Lottery DApp from Multi-Randomness Extraction

Yu-Chi Chen, Song-Yi Hsu, Ting-Wei Chang, and Ting-Wei Wu
Department of Computer Science and Engineering
Yuan Ze University
Taoyuan, Taiwan
wycchen@saturn.yzu.edu.tw,
{s1076022,s1041534,s1041502}@mail.yzu.edu.tw

***Abstract*—Randomness is a significant ingredient to achieve fairness on many applications (i.e., gambling). On the use of blockchain and smart contract, we observe that some existing lottery and gambling games involve one single source to produce randomness. As known, Fomo3D uses the game state to determine the winner, whereas TrueFlip uses the blockchain state (at a specific point in time). Those system do not rely on significant randomness extraction. In this paper, we present a lottery DApp and implement it by using smart contracts. In this DApp, the randomness is determined not only from game and blockchain states, but also from committees.**

***Index Terms*—Blockchian, Smart contracts, DApp, Randomness extraction**

I. Introduction

Blockchain [9] is a technology which offers many functionailities such as recording data without being tampered or reaching a distributed and permissionless consensus. It relies on cryptography to handle security of user and data, and then the such procedure can accurately work by the anaylsis from game theory and distributed algorithm. Blockchain network is based on a P2P network with plenty of nodes. Each node maintains the consistency of data on the blockchain. For data sharing over blockchain network, nodes do not rely on additional and centralized management mechanism. Instead, data are only distributed to nodes in the blockchain network, and then each node is in charge to verify the correctness, broadcast to the others, and manage the data on its side. Nodes can obtain consensus without any third-party intervention, and in particular this manner induces *relatively low cost* to achieve the trust and security. Bloclchain was named by Satoshi Nakamoto [9], and acts as the heart of cryptocurrecy (i.e., Bitcoin and Ethereum).

To generalize the activities of transactions, smart contract (SC) [5], [8] is the formulation on blockchain. In general, SC is built by some programming languages, and mainly is used for providing validation and executing the function. The SC has many faces, which can interact with other contracts, make decisions, store data and complete transactions, so it can achieve some applications like gambling [4]. The operations of smart contract will trigger the functions on the contract automatically if some conditions hold. Typically, a user will put the assets and deploy a SC in to the block on blockchain. As the transaction broadcast, the SC will be distributed to the nodes in blockchain network. When the other contract trigger the conditions in this SC, the inside functions will be executed. Now Ethereum is the well-prepared platform for smart contracts, and the applications produced by smart contracts are called *DApp*.

978-1-7281-1328-9/19/$31.00 ©2019 IEEE

Randomness is a critical factor to achieve fairness on lottery and gambling games. We observe that some existing lottery and gambling games involve one single sources to produce randomness; for example, Fomo3D [3] uses the game state to determine the winner, whereas TrueFlip [2] uses the blockchain state (at a specific point in time). In this paper, we follow the notion, randomness extraction, in theoretical computer science to build our lottery DApp in a simple way. Intuitively, inspired from the above games, the randomness usually comes from game and blockchain states; however, these are not guaranteed to be truly random. The n-bit output randomness might not be nice to provide fairness for the whole games. Hence, we use seeded randomness extraction which needs only $\log(n)$-bit purely random number as a seed, and then rely on the committee protocol to produce the such seed. If the committee protocol returns the acceptable seed, then we can have a nice randomness for our lottery DApp. Finally, the committee protocol is implemented by multi-party jointly to produce a random number, and the committee members can be elected by using hybrid consensus protocol [10]. The abstraction model of our system can be used to achieve the other styles of games.

II. Smart Contract-based Lottery System

We build a fair lottery system [7], and thus by using the concept of pseudorandomness [6], we propose a smart contract-based decentralizing system. This system is mainly composed of three entities: user, owner, and smart contract. In the system implementation, we use Solidity (contract-oriented programming language) [8] to prepare the smart contract on the Ethereum [1] blockchain. In the following, we will show the system overview, highlight the smart contract, and finally analyze the randomness.

A. Overview

The owner creates a smart contract and deploys it on the blockchain at first. When user joins at the first time, the smart contract (SC) will add the user into the system. The SC will send the associated permissions. Upon receiving permissions,

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:56:13 UTC from IEEE Xplore. Restrictions apply.

the user can choose a number which user want to bet, and then this number will be stored in blockchain. Moreover, the user can also send a query to the SC for his choice in this round. Finally in each round, the SC will get winning number, where the winning number will be randomly determined, and the such randomness is defined by 1) players 2) blockchain and 3) committees. Then, this suffices to determine the winner list, and then the SC will send back the reward to the winners according to the winner list.

