

Vivekanand Education Society's Institute of Technology
Department of Computer Engineering



Subject: Blockchain

Semester:7

Class: D17B

| | |
|------------------------|--|
| Roll No: 31 | Name:Tithi Jhamnani |
| Practical No: 7 | Title: Implement the embedding wallet (Metamask) and transaction using Solidity |
| DOP: | DOS: |
| Grades: | LOs Mapped: LO2 |
| Signature: | |

AIM: Implement the embedding wallet (Metamask) and transaction using Solidity

Lab Objectives: To explore Blockchain concepts.

Lab Outcomes (LO): Design Smart Contract using Solidity (LO2)

Task to be performed:

1. Set Up MetaMask:
 - a. Install MetaMask
 - b. Create or Import an Account:
 - c. Fund Your Wallet : Sepolia Testnet (0.5 ETH per day) / RSK Testnet (0.05 RBTC per day)
2. Connect the Sepolia Testnet / RSK Testnet to Remix IDE
3. Create a Simple Solidity Smart Contract based on the MiniProject chosen
4. Compile and Deploy the Smart Contract.
5. Check the transaction details on the RSK Explorer
6. Interact with the smart contract

Theory:

What is a Metamask?

MetaMask is a popular cryptocurrency wallet and browser extension that allows users to manage their Ethereum-based assets, interact with decentralized applications (DApps), and access the Ethereum blockchain from within their web browsers. It provides a user-friendly interface for handling cryptocurrency transactions and interacting with various decentralized services and platforms.

Key features and functions of MetaMask:

1. Wallet: MetaMask functions as a digital wallet that stores Ethereum and Ethereum-based tokens (ERC-20 and ERC-721 tokens). Users can send, receive, and manage their cryptocurrencies through MetaMask.
2. Browser Extension: MetaMask is typically used as a browser extension for popular web browsers like Chrome, Firefox, and Brave. It seamlessly integrates with your browser, allowing you to interact with DApps directly from your browser window.
3. DApp Integration: MetaMask allows users to interact with decentralized applications (DApps) on the Ethereum blockchain. When you visit a DApp-enabled website, MetaMask prompts you to connect your wallet, enabling you to use the DApp's features and make transactions.

4. **Security:** MetaMask prioritizes security. Users create a unique passphrase to secure their wallet, and private keys are stored locally on their device, rather than on a centralized server. It also provides options for hardware wallet integration for added security.

5. **Network Management:** Users can connect to different Ethereum networks (Mainnet, Testnet, and custom networks) within MetaMask, allowing them to test and interact with DApps on various Ethereum test networks before using real assets on the mainnet.

6. **Transaction History:** MetaMask provides a transaction history that displays all the transactions made from the wallet, including their status, gas fees, and transaction details.

7. **Token Support:** In addition to Ethereum (ETH), MetaMask supports a wide range of ERC-20 and ERC-721 tokens. Users can add custom tokens to their wallets as needed.

What is a test net?

A testnet (short for "test network") is a separate and parallel blockchain network used by developers and users to test new features, applications, and smart contracts without risking real cryptocurrency or assets. Testnets are essential for ensuring the functionality and security of blockchain software and applications before deploying them on the mainnet, which is the live, production blockchain network where real transactions occur.

Key characteristics and purposes of testnets:

1. **No Real Value:** Testnets use tokens or cryptocurrencies that have no real-world value. These tokens are often obtained for free from a faucet or developer tools. Since they have no monetary value, users can experiment with them without any financial risk.

2. **Sandbox Environment:** Testnets create a controlled environment for developers and users to test blockchain applications, smart contracts, and software changes without affecting the mainnet. This allows developers to identify and resolve bugs, security vulnerabilities, and other issues before deploying on the live network.

3. **Protocol Upgrades:** Developers often use testnets to test protocol upgrades or changes to the blockchain's underlying software. This helps ensure that these changes are compatible with existing applications and do not disrupt the network.

4. Development and Testing: Blockchain developers use testnets to build and test decentralized applications (DApps) and smart contracts. This enables them to fine-tune their code and ensure it functions as intended before releasing it to a wider audience.

