5/22/22

SOLIDIFI

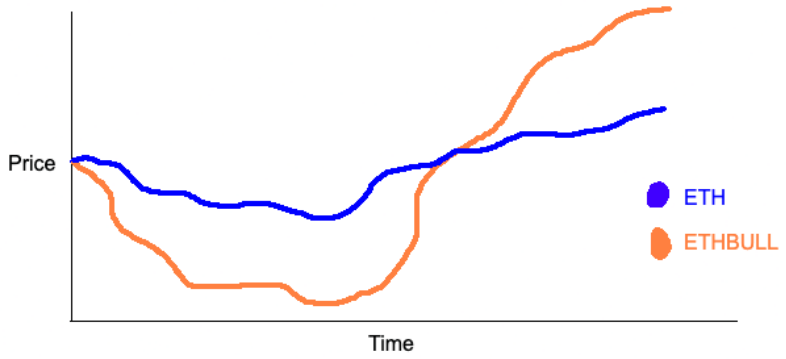
by Jacob Janak

Solidifi is a protocol for a decentralized leveraged token that allows investors to hold a short or long position in ETH without suffering from decay.

- Contract address: 0xEcb6f4CE53a36943B801659c9719d84eca970eD6
- Contract network: Rinkeby
- Website: https://jacobjanak.github.io/solidifi/
- Test website: https://jacobjanak.github.io/solidifi/test/
- GitHub: https://github.com/jacobjanak/solidifi

I. What is a leveraged token?

A leveraged token is a token that gives its holder a leveraged position in the asset it represents. For example, a 3x ETHBULL leveraged token would give you 3x the gains or losses of ETH. That means that, for every $1 that ETH's price changes, the price of ETHBULL will change by $3. To illustrate this, let's look at this chart:



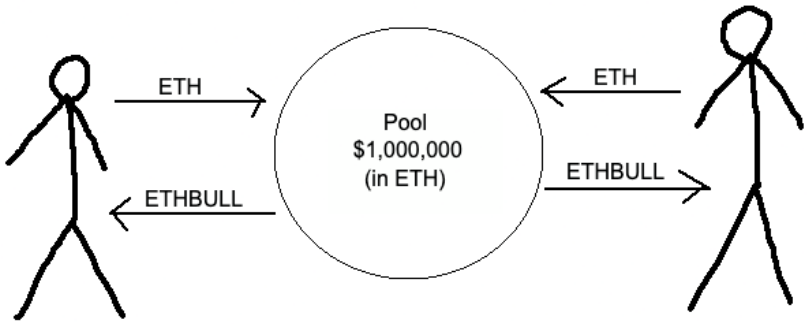
As you can see, the price of ETHBULL goes up when ETH goes up and it goes down when ETH goes down. The difference is that it goes up by a lot more than ETH and down by a lot more than ETH. Essentially, it is an exaggerated version of ETH. These extreme price movements are not suitable for many investors. But, there is definitely a market for these tokens. The leverage industry, as a whole, is a multi-billion dollar market and leveraged tokens are a small-but-growing part of this industry. This Koinly article gives a more in-depth description of leveraged tokens, for those wanting to learn more. I will discuss some of the mechanics behind these leveraged tokens in the next section.

II. Motivation

Simply put, every leverage token currently available on the market is bad. One issue affecting leveraged tokens is that almost all of them are centralized. According to CoinGecko's listings, there are over 130 leveraged tokens and ETFs on the market, only 2 of which are decentralized. This is an issue because centralized leverage tokens suffer from regulation. In just the last year, I have seen many exchanges, including Binance, stop offering leveraged tokens due to regulatory pressure. Today, in many countries, the only way for people to access leveraged tokens is through decentralized means. But, there are very few decentralized options on the market. The reason for this is that it's very difficult to create one. I will offer the following readings for anyone interested in learning more about the difficulty of making a leveraged token decentralized:

- This article by Phoenix Finance describes their protocol for their leveraged tokens.
- This two-part article on CoinGecko describes the decentralized levered token industry as whole, including all the challenges it currently faces. You will need a CoinGecko premium account to read the whole article.

