Bitcoin Anonymous

Privacy Token Protocol for Binance Smart Chain

Version 1.0

The Bitcoin Anonymous Team

June 2021

OVERVIEW

Privacy is a right. The Bitcoin Anonymous team is looking to establish a privacy focused protocol on the Binance Smart Chain (BSC). Bitcoin Anonymous will achieve this level of privacy by anonymizing the native BEP-20 token (BTCA) and Smart Chain token (BNB). In the future Bitcoin Anonymous will include and add privacy to other BEP-20 tokens to extend the application's use-case.

Privacy is accomplished by utilizing zero-knowledge proofs via smart contracts. These smart contracts are immutable and act as the "vault" to store the deposited tokens which can later be withdrawn with no trace to the originating depositor. Thus creating a private and secure way to send funds to any BSC supported address.

Bitcoin Anonymous has chosen to do this via decentralized desktop wallet apps vs a centralized server dApp. This allows the users of the protocol to feel confident that their deposits and withdrawals will never fail even if the maintainers of the project are not up-to-date with server infrastructure. The only requirement will be the RPC access to the smart chain.

PROTOCOL

The Bitcoin Anonymous privacy protocol is achieved via a multi-step process. Each step is necessary to provide the most secure way to privatize the token. The first step is to deposit tokens into the smart contract vault in a chosen fixed amount. The second step is to withdraw the tokens from the vault to a designated recipient. BNB is required to both deposit tokens and withdraw them in order to cover the gas fees associated with interacting with smart contracts. It is advised to always withdraw tokens to a different BSC wallet address regardless of whether the recipient is yourself or another wallet holder.

Initialization

The protocol is developed by utilizing SNARK proofs and Pedersen hash functions. Specifically the MiMC hash function. The Merkle tree used is at a height of 20. All tree leafs are initialized with 0 values. The zero values will be replaced as deposits are added into the contract. The smart contract acting as the vault stores 100 root values in its history array. The last added Merkle Tree leaf to the root is necessary to create the next root.

Deposit

To deposit tokens into the smart contract, the application will generate two random bigint numbers, known as the nullifier and the secret. The numbers are concatenated and used to create the commitment and nullifier hash. The unhashed nullifier and secret are then part of the specific deposit note that is used provided to the user. The deposit note is in the format of CURRENCY-AMOUNT-INTEGER. The application will then call a function on the specific smart contract vault related to the value of the deposit. The function accepts the commitment and the deposit amount specified if the tree is not full. The deposit is included into the Merkle Tree as a new non-zero leaf.

Withdrawal

To withdraw a token, the application utilizes the deposit note and interacts directly with the same smart contract specified during the deposit process based on the fixed value. The user will enter the deposit note, and recipient address into the application. The application will parse the note into the different values including, the currency, token amount and unhashed nullifier and secret. The nullifier and secret are once again used to create the commitment and nullifier hash. The application requires the deposit history of the smart contract in order to recreate the Merkle tree and search the leaves for the commitment. Once found, the nullifier hash is used to compute a proof that the tokens are unspent and available in the tree. The smart contract verifies the proof and uniqueness of the nullifier hash. Once proved, the tokens are transferred to the recipient and the nullifier hash is added to the list of nullifier hashes.

SECURITY

Bitcoin Anonymous claims the following security statements:

- Only tokens deposited into the contract vaults can be withdrawn.
- Tokens can be withdrawn only once.
- Tokens can only be withdrawn with a valid formatted deposit note.
- Proofs are binding. The same proof cannot be used with a different nullifier hash, recipient address, or token amount.

REFERENCES

- Tornado Cash Privacy Solution. https://github.com/tornadocash
- Typhoon Network. https://typhoon.network
- Iden3: Pedersen Hash. https://iden3-docs.readthedocs.io/en/latest/iden3_repos/research/publications/zkproof-standards-workshop-2/pedersen-hash/pedersen.html
- Iden3: Circomlib and Solidity Verifier generation. https://github.com/iden3/circomlib/blob/master/src/mimcsponge_gencontract.js
- Binance Smart Chain. https://www.binance.org/en/smartChain