ADVANCE CYBER PROJECT

Topic : Creating Cryptocurrency using Blockchain Technology

SUBMITTED BY **SUBMITTED TO**

Rishab Banthia - 18BCN7076 **Prof. Ajith Jubilson**
Dept. of Computer Science

INTRODUCTION

A cryptocurrency is a digital or virtual currency that is secured by cryptography, which makes it nearly impossible to counterfeit or double-spend. Many cryptocurrencies are decentralized networks based on blockchain technology—a distributed ledger enforced by a disparate network of computers. A defining feature of cryptocurrencies is that they are generally not issued by any central authority, rendering them theoretically immune to government interference or manipulation.

KEY TAKEAWAYS

- A cryptocurrency is a form of digital asset based on a network that is distributed across a large number of computers. This decentralized structure allows them to exist outside the control of governments and central authorities.
- The word “cryptocurrency” is derived from the encryption techniques which are used to secure the network.
- Blockchains, which are organizational methods for ensuring the integrity of transactional data, is an essential component of many cryptocurrencies.
- Many experts believe that blockchain and related technology will disrupt many industries, including finance and law.
- Cryptocurrencies face criticism for a number of reasons, including their use for illegal activities, exchange rate volatility, and vulnerabilities of the infrastructure underlying them. However, they also have been praised for their portability, divisibility, inflation resistance, and transparency.

TYPES OF CRYPTOCURRENCY

Coins refer to cryptocurrencies built on their independent blockchain network. The most famous example is Bitcoin (BTC), which is also the world’s largest cryptocurrency by market capitalization.

Bitcoin is powered by its native blockchain network. Similarly, Litecoin (LTC) and Ethereum (ETH) function on their respective blockchains. These blockchains may differ in their size, rules, miners, performance, etc.

Some of the popular coins are Bitcoin (BTC), Ripple (XRP), Ethereum (ETH), Dogecoin (DOGE), and Litecoin (LTC).

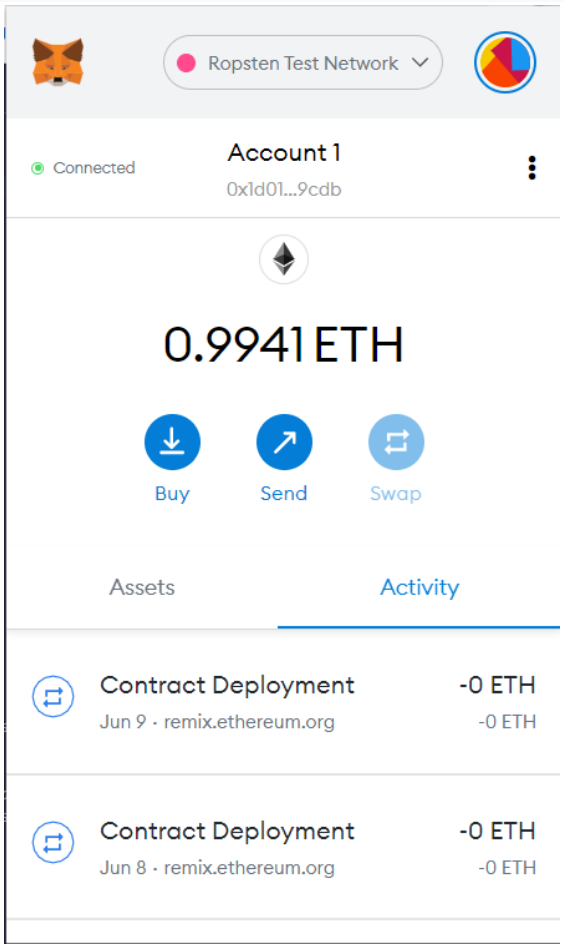
In the meantime, tokens refer to cryptocurrencies that don't have a blockchain network of their own. Instead, these cryptocurrencies are built on another blockchain. Users can create digital tokens using one of the many platforms in the DeFi (Decentralized Finance) ecosystem.

Ethereum is one of the most popular choices, thanks to its support for smart contracts. Most of the digital tokens found today are ERC-20 tokens since the Ethereum platform easily enables creating tokens on top of the Ethereum blockchain.

Currently, thousands of tokens exist in the market. Tether (USDT), USD Coin (USDC), DAI, UMA, and Basic Attention Token (BAT) are some of the commonly-used digital tokens out there. These tokens may have powers other than value transfer.

