Web3Auth MPC Core Kit Web SDK

Web3Auth's@web3auth/mpc-core-kit SDK is simple and easy-to-use SDK, which helps you implement the Web3Auth MPC Features while giving you the flexibility to customize the UI and UX of the authentication process.

This Documentation is based on the2.3.0-0 Pre Release

SDK Version.â

Requirementsâ

- This is a frontend SDK and can only run in a browser environment
- Basic knowledge of JavaScipt
- Supports all major JavaScript Frameworks, Libraries and Bundlers

note The minimumpricing plan to use this SDK in a production environment is theEnterprise Plan . However, you can use this SDK with all features enabled in the development environment for free.

Web3Auth MPC Infrastructure Componentsâ

With the Web3Auth infrastructure, your key is divided into multiple parts and stored across your devices and our Auth Network. This ensures that your key is always available and never stored in a single place. While in the traditional Web3Auth SDK, your key was dynamically reconstructed in the frontend using threshold signatures, with the new Web3Auth MPC (Multi Party Computation) architecture, it isnever reconstructed . Instead, these partial keys are stored across different locations, and your device is used to make partial signatures for your message/ transaction. These are finally returned to the frontend where using TSS (Threshold Signature Scheme), these signatures are combined to make a final signature. You can use this finally signed message/transaction to make a transaction on the blockchain.

The Threshold Signature Scheme (TSS) is a cryptographic primitive for distributed key generation and signing. The use of TSS in Web3Auth's Auth network is a new paradigm that can provide numerous benefits, especially in terms of security.

As you can notice in this diagram above, the final output, i.e., the User's TSS Account, is generated in multiple stages within the infrastructure. Since this is a TSS- MPC based infrastructure, you don't get back a private key, but signatures that can be used to make transactions on the blockchain. Let's understand each of these stages in detail.

Factorsâ

Social Login Factorâ

This is the primary way for a user to access their account. This step involves authentication using a user's preferred social login provider. The idToken received from the social login provider here is passed to the Web3Auth Auth Network to generate the TSS Shares in the Nodes. By default, these nodes have a threshold of 3/5 that can be customized according to requirements. When a user logs in, the Auth Network generates signatures corresponding to the TSS Shares in the nodes and returns them to the user's end. These signatures are then used alongside other shares to generate the final TSS Account signatures.

Device Factor (Index 2)â

This is the second factor used to access the user's account. This step involves the generation of a TSS Share on the user's device and using that to generate a final signature for the TSS Account alongside the social login factor. This ensures the user logs in using their trusted device and maintains a proper non-custodial setup.

Backup Factor (Index 3)â

A user has a choice to generate as many backup factors as needed to access their account. This step involves the generation of a TSS Share on the user's end and storing them in whichever location they prefer. This share can be used similarly to the device share to generate a final signature for the TSS Account alongside the social login and/or device factors.

Thresholdâ

The threshold is the minimum number of shares required to generate a final signature for the TSS Account. This threshold, by default, is set to 3/5 on the Auth Network and 2/3 for the user's device front. This ensures high availability and ease of access on both ends alongside optimum security. Both these thresholds can be customized according to the requirements.

Components of a Factorâ

TSS Sharesâ

The TSS Shares are the main component needed for the generation of the final working signature of the user. These shares are generated using distributed key generation and are stored in the Auth Network and the user's device. Since these shares are generated using MPC, they arenever reconstructed and always stay decentralized and secure.

However, we have exposed a function to reconstruct the TSS Key from the shares in the SDK. This function has been marked unsafe since there is no direct use case for it other than key exports in the case the user wants to own it completely.

Metadata Key & Sharesâ

The Metadata Key closely mimics the storage of the TSS Shares. The only difference is that the metadata key is always reconstructed and used for its encryption/decryption capabilities. It utilizes basic Shamir's Secret Sharing and is initially generated on a users' front end. The metadata key is utilised for the encryption/decryption of the user metadata stored in the Web3Auth Metadata Server.

One metadata share gives you theread access to the metadata server, while two or more (as the threshold is reached) give you thewrite access.

Factor Keysâ

To enable refresh, deletion, and rotational capabilities on thetssKey , we also introducefactorKeys . Factor Keys are randomly generated keys unique to each factor-generated user's device and backups, like users' phones, chrome extension, on their cloud, assisting third parties, etc.

As share to the TSS Key and/or Metadata Key may rotate, Factor Keys allow a consistent secret to being saved in these different locations.

Understanding the MPC Core Kit SDK Flowâ

By default, for a new user, the MPC Core Kit SDK starts in a 2/2 flow. This means that the user will have to generate a social login factor and a hashed cloud factor. This hashed cloud factor is derived on the SDK front end and stored in the encrypted metadata server. This enables the user to start the login process from any device without creating an MFA factor.

This is done to make sure the user can access their account from any device without having to generate a new factor. However, this makes the initial onboarding, semi-custodial. To make the onboarding completely non-custodial, the user can use theenableMfa feature to generate a device and backup factor and delete the hashed cloud factor. This makes the user's account completely non-custodial in 2/3 setup and enables them to access their account from any device using the backup share. Device share is saved on trusted device and can be used to access the account from that device seamlessly without having to generate a new factor.

Resourcesâ

- Quick Start
- : Get Started with an easy to follow integration of
- Web3Auth
- Example Applications
- : Explore our example applications and try the SDK yourself.
- Troubleshooting
- : Find quick solutions to common issues faced by developers.
- Source Code
- : Web3Auth is open sourced. You can find the source code on our GitHub repository.
- Community Support Portal
- : Join our community to get support from our team and other developers.

Helper SDKsâ

Common Types and Interfacesâ

@web3auth/base

â

This package gives access to common types and interfaces for Web3Auth. This comes in handy by providing you with a

standard way of importing the values you need to work with the SDKs. We highly recommend using it while working with Typescript. Edit this page Previous SafeAuth Next Install