an average computer will take to find an output. VDFs are interesting in their own right, but they shine in combination with RANDAO. Instead of simply using the result of RANDAO as our random number, we can first feed this result into a VDF and use the output as our final random number.



IV. SYSTEM ARCHITECTURE

Our design makes an effort to take into account both third-party independence and security. DeLottery aims to eliminate unnecessary steps from the lottery event process in order to make it as similar to the basic lottery process as possible. Here, we demonstrate how DeLottery was created utilizing a blockchain with smart contracts. Figure 1 depicts the structure of the high phase system. Blockchain connects users by leveraging their digital devices as chain nodes. And the same blockchain hosts the lottery smart contract. One smart contract that has been deployed on an arbitrary account is shared by all players that take part in the same lottery event. Then communication between the lottery system and the players begins.

It's no secret that the lottery event process can be complicated and drawn-out. That's why we at DeLottery have sought to make it simpler by utilizing blockchain technology with smart contracts. Our design makes an effort to take into account both third-party independence and security, so participants can rest assured their data is safe. Figure 1 depicts the structure of our high phase system which connects users through digital devices as chain nodes on a single blockchain hosting the lottery smart contract. This allows for a streamlined experience, eliminating unnecessary steps from traditional lotteries while still maintaining its basic nature – all without sacrificing safety or security!

The deployed arbitrary account shared between players in each lottery event ensures that everyone has access to all necessary information regarding their participation in any given game - such as ticket numbers, prize amounts etcetera - without having to worry about potential interference from outside sources who may not have been intended recipients of said information.. In addition, this also eliminates any need for manual verifications or record keeping; everything is securely stored within the same network allowing easy access when needed.

Consider a lottery system with a large number of users, a user who deploys the contract at random, and users who can interact with the contract.

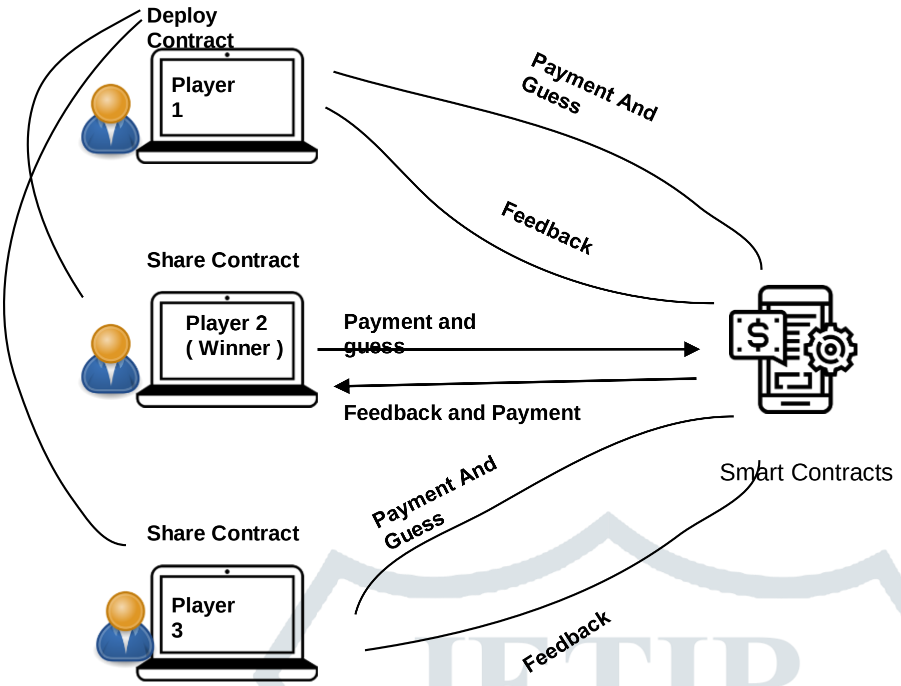
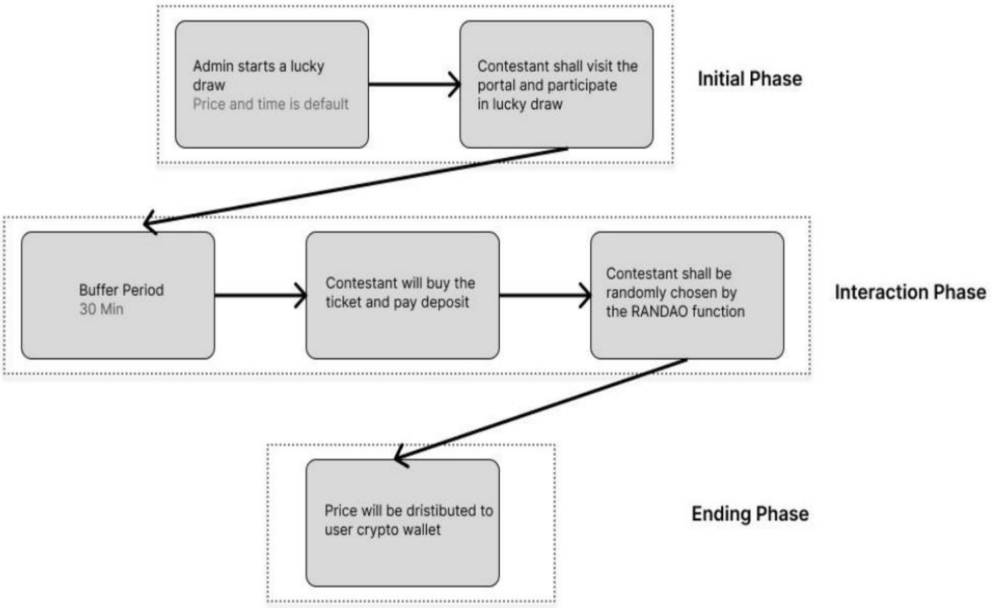


