Sortes Whitepaper v0.8 EN

Sortes: A decentralized social welfare system

Author: Alhaitham Sakamoto, Edward Seldon

Keyword: Lottery, Bitcoin, Smart contract, Layer 3.

1. Introduction

We propose a charitable social welfare system based on Ethereum smart contracts, which replaces human factors with machines in the traditional lottery system, achieving true fairness, transparency, and decentralization. Currently, the lottery system relies entirely on centralized institutions as organizers to generate and publish game results, which has inherent weaknesses based on a trust model, making it impossible to produce completely random prizes, depriving ordinary people of the opportunity to seek a chance to change their lives with a one-in-a-million chance. This paper discusses the prize pool mechanism, lottery principle, and technical implementation of our proposed new lottery system, and studies its scalability and broader application scenarios, which will allow anyone to participate freely in contributing to the lottery prize pool, operating lottery sales nodes, and participating in the lottery fairly.

2. Prize Pool

2.1 Generation of Prize Pool

Sortes differs from other lottery systems that are issued and operated by centralized organizations. The generation of its prize pool is achieved through a completely decentralized way: any user can contribute to this prize pool by depositing Bitcoin (wBTC) on the Ethereum blockchain into the contract, and directly obtain the vast majority of the revenue from lottery sales. When a user deposits wBTC into the prize pool, the contract will issue a corresponding certificate Xbit based on the deposited amount. Xbit is both a deposit certificate obtained after wBTC is deposited, and an exchange certificate for withdrawing wBTC from the pool. It serves as a record-keeping certificate for depositing and withdrawing Bitcoin in the prize pool. The only way to issue Xbit tokens is to deposit Bitcoin into the contract, and the rules are as follows:

$$Xbit = Bitcoin_{depo} \cdot \frac{Xbit_{total}}{Bitcoin_{inpool}} \quad \text{(a)}$$

For the case where Bitcoin in the prize pool is 0, Xbit = Bitcoindepo, which means the amount of Xbit received equals the amount of Bitcoin deposited. As the lottery process causes fluctuations in BitcoinInpool, the exchange rate between Xbit and Bitcoin usually fluctuates as well. Changes in the prize pool are described in section 2.4 Prize Pool Changes.

2.2 Withdrawal of BTC from the Prize Pool

(a) The formula is also the theoretical value of Xbit token. When Xbit holders want to withdraw the deposited bitcoin from the prize pool, the formula becomes (b). By destroying Xbit, users will be able to withdraw wBTC from the prize pool in the following quantity:

$$Bitcoin_{withdraw} = Xbit \cdot \frac{Bitcoin_{inpool}}{Xbit_{total}} \quad \text{(b)}$$

2.3 Minimum Runnable Prize Pool

In order to ensure the stable operation and user experience of the Sortes protocol, a minimum prize pool that can be redeemed needs to be established to avoid situations where the prize pool has a low amount of winnings and cannot be redeemed. This can also reduce the impact of the lottery process on the pool. Sortes sets the minimum lottery pool size that can run to be 10 wBTC. When the total amount of Bitcoin in the prize pool is less than 10 wBTC, the lottery function will be temporarily suspended until new Bitcoin is deposited to make the total prize pool amount exceed 10 wBTC.

2.4 Changes in Prize Pool

When users consume USDT to participate in the lottery, this part of the USDT will be instantly converted into wBTC and entered into the total prize pool through decentralized liquidity pools such as Uniswap. This increases the amount of wBTC in the total prize pool. Since this process does not issue new Xbits, it also means that all Xbit holders share the income of this part of the wBTC, thereby increasing the exchange rate of Xbit to wBTC.

$$Bitcoin'_{inpool} = Bitcoin_{inpool} + convert_{BTC}(USDT)$$

On the contrary, if the user wins during the lottery process and it is in wBTC, then the amount of wBTC in the prize pool will decrease. This process does not destroy the issued Xbit, but dilutes the wBTC amount held by Xbit holders, thereby reducing the exchange rate of Xbit to wBTC.

$$Bitcoin'_{inpool} = Bitcoin_{inpool} - Bitcoin_{Awarded}$$

Combining the above two equations, we can see that the formula for the change in the prize pool during the lottery process is:00-*..3

$$Bitcoin'_{inpool} = Bitcoin_{inpool} + convert_{BTC}(USDT) - Bitcoin_{Awarded} \quad \text{(c)}$$

For cases where a user wins a prize that is not from the prize pool but is instead a commemorative NFT, the pool will not be reduced.

