D-Lotto: The Lottery DApp with Verifiable Randomness

Kunal Sahitya and Bhavesh Borisaniya

Abstract The true ingredient of any lottery system to be successful is randomness of its underlying algorithm. Since the introduction of gambling and lottery industry into cryptocurrency market, trust and security have been less of a concern for organizations, unlike randomness and verifiability. Many of the existing lottery designs use dynamic attributes of either game or blockchain to introduce randomness in its algorithm and only a handful of them are verifiable. In this paper, we introduce a lottery system design having novel random function, which uses combination of game and blockchain state to produce randomness in underlying algorithm and is verifiable. Proposed lottery DApp, built on Ethereum platform, includes smart contracts that help system to achieve properties like decentralization, transparency, and immutability. These properties combined with randomness and verifiability lead to this significant lottery design to be unique of its kind.

Keywords Lottery · Ethereum · DApp · Randomness · Verifiability

1 Introduction

Cryptocurrency and lottery shares similarities at Indian subcontinent of being topics for discussion and partial acceptance rather than having wholehearted technological embrace. While on the other hand, many of the European countries already have blockchain-based fortune lottery companies like FireLotto [1] and Kibo [2] making trillion dollar profit-count annually [3]. Lottery is legalized in less than half states of India, but surprisingly, it estimates to generate per annum revenue of around fifty thousand crores alone for states and companies despite taking hit from tax increment by government [4]. Online lotteries have seen its fair share of failures like

K. Sahitya (✉) · B. Borisaniya
Shantilal Shah Engineering College, Bhavnagar, Gujarat, India
e-mail: kunal2sahitya@gmail.com

B. Borisaniya
e-mail: borisaniyabhavesh@gmail.com

K. Kotecha et al. (eds.), *Data Science and Intelligent Applications*,
Lecture Notes on Data Engineering and Communications Technologies 52,
https://doi.org/10.1007/978-981-15-4474-3_4

HotLotto scandal [5], which included theft of 14.3 Million US dollars by deployment of self-destructing malware to affect randomness of underlying algorithm and many more.

A typical lottery system includes three phases: announcement of lottery, purchase of lottery and lottery drawing, and distribution of winnings. A traditional process is physical, tiresome, and error-prone. On the other hand, e-lotteries have improved in terms of speed and security; however, still the system lacks transparency and is vulnerable to single point of failure. Now has come an era of distributed, fully transparent, and immutable lottery systems which work without a slightest intervention of third-party organizers. Formal-mentioned systems are either independent distributed peer-to-peer lotteries or they are built on blockchain platforms like Ethereum, Neo, Cardano, Zilliqa, etc. Highlights of such systems include fully decentralized, distributed, peer-to-peer, transparent, immutable, based on fiat as well as cryptocurrencies, and operable worldwide. To solve the disbelief in participants toward any electronic lottery system, it must ensure qualities like, unpredictability, intrigue-resistivity, immutability, open membership, verifiability, and auditability [6]. Hence, it is hard to achieve transparency while ensuring the trust in electronic lotteries. When blockchain platforms combined with lottery system design can help with decentralization and immutability, and it is hard to achieve trustworthy randomness in that case. The design of lottery system discussed in this paper safeguards all of the above criteria that make it novel.

The rest of the paper is organized as follows: Section 2 discusses the background of lottery and related work in terms of blockchain-based lotteries. Section 3 focuses on actual system design and its underlying algorithm description. Properties of proposed system as whole and related advantages are discussed in Sect. 4 with conclusion and references at the end.

2 Background Theory and Related Works

Ethereum is an open source, distributed, and decentralized platform, which supports smart contracts. It is mainly known for its adoption of detailed scripting languages in cryptocurrency domain which can be used to write smart contracts. Cyptocurrency of Ethereum is known as ether. Smart contract is a tamper-proof, immutable program executing the terms mentioned in its transactional protocol. Smart contracts run on virtual machine (VM) supported by every node running on the network in a distributed manner. Ethereum supports its own higher level programming language called Solidity to write smart contracts. DApp is nothing but combination of one or more smart contracts to develop a piece of decentralized software. Inspired from these ideas, proposed system is designed.