Figure 1

At the very beginning of the lottery procedure, the arbitrary host is the only account deployed with DeLottery. We design the lottery system using a high phase finite state machine diagram shown in the Figure 2. The seven steps in DeLottery procedure can be summarized in three phases with each phase possessing two steps: initial phase, interaction phase, and ending phase. Phases are indicated by different rows in Figure 2. Based on the diagram and phases we designed, to store the critical data of the lottery system and users and to allow the users to interact with the system, the contract should have the following several variables as shown in Table 2. These variables are set to support the complete lottery process, and to ensure that the system is decentralized. As a typical lottery process, it contains all variables in Table 1, as well as specialized variables included in DeLottery.


Admin starts a lucky draw
Price and time is default
Contestant shall visit the portal and participate in lucky draw
Initial Phase
Buffer Period
30 Min
Contestant will buy the ticket and pay deposit
Contestant shall be randomly chosen by the RANDAO function
Interaction Phase
Price will be dristibuted to user crypto wallet
Ending Phase

Our design tries to take both security and independence of a third party into consideration. The aim of our project is to simplify the Lucky Draw event procedure to be as close to the core Lucky Draw procedure as possible, avoiding redundant steps. Here we show how our project is designed by using smart contracts in blockchain. The high phase system structure is shown in Figure 5. All the Contestants and participants are connected in blockchain using their digital devices as nodes on the chain with the help of the digital wallet Metamask and platform that we are building. And the lucky draw smart contract is deployed on the same blockchain. All contes-tants participate in the same lucky draw event, so they share one smart contract deployed on an admin account that is deployed on a third web platform. Then the contestants and the lucky draw system start interacting. Consider that there are many users in a lucky draw system, this contract is deployed by an admin user and 100 users can participate in this contract.