IMPLEMENTATION

Creating a meta mask wallet and having a test ether in the ropsten test network.



Next going to remix and create two files Cyber. sol and EIP20Interface.sol

FILE EXPLORERS

Workspaces

Browser

artifacts

Cyber_metadata.json

Cyber.json

Cyber.sol

EIP20Interface.sol

Home | Cyber.sol | EIP20Interface.sol

```
pragma solidity ^0.4.21;

import "./EIP20Interface.sol";


contract Cyber is EIP20Interface {

    uint256 constant private MAX_UINT256 = 2**256 - 1;
    mapping (address => uint256) public balances;
    mapping (address => mapping (address => uint256)) public allowed;
    /*
    NOTE:
    The following variables are OPTIONAL vanities. One does not have to include them.
    They allow one to customise the token contract & in no way influences the core functionality.
    Some wallets/interfaces might not even bother to look at this information.
    */
    string public name;                   //fancy name: eg Simon Bucks
    uint8 public decimals;                //How many decimals to show.
    string public symbol;                 //An identifier: eg SBX

    function Cyber(
        uint256 _initialAmount,
        string _tokenName,
        uint8 _decimalUnits,
        string _tokenSymbol
    ) public {
        balances[msg.sender] = _initialAmount;               // Give the creator all initial tokens
        totalSupply = _initialAmount;                        // Update total supply
        name = _tokenName;                                   // Set the name for display purposes
        decimals = _decimalUnits;                            // Amount of decimals for display purposes
        symbol = _tokenSymbol;                               // Set the symbol for display purposes
    }

    function transfer(address _to, uint256 _value) public returns (bool success) {
```

