Keys and Addresses

Sui adheres to widely accepted wallet specifications in the cryptocurrency industry, including BIP-32 and its variation SLIP-0010, BIP-44, and BIP-39, to facilitate key management for users. At present, Sui supports pure Ed25519, ECDSA Secp256k1, ECDSA Secp256r1, and multisig for signed transactions.

Follow the relevant link for more information on each wallet specification:

Sui follows SLIP-0010 for managing wallets that support the Ed25519 (EdDSA) signing scheme.

For managing wallets that support the Ed25519 (EdDSA) signing scheme, Sui follows SLIP-0010, which enforces wallets to always derive child private keys from parent private keys using the hardened key path.

Sui follows BIP-32 for managing wallets that support the ECDSA Secp256k1 and ECDSA Secp256r1 signing scheme.

BIP-32 defines the hierarchical deterministic wallet structure to logically associate a set of keys. Grouping keys in this manner reduces the overhead of keeping track of a large number of private keys for a user. This method also lets custodians issue distinct managed addresses for each user account under one source of control. Using BIP-32 decouples the private key derivation from the public key derivation, enabling the watch-only wallet use case, where a chain of public keys and its addresses can be derived, while the private key can be kept offline for signing.

BIP-44 further defines the five levels of the derivation path with their exact meanings: m / purpose' / coin_type' / account' / change / address_index . In this structure, the slashes indicate new levels, or children, in the hierarchy.

The purpose level is generally set to 44, corresponding to the BIP number. In Sui, however, the purpose level distinguishes different signing schemes: 44 is set for Ed25519, 54 for ECDSA Secp256k1 and 74 for Secp256r1. While it is non-standard to set the purpose level to a value that is not 44, it is common to use the purpose field to distinguish different signing schemes. BIP-49 and BIP-84, for example, are used to identify script types in Bitcoin. Sui chose 54 to indicate ECDSA Secp256k1 because there is no existing BIP under 54, avoiding confusion with any Bitcoin standard.

The coin_type value is managed with a repository of all other cryptocurrencies. Both signature schemes use Sui's registered coin_type, 784 (SUI on a telephone keypad).

The account level is usually used for logically separating user accounts and creating specific account categories.

It is generally accepted that, while account-based currencies define only the first three levels, UTXO-based currencies add change and address level definitions. Because Sui's object-oriented data model is neither UTXO nor account-based (it in fact combines both), it employs all five levels for maximum compatibility.

After Sui defines the deterministic way to derive the master key from a seed, BIP-39 is introduced to make the seed more human-readable and memorizable using mnemonics. Sui accepts 12, 15, 18, 21, and 24 words from the BIP-39 word list that is properly checksummed, corresponding to 128, 160, 192, 224, and 256 bits of entropy. Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

For deriving a 32-byte Sui address, Sui hashes the signature scheme flag 1-byte concatenated with public key bytes using the BLAKE2b (256 bits output) hashing function. Sui address currently supports pure Ed25519, Secp256k1, Secp256r1, and MultiSig with corresponding flag bytes of 0x00, 0x01, 0x02, and 0x03, respectively.

Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

See more test vectors for pure Ed25519 or ECDSA Secp256k1 .

Key derivation scheme

Sui follows SLIP-0010 for managing wallets that support the Ed25519 (EdDSA) signing scheme.

For managing wallets that support the Ed25519 (EdDSA) signing scheme, Sui follows SLIP-0010, which enforces wallets to always derive child private keys from parent private keys using the hardened key path.

Sui follows BIP-32 for managing wallets that support the ECDSA Secp256k1 and ECDSA Secp256r1 signing scheme.

BIP-32 defines the hierarchical deterministic wallet structure to logically associate a set of keys. Grouping keys in this manner reduces the overhead of keeping track of a large number of private keys for a user. This method also lets custodians issue distinct

managed addresses for each user account under one source of control. Using BIP-32 decouples the private key derivation from the public key derivation, enabling the watch-only wallet use case, where a chain of public keys and its addresses can be derived, while the private key can be kept offline for signing.

BIP-44 further defines the five levels of the derivation path with their exact meanings: m / purpose' / coin_type' / account' / change / address_index . In this structure, the slashes indicate new levels, or children, in the hierarchy.

The purpose level is generally set to 44, corresponding to the BIP number. In Sui, however, the purpose level distinguishes different signing schemes: 44 is set for Ed25519, 54 for ECDSA Secp256k1 and 74 for Secp256r1. While it is non-standard to set the purpose level to a value that is not 44, it is common to use the purpose field to distinguish different signing schemes. BIP-49 and BIP-84, for example, are used to identify script types in Bitcoin. Sui chose 54 to indicate ECDSA Secp256k1 because there is no existing BIP under 54, avoiding confusion with any Bitcoin standard.