A random number is generated and recorded on the blockchain Since the random number generator is based on the blockchain, the algorithm relies on recent random blockchain transactions. It pulls a specified amount and order of numbers to generate each winning number sequence. Because nobody is aware of the next transaction in the blockchain, the Lucky Draw platform adds an extra layer of randomness to the selection process. Once the random number is generated and matched to the player's ticket number, they are awarded and money is automatically sent to their respective wallets. The funds to be distributed to players are defined in the smart contracts. Therefore, the decided commissions and funds are paid out to every player on the platform. Moreover, the smart contract code is available publicly on the platform. Players can check the rules defined in the contracts to confirm if the funds are distributed in a fair way or not .

Players can trace back the history of records of transactions Since the transactions are recorded on the blockchain, players can trace back the history to know who had won the jackpot and if the commissions and wins are paid out as defined in the smart contracts.

At the very beginning of the Lucky Draw procedure, the admin host is the only account deployed with the Lucky draw system. We design the lucky draw system using a high phase finite state machine diagram shown in Figure 5. The six steps in the Lucky Draw procedure can be summarized in three phases with each phase possessing two steps: initial phase, interaction phase, and ending phase. we designed, to store the critical data of the Lucky Draw system and users and to allow the users to interact with the system.

Admin announces the ticket openings and deploys the smart contracts Admin announces the ticket openings on the platform and the notification is sent to the users. They also deploy smart contracts, which contain predefined rules for the Lucky Draw game to bring fairness and transparency to the ecosystem. Smart contracts ensure what information should be shared with which stakeholder in the system, providing privacy and disclosure of data. Since the players can buy the tickets with cryptocurrencies, their identities remain anonymous. The transactions stored on a public blockchain allows traceability and makes it easier to resolve disputes/scandals related to lucky draws.

A random number is generated and recorded on the blockchain Since the random number generator is based on the blockchain, the algorithm relies on recentrandom blockchain transactions. It pulls a specified amount and order of numbers to generate each winning number sequence. Because nobody is aware of the next transaction in the blockchain, the Lucky Draw platform adds an extra layer of randomness to the selection process. Once the random number is generated and matched to the player's ticket number, they are awarded and money is automatically sent to their respective wallets. The funds to be distributed to players are defined in the smart contracts. Therefore, the decided commissions and funds are paid out to every player on the platform. Moreover, the smart contract code is available publicly on the platform. Players can check the rules defined in the contracts to confirm if the funds are distributed in a fair way or not. Players can trace back the history of records of transactions Since the transactions are recorded on the blockchain, players can trace back the history to know who had won the jackpot and if the commissions and wins are paid out as defined in the smart contracts.