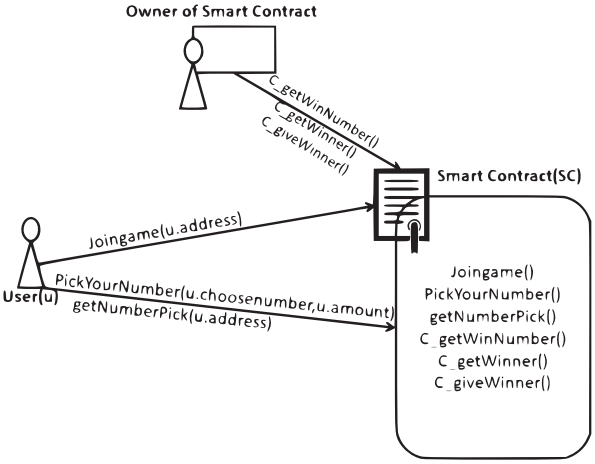

Fig. 1. System overview

B. Smart Contract

We highlight the functions provided by our smart contract[1] to meet the requirements.

a) joingame(u.address): This function which adds the user into the system can only be executed by the user. It takes the user's address(u.address) as input.

b) PickYourNumber(u.choosenumber,u.amount): This function which adds the number that user choose into the system can only be executed by the user. It takes the number choose by the user (u.choosenumber)and the amount that choose by user(u.amount) as inputs.

c) getNumberPick(u.address): This function which returns the number and amount that choose by the user can only be executed by the user. It takes the user's address(u.address) as input.

d) C_getWinnumber(): This function which return the Winning number from the system can only executed by the owner.

e) C_getWinner(): This function which return the Winner list from the system can only executed by the owner.

f) C_givewinner(s.Winning_number,s.Winner_list): This function which give the Winners total award can only executed by the owner. It takes the Winning number(s.Winning_number) and Winner list(s.Winner_list) as inputs.

[1]The code is located in Github. https://github.com/cislab-yzu/LotteryDApp

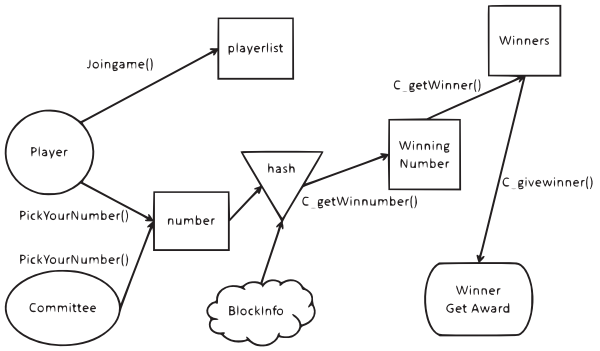

Fig. 2. Details of the smart contract

C. Randomness

We use multi-randomness methods to decide random numbers which comes from the following three ways.

a) From committees: A group of specific players is selected by paying or voting to form a committee, and then the committee member selects the number of random seed.

b) From blockchain: We use the Ethereum to select random numbers. Ethereum has the hash function keccak256, which uses the SHA3 (Secure Hash Algorithm 3). A hash function basically convert a string into a random 256-bit hexidecimal number. A tiny change in the input will cause a large change in the hash.

c) f(choice): All players selects the number of random seed.

We use two district sources (including lottery and blockchain states) and a seed(committees) to extract randomness by using hash functions. For implementation on the smart contract, we use hash function with input concatenating committee's output, blockchain state, and game state. Theoretically, to produce a n-bit random number as output, we require the following assumptions:

- hash functions are almost perfect randomness extractor;
- the sources are k-source (such that the random variable X is k-source if min-entropy of X is least than k);
- the seed is pure random.

By the system setting, the seed is honestly produced by committees. The states of blockchain and lottery are implied by computation power and lottery stake, and thus our system can work if these two parts are not corrupted.

III. Conclusions

In this paper, we present a lottery DApp and implement it by using smart contracts. In this DApp, the randomness is determined not only from game and blockchain states, but also from committees. In the future work, we will aim for investigating the committee protocol for simplification.

Acknowledgements

This work supported in part by Ministry of Science and Technology of Taiwan, under grant 106-2218-E-115-008-MY3, and Innovation Center for Big Data and Digital Convergence, Yuan Ze University.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:56:13 UTC from IEEE Xplore. Restrictions apply.

REFERENCES

[1] The ethereum. https://www.ethereum.org/, 2017.

[2] True flip white paper. https://trueflip.io/TrueFlip_WP.pdf, 2017.

[3] Fomo3d. https://fomo3d.hostedwiki.co/, 2018.

[4] Vanchai Ariyabuddhiphongs. Lottery gambling: A review. *Journal of Gambling Studies*, 27(1):15–33, 2011.

[5] Massimo Bartoletti and Livio Pompianu. An empirical analysis of smart contracts: platforms, applications, and design patterns. In *International Conference on Financial Cryptography and Data Security*, pages 494–509. Springer, 2017.

[6] Stéphane Grumbach and Robert Riemann. Distributed random process for a large-scale peer-to-peer lottery. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 34–48. Springer, 2017.

[7] Pramote Kuacharoen. Design and implementation of a secure online lottery system. In *International Conference on Advances in Information Technology*, pages 94–105. Springer, 2012.

[8] Da-Yin Liao and Xuehong Wang. Design of a blockchain-based lottery system for smart cities applications. In *Collaboration and Internet Computing (CIC), 2017 IEEE 3rd International Conference on*, pages 275–282. IEEE, 2017.

[9] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[10] Rafael Pass and Elaine Shi. Hybrid consensus: Efficient consensus in the permissionless model. In *International Symposium on Distributed Computing*, page 6, 2017.

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on April 15,2023 at 06:56:13 UTC from IEEE Xplore. Restrictions apply.