FILE EXPLORERS

Workspaces

Browser

artifacts

Cyber_metadata.json

Cyber.json

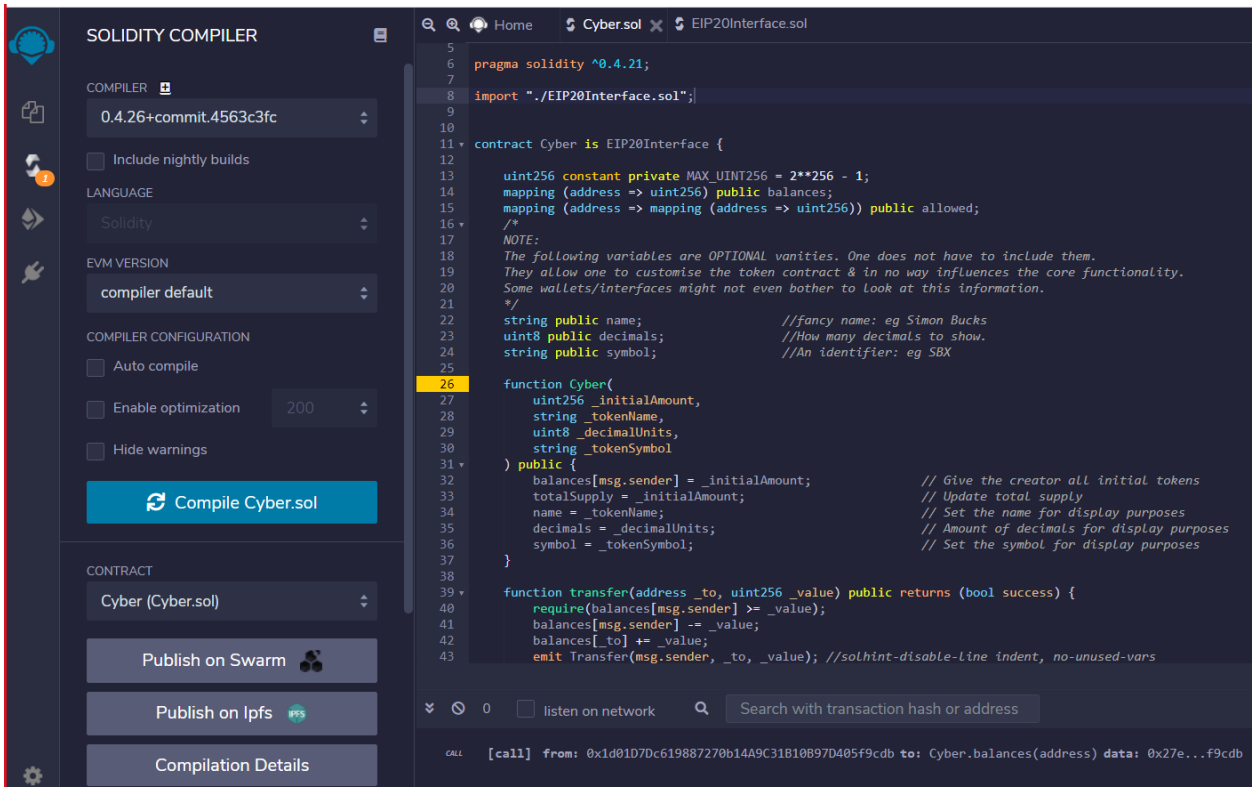
Cyber.sol

EIP20Interface.sol

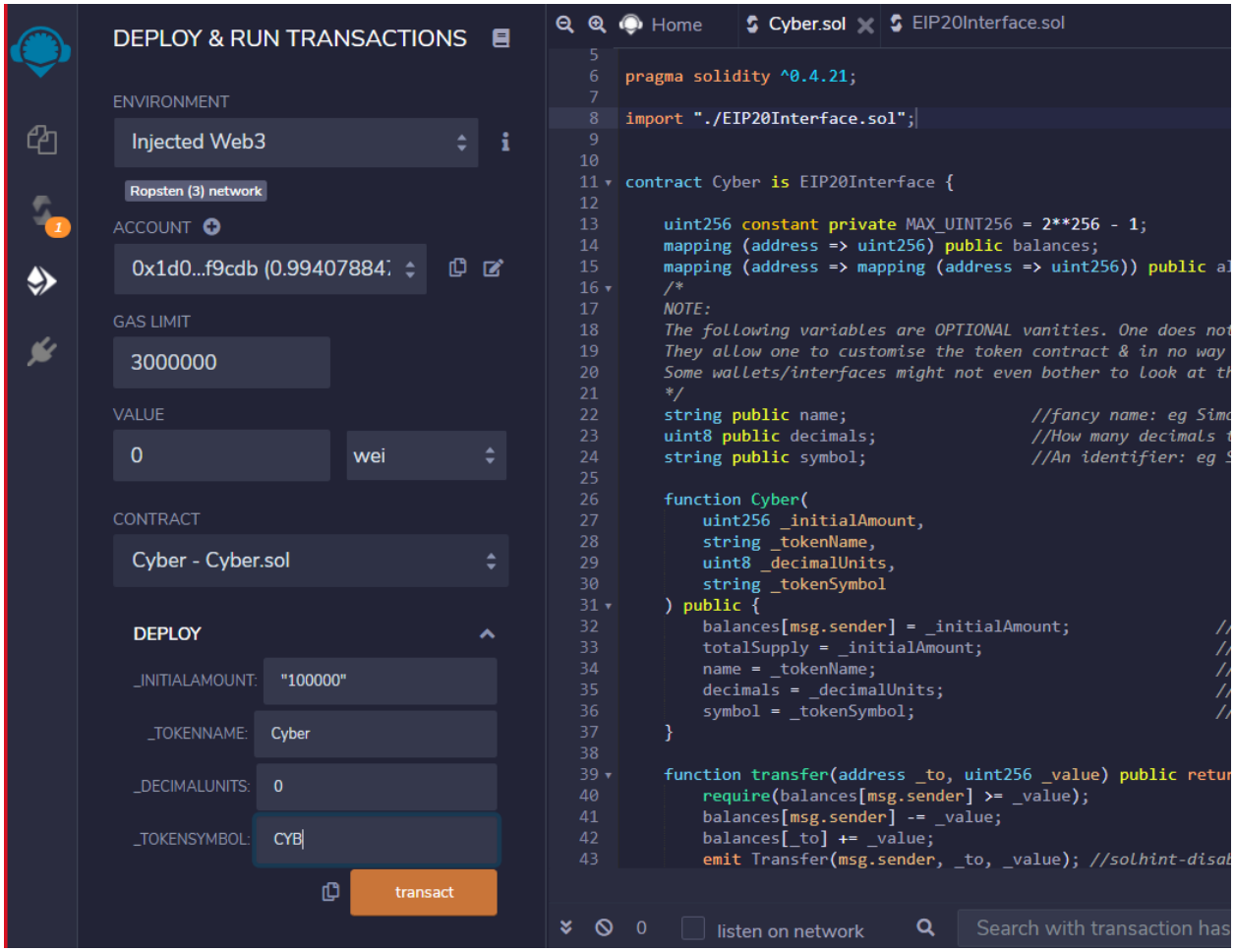
Home | Cyber.sol | EIP20Interface.sol

```
contract EIP20Interface {
    /* This is a slight change to the ERC20 base standard.
    function totalSupply() constant returns (uint256 supply);
    is replaced with:
    uint256 public totalSupply;
    This automatically creates a getter function for the totalSupply.
    This is moved to the base contract since public getter functions are not
    currently recognised as an implementation of the matching abstract
    function by the compiler.
    */
    /// total amount of tokens
    uint256 public totalSupply;

    /// @param _owner The address from which the balance will be retrieved
    /// @return The balance
    function balanceOf(address _owner) public view returns (uint256 balance);

    /// @notice send `_value` token to `_to` from `msg.sender`
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transfer(address _to, uint256 _value) public returns (bool success);

    /// @notice send `_value` token to `_to` from `_from` on the condition it is approved by `_from`
    /// @param _from The address of the sender
    /// @param _to The address of the recipient
    /// @param _value The amount of token to be transferred
    /// @return Whether the transfer was successful or not
    function transferFrom(address _from, address _to, uint256 _value) public returns (bool success);

    /// @notice `msg.sender` approves `_spender` to spend `_value` tokens
    /// @param _spender The address of the account able to transfer the tokens
    /// @param _value The amount of tokens to be approved for transfer
```