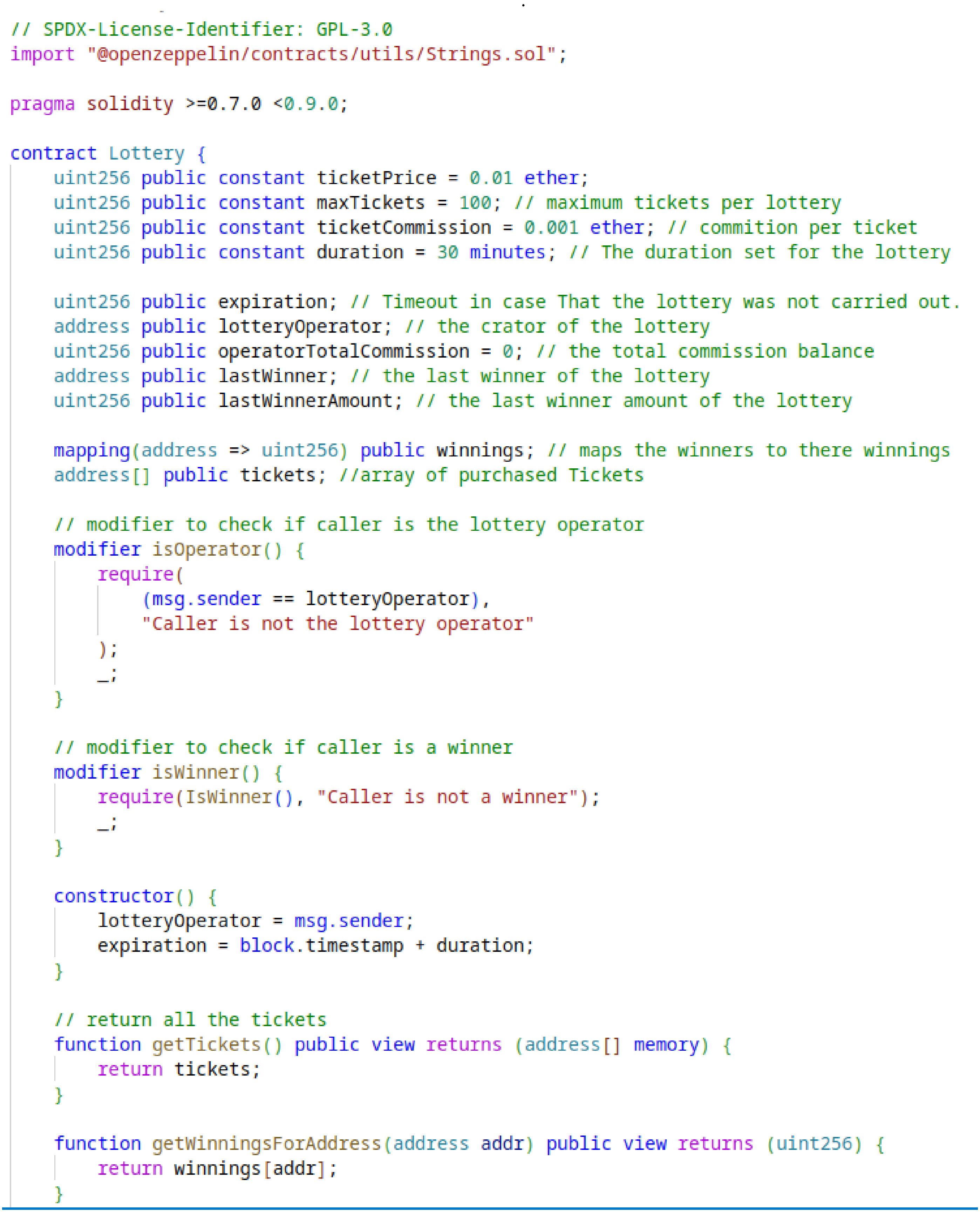
V. IMPLEMENTATION AND RESULT

A smart contract is a self-executing program that runs on a blockchain and can automate the execution of digital contracts without the need for intermediaries. Solidity is one of the most popular programming languages for writing smart contracts on the Ethereum blockchain. Here is some information about smart contracts and Solidity: Smart contracts are stored on the blockchain and are executed automatically when certain conditions are met. They can be used to implement various types of agreements, including financial contracts, supply chain agreements, and voting systems. Smart contracts are immutable, which means that once deployed to the blockchain, their code cannot be changed or tampered with. Solidity is a high-level programming language that is specifically designed for writing smart contracts on the Ethereum blockchain. It is a statically typed language, meaning that variables must be declared with a specific data type. Solidity supports various data types, including integers, strings, booleans, and arrays. It also includes features for inheritance, interfaces, and libraries, allowing for modular and reusable code. Solidity code is compiled into bytecode, which can be executed on the Ethereum Virtual Machine (EVM). Solidity also includes a testing framework that allows developers to write and run tests for their smart contracts. To deploy a smart contract written in Solidity, developers must use a blockchain development platform such as Remix or Truffle.