The coin_type value is managed with a repository of all other cryptocurrencies. Both signature schemes use Sui's registered coin_type, 784 (SUI on a telephone keypad).

The account level is usually used for logically separating user accounts and creating specific account categories.

It is generally accepted that, while account-based currencies define only the first three levels, UTXO-based currencies add change and address level definitions. Because Sui's object-oriented data model is neither UTXO nor account-based (it in fact combines both), it employs all five levels for maximum compatibility.

After Sui defines the deterministic way to derive the master key from a seed, BIP-39 is introduced to make the seed more human-readable and memorizable using mnemonics. Sui accepts 12, 15, 18, 21, and 24 words from the BIP-39 word list that is properly checksummed, corresponding to 128, 160, 192, 224, and 256 bits of entropy. Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

For deriving a 32-byte Sui address, Sui hashes the signature scheme flag 1-byte concatenated with public key bytes using the BLAKE2b (256 bits output) hashing function. Sui address currently supports pure Ed25519, Secp256k1, Secp256r1, and MultiSig with corresponding flag bytes of 0x00, 0x01, 0x02, and 0x03, respectively.

Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

See more test vectors for pure Ed25519 or ECDSA Secp256k1 .

Key derivation path

BIP-44 further defines the five levels of the derivation path with their exact meanings: m / purpose' / coin_type' / account' / change / address_index . In this structure, the slashes indicate new levels, or children, in the hierarchy.

The purpose level is generally set to 44, corresponding to the BIP number. In Sui, however, the purpose level distinguishes different signing schemes: 44 is set for Ed25519, 54 for ECDSA Secp256k1 and 74 for Secp256r1. While it is non-standard to set the purpose level to a value that is not 44, it is common to use the purpose field to distinguish different signing schemes. BIP-49 and BIP-84, for example, are used to identify script types in Bitcoin. Sui chose 54 to indicate ECDSA Secp256k1 because there is no existing BIP under 54, avoiding confusion with any Bitcoin standard.

The coin_type value is managed with a repository of all other cryptocurrencies. Both signature schemes use Sui's registered coin_type, 784 (SUI on a telephone keypad).

The account level is usually used for logically separating user accounts and creating specific account categories.

It is generally accepted that, while account-based currencies define only the first three levels, UTXO-based currencies add change and address level definitions. Because Sui's object-oriented data model is neither UTXO nor account-based (it in fact combines both), it employs all five levels for maximum compatibility.

After Sui defines the deterministic way to derive the master key from a seed, BIP-39 is introduced to make the seed more human-readable and memorizable using mnemonics. Sui accepts 12, 15, 18, 21, and 24 words from the BIP-39 word list that is properly checksummed, corresponding to 128, 160, 192, 224, and 256 bits of entropy. Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

For deriving a 32-byte Sui address, Sui hashes the signature scheme flag 1-byte concatenated with public key bytes using the BLAKE2b (256 bits output) hashing function. Sui address currently supports pure Ed25519, Secp256k1, Secp256r1, and MultiSig with corresponding flag bytes of 0x00, 0x01, 0x02, and 0x03, respectively.

Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

See more test vectors for pure Ed25519 or ECDSA Secp256k1 .

Mnemonics support

After Sui defines the deterministic way to derive the master key from a seed, BIP-39 is introduced to make the seed more human-readable and memorizable using mnemonics. Sui accepts 12, 15, 18, 21, and 24 words from the BIP-39 word list that is properly checksummed, corresponding to 128, 160, 192, 224, and 256 bits of entropy. Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

For deriving a 32-byte Sui address, Sui hashes the signature scheme flag 1-byte concatenated with public key bytes using the BLAKE2b (256 bits output) hashing function. Sui address currently supports pure Ed25519, Secp256k1, Secp256r1, and MultiSig with corresponding flag bytes of 0x00, 0x01, 0x02, and 0x03, respectively.

Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

See more test vectors for pure Ed25519 or ECDSA Secp256k1 .

Address format

For deriving a 32-byte Sui address, Sui hashes the signature scheme flag 1-byte concatenated with public key bytes using the BLAKE2b (256 bits output) hashing function. Sui address currently supports pure Ed25519, Secp256k1, Secp256r1, and MultiSig with corresponding flag bytes of 0x00, 0x01, 0x02, and 0x03, respectively.

Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

See more test vectors for pure Ed25519 or ECDSA Secp256k1 .

Example

Sui Wallet and SDKs provide a flexible interface to sign transactions with various signing schemes.

See more test vectors for pure Ed25519 or ECDSA Secp256k1 .