Besides centralization, the other issue faced by leverage tokens is decay. To understand the effects of decay, let's look at the price chart of ETHBULL. It is supposed to match ETH's price movement with 3x leverage. But, the price is trending down over time, despite ETH's price increasing. ETHBULL was worth $500 per token at the start of 2020. Today, it is worth only $65. On the other hand, ETH was worth $130 at the start of 2020. Today, it is worth $2,000. Why is it that ETHBULL has decreased by -87% in the same time period that ETH has increased by over 1500%? Shouldn't ETHBULL have increased by 4500%, since it promises returns equal to 3x that of ETH? The reason for this discrepancy is decay. Every leveraged token that has ever existed suffers from decay. They decay over time because they rely on outside sources to artificially adjust the price of their tokens. To understand decay, let's take a look at this diagram:



This diagram describes every leveraged token protocol currently available on the market. As you can see, there are some users who are depositing their ETH into a pool. In return, they get ETHBULL tokens. The users can always return their ETHBULL tokens to the pool and withdraw their ETH whenever they want. Let's say ETH's price doubles. Now, the pool is worth $2,000,000. That means the market cap of all the ETHBULL tokens is also $2,000,000, since

the ETHBULL tokens are worth whatever is in the pool. We want to make sure that ETHBULL has 3x leverage compared to ETH. That means that, since ETH just went up 200%, we need ETHBULL to go up 600%. In order for ETHBULL to increase by 600% from its original value, we need there to be $6,000,000 in the pool. Right now, there is only $2,000,000 in the pool. Where does the extra $4,000,000 come from? Well, different leveraged tokens have different ways of acquiring this money. Some use flash loans while others have a dedicated liquidity pool that they take money from. Regardless of the method, acquiring this money incurs interest. You cannot get money from an outside source without paying interest. Since these leveraged token protocols are always borrowing money, they are always paying interest, which causes decay over time. Even if ETH's price continues to go steadily upwards for many more years, the ETHBULL protocol will be paying so much interest that the token will actually be losing value. Decay does not necessarily impact short-term investors, but it completely destroys any long-term value the token has.

Solidifi fixes both of these issues that are faced by leveraged tokens: it is free from centralization and it is free from decay. Solidifi offers long-term investors a leveraged token that they can hold without needing to do KYC verification. No one on the market currently offers this.

III. Non-technical project description

For everyday investors who want to use my contract without fully understanding the protocol, the process is extremely easy. Of course, you must have an Ethereum wallet that contains some ETH. To get started, you simply deposit some ETH into the contract using the website: jacobjanak.github.io/solidifi/. You can either deposit ETH into the up side of the pool or the down side of the pool, depending on where you think ETH's price will go. If you guess correctly, you will be able to withdraw more ETH than you put in. If you are wrong, you will withdraw less ETH than you put in. Withdrawing is simple: just indicate what percent of your holdings you want to withdraw and click the withdraw button. Currently, there are no fees. Eventually, there will be a deposit and withdrawal fee, but no continuous fee for keeping your ETH in the protocol. Gas fees are a factor to consider, but my contract was designed with efficiency in mind so that the gas fees aren't too high. Currently, the gas fee is less than 0.0003 ETH ($0.60 USD) per deposit or withdrawal. Please note that ETH's price is determined according to its value in USD.

IV. Technical project description

In short, the protocol works by moving ETH from the losers to the winners. The winners are the people who correctly guessed ETH's price movement and the losers are the people who guessed wrong. Thus, there is no need for an outside pool. The money is all moved around internally. Without an outside pool, there is no decay factor (as discussed in section II). This is the beauty of my design. The money just gets moved from some investors to other investors. Here is how this money movement works:

This rectangle represents the contract. The red side represents the ETH that has been bet on ETH's price going down. The green side represents the ETH that has been bet on ETH's price going up. As you can see, the green side looks larger than the red side. That means that more ETH is bet on ETH's price going up. Now, let's see what happens if ETH's price goes up:

Since ETH's price went up, the green side got bigger. That means that the people who deposited ETH into the green side can now withdraw more ETH than they originally deposited. If the price had gone down instead of up, then the red side of the pool would have gotten bigger. If you want to experiment more with this system, you can visit the Solidifi test website.

The overall size of the pool does not change unless people deposit or withdraw ETH. The only thing that changes is the line between the green side and the red side. We call this line the divider. The divider is moved according to a function named updateDivider. UpdateDivider is called before every deposit and every withdrawal to ensure that everyone deposits/withdraws exactly the right amount of ETH according to ETH's current price. Here is how updateDivider works:

1. It checks that there is ETH in both sides of the pool.
2. The contract retrieves ETH's current price (in USD) using a ChainLink oracle.
3. It checks that the current price is different from the last price it retrieved.
4. The divider is moved according to this function:

$$divider \ -= \frac{1}{4} \cdot totalETH \cdot log\left(\frac{price}{lastPrice}\right)$$

Let's go over this function. First, we multiply the total amount by ¼ simply to scale down the movement of the divider. Without scaling it down, I've found that the moves are too extreme. Then, we multiply the move by totalETH, which represents the total amount of ETH that is in the contract. We do this because we want the size of the moves to scale with the size of the pool. Lastly, we see the most important part of the function, the log. We take price divided by lastPrice because we want the movement of the divider to be scaled according to the relative change in price. After all, a 50% change in price should cause a much more dramatic move than a 5% change in price. We take the log of this ratio because log has this beautiful property:

$$log\left(\frac{a}{c}\right) = log\left(\frac{a}{b}\right) + log\left(\frac{b}{c}\right)$$

This property is crucial to the pool. It says that the pool will look the same at point A regardless of how ETH's price got to point A. For example, whether ETH's price increased to $10,000 in one day or two years, it doesn't matter. The divider will be in the same spot regardless of how ETH's price arrived at point A. Without log, we cannot achieve this property. For the sake of efficiency, we use log 2, as opposed to other types of logs.