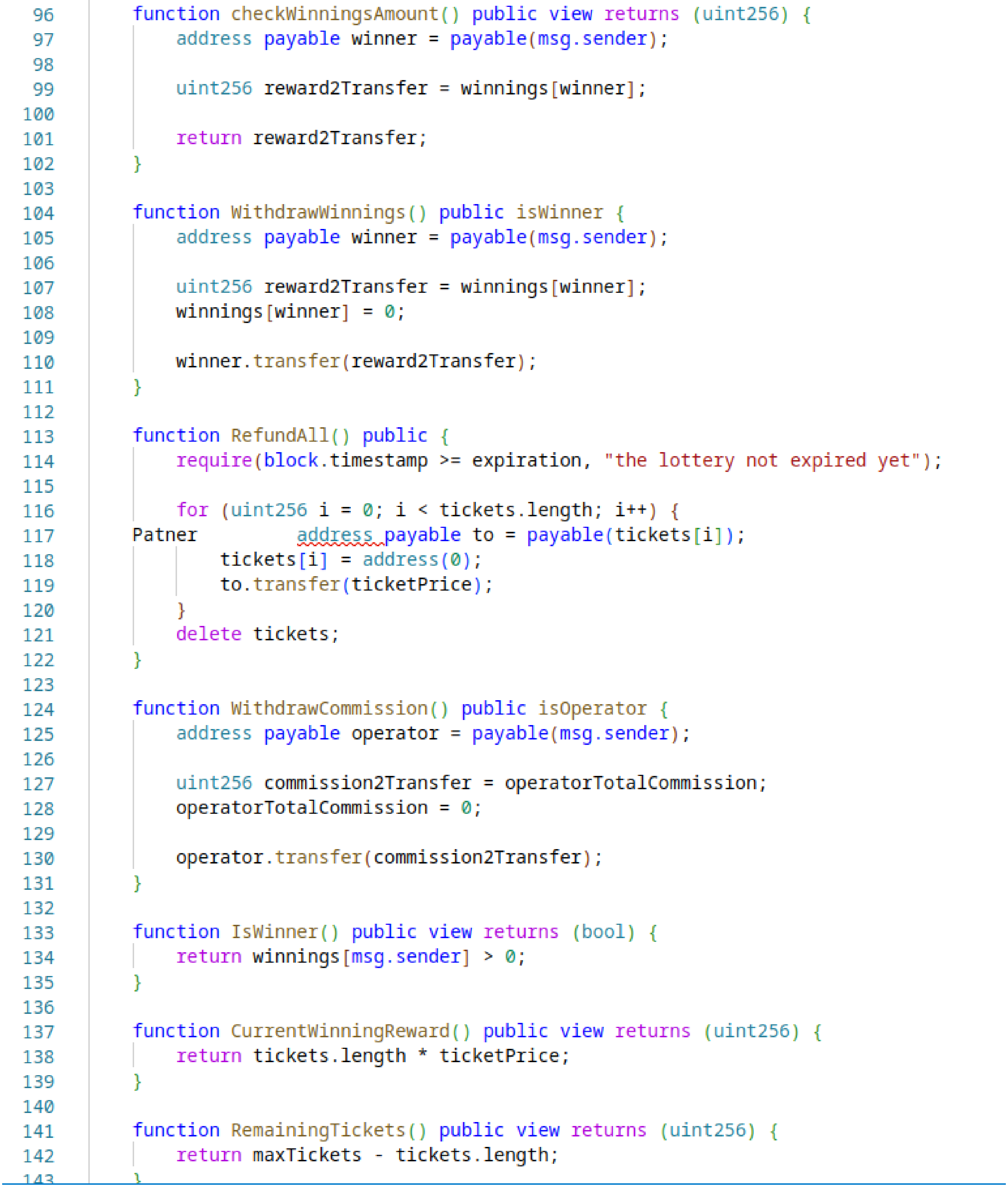
Randao is a secure and unbiased RNG. It is based on the Verifiable Random Function (VRF) protocol, which is a secure and efficient way to generate random numbers. Randao is used in a variety of applications in blockchains, such as block production, staking, and governance. Randao is a complex RNG, which can make it difficult to understand and use. It can also be slow and energy-intensive, especially for large numbers of participants.

