YearnVaultShareTWAPOracle

Introduction

VMEX’s strategies target underutilized, yield-bearing assets like LP tokens, which are typically considered higher risk by traditional lending protocols. VMEX's isolated tranches enable users to benefit from these assets by combining the advantages of borrowing against LP tokens and staking them for additional yield. This innovative approach to lending allows users to maximize their returns through leveraged yield farming, earn higher yields on stablecoin deposits, and passively improve their health factor as rewards accrue

When users deposit Curve LP tokens as collateral (for borrowing purposes) and stake them in Yearn vaults, a **Yearn Vault Position’s price oracle is essential for**:

1. **Collateral Value** : The lending protocol must accurately value the collateral (Curve LP token) to determine the borrowing capacity and collateralization ratio for the user.

2. **Liquidation Thresholds** : Determining when a position's value drops, requiring liquidation to protect the system.

3. **Risk Assessmen**t : The lending protocol needs up-to-date pricing information to assess and manage the risk associated with lending against volatile and yield-bearing assets like Curve LP tokens.

4. **Vault Monitoring** : Tracking Yearn vault performance, accrued rewards, and yields to inform user decisions.

Design

The YearnVaultShareTWAPOracle smart contract calculates the Time-Weighted Average Price (TWAP) of a Yearn Vault share by taking into account **the prices of the underlying tokens in a Curve pool**. The process starts with **obtaining the price data for each underlying token in the Curve pool** from the respective Chainlink price feeds. These **prices are then used to calculate the value of a single unit of the Curve LP token**, which is a representation of the combined value of the underlying tokens within the pool.

Next, the smart contract fetches the total value of the Curve LP tokens held in the Yearn Vault, which represents the total value of the underlying tokens managed by the vault. This value is then divided by the total number of Yearn Vault shares issued, giving the value of a single Yearn Vault share.

Manipulation Resistance:

Undercollateralized Loan Attack:

In this attack, a bad actor manipulates the price of an asset, allowing them to borrow more of the loan asset using their attack capital as collateral. By inflating the price of the collateral asset on the lending protocol, the attacker can borrow more of the loan asset and sell it in the open market for a profit. The success of this attack depends on the attacker's ability to de-manipulate the price without other users front-running them.

Liquidation Attack:

A liquidation attack occurs when a bad actor manipulates the price of a collateral asset or loan asset to make a loan appear undercollateralized. Acting as a liquidator, the attacker then settles the loan in the loan asset and claims the collateral asset for a profit. The attacker competes with other rational actors to execute this attack and must outbid their transactions to get included in the next block.

Spot Price Manipulation Attack:

In a spot price manipulation attack, the bad actor manipulates the price of an asset on the lending protocol's reference automated market maker (AMM). If the lending protocol uses a naive spot price, the attacker can manipulate and de-manipulate the price within a single blockchain transaction, making the attack cheap. Flash loans, which allow large amounts of assets to be borrowed without collateral, can exacerbate this issue.

Time-weighted average price (TWAP) oracles, help mitigate these attacks by making price manipulation more expensive and giving arbitrageurs the opportunity to front-run de-manipulation transactions.

YearnVaultShareTWAPOracle Implementation:

YearnVaultShareTWAPOracle records the price of a specific trading pair before the first trade of each block. The price is multiplied by the number of seconds since the last update, resulting in observation pi. All observations are stored in an accumulator that reflects the sum of the spot price at each second in the contract's history.
TWAP Calculation:
An external caller (e.g., lending protocol) checkpoints the accumulator's value at times t1 and t2. Using these values, it calculates the TWAP from t1 to t2 (with LT = t2 - t1) as:
TWAP_t1,t2 = (a_t2 - a_t1) / LT

Trade-offs:

TWAP oracles involve a trade-off between manipulation resistance and price freshness. A larger LT in the TWAP calculation increases manipulation resistance, while a shorter TWAP follows the spot price more closely. However, using a longer duration TWAP may result in the oracle not

reflecting the true spot price, causing the lending protocol to fail to respond to real market conditions that can lead to under-collateralized loans.
In the context of YearnVaultShareTWAPOracle, we assume that the TWAP uses the arithmetic mean over its range, providing a balance between manipulation resistance and price accuracy.

Smart Contract

The YearnVaultShareTWAPOracle smart contract calculates the Time-Weighted Average Price (TWAP) of a Yearn Vault share by utilizing Chainlink price feeds for the underlying assets in a Curve pool. This ensures that the price data is accurate, reliable, and resistant to tampering. The contract design is modular, allowing seamless integration with other smart contracts or protocols, such as Curve pools and Yearn vaults, by adhering to standardized interfaces like AggregatorV3Interface, ICurvePool, and IYearnVault.

The contract has several key dependencies:

Curve Pool: The oracle retrieves data on the underlying assets' balances and LP token supply from the Curve pool contract. This enables the oracle to calculate the weighted average price of the underlying coins in the Curve pool and derive the Yearn Vault position's USD value.

Yearn Vault: The contract interacts with the Yearn Vault to obtain the latest price per share data. This information is essential for determining the Yearn Vault share's USD price, which is then recorded in the observations array.

Chain Link Price Feeds: To obtain accurate and reliable price data for the underlying assets in the Curve pool, the contract relies on Chain Link price feeds. These price feeds are considered highly secure and are widely used in the DeFi ecosystem.