The proof of first lottery drawn by human kind can be found in fifteenth century. Modern lotteries run by state government began around 60s in New Hampshire, USA, to generate revenue without incrementing taxes [7]. Hence, the lottery has been around since 600 years without changing a lot other than its forms. Computer has surely taken place of traditional lotteries as in storing the data of participants and drawing winners to transferring funds, but at the heart of it, basics remain the same. Many new forms of lotteries can be found in gambling ranging from instantaneous reward, lotto, number games, and scratch lottery, etc. [7].

Blockchain-based lottery systems are still finding its way in the maze of this new combination of cryptocurrencies and lottery scheme. Liao and Wang [8] have described blockchain-based lottery scheme as smart city application. They used a cryptographic model called hawk, to hide sensitive information from the participants on blockchain. Authors assert this model as lightweight enough to be implemented using IoT devices in new smart cities applications. Jia et al. [9] has proposed a lottery model built using smart contracts in Solidity language and deployed on Ethereum blockchain network. It uses existing RANDAO algorithm [6] to generate random numbers for lottery drawings. This model asserts resistant against Sybil and Node Attacks, and runs without any third-party intervention. Chen et al. [10] define a lottery DApp which determines its randomness not only from game state and blockchain state, but also using a specific set of people called 'committee' from the list of joined participants. Other good electronic lottery examples include [11, 12], and [13] that show properties like multi-level hash chain result drawing, verifiability, and distributed architecture, respectively.

Cryptocurrency and blockchain technology together can surely transform the era of modern gambling. Though, there have been a handful of attempts to develop lottery DApp, proposed system is unique due to its verifiability and novel randomness algorithm.

3 D-lotto System Design

D-lotto is a lottery DApp design that ensures lottery procedure to be fair, transparent, and verifiable. It is important to stick as close as possible to basic lottery system design while improving it. Hence, D-lotto has kept overall lottery design untouched as depicted in Fig. 1. While dealing with D-lotto, any dealer can initiate the new lottery using D-lotto interface using their electronic device. Any interaction medium including WebApp, DesktopApp, or an android application can be used as D-lotto interface. Participants have separate interface instance for joining the same lottery and they have that particular smart lottery contract in common. Also, it is not mandatory for dealer to not participate in lottery. He can behave same as other participants once the lottery is initiated, and that shows the real strength of immutability in smart contracts. Because not even creator of smart contracts can tamper them once they are deployed on network. Hence, if an individual wants to participate, we can treat even the dealer/organizer of lottery as any other participant after deployment of D-lotto. Below are the phases of D-lotto system design.