Overall, Randao is a secure and reliable RNG that is essential for the security and functionality of blockchains. However, it is important to be aware of the limitations of Randao, such as its complexity, performance, and energy consumption.

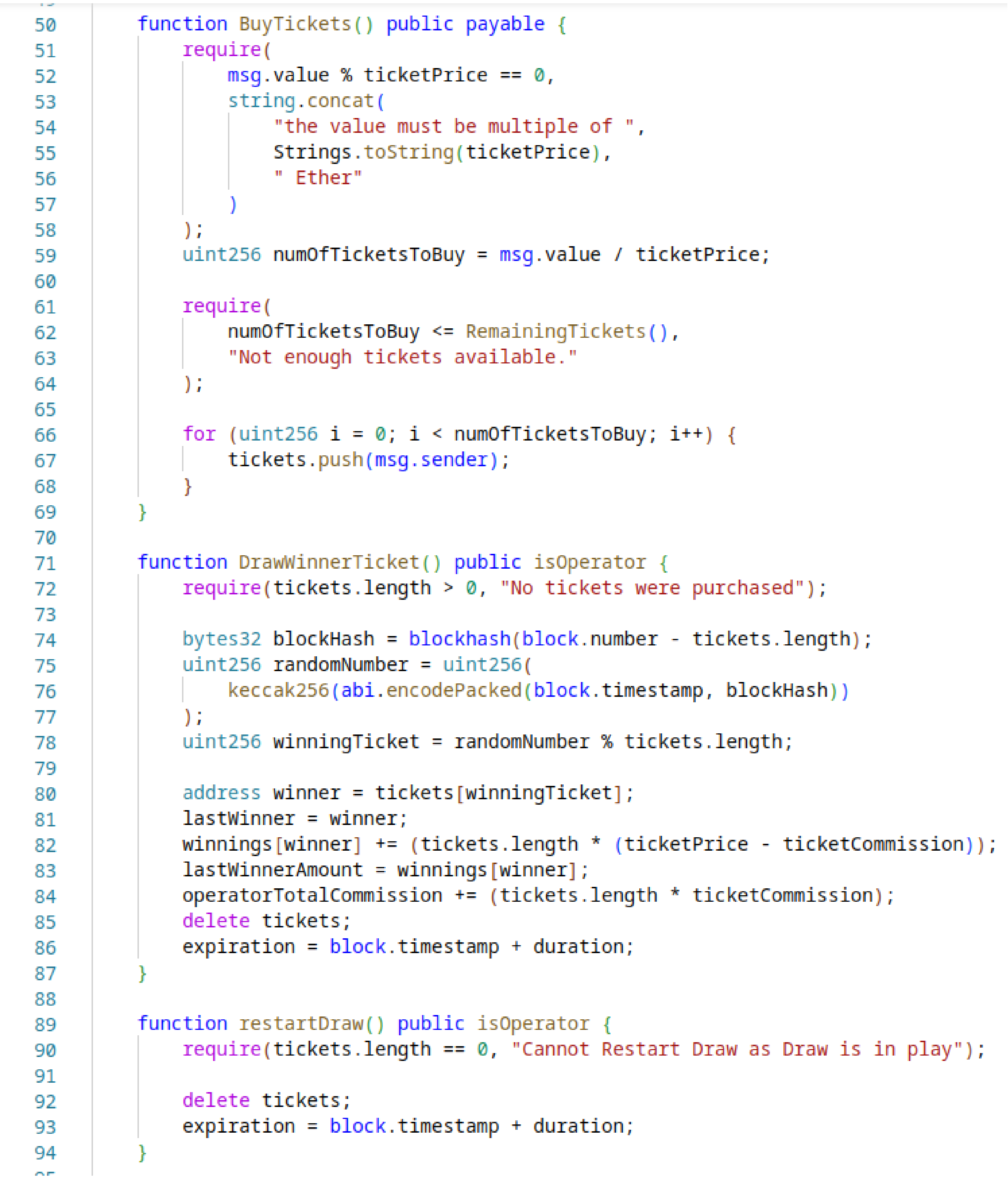
A. Solidity Code for Smart Contract:



```
// SPDX-License-Identifier: GPL-3.0
import "@openzeppelin/contracts/utils/Strings.sol";

pragma solidity >=0.7.0 <0.9.0;

contract Lottery {
    uint256 public constant ticketPrice = 0.01 ether;
    uint256 public constant maxTickets = 100; // maximum tickets per lottery
    uint256 public constant ticketCommission = 0.001 ether; // commition per ticket
    uint256 public constant duration = 30 minutes; // The duration set for the lottery

    uint256 public expiration; // Timeout in case That the lottery was not carried out.
    address public lotteryOperator; // the crator of the lottery
    uint256 public operatorTotalCommission = 0; // the total commission balance
    address public lastWinner; // the last winner of the lottery
    uint256 public lastWinnerAmount; // the last winner amount of the lottery

    mapping(address => uint256) public winnings; // maps the winners to there winnings
    address[] public tickets; //array of purchased Tickets

    // modifier to check if caller is the lottery operator
    modifier isOperator() {
        require(
            (msg.sender == lotteryOperator),
            "Caller is not the lottery operator"
        );
        _;
    }

    // modifier to check if caller is a winner
    modifier isWinner() {
        require(IsWinner(), "Caller is not a winner");
        _;
    }

    constructor() {
        lotteryOperator = msg.sender;
        expiration = block.timestamp + duration;
    }

    // return all the tickets
    function getTickets() public view returns (address[] memory) {
        return tickets;
    }

    function getWinningsForAddress(address addr) public view returns (uint256) {
        return winnings[addr];
    }
```



```
    function checkWinningsAmount() public view returns (uint256) {
        address payable winner = payable(msg.sender);

        uint256 reward2Transfer = winnings[winner];

        return reward2Transfer;
    }

    function WithdrawWinnings() public isWinner {
        address payable winner = payable(msg.sender);

        uint256 reward2Transfer = winnings[winner];
        winnings[winner] = 0;

        winner.transfer(reward2Transfer);
