SecretPath + WalletConnect integration

Learn how to connect EVM compatible chains with Secret Network using WalletConnect.

SecretPath WalletConnect

SecretPath connects Secret Network to20+ EVM chains , enabling public EVM chains to call functions on Secret Network while preserving the privacy of the inputs and validity of the outputs.

WithWalletConnect , you can create a seamless user experience with one wallet login thatallows users to interact with Secret Network smart contracts on every SecretPath-connected chain.

Seehere for a fullstack SecretPath + WalletConnect demo In this tutorial, you will learn how to configure SecretPath and WalletConnect in React so you can create seamless UI for confidential cross-chain computation :D

Getting Started

1. Sign in to WalletConnect with your crypto wallet andcreate a project id
2. for your project.
3.

When you create your project, select project type "App - Enable users to access your app from any wallet" Create a WalletConnect Project 1. Select "Web3Modal" for type of project 2.

Select Web3Modal Product 1. Select "React" platform to follow along with this tutorial 2.

Select WalletConnect platform 1. Clone the SecretPath WalletConnect repository: 2.

```

Copy git clone https://github.com/writersblockchain/secretpath-walletconnect/tree/main

```

1. install the dependencies:
2.

```

Copy cd frontend npm install

```

1. Update the WalletConnectproject-id
2. with your project-id
3.

Now you're ready to use WalletConnect with SecretPath!

Configuring WalletConnect for SecretPath

Now that you have your environment properly configured, let's understand how everything is connected. Runnpm run start to see your WalletConnect application:

```

Copy npm run start

```

Out of the box, you can use this React application as a starting point for your SecretPath + WalletConnect applications.

Click "Connect Wallet" and sign in to your Metamask or EVM wallet. Click on the network you are currently connected to in order to see all of the EVM networks available:

All of these 20+ networks are connected to SecretPath, which means any of these chains can call functions on Secret Network smart contracts

Now let's breakdown each of the parameters that are required to properly configure our WalletConnectWeb3Modal .

Web3Modal Parameters

We must pass 5 parameters to Web3Modal:chainImages ,ethersConfig ,chains ,projectId , andmetadata .

chainImages

chainImages is an object that associates EVM chain IDs with images that are displayed in the walletConnect UI. You can add additional chain IDs and images for each additional chain you would like included in your WalletConnect application. Currently, every chain that is supported by SecretPath is included.

ethersConfig

ethersConfig is an object that contains boilerplate ethers configuration code.

chains

chains contains the RPC info for each chain that you connect to SecretPath. The chain info is imported from thechains.js file in the config folder. If you have added RPC info for a chain to Metamask before, this should look familiar.

projectId

Your uniqueproject id generated by WalletConnect.

metadata

Metadata unique to your dapp.

Putting it all together: SecretPath + WalletConnect

Now that you have your WalletConnect interface enabled to your liking, let's understand how it connects the EVM to Secret Network with SecretPath.

OpenApp.js . Notice that there are 3 functions working in tandem here:

1. useEffect
2. , which returns the currently enabled chain ID (so every time the user switches to a new chain, this is saved in the application state)
3. requestRandomness
4.
    - this is our SecretPath function which requests a verifiable random number from a Secret smart contract. It takes the parameterchainId
5. , because the application must know whichEVM gateway contract
6. to execute based on which EVM chain is currently connected.
7. querySecret
8.
    - a query function that queries the random number stored in the Secret Network smart contract
9.

Finally, on click,requestRandomness calls therequest_random handle in the Secret smart contract, which returns a random number between 1-200. SecretPath knows which EVM gateway contract to execute based on which chain ID is currently enabled by WalletConnect (you can see the if/else logichere).

Summary

✨ SecretPath integrates Secret Network with over 20 EVM chains, enabling public EVM chains to execute functions on Secret Network while preserving privacy and ensuring output validity. Using WalletConnect, developers can create a seamless user experience, allowing users to interact with Secret Network smart contracts across all SecretPath-connected chains through a single wallet login. This tutorial provides a step-by-step guide to configuring SecretPath and WalletConnect in a React application, starting from signing in with WalletConnect and creating a project, to setting up the development environment. Key configurations include specifying parameters for Web3Modal, such as chainImages, ethersConfig, chains, projectId, and metadata. The tutorial also explains how to connect EVM to Secret Network using functions likerequestRandomness andquerySecret within the app, ensuring smooth and confidential cross-chain computations ✨

Last updated3 days ago On this page * SecretPath * Getting Started * WalletConnect * Configuring WalletConnect for SecretPath * Web3Modal Parameters * Putting it all together: SecretPath + WalletConnect * Summary

Was this helpful? Edit on GitHub Export as PDF