Aliera: Hedged-Loss Lottery Betting Platform

12/12/2025
https://aliera.bet

1. Introduction

This paper provides a comprehensive analysis of the Aliera hedged-loss lottery betting platform. Aliera introduces an innovative model to the lottery and betting sectors by incorporating a unique "hedged-loss" mechanism designed to reduce player risk and enhance user retention. This paper covers the platform's core mechanics, technical architecture, economic model, and security features.

2. Paper Summary

This section summarizes the key information about the platform and implemented mechanics.

2.1. Platform Overview

Aliera is a provably fair on-chain lottery betting platform with a central innovation: a hedged-loss mechanism where players receive a portion of their entry back even if they do not win. The randomness for winner selection is sourced from Chainlink's Verifiable Random Function (VRF), ensuring transparency and tamper-resistance.

2.2. Core Mechanics

The lottery betting loop is structured around a few simple steps:

1. Deposit and Tokenization: Players deposit USDT, USDC or ETH and receive LUCK tokens at a 1:1 ratio. These deposits directly fund the prize pool for the current round.
2. Lottery Entry: Participants burn their LUCK tokens to enter a round, with each token representing one ticket.
3. Winner Selection: A winner is chosen using the Chainlink VRF, receiving the entire prize pool minus a protocol fee.
4. Refund Claim: Non-winners can claim a 50% refund of their burned LUCK tokens, which can be used in future rounds.

2.3. The Luck Cycle

The "Luck Cycle" is an important part of the platform as the refund mechanism creates a sustainable and engaging player experience. A player who loses a round can combine their refunded LUCK with a new deposit to re-enter with the same number of tickets, effectively reducing their cost of participation over time.

2.4. Key Features

The platform's key features are:

- Provably Fair: Transparent and verifiable randomness through Chainlink VRF.
- Hedged Loss: The 50% refund system.
- Flexible Entry: Support for USDT, USDC, native ETH or any ERC20 token (via KyberSwap), and accumulated LUCK.
- Transparent Prize Pool: All deposits contribute directly to the prize pool.
- Low Fees: A protocol fee with a maximum of 10%, initially set to 1%.

2.5. Security and Roles

Security is a primary concern, addressed through:

- Whitelisted Access: Participation is limited to verified users.
- Reentrancy Protection: Smart contracts are designed to prevent reentrancy attacks.

The platform defines several key roles, including Payer, User/Beneficiary, Coordinator, Gatekeeper, and Admin, each with distinct responsibilities.

3. Technical Specifications

This section details the smart contract specifications, including data structures, functions, and operational parameters.

3.1. Round Lifecycle and States

The lottery operates in three distinct states:

- ACTIVE: The round is open for entries, and players can burn LUCK tokens.
- VRF_REQUESTED: The round has ended, and the system is awaiting a random number from the Chainlink VRF.
- FINALIZED: A winner has been selected, the prize is distributed, and refunds are available for non-winners.

3.2. Core Data Structures

The platform relies on two primary data structures to manage its state:

- Round: A struct that stores all information related to a single lottery round, including start and end times, total tickets, prize pool, and winner details.
- PlayerRoundData: A struct that tracks a player's participation in a specific round, including the amount of LUCK burned, claimable refund, and whether the refund has been claimed.

3.3. Smart Contract Functions

The LotteryGame contract exposes a rich set of functions for players and the coordinator:

Player Functions:

Function	Description
depositAndMint(uint256 usdtAmount)	Deposits USDT and mints an equivalent amount of LUCK tokens.
enterRound(uint256 luckAmount)	Burns LUCK tokens to enter the current lottery round.
claimRefund(uint256 roundId)	Claims a 50% LUCK refund for a lost round.
claimRefundBatch(uint256[] memory roundIds)	Claims refunds from multiple rounds in a single transaction for gas efficiency.

Coordinator Functions:

Function	Description
startRound()	Initiates a new lottery round.
endRound()	Ends the current round and requests randomness from Chainlink VRF.
forceEndRound()	Allows the coordinator to end a round prematurely in case of an emergency.

3.4. Events and Error Handling

The contracts emit a comprehensive set of events for all significant actions, such as RoundStarted, PlayerEntered, WinnerSelected, and RefundClaimed. This allows for transparent and reliable off-chain monitoring. The contracts also include a full set of custom error messages to provide clear reasons for failed transactions.

4. Platform Architecture Analysis

This section provides a deep dive into the system's architecture, economic model, and security mechanisms.

4.1. Core Innovation: The Hedged-Loss Mechanism

The primary innovation of the Aliera platform is the "hedged-loss" mechanism, which is designed to mitigate the risk for players and encourage sustained participation. This is achieved through a recursive refund cycle:

1. Initial Entry: A player deposits USDT and receives an equivalent amount of LUCK.
2. Participation: The player burns LUCK tokens to enter a lottery round.
3. Loss and Refund: If the player does not win, they can claim a 50% refund of the LUCK tokens they burned.
4. Reinvestment: The refunded LUCK can be used to enter future rounds, reducing the need for new capital.

This creates a compounding benefit for players, as the effective cost of participation decreases with each non-winning round.

4.2. System Architecture and Smart Contracts

The platform's architecture is centered around a suite of smart contracts that work in concert:

- LotteryGame Contract: The core contract that manages the lottery's logic, including the round lifecycle, player entries, prize distribution, and the integration with Chainlink VRF.
- LotteryPeriphery Contract: This contract serves as a user-friendly wrapper, simplifying interactions for the end-user. It includes functions for easy entry and handles token swaps from any ERC20 token to USDT through an integration with KyberSwap.
- LuckToken and USDT Contracts: These are the ERC20 tokens that facilitate the platform's economic model. USDT is used for deposits and prizes, while LUCK is the in-game currency for entries and refunds.

4.3. Economic Model and Sustainability

The economic model is designed for long-term sustainability:

- Prize Pool: The prize pool is funded directly by player deposits of USDT or other tokens. Since refunds are issued in LUCK tokens, the USDT prize pool is not depleted by the refund mechanism.
- Player Economics: The expected value for a player is significantly improved compared to traditional lotteries. A loss results in a 50% recovery of the entry cost in the form of LUCK tokens, which can be reinvested.
- Protocol Revenue: The platform sustains itself by collecting a small protocol fee from the prize pool of each round.

4.4. Security and Access Control

Security is a critical aspect of the platform's design:

- Role-Based Access Control: The system employs a set of distinct roles (Admin, Coordinator, Gatekeeper) to manage permissions and control access to sensitive functions.
- Whitelist System: In its alpha phase, the platform restricts participation to whitelisted users, preventing unauthorized access and potential Sybil attacks.
- Smart Contract Security: The contracts are designed with reentrancy protection and include mechanisms for emergency pause and settlement.

5. Conclusion

Aliera hedged-loss lottery betting platform represents a significant advancement in the design of blockchain-based lottery and betting applications. Its innovative refund mechanism fundamentally changes the economic incentives for players, fostering a more sustainable and engaging experience. The platform's architecture is robust, with a clear separation of concerns, and its commitment to fairness through the integration of Chainlink VRF.