    }

    function RefundAll() public {
        require(block.timestamp >= expiration, "the lottery not expired yet");

        for (uint256 i = 0; i < tickets.length; i++) {
    Patner         address payable to = payable(tickets[i]);
            tickets[i] = address(0);
            to.transfer(ticketPrice);
        }
        delete tickets;
    }

    function WithdrawCommission() public isOperator {
        address payable operator = payable(msg.sender);

        uint256 commission2Transfer = operatorTotalCommission;
        operatorTotalCommission = 0;

        operator.transfer(commission2Transfer);
    }

    function IsWinner() public view returns (bool) {
        return winnings[msg.sender] > 0;
    }

    function CurrentWinningReward() public view returns (uint256) {
        return tickets.length * ticketPrice;
    }

    function RemainingTickets() public view returns (uint256) {
        return maxTickets - tickets.length;
    }
```



```
    function BuyTickets() public payable {
        require(
            msg.value % ticketPrice == 0,
            string.concat(
                "the value must be multiple of ",
                Strings.toString(ticketPrice),
                " Ether"
            )
        );
        uint256 numOfTicketsToBuy = msg.value / ticketPrice;

        require(
            numOfTicketsToBuy <= RemainingTickets(),
            "Not enough tickets available."
        );

        for (uint256 i = 0; i < numOfTicketsToBuy; i++) {
            tickets.push(msg.sender);
        }
    }

    function DrawWinnerTicket() public isOperator {
        require(tickets.length > 0, "No tickets were purchased");

        bytes32 blockHash = blockhash(block.number - tickets.length);
        uint256 randomNumber = uint256(
            keccak256(abi.encodePacked(block.timestamp, blockHash))
        );
        uint256 winningTicket = randomNumber % tickets.length;

        address winner = tickets[winningTicket];
        lastWinner = winner;
        winnings[winner] += (tickets.length * (ticketPrice - ticketCommission));
        lastWinnerAmount = winnings[winner];
        operatorTotalCommission += (tickets.length * ticketCommission);
        delete tickets;
        expiration = block.timestamp + duration;
    }

    function restartDraw() public isOperator {
        require(tickets.length == 0, "Cannot Restart Draw as Draw is in play");

        delete tickets;
        expiration = block.timestamp + duration;
    }
```

