Bitcoin Blink
Ajay joshua

One-way bridge with Merkle chain v0.1

OVERVIEW

A one-way bridge between two cryptocurrency chains utilizes a Merkle chain, This method enables the secure and reliable transfer of cryptocurrency between the two chains without relying on a centralized API. To transfer coins across the bridge, the proof is generated with the position of the data and the node root. This proof is used to validate the transfer of coins on the other chain using a validator function that reconstructs the Merkle chain. If the node root provided in the proof is valid, a new coin is minted on the other chain.

Problems of current crypto bridges

Despite their potential benefits, current crypto bridges face several problems that can impact users in significant ways. One major issue is the lack of security and reliability of some bridges, which can result in loss of funds and other negative consequences.

The Poly Network suffered a hack that resulted in the loss of over $600 million worth of cryptocurrency. The hack was possible due to vulnerabilities in the bridge mechanism that enabled the attacker to exploit flaws in the network's smart contracts.

These incidents highlight the importance of ensuring the security and reliability of crypto bridges. To address these problems, it is essential to develop and implement robust security and risk management protocols for crypto bridges. This will help to ensure that users can transfer their assets into bitcoin blink with confidence and peace of mind.

GOALS

1. To construct a One-way bridge that will enable a trustless one-way transfer of cryptocurrency into the bitcoin blink chain
2. Unlike other bridges that are popular in practice today, we provide an API-free solution which would be relatively more secure and decentralized.
3. Ensuring and Facilitating the easy and more secure adaptation of bitcoin blink with cross-chain interoperability
4. Finally providing an easy way of migrating into Bitcoin Blink

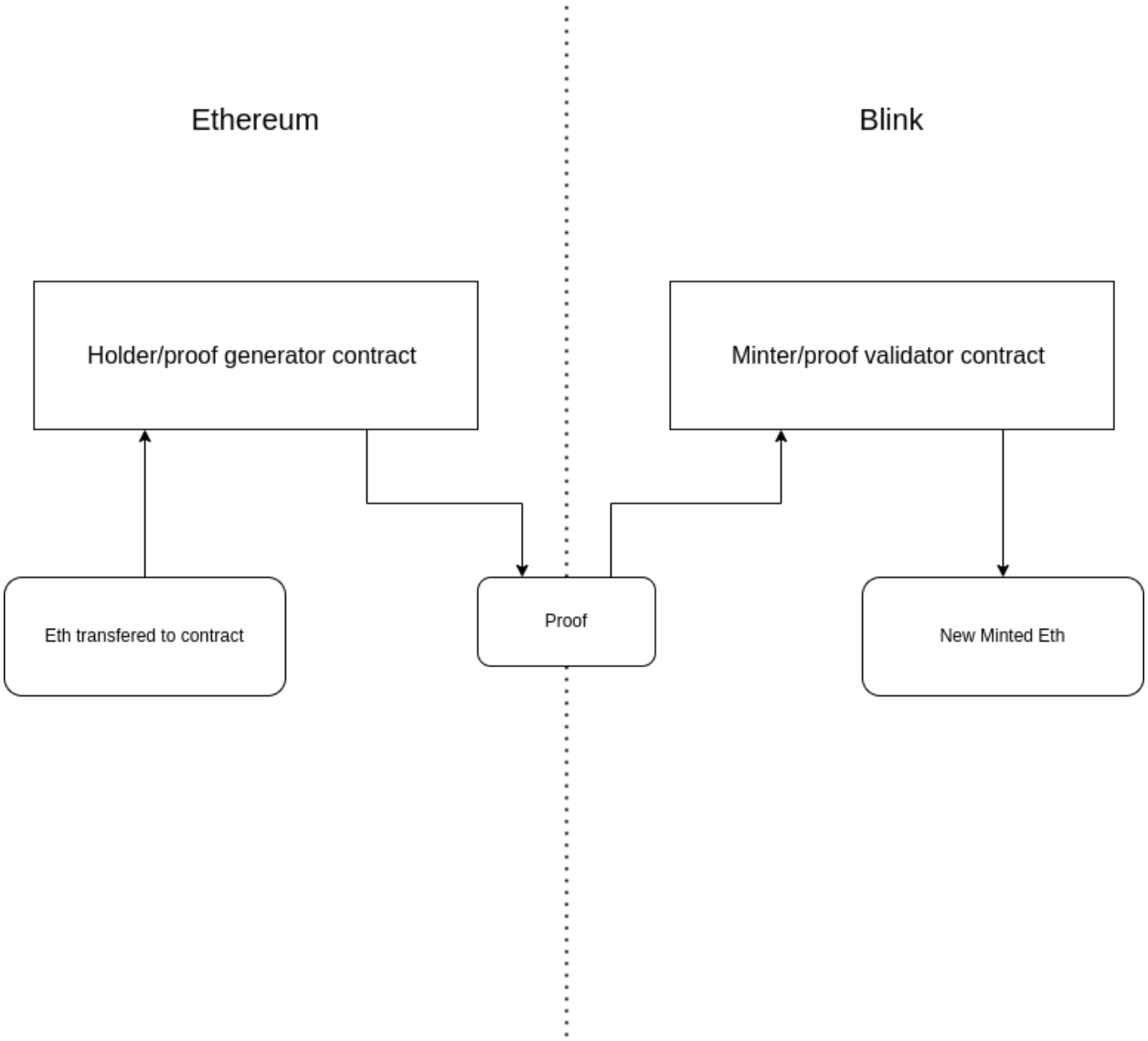
Working of One-way bridge

Concept

In a more generalized way, the protocol works in 2 distinct steps

- Proof generation (User initiating the transfer)
- Proof validation (minting / climbing of the value of the transfer currency)