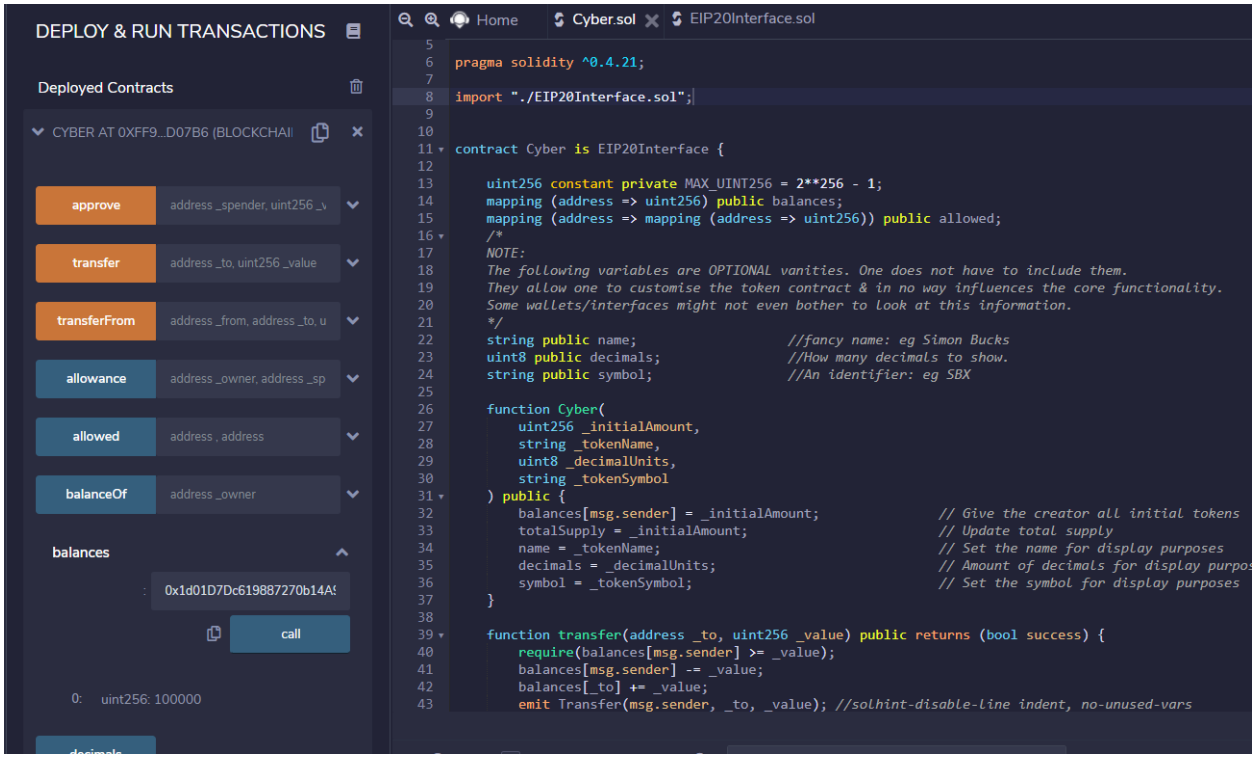
Next, compiling the Cyber. sol



After that deploying around 100000 crypto.