VI. CONCLUSION AND FUTURE SCOPE

To create a lucky draw based decentralized application (Dapp) on Ethereum /Polygon network blockchain for increasing transparency and reducing frauds in the lucky draw industry. Once the contract has been deployed by the administrator, there will be a minimum contribution amount for players to register in the game and a price pool will be maintained by the smart contract. Lucky draws have been a classical form of entertainment and charity for centuries. We use smart contracts and blockchain in decentralized, intelligent, and secure systems for lucky draw industries. Moreover, we are inspired by the algorithm of RANDAO, an outstanding way of random number generation in the blockchain scenario.

The future scope of lucky draw system is promising and has the potential to revolutionize the way lucky draw are conducted.Here are some possible scenarios for the future scope of lucky draw system using Blockchain.

1. Transparency and Security: One of the key advantages of using blockchain in a lucky draw system is transparency and security. Blockchain, being a decentralized and immutable ledger, can ensure that the lucky draw process is transparent, fair, and tamper-proof. Participants can verify the results of the lucky draw on the blockchain, which eliminates any suspicion of fraud or manipulation. This increased transparency and security can boost participants' trust in the lucky draw system and attract more participation.
2. Global Participation: Blockchain-based lucky draw systems have the potential to attract participants from around the world. Traditional lucky draw systems often have geographical limitations, and participants need to be physically present or meet certain criteria to participate. With blockchain, participants can join the lucky draw from anywhere in the world, as long as they have internet access and meet the requirements set by the smart contract. This can greatly expand the reach of lucky draw systems and increase participation from a global audience.
3. Decentralized Governance: Blockchain-based lucky draw systems can also introduce decentralized governance models, where decisions related to the lucky draw process, such as rules, prize distribution, and future developments, are made collectively by the community through voting mechanisms. This can ensure that the lucky draw system is not controlled by a single entity but is instead governed by
4. Integration with Smart Devices: With the advent of the Internet of Things (IoT) and smart devices, blockchain-based lucky draw systems can be integrated with various smart devices, such as smartwatches, smart TVs, and smart speakers, to enable seamless participation in the lucky draw process. Participants can interact with the lucky draw system using voice commands, gestures, or other IoT-enabled interactions, making it more convenient and accessible.
5. In conclusion, the future scope of a lucky draw system using blockchain is promising and can bring about significant improvements in terms of transparency, security, global participation, tokenized rewards, decentralized governance, integration with smart devices, and enhanced user experience. As blockchain technology continues to mature and gain widespread adoption, the potential for innovation and disruption in the lucky draw industry using blockchain is significant.

REFERENCES

[1] Antonopoulos, A. (2017). Mastering Bitcoin: Programming the Open Blockchain. USA: O' Reilly media.

[2] Binance Academy. (2018). Proof of Stake Explained. Retrieved 03 01, 2021,

[3] Ehrsam, F. (2017). Blockchain Governance: Programming Our Future. Retrieved 05 27, 2021, from medium.com/@FEhrsam/blockchain-governanceprogrammingour-future-c3bfe30f2d74.

[4] Ethereum.org. (2021). Gas and Fees. Retrieved 06 10, 2021, from ethereum.org/it/developers/docs/gas/

[5] Mathews Emmanuel, Nimmy Chacko and T Anagha, A Blockchain based SmartCon-tract Digitized Lottery Scheme.

[6] Da-Yin Liao,Xuehong Wang, Design of a Blockchain-Based Lottery System for Smart Cities Applications.