5. Community Engagement: Testnets also serve as a way to engage the blockchain community in the testing process. Users can participate in testnet activities, report issues, and provide feedback to improve the reliability and security of the network.

Examples of testnets include:

- Ethereum Testnets: Ethereum has several testnets, such as Ropsten, Rinkeby, and Goerli, where developers can test smart contracts and DApps before deploying them to the Ethereum mainnet.

- Bitcoin Testnet: Bitcoin has a testnet called "Testnet" (often referred to as Bitcoin Testnet3) where developers can experiment with Bitcoin-related software and transactions.

- Binance Smart Chain Testnet: Binance Smart Chain, a blockchain network compatible with Ethereum, also has testnets for developers to test their projects.

Testnets are a critical part of the blockchain development process, as they help ensure that blockchain applications and protocols are secure, reliable, and functional before they are exposed to real assets and users on the mainnet.

List the steps to connect a Metamask with a Remix IDE for performing transactions.

To connect MetaMask with Remix IDE for performing transactions on the Ethereum blockchain, you can follow these steps:

1. Install MetaMask:

If you haven't already, install the MetaMask browser extension by visiting the official MetaMask website and following the installation instructions for your preferred web browser (e.g., Chrome, Firefox).

2. Create or Import an Ethereum Wallet:

After installing MetaMask, you'll need to either create a new Ethereum wallet or import an existing one. Follow the on-screen prompts to set up your wallet and securely store your seed

phrase.

3. Connect to the Correct Network:

Ensure that MetaMask is connected to the Ethereum network you want to use (e.g., Mainnet, Ropsten Testnet). You can switch networks by clicking the network name at the top of the MetaMask extension and selecting the desired network from the dropdown menu.

4. Fund Your MetaMask Wallet:

To perform transactions, you'll need some ETH in your MetaMask wallet to cover gas fees. If your wallet doesn't have any ETH, you can purchase it on a cryptocurrency exchange and then transfer it to your MetaMask address.

5. Open Remix IDE:

Visit the Remix IDE website (<https://remix.ethereum.org/>) in your web browser. Remix is a web-based development environment for Ethereum smart contracts.

6. Select the Right Environment:

In Remix IDE, you can choose different environments for your development and testing. To connect Remix to MetaMask, select the "Solidity" environment.

7. Configure Remix to Use MetaMask:

To connect Remix with MetaMask, you need to configure the Remix settings:

- In Remix IDE, look for the "Settings" tab on the left panel.
- Under "Plugin," select "Solidity Compiler."
- In the "General Settings" section, check the box labeled "Auto Compile" to automatically compile your smart contracts.
- In the "Plugin Settings" section, find the "Environment" option and select "Injected Web3" from the dropdown menu.

8. Connect Remix to MetaMask:

With Remix configured to use MetaMask, you should see a "Connect to MetaMask" button or similar option within Remix. Click this button to establish the connection.

9. Authorize Remix in MetaMask:

A popup window from MetaMask will appear, asking you to authorize Remix to connect to your MetaMask wallet. Confirm the connection by clicking "Connect" or "Approve."

10. Select the MetaMask Account:

If you have multiple Ethereum accounts in MetaMask, you may need to select the specific account you want to use within Remix.

11. Start Developing and Transacting:

With Remix now connected to MetaMask, you can start writing, deploying, and interacting with Ethereum smart contracts in the Remix IDE. When you initiate transactions, MetaMask will prompt you to confirm and pay any associated gas fees.

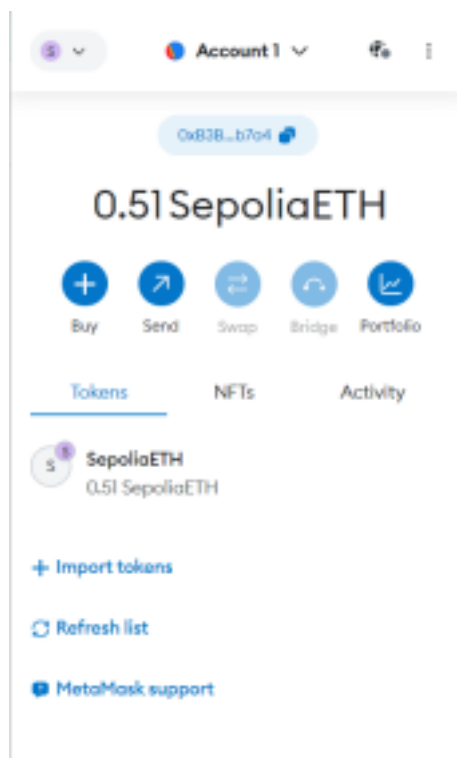
12. Monitor Transactions:

