Hashing

A cryptographic hash function is a widely used cryptographic primitive that maps an arbitrary length input to a fixed length output, the hash value. The hash function is designed to be a one-way function, which means that it is infeasible to invert the function to find the input data from a given hash value, and to be collision resistant, which means that it is infeasible to find two different inputs that map to the same hash value.

The Sui Move API supports the following cryptographic hash functions:

The SHA2-256 and SHA3-256 hash functions are available in the Move Standard Library in the std::hash module. The following example shows how to use the SHA2-256 hash function in a smart contract:

The Keccak256 and Blake2b-256 hash functions are available through the sui::hash module in the Sui Move Library. An example of how to use the Keccak256 hash function in a smart contract is shown below. Notice that here, the input to the hash function is given as a reference. This is the case for both Keccak256 and Blake2b-256.

Usage

The SHA2-256 and SHA3-256 hash functions are available in the Move Standard Library in the std::hash module. The following example shows how to use the SHA2-256 hash function in a smart contract:

The Keccak256 and Blake2b-256 hash functions are available through the sui::hash module in the Sui Move Library. An example of how to use the Keccak256 hash function in a smart contract is shown below. Notice that here, the input to the hash function is given as a reference. This is the case for both Keccak256 and Blake2b-256.