Considering an example A user is transferring his eth to bitcoin blink eth



Block Diagram of One-way bridge

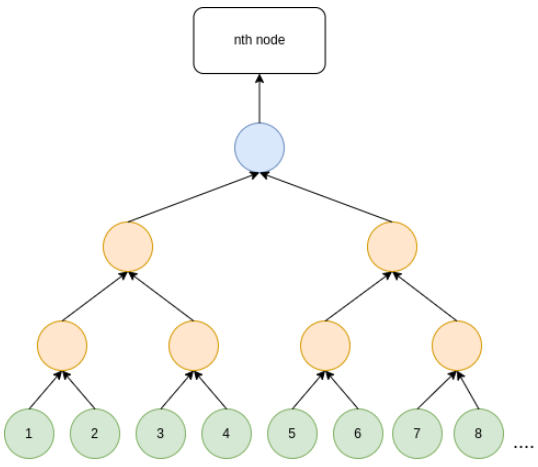
Proof generation (Holder)

- The proof generation or the holder contract is deployed on other networks is the first stage of the one-way bridge
- The contract is capable of receiving and holding the cryptocurrency that's sent to it
- Once the contract receives the amount and verifies the transaction the proof generation is initiated
- Merkel chain is used in the process of making and validating the proof (this will be discussed in detail in later sections)
- The generated proof then can be used to mint the same value of the token in the bitcoin blink chain

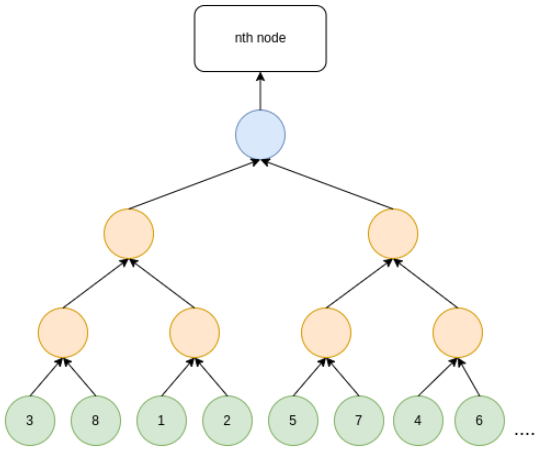
Proof validation (Minter)

- The Minter contract is deployed on the Bitcoin blink chain and is used to validate and mint the tokens from other chains this marks the final stage of the cross-chain transfer of a token
- The contract can mint from constructing and validating a Merkel chain proof
- Due to the one-way nature of the bridge reversing the transaction is not an option altho there is a solution discussed in the later section that allows the user to claim the eth back in case of an error
- The maker contract outputs a p2pk output and it is verified that the transferred token and the minted token id are the same

Merkel chain for generation and validation



generated node



recreated node

Note: the numbering in the Merkle leaf does not represent the leaf number

A Linked list data structure with each node as a Merkle tree and the Merkle root is assigned as the node root and used for validating each node. In each node, the Merkel tree is constructed with 8 data.

Generation:

During the Generation part of the process, the request to transfer the token is obtained and arranged in generating sets of 8 and then a node is generated in the chain and each user gets a proof with the leaf position and the node root.

Users can show this proof to mint on the Bitcoin blink chain. Taking into account the number of transactions happening on each network in Ethereum (Transactions Per Day is at a current level of 1.043M) it would require 8 transactions to make a node and get proof. it would require the user to wait till the set of 8 translation requests happens to get proof it would be a more secure and more decentralized approach to a bridge.

Validation:

In the process of validation the position of the leaf, the node it belongs to and the node root is taken as the proof and the existing Merkle chain on the minting chain is updated the node root on the proof is verified the token is minted on the chain and is sent to the user

This would require the user to wait till all 8 users claim their proof but unlike the generation, this usually happens seamlessly since all the users who minted will claim the token instantly

Considering a point of failure it would be possible to provide proof of the node root on the minting chain and prove that the node root of the particular node is not the same as the node root on the generation chain the user could claim the amount they have sent to the contract but the whole node will be deleted and the proof should go null and void this could be done on node level by constantly verifying the other chains Merkel chain and syncing it

It works on the concept that if we arrange all the data precisely as we do in the generation the output node root should be the same and is possible in the Merkel chain because of its flexible nature and even when one node root is failed the chain keeps working and the other transactions are verified

Solidity codes:

For Proof generation

```
pragma solidity ^0.8.7;


contract TransferHash {


   function transfer(int token_code) public payable returns (bytes32) {

       return GenerateProof(msg.value,msg.sender,token_code);

   }

}
```

For Proof validation

```
pragma solidity ^0.8.0;

import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

import "@openzeppelin/contracts/access/Ownable.sol";

contract Bitcoin is ERC20, Ownable {

   constructor() ERC20("bitcoin", "BTC") {}

   function mint(address to, uint256 amount) public {

       _mint(to, amount);

   }

   function mintTokensAndTransfer(uint256 amount,address recipient,

      bytes32 proof) public {

       require(ValidateProof(proof));

       mint(recipient, amount);

   }

}
```

Summary

A one-way bridge is a mechanism for transferring cryptocurrency between two chains using a secure and reliable Merkel chain. The Merkle chain is a linked list of Merkle trees, each with its node root that represents the Merkle root of its corresponding chain. To transfer coins across the bridge, a proof is generated with the position of the data and the node root, which is used to validate the transfer of coins on the other chain using a validator function that reconstructs the Merkel chain. The one-way bridge mechanism ensures the integrity of the transfer process, enables cross-chain interoperability, and provides a secure and efficient way to connect different crypto chains, while also encouraging innovation and collaboration within the blockchain ecosystem.