3. Lottery Algorithm

3.1 Setting the Payout Ratio

Most lottery systems set a payout ratio, which is typically defined as

$$R = \frac{Expectation_{Award}}{Bet_{Amount}} \quad \text{(d)}$$

Among them, it is the capital consumed for lottery and the expected bonus obtained.

$$Expectation_{Award} = \sum_{n=1}^{k} P_n * Award_n$$

Most of the current lottery systems have a certain degree of fluctuation in their R values. In some lottery systems, R can even exceed 1 as the jackpot accumulates. The initial payout ratio in the Sortes protocol is 0.9, and future payout ratios may be determined through community governance voting. In contrast, Powerball's payout ratio is only 0.32-0.43.

Ref: https://bigthink.com/starts-with-a-bang/math-of-powerball/#:~:text=If you want to calculate,total worth of each ticket.

3.2 Setting and Probability of Initial Reward Tiers

Assuming the user spends **K (USDT)** to participate in the lottery, in order to maintain the stability of the prize pool fluctuations, it is necessary to limit K to no more than 1% of the total value of the prize pool.

$$K \leq 0.01 * value_{pool}$$

$$Value_{pool} = Amount_{BTC} * Price_{BTC}$$

The *PriceBTC* is obtained from the oracle.

In a simple example for the reward levels for a 10U lottery prize can be divided into the following categories:

Payout	Odds	Expected Value
Exp	~79%	$0
$20	20%	$4
$200	1%	$2
$1000	0.1%	$1
$10000	0.005%	$0.5
10% value_pool	5 / value_pool	$0.5
Sum	100%	$8

The value of Exp was not calculated in the above discussion, but in actual operation, the value of Exp obtained will be adjusted based on the Ethereum Gas fee rate of the main network, which will affect the ratio of airdropped tokens that may be obtained later, compensating for the cost of burning Gas for users during the lottery process. This maintains the actual payout ratio at around 0.85 (or even higher), maximizing the profits of lottery participants.

In a more general situation which we intend to provide as standard contract API for registration, the possibility table need to satisfy R < 0.8.

If the process of generating rewards is random enough, then in the case of sufficient lottery draws, the prize pool theoretically continues to increase. It can be known that for those who deposit Bitcoin into the prize pool, they can obtain a low-risk and stable return, and Xbit is theoretically a BTC-denominated income certificate.

4. Lottery as a Service (LaaS)

Lottery as a Service (LaaS) refers to a business model in which a third-party provider offers lottery-related services and tools to organizations or individuals through a subscription. This enables organizations or individuals to set up, manage, and run their own lottery games without having to build or maintain the necessary infrastructure and technology from scratch. LaaS providers typically offer a range of services, including lottery platform development, ticket sales, payment processing, regulatory compliance, and draw management. By outsourcing these services to a LaaS provider, organizations can focus on marketing, customer acquisition, and other strategic aspects of their lottery business.

It is worth noting that the core service provided by LaaS is the offering of a probability pool. The probability pool refers to the collection of all lottery revenues after participants purchase tickets. The funds in this pool are used to pay various prizes, as well as to pay a certain percentage of fees to operators, LaaS providers, and others. By using the probability pool offered by LaaS, customers can achieve better risk management while ensuring the fairness and transparency of lottery activities.