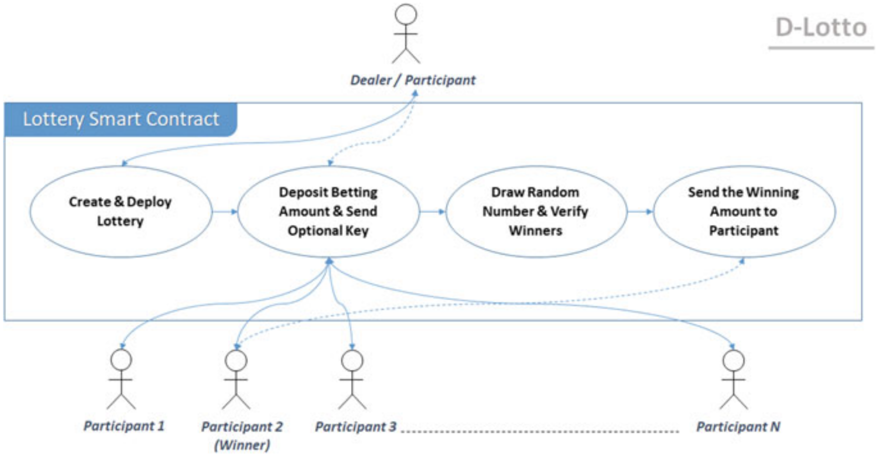

Fig. 1 D-lotto usecase diagram

3.1 Announcement of Lottery

This phase resembles with core lottery design facing slight variation. Dealer declares the own lottery scheme using D-lotto interface. Smart contract of that particular lottery instance is deployed on network by dealer after paying required transaction fee for same. Dealer defines and discloses everything from betting amount to number of maximum participants, pooling amount, winning amount, maintenance/dealer's share, lottery starting and ending time, etc., in detail for participants joining. Now, ideal betting share of each participant can be decided by following equation.

$$B_s = \frac{W_a + D_p + D_c}{T_g} \times 10^{1/\ln(S_f)} \tag{1}$$

Here, B_s is betting share, W_a is winning amount, D_p is dealer profit (Optional), D_c is lottery DApp deployment charges, and B_s, W_a, D_p, $D_c \in R+$. T_g defines number of tickets generated and S_f is a security factor, where $T_g \in N$ and $S_f \in [1, 2]$.

Security factor is used to keep betting share high, so that counterfeiting using dummy nodes and addiction to game can be avoided. Generated extra revenue can be used to declare extra jackpot prices or a consolation price for all the participants. After deciding details of lottery, dealer pays the transaction charges to deploy contract on Ethereum network and lottery goes live, provided having its operative interface. Individuals can have its own strategy to promote particular lottery scheme.

3.2 Purchase of Lottery

Depositing Betting Amount: After deployment of lottery by its dealer, system starts accepting the participant requests at a dealer specified start time through smart contract. Participants can simply join the lottery by paying the mentioned betting share calculated using Eq. 1. Single participant can buy more than one lottery tickets. If a participant is buying Ts tickets, his payment can be managed as per following equation:

$$T_s = \frac{P_a - R_a}{B_s} \tag{2}$$

Here, T_s is tickets share, R_a is returned amount, and P_a is paid amount. If a participant is not paying in whole multiples of B_s, he will not be awarded a full ticket and his remaining amount R_a will be sent back to his account by smart contract itself. For example, if a participant pays 110 ETH for 50 ETH B_s in lottery, he will be awarded two tickets as per Eq. 2 and his remaining amount of 10 ETH will be sent back to his address.

Sending an Optional Key: To participate in randomness drawn by D-lotto drawing algorithm 1, every participant can submit an optional random key of k hexadecimal bits or $k \times 4$ binary bits of his choice. Value of k can be arbitrary or can be best determined based on the factors in D-lotto drawing algorithm. Despite of buying more than one tickets, every participant can send only one random key to D-lotto using an interface. Keys will be stored and processed by smart contracts. Here, if any participant or all the participants choose not to send the key, it would not stop algorithm proceedings and will not affect its fair share of randomness.

3.3 Lottery Drawing and Distribution of Winnings

Generating Random Numbers for Lottery Drawing: Random number generation algorithm for D-lotto is mentioned in algorithm 1. As shown in the algorithm, every key submitted by participant is XORed by algorithm to produce participant key P_{key}. As discussed earlier, either value of key can be kept arbitrary with cap of 40 hexadecimal digits (160 binary bits) or it can be kept of a predetermined value for all participants. Participants failing to submit key of described parameters, will not be participating in randomness of D-lotto drawing algorithm. Another job is to produce block key B_{key}, which is nothing but the XORing of every block hash produced before $E_t + \delta$ time. Here, δ is a small amount of time compared to lottery buffer time, which is used to incorporate natural randomness underlying Ethereum blockchain. Blocks found during δ time also contribute to generation of random number, which is not known by any individual in advance. This feature gives D-lotto an edge over other lottery DApps for providing randomness.

