Offline Signing

Sui supports offline signing, which is signing transactions using a device not connected to a Sui network, or in a wallet implemented in a different programming language without relying on the Sui key store. The steps to implement offline signing include:

You must serialize transaction data following Binary Canonical Serialization (BCS). It is supported in other languages.

The following example demonstrates how to serialize data for a transfer using the Sui CLI . This returns serialized transaction data in Base64. Submit the raw transaction to execute as tx_bytes .

Beginning with the Sui v1.24.1 release , the --gas-budget option is no longer required for CLI commands.

The console responds with the resulting value.

All other CLI commands that craft a transaction (such as sui client publish and sui client call) also accept the --serialize-unsigned-transaction flag used in the same way.

You can sign the data using the device and programming language you choose. Sui accepts signatures for pure Ed25519, ECDSA secp256k1, ECDSA secp256r1 and native multisig. To learn more about the requirements of the signatures, see Sui Signatures .

This example uses the sui keytool command to sign, using the Ed25519 key corresponding to the provided address stored in sui.keystore . This command outputs the signature, the public key, and the flag encoded in Base64. This command is backed by fastcrypto. sui keytool sign --address --data

You receive the following response:

To ensure the signature produced offline matches with Sui validity rules for testing purposes, you can import the mnemonics to sui.keystore using sui keytool import . You can then sign with it using sui keytool sign and then compare the signature results. Additionally, you can find test vectors in ~/sui/sdk/typescript/test/e2e/raw-signer.test.ts .

To verify a signature against the cryptography library backing Sui when debugging, see sigs-cli .

After you obtain the serialized signature, you can submit it using the execution transaction command. This command takes --tx-bytes as the raw transaction bytes to execute (see output of the previous sui client transfer command) and the serialized signature (Base64 encoded flag || sig || pk , see output of sui keytool sign). This executes the signed transaction and returns the certificate and transaction effects if successful.

You get the following response:

Alternatively, you can use the active key in Sui Keystore to sign and output a Base64-encoded sender signed data with flag --serialize-signed-transaction .

The console responds with the resulting value.

After you obtain the signed transaction bytes, you can submit it using the execute-combined-signed-tx command. This command takes --signed-tx-bytes as the signed transaction bytes to execute (see output of the previous sui client transfer-sui command). This executes the signed transaction and returns the certificate and transaction effects if successful.

Serialize data for a transfer

You must serialize transaction data following Binary Canonical Serialization (BCS). It is supported in other languages.

The following example demonstrates how to serialize data for a transfer using the Sui CLI . This returns serialized transaction data in Base64. Submit the raw transaction to execute as tx_bytes .

Beginning with the Sui v1.24.1 release , the --gas-budget option is no longer required for CLI commands.

The console responds with the resulting value.

All other CLI commands that craft a transaction (such as sui client publish and sui client call) also accept the --serialize-unsigned-transaction flag used in the same way.

You can sign the data using the device and programming language you choose. Sui accepts signatures for pure Ed25519, ECDSA secp256k1, ECDSA secp256r1 and native multisig. To learn more about the requirements of the signatures, see Sui Signatures .

This example uses the sui keytool command to sign, using the Ed25519 key corresponding to the provided address stored in sui.keystore . This command outputs the signature, the public key, and the flag encoded in Base64. This command is backed by fastcrypto. sui keytool sign --address --data

You receive the following response:

To ensure the signature produced offline matches with Sui validity rules for testing purposes, you can import the mnemonics to sui.keystore using sui keytool import . You can then sign with it using sui keytool sign and then compare the signature results. Additionally, you can find test vectors in ~/sui/sdk/typescript/test/e2e/raw-signer.test.ts .

To verify a signature against the cryptography library backing Sui when debugging, see sigs-cli .

After you obtain the serialized signature, you can submit it using the execution transaction command. This command takes --tx-bytes as the raw transaction bytes to execute (see output of the previous sui client transfer command) and the serialized signature (Base64 encoded flag || sig || pk , see output of sui keytool sign). This executes the signed transaction and returns the certificate and transaction effects if successful.

You get the following response:

Alternatively, you can use the active key in Sui Keystore to sign and output a Base64-encoded sender signed data with flag --serialize-signed-transaction .

The console responds with the resulting value.

After you obtain the signed transaction bytes, you can submit it using the execute-combined-signed-tx command. This command takes --signed-tx-bytes as the signed transaction bytes to execute (see output of the previous sui client transfer-sui command). This executes the signed transaction and returns the certificate and transaction effects if successful.

Sign the serialized data

You can sign the data using the device and programming language you choose. Sui accepts signatures for pure Ed25519, ECDSA secp256k1, ECDSA secp256r1 and native multisig. To learn more about the requirements of the signatures, see Sui Signatures .

This example uses the sui keytool command to sign, using the Ed25519 key corresponding to the provided address stored in sui.keystore . This command outputs the signature, the public key, and the flag encoded in Base64. This command is backed by fastcrypto. sui keytool sign --address --data

You receive the following response:

To ensure the signature produced offline matches with Sui validity rules for testing purposes, you can import the mnemonics to sui.keystore using sui keytool import . You can then sign with it using sui keytool sign and then compare the signature results. Additionally, you can find test vectors in ~/sui/sdk/typescript/test/e2e/raw-signer.test.ts .

To verify a signature against the cryptography library backing Sui when debugging, see sigs-cli .

After you obtain the serialized signature, you can submit it using the execution transaction command. This command takes --tx-bytes as the raw transaction bytes to execute (see output of the previous sui client transfer command) and the serialized signature (Base64 encoded flag || sig || pk , see output of sui keytool sign). This executes the signed transaction and returns the certificate and transaction effects if successful.

You get the following response:

Alternatively, you can use the active key in Sui Keystore to sign and output a Base64-encoded sender signed data with flag --serialize-signed-transaction .

The console responds with the resulting value.

After you obtain the signed transaction bytes, you can submit it using the execute-combined-signed-tx command. This command takes --signed-tx-bytes as the signed transaction bytes to execute (see output of the previous sui client transfer-sui command). This executes the signed transaction and returns the certificate and transaction effects if successful.

Execute the signed transaction

After you obtain the serialized signature, you can submit it using the execution transaction command. This command takes --tx-bytes as the raw transaction bytes to execute (see output of the previous sui client transfer command) and the serialized signature (Base64 encoded flag || sig || pk , see output of sui keytool sign). This executes the signed transaction and returns the certificate and transaction effects if successful.

You get the following response:

Alternatively, you can use the active key in Sui Keystore to sign and output a Base64-encoded sender signed data with flag --serialize-signed-transaction .

The console responds with the resulting value.

After you obtain the signed transaction bytes, you can submit it using the execute-combined-signed-tx command. This command takes --signed-tx-bytes as the signed transaction bytes to execute (see output of the previous sui client transfer-sui command). This executes the signed transaction and returns the certificate and transaction effects if successful.

Alternative: Sign with Sui Keystore and Execute Transaction

Alternatively, you can use the active key in Sui Keystore to sign and output a Base64-encoded sender signed data with flag --serialize-signed-transaction .

The console responds with the resulting value.

After you obtain the signed transaction bytes, you can submit it using the execute-combined-signed-tx command. This command takes --signed-tx-bytes as the signed transaction bytes to execute (see output of the previous sui client transfer-sui command). This executes the signed transaction and returns the certificate and transaction effects if successful.