4.1 Lottery Nodes

A robust lottery sales system is essential for the smooth operation of any lottery. LaaS enables third parties to access a shared prize pool through an open API, which acts as a "probability depth" that offers verifiable winning chances for various third-party lottery systems. By calling the API, lottery sales nodes can distribute the lottery pool within the protocol's scope. The lottery sales nodes can sell lottery tickets and potential profits by utilizing the LaaS provider's smart contract API.

All lottery sales nodes share the same prize pool, and each site independently holds and sets the prize distribution and allocation ratios. The lottery sales nodes, as allowed by the protocol, can distribute rewards generated through node bets. This LaaS model streamlines lottery operations, allowing multiple parties to participate and benefit from a shared, transparent, and verifiable prize pool.

4.2 Node revenue sharing mode

Lottery nodes can share the revenue of users who are purchasing the lottery in their own website that integrate Sortes protocol and have registered on the smart contract for revenue sharing ratio. The allowed arrange of revenue sharing ratio is from zero to 20 percent.

5. Technical Implementation

5.1 Contract Structure

The contract mainly consists of four parts:

1. Based on the ERC20 standard, it provides the circulation function of standard tokens (i.e., Xbit).
2. It uses the on-chain async random method to obtain the ability to generate random numbers, thus ensuring the security and fairness of the lottery.
3. It allows vendors to define outcomes and probabilities, which must meet preset constraints.
4. It has the ability to call swap to help users automatically exchange different digital currencies.
5. The main functions of the contract's basic functions are implemented:

    a. Deposit WBTC and mint Xbit

    ```
    function save(uint256 amount_wbtc) public
    ```

    After authorization, users transfer WBTC to the contract address and mints Xbit at the same time. The amount of minted Xbit is determined by the total amount of WBTC in the current contract address and the total circulation of all Xbit.

    b. Exchange Xbit for WBTC

    ```
    function withdraw(uint256 amount_xbit) public
    ```

    After authorization, users destroy Xbit and exchanges it back to WBTC. The amount of WBTC obtained is determined by the total amount of WBTC in the current contract address and the total circulation of all Xbit.

    c. Register a swap

    ```
    function registerSwap(Swap memory swap) public returns (uint256)
    ```

    Vendors can register self-defined swaps, which must meet preset constraints:

    - sum of expectations must be lower than the upper bound
    - sum of probabilities must be no more than 1
    - relative reward must be less than a amount
    - absolute reward must be less than a amount

    d. List swaps

    ```
    function listSwaps(address owner) public view returns (Swap[] memory)
    ```

This function is used to list all swaps created by the given owner.

e. Deposit USDT and participate in the lottery to obtain WBTC and exp reward

`function safeSwap(uint256 amount_usdt, uint256 swapId) returns (uint256)`

After authorization, users transfer USDT to the contract address and participates in the lottery activity, which may result in a certain amount of WBTC. The amount of WBTC obtained is jointly determined by the amount of USDT transferred, the exchange rate between USDT and WBTC, the reward tier and probability defined in the selected swap, and the random number generated by the on-chain async random method.

f. Reveal swap outcomes

`function reveal(uint256 requestId) public`

For safety, the results of a swap are not generated immediately. Instead, users must call the reveal function to show and claim the outcomes of a swap.

5.2 Currency Exchange

During the lottery process, users may choose to deposit USDT but receive wBTC, requiring a mechanism for currency exchange to more accurately calculate rewards. Currently, we use the Uniswap Protocol in our smart contract for currency exchange and do not implement internal exchange functionality, thus fully utilizing the market efficiency and fairness of existing pools.

Ref: https://uniswap.org/

6. Future of Sortes

Sortes envisions revolutionizing charitable donations through the use of blockchain technology by creating a decentralized, transparent, and participatory social welfare system. This system allows individuals to participate in a decentralized lottery, where funds raised through ticket sales are directed towards charity initiatives.

Sortes could redefine how charitable funds are raised and distributed, democratizing philanthropy and making it a more inclusive and trustworthy process. As blockchain technology matures, the system will likely incorporate additional features such as token rewards, partnerships with NGOs, and the ability to support multiple currencies, further enhancing its reach and impact.