After calculating P_{key} and B_{key}, evaluate f_{SHA} (.) for producing seed. Given function represents hashing using any algorithm from family of Secure Hash Algorithms (SHAs). We suggest using Keccak-256 [14] algorithm from SHA-3 standards, as it is used by Ethereum in Proof-of-Work (PoW) consensus for calculation of Nonce based on random hash addresses. Now, produced natural random seed is the only requirement of any pseudorandom number generator (PRNG). f_{PRNG} (.) function is used to calculate a set of W pseudo random numbers, where W is the set of winning tickets determined in the lottery. We recommend using ISAAC [15], ChaCha20 [16] or HC-256 [17] for PRNG, as they produce considerable pure randomness and are cryptographically secure to meet the application requirements. These random numbers can be capped by performing modulo operation with number of tickets being sold. This random number generation stage will be drawing out specified number of winners for lottery scheme. Every parameter in the described algorithm is public and/or traceable after all, because smart contracts work in fair and transparent manner. This makes algorithm verifiable after lottery completion and gives participants fair and transparent lottery proceedings without intervention of any third party including dealer itself.

Algorithm 1 D-lotto Drawing Algorithm

```
Require:
    Participant state $P_{ki}$.
    Blockchain state $B_{kh}$.
Ensure:
    A set of random numbers W ( W = $w_1, w_2, \ldots, w_n$ )

Initialize $P_{key}$ with 40 bits of zeroes or ones.
Initialize $B_{key}$ with block hash including deployed lottery DApp.
Initialize $h$ with latest block height at lottery start time.
if $C_t = E_t + \delta$ then
    while ($P_{ki} \neq$ NULL) && ($i <$ no. of participants) do
        $P_{key} \leftarrow P_{key} \oplus P_{ki}$ ;   $i \leftarrow i + 1$
    end while
    while $h <$ Latest block height at $C_t$ do
        $B_{key} \leftarrow B_{key} \oplus B_{kh}$ ;   $h \leftarrow h + 1$
    end while
end if
Seed $\leftarrow f_{SHA}$ ($P_{key} \oplus B_{key}$)
W = $f_{PRNG}$(Seed)
return W
```

Distribution of Shares: After drawing out winners using RNG algorithm specified in previous state, smart contract will send the share of their money to winning participants on earlier verified receiving addresses. If a dealer has participated in lottery as a player and has been listed among winners, he will get his fair share from it. As mentioned earlier, it will be pretty clear to every participant about dealer's share/profit in the lottery since beginning, so remaining money will be transferred to dealer on his mentioned receiving address in smart contract. Even if the dealer chooses to avoid taking dealer's share, he will be paid back with lottery deployment charges bared by him earlier.

4 Comparison and Advantages

D-lotto, being a decentralized, Peer-to-Peer (P2P) application has its own advantages over traditional lottery systems and any electronic lottery system as described in Table 1. Along with low operational cost and quick response rate, it has no third-party intervention in any way. D-lotto specific additional unique features can be mentioned in terms of auditability and verifiability. Because of its underlying blockchain technology, D-lotto also incorporates blockchain technology advantages along with its architectural design benefits. Few of them are discussed below.

Transparency and Trust: Leaning toward decentralization and distributed computing has its own set of advantages. D-lotto, being built on public decentralized ledger, is totally transparent system. As it is based on distributed computing and uses smart contracts, need of third party and/or control organization has been entirely wiped out. This phenomenon builds trust among its users despite of no connection whatsoever with each other.

Auditability and Verifiability: Auditability in D-lotto is, being able to see every penny flowing through system without any privacy barriers. In public smart contracts, D-lotto can be traced down for its fund transfers and users can verify it with declarations of fund distribution at the time of lottery announcement. Verifiability

Table 1 Comparison of D-lotto with other lottery approaches

Property	Traditional lottery	E-lottery	D-lotto
Unpredictable	Yes	Yes	Yes
Membership	Private	Private/Public	Public
Third-party intrigue	High	Moderate	None
Operational cost	High	Moderate	Low
Response rate	Slow	Moderate	Quick
Auditability	No	Yes/No	Yes
Verifiability	No	No	Yes

can be described as any public procedure with proof of no conspiracy while drawing out lottery numbers. As described earlier, D-lotto uses game state and blockchain state to construct verifiable RNG algorithm. Hence, verifiability makes D-lotto is unique among other DApp lottery designs.