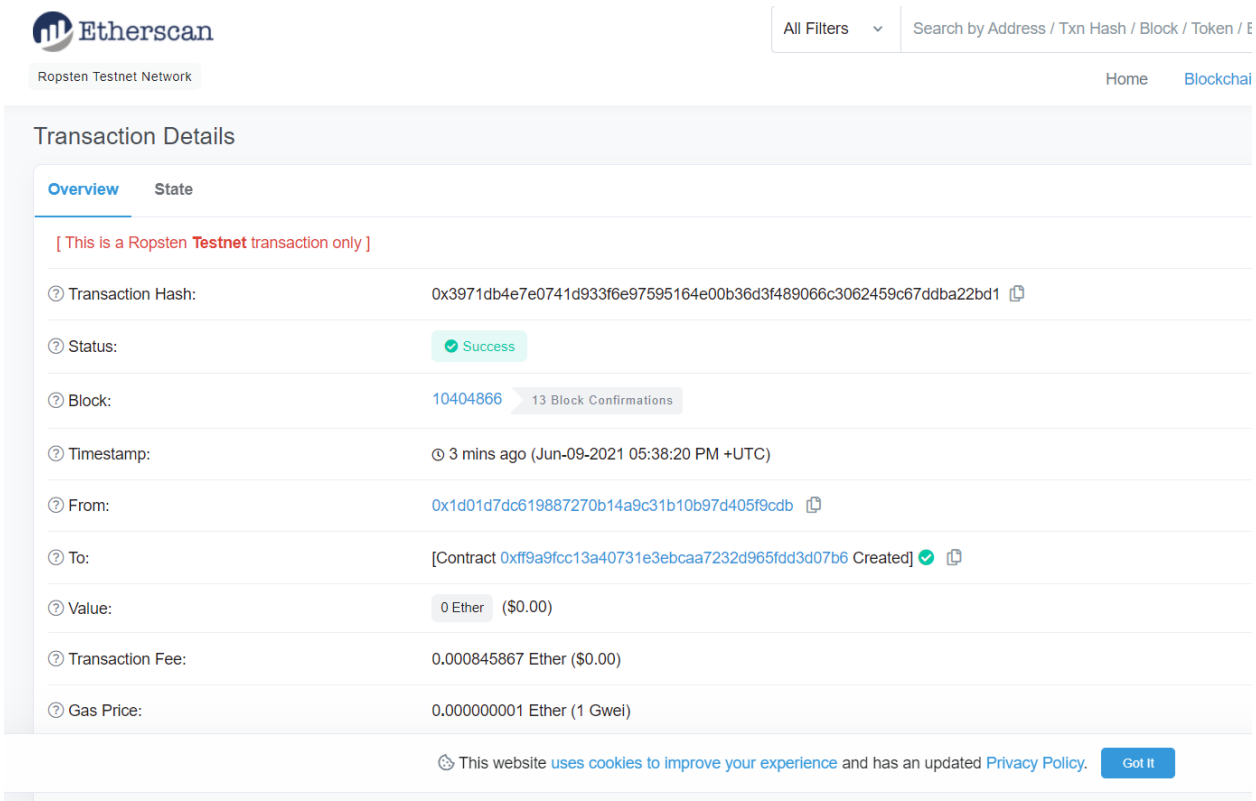


Check the transaction detail and deployment contract



You have successfully created your Cyber test crypto.

Now to check go to ropsten.etherscan.io



This proves that our transaction was successful.

FUTURE GOALS

- Create a cryptocurrency token on the Binance Smart chain.
- Give liquidity to cryptocurrency using the platform Pancake Swap
- Giving it value and increasing the market capitalization and adding liquidity to existing crypto that we have and use it for trading.
- This would help people to become cashless and try to pay using cryptocurrencies which will be the future of payment where our cryptocurrency will use very little fee for the transaction and make it more profitable after investing in it.

CONCLUSION

We have successfully tested the crypto creation and have performed this task on the remix for creating the crypto and ropsten test network which acted as the testing network and confirmed it using the ropsten.etherscan.io platform.