The contract maintains a **dynamic array of historical observations**, which contains timestamps, cumulative prices, and prices at the given timestamps. A public function allows updating these observations, ensuring that new price data is periodically added, keeping the oracle up-to-date with the latest market information.
Using a **dynamic array** in the YearnVaultShareTWAPOracle smart contract provides **several benefits:**

- Scalability: Improves the scalability of the contract by accommodating a growing number of price data points over time.
- Flexibility: The smart contract can easily **adapt to different time-weighted average price** (TWAP) calculation requirements. It can be configured to use **varying observation windows or different aggregation methods**, depending on the specific use case or risk management strategy. This flexibility makes the smart contract more versatile and adaptable to changing market conditions.

The **binary search algorithm** implemented within the contract helps efficiently retrieve historical price data by finding the closest price observation for a given timestamp. This efficient approach significantly reduces gas consumption and optimizes the performance of the TWAP calculation.

Limitations and Attack Vectors

Single-block attack vulnerability:

A well-capitalized attacker can manipulate a TWAP oracle within a single block by executing a series of transactions that change the price of an asset. This manipulation can be performed by using large capital to execute a series of trades on an automated market maker (AMM) platform like Uniswap, altering the asset's price. The attacker can then execute other transactions, such as exploiting lending protocols, that depend on the manipulated TWAP price. Since the attacker's trades occur within a single block, it becomes challenging for arbitrageurs to react and correct the price.

Multi-block MEV (MMEV) attacks:

An attacker can control the transaction ordering over multiple blocks to manipulate TWAP oracles at a lower cost. This involves an attacker specifying a transaction ordering over not just one but multiple blocks in a row, allowing them to bypass competing with arbitrageurs. This type of attack, called MMEV, takes advantage of the fact that miner extractable value (MEV) can be exploited through transaction ordering. In this case, MMEV can be used to manipulate TWAP oracles and execute under-collateralized loan or liquidation attacks on lending protocols.

Selfish mining:

In MMEV attacks, an attacker can use selfish mining techniques to control two consecutive blocks, reducing the manipulation capital required for a single-block attack. Selfish mining involves a miner withholding mined blocks and maintaining a private chain to later publish it judiciously, extracting more value than their share of hash power would warrant. By controlling two consecutive blocks, the attacker can execute a series of transactions that manipulate the TWAP oracle without being vulnerable to arbitrage.

Significant manipulation capital:

MMEV attacks can be limited by the high manipulation capital required to execute them. To manipulate the TWAP oracle, an attacker needs to invest a significant amount of funds temporarily to execute the necessary trades on an AMM platform. This capital requirement scales linearly with the total liquidity and with the square root of the TWAP length and the targeted price deviation. The manipulation capital required might be a more significant limiting factor for an attacker than the cost in fees.

Proof-of-Stake collusion:

In Proof-of-Stake systems, two block proposers could collude to perform MMEV-style oracle manipulation. Since block proposers are known in advance, they can work together to control transaction ordering over consecutive blocks, enabling MMEV attacks. These attacks do not go against standard consensus rules of blockchains, and hence, colluding proposers will escape slashing or detection, making it a challenging attack vector to counter.

Stake grinding attacks:

In Proof-of-Stake systems using verifiable randomness functions, block proposers could try to improve their odds of proposing two blocks in a row to enable MMEV-style attacks. This is called a "grinding attack," where the proposers attempt to manipulate the randomness used for block proposal selection. By proposing consecutive blocks, attackers can control transaction ordering and manipulate the TWAP oracle.

Future Scope & Improvements

Median as a better choice for oracle design :

Using a median-based oracle can offer some advantages in terms of manipulation resistance. A median oracle calculates the median price of an asset within a given time range, making it less sensitive to extreme price fluctuations. However, it is important to note that a median is less resilient to multi-block attacks than a TWAP. Manipulating a median only requires control over half the blocks it encompasses, making the multi-block attack 50% cheaper compared to when using an average.

One economic solution could be to use a median of averages by splitting the block range of interest into 'n' smaller ranges using 'n+1' checkpoints and calculating the average for each range. To manipulate the median in this case, the majority of the ranges would need to contain at least one manipulated block. This would offer more resistance to manipulation compared to a simple median or average.

However, implementing a median oracle can be impractical, as the calling contract would need to store checkpoints of the accumulator for every single block and load all checkpoints from storage to calculate the median. This process would be very expensive in terms of gas costs.

Further research is required to analyze the properties of a median of averages in worst-case scenarios compared to standard median or average oracles.

Geometric Mean

When comparing Time-Weighted Average Price (TWAP) oracles that use arithmetic and geometric means, the geometric mean has an important advantage: manipulating the geometric mean by adjusting the price of an asset in a single block is more expensive than multi-block manipulation. This means that Uniswap V3 oracles, which use the geometric mean, are not affected by the single-block attack described in the research, while Uniswap V2 oracles, which use the arithmetic mean, are vulnerable.

Using the Maximum Mean Extractable Value (MMEV) approach to avoid arbitrageurs while executing the multi-block attack on the geometric mean could reduce its costs significantly. However, this would likely require controlling many blocks within the TWAP period, not just two, making the attack more difficult and expensive.