Reduced Fraud and Increased Accessibility: Blockchain technology possesses features like immutability and distributed computing. Hence, it automatically gifts D-lotto system with reduction in fraud, saving it from significant amount of loss. D-lotto uses cryptocurrency as medium of exchange in secure financial transactions, which helps it to remove geographical boundaries along with some legal aspects to participate in a lottery. Anyone is good to operate just by using good Internet connection.

5 Conclusion

Lottery can be used as a tool of entertainment and fund-raising apart from gambling. Hence, a novel lottery DApp design is proposed here on Ethereum platform using smart contracts. Ethereum as a platform ensures the availability of live blockchain parameters needed to build randomness in D-lotto drawing algorithm proposed here. D-lotto has an edge over other existing blockchain-based lottery systems in a manner that it uses game sate and blockchain state both to produce its verifiable randomness. Former property is a major contribution in trust and transparency of the system. D-lotto can be claimed as the novel system design in Ethereum-based lottery research due to its verifiable randomness.

References

1. Firelotto—white paper. https://firelotto.io/whitepaper_en.pdf. Last accessed 2019/11/30
2. Kibo—ethereum smart contracts based lottery. https://kiboplatform.net/en/landing.html. Last accessed 2019/11/30
3. Takya RA (2019) Blockchain lottery platform transforming lottery industry—bringing fairness to the lottery ecosystem. https://www.leewayhertz.com/blockchain-lottery-revolutionize-lottery-industry/. Last accessed 2019/11/30
4. Sharma M (2019) Impact of a streamlined lottery industry on the job market & development in india. https://www.stoodnt.com/blog/impact-of-a-streamlined-lottery-industry-on-the-job-market-development-in-india/. Last accessed 2019/11/30
5. Rodgers G (2015) Guilty verdict in hot lotto trial. https://www.desmoinesregister.com/story/news/crime-and-courts/2015/07/20/hot-lotto-verdict/30411901. Last accessed 2019/11/30
6. Randao: Verifiable random number generation. https://www.randao.org/whitepaper/Randao_v0.85_en.pdf. Last accessed 2019/11/30
7. Ariyabuddhiphongs V (2011) Lottery gambling: a review. J Gambl Stud 27(1):15–33
8. Liao D, Wang X (2017) Design of a blockchain-based lottery system for smart cities applications. In: IEEE 3rd international conference on collaboration and internet computing (CIC), pp 275–282

9. Jia Z, Chen R, Li J (2019) Delottery: a novel decentralized lottery system based on blockchain technology
10. Chen Y, Hsu S, Chang T, Wu T (2019) Lottery DApp from multi-randomness extraction. In: IEEE international conference on blockchain and cryptocurrency, pp 78–80
11. Liu Y, Liu H, Hu L, Tian J (2006) A new efficient e-lottery scheme using multi-level hash chain. In: 2006 international conference on communication technology, pp 1–4
12. Kuacharoen P (2012) Design and implementation of a secure online lottery system. In: Advances in information technology. Springer, Heidelberg, Berlin, pp 94–105
13. Grumbach S, Riemann R (2017) Distributed random process for a large-scale peer-to-peer lottery. In: Distributed applications and interoperable systems. Springer, Cham, Berlin pp 34–48
14. Bertoni G, Daemen J, Peeters M, Van Assche G (2013) Keccak In: Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, pp 313–314
15. ISAAC: a fast cryptographic random number generator. http://burtleburtle.net/bob/rand/isaacafa.html. Last accessed 2019/11/30
16. Bernstein DJ (2008) Chacha, a variant of salsa20. In: Workshop record of SASC, vol 8, pp 3–5
17. Wu H (2004) A new stream cipher hc-256. In: Fast software encryption. Springer, Berlin, pp 226–244