After submitting transactions in Remix, you can monitor their progress and view transaction details in MetaMask's transaction history.

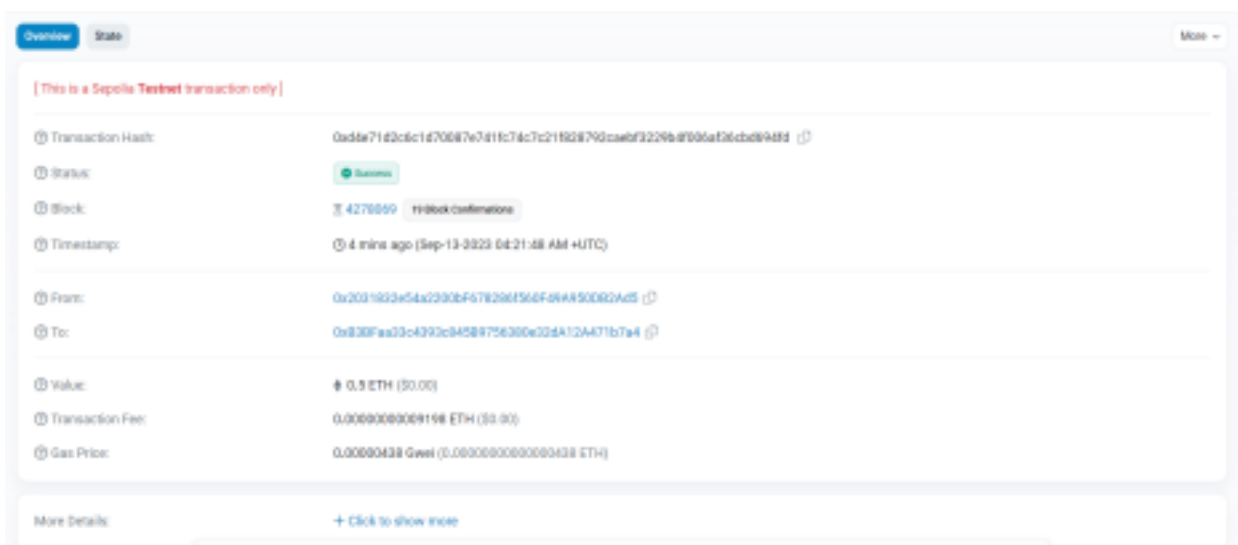
Conclusion:

Connecting MetaMask with Remix IDE streamlines Ethereum smart contract development and testing. This integration facilitates secure transactions, providing developers with a user-friendly interface to interact with the Ethereum blockchain. By following the steps outlined above, users can harness the power of Remix IDE while leveraging MetaMask's wallet functionalities, making it easier to create, deploy, and experiment with smart contracts and decentralized applications before deploying them on the live Ethereum network.