After the divider is moved, the last step is to check for liquidations. Liquidations should never happen, since my protocol is designed to be anti-liquidation, as I will explain in the next paragraph. Liquidations occur when the divider is greater than or equal to the total amount of ETH in the contract or when the divider is less than or equal to zero. To liquidate the bulls, for example, we reset the total number of bull LP tokens to zero and we increment a variable called "bullx" by one, effectively making all the old bull LP tokens unusable. The EVM does not offer an easy way of deleting all values from a mapping, so we have to use this "bullx" method. Again, this should never happen, but we still need to include the code to handle it, just in case.

As I mentioned, liquidations should never happen. In fact, I named this project Solidifi because solidify is the opposite of liquidate. The name Soldifi is also a reference to DeFi and Solidity. So, how are liquidations prevented? Well, we need to keep the pool balanced. We need the green side and the red side to be roughly the same size. To accomplish this, we use a fee structure. We charge higher fees for deposits/withdrawals that imbalance the pool and we charge lower fees for deposits/withdrawals that help balance the pool. For example, if 80% of the pool is green ETH, you will pay a lot of fees if you try to deposit more ETH into the green side. Withdrawing red ETH will also incur high fees, since we need to keep as much red ETH in the pool as possible. The fee structure is designed to basically prevent any deposits or withdrawals that cause the pool to have a 90:10 imbalance. We will not actually prevent people from moving their ETH, but the fees will be so high that it isn't financially smart to do so. This prevents liquidation attacks where a bad actor could deposit massive amounts of ETH into one side of the pool, causing the other side of the pool to be liquidated after even a small movement in ETH's price. If someone attempts a liquidation attack, they will pay such a high fee that they will actually lose more money than they can steal. Currently, fees are not actually implemented. I have not been able to come up with the right fee structure, yet. This is the logical next step in development.

Besides updateDivider, there are four other relevant functions: buyBull, buyBear, sellBull, and sellBear. These functions allow you to buy or sell your green (bull) ETH or red (bear) ETH. They simply use LP tokens to keep track of what share of the pool each user owns. LP tokens are commonplace in the world of cryptocurrency. I would recommend this article for anyone wanting to learn more about LP tokens.

Lastly, I should mention the math library that is used in my contract. Many math functions, including anything involving floating point numbers, are not supported by the EVM. So, I have imported ABDK's math library to allow the contract to do floating-point division and logarithmic functions. This library converts integers into a signed hexadecimal numerator, where the denominator is always 2^64. Since the denominator is the same for all numbers, it does not have to be stored. Having the same denominator also allows for easy addition and subtraction. The point of using a denominator is that we can use floating point numbers, since any number that is not a multiple of 2^64 will represent a floating point number.

V. Risks

- Liquidation attacks can occur if someone deposits massive amounts of ETH into one side of the pool. A proper fee structure can prevent this.
- Anyone who has enough money to manipulate ETH's price can exploit my protocol. This person would have to be a billionaire.
- Reliance on an Oracle is always a risk. My project is only as safe as ChainLink.
- There is always a delay between the price of ETH and the value retrieved from the oracle, which could present an opportunity for attackers. One simple way to prevent these attacks is to put a time delay on all deposits and withdrawals.
- It is counterintuitive to have people betting ETH on ETH's price decreasing. People who own ETH generally do not think that ETH's price will decrease. However, if they bet on ETH's price going down but the price actually goes up, they will still make money overall because they own ETH. This means that people can use my protocol if they want to lower their risk without selling their ETH. I think this is a useful option that doesn't currently exist in the market. If the protocol still struggles to find people willing to bet ETH on ETH's price going down, we can simply use USDC instead of ETH for all deposits and withdrawals.

VI. Future improvements

- The fee structure still needs to be designed and implemented. Currently, we just prevent deposits and withdrawals that cause the protocol's balance to exceed a certain ratio.
- We should make the LP tokens an actual ER20 coin. Currently, the LP tokens aren't real tokens and, thus, cannot be traded.
- General updates to the aesthetic of the website need to be made.