Ethereum Smart Contract Ponzi Schemes: Part 2

Tree Shaped Ponzi

Alex Roan Follow
Feb 21, 2020 · 5 min read ★

*If you haven't read part one of this series on Ethereum Smart Contract Ponzi Schemes, you can read it **here**.*

In the previous post, I explained how Ethereum is being used to modernize the nefarious world of Ponzi schemes, using a simple Chain Ponzi. In the Chain Ponzi, each new member is linked only to the member previously joined, with the pitch being "we guarantee to double your money". In this post, I'm going to explain the Tree-Shaped Ponzi.

Just as the name suggests, rather than link each new member of the scheme to the previous member in a 1:1 ratio, a tree-shaped scheme branches out from the original root. Picture it as a Pyramid.

The root node is at the top, and new members join with knowledge of an address of a previously joined member of the scheme when investing money in.

Here's how it works.

Referral Bonus Platform!

As the owner of the scheme, I deploy the contract and advertise my scheme as a **ONCE IN A LIFETIME INVESTMENT OPPORTUNITY, 50% REFERRAL BONUSES FOR GETTING YOUR FRIENDS ONBOARD!**

I post links everywhere to where people can join my referral scheme and finally, I get a bite.

1. Alice uses my referral code (my ether address) and invests 1 ETH. This goes all to me because I am her parent node and the root node of the tree.
2. She knows that for every new member she brings in, she gets a 50% referral bonus, so she invites Bob.
3. Bob, using Alice's referral code (her ether address) invests 1 ETH. Bob's parent node is nowAlice, and her parent node is me. The contract distributes Bob's invested money UP the tree, halving each time until the root is hit. So, Alice gets 0.5 ETH, I get 0.25 ETH, and because I'm the root, I get the remaining 0.25 ETH also. I stand at +1.5 ETH, Alice stands at -0.5 ETH and Bob is at -1 ETH
4. Alice manages to invite another member, Charlie, who invests 1 ETH. 0.5 ETH goes to Alice, 0.25 ETH to me, and the remaining 0.25 ETH to me as well. I'm at +2 ETH, Alice has broken even, Bob still at -1 ETH.
5. Bob has invited Dave, who invests 1 ETH. Bob gets 0.5 ETH, Alice gets 0.25 ETH (because she's Bob's parent), and I get 0.125 ETH plus the remaining 0.125 ETH. My coffers now at +2.25 ETH, Alice: +0.25 ETH, Bob: -0.5 ETH.

The main draw for participants here is that they are directly rewarded for the more people *they* can sign up using their referral code. If other existing members who aren't branched from them beat them to the punch, they don't receive any commission at all.

Members are also rewarded for the number of new members that *their* new members bring in, as the referral cascades upwards. Using some more commonly known terms: Each member is rewarded for the performance of their ***Downline***, and their ***Upline*** is rewarded from them.

This is likely sounding very familiar to people who have participated in multi-level marketing schemes.

Code

The full project for this code can be found here

```
pragma solidity ^ 0.5.0;

import "@openzeppelin/contracts/math/SafeMath.sol";

contract Tree {

using SafeMath for uint;

struct User {

address payable inviter;

address payable self;

}

mapping(address => User) public tree;

address payable public top;

constructor() public {

tree[msg.sender] = User(msg.sender, msg.sender);

top = msg.sender;

}

function enter(address payable inviter) external payable {

require(msg.value >= 1 ether, "Must be at least 1 ether");

require(tree[msg.sender].inviter == address(0), "Sender can't already
exist in tree");

require(tree[inviter].self == inviter, "Inviter must exist");

tree[msg.sender] = User(inviter, msg.sender);

address payable current = inviter;

uint amount = msg.value;

while(current != top) {

amount = amount.div(2);

current.transfer(amount);

current = tree[current].inviter;

}

top.transfer(amount);

}

}
```

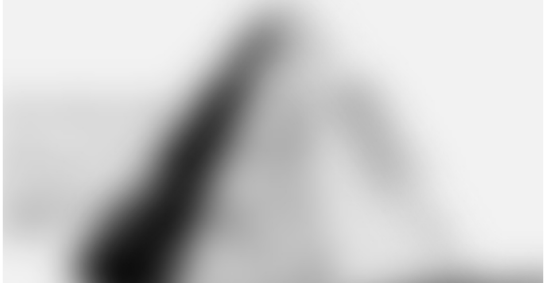
Much like the Chain Ponzi contract, this is a simple piece of code.

Once the constructor is called by the creator of the contract, the ***enter()*** function is used every time a new member joins the scheme. It takes one parameter representing the address of the referer and uses it to create a new node in the ***tree*** using the ***User*** struct. It then distributes referral bonuses up the tree immediately.

Conclusion

Unlike the Chain Ponzi which offers to double your money, the Tree-shaped Ponzi offers a much higher limit on the potential profits each member can earn. So long as you can attract a lot of participants in your ***downline***, and encourage those to do the same, you will make money without much effort thereafter.

This will sound very familiar to multi-level marketers because it's the same model for profit. The issue is that because of the perceived benefits of joining this scheme, the tree is likely to grow exponentially.



Each member has 6 child members.

The diagram above shows the number of people that need to be invested in the scheme if at each level every member signs up 6 members with their referral code. By level 13, the number required to keep the scheme moving forward is almost twice that of the global population.

If the scheme ever gets to that point, the only winners will be the early investors, and they win **BIG**. Unfortunately for the newest members who can find any more people to add to their downline, their money's gone forever, distributed up the tree and dispersed amongst the early adopters.

If you haven't read the first in the series please check it out here!

Want to read the next one, part 3 is available now!

. . .

This article was inspired by a paper entitled "Dissecting Ponzi schemes on Ethereum: identification, analysis, and impact" by Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, Roberto Saia.

. . .

Learn More

> If you enjoyed this post and want to learn more about Smart Contract security, Blockchain Development or the Blockchain Space in general, I highly recommend signing up to the **Blockgeeks platform**. They have courses on a wide range of topics in the industry, from Coding to Marketing to Trading. It has proven to be an invaluable tool for my development in the Blockchain space.

WRITTEN BY

Alex Roan

Engineer at Chainlink Labs.

Follow

BlockCentric

Insights and Analysis in Blockchain Development and The Blockchain Industry.

Follow