Metamask Wallet with test ether in Sepolia



Sepolia Testnet Transaction Details



Adding RSK Testnet manually

General Networks > Add a network > Add a network manually

Advanced

Contacts

Snap

Security & privacy

Alerts

Networks

Experimental

About

Network name

New RPC URL

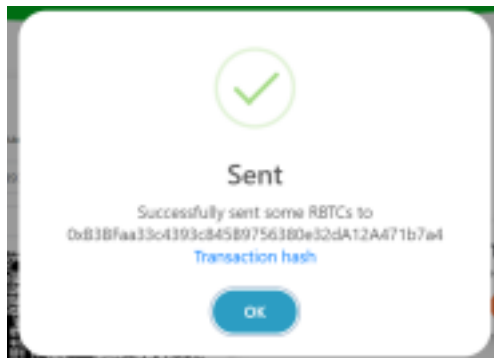
Chain ID

Currency symbol

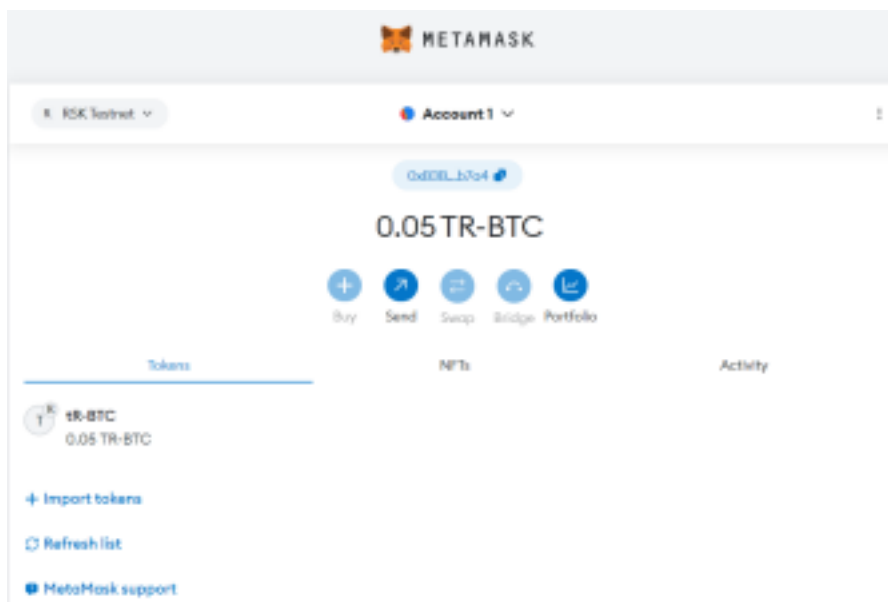
The network with chain ID 31 may use a different currency symbol (tRBTC) than the one you have entered. Please verify before continuing.

Block explorer URL (Optional)

Cancel Save



Metamask Wallet with test ether in RSK Testnet



Details for deploying a contract in Remix IDE

RSK Testnet

Account 1

New contract

https://remix.ethereum.org

CONTRACT DEPLOYMENT

DETAILS DATA

EDIT

Estimated gas fee ⓘ

0.00000929

0.000009 TR-BTC

Site suggested

Max fee: 0.00000929 TR-BTC

Total

0.00000929

0.00000929 TR-BTC

Amount + gas fee

Max amount: 0.00000929 TR-BTC

Reject

Confirm

Successful deployment of contract using Metamask account

✓ [block:417666 txIndex:4] from: 0x835...1b7a4 to: Storage.constructor value: 0 wei data: 0x05...20033 logs: 0 hash: 0xc53...de323

Debug

status

true Transaction mined and execution succeed

transaction hash

0xad0e07f6a034a00a25b9629e393d05a120aef74d93afec90a233070ef ⓘ

block hash

0xc91edc7207968e556b642a28e613e0d756d77be7f55e6817aeeb0de915 ⓘ

block number

417666 ⓘ

contract address

0x7220e060f00620041809527527f0312807135 ⓘ

from

0xb08f6a13b4381b4580776100b032032a471b7a6 ⓘ

to

Storage.constructor ⓘ

gas

242525 gas ⓘ

transaction cost

242525 gas ⓘ

input

0x05...20033 ⓘ

decoded input

() ⓘ

Details of Contract deployment

Contract deployment

✕

Status

Confirmed

[View on block explorer](#)

[Copy transaction ID](#)

From

 0x838...b7a4



To

New contract

| Transaction | |
|-------------------|-------------------|
| Nonce | 0 |
| Amount | -0 TR-BTC |
| Gas Limit (Units) | 142525 |
| Gas Used (Units) | 142525 |
| Gas price | 0.065164 |
| Total | 0.00000929 TR-BTC |

+ Activity log

+ Transaction data