Getting Started

A core component of the Fhenix ecosystem is theFHE.sol solidity library.

FHE.sol is a Solidity library designed to facilitate the use of fully homomorphic encryption within Ethereum smart contracts. FHE enables computations to be performed on encrypted data (ciphertexts) without needing to decrypt them first. The results of such computations, when decrypted, are identical to what would have been obtained if the operations had been performed on the unencrypted data (plaintexts).

To find a full list of functions and their descriptions, please refer to the FHE.sol documentation .

Installation

To get started withFHE.sol , you need to install it as a dependency in your Solidity project. You can do this usingnpm ,yarn or our personal favorite -pnpm . Open your terminal and navigate to your project's directory, then run the following:

- npm
- yarn
- pnpm

npm install @fhenixprotocol/contracts; yarn install @fhenixprotocol/contracts; pnpm install @fhenixprotocol/contracts;

Usage

Key Concepts and Types

euint

- Encrypted Unsigned Integer
- Description
- : Represents an encrypted unsigned integer. This type is used for encrypted variables within smart contracts.
- The currently supported types are:ebool
- ,euint8
- ,euint16
- &euint32
- .
- Usage
- : Store and manipulate encrypted values within smart contracts.

inEuint

- Input Encrypted Unsigned Integer
- Description
- : A type used for passing encrypted values as function arguments. It's the format in which encrypted data is input into the smart contract functions that process encrypted values.
- The currently supported types areinEuint8
- ,inEuint16
- &inEuint32
- .
- Usage
- : Pass typed encrypted values as function arguments.

Core Functions of FHE.sol

asEuint

- Convert to Encrypted Unsigned Integer
- Purpose
- : Converts a plaintext number or aninEuint
- encrypted input into aneuint
- type.

decrypt

- Decrypt Encrypted Data
- Purpose
- : Decrypts aneuint
- encrypted value back to its plaintext form. If the value should only be revealed to a specific address, thesealoutput
- function should be used instead. Learn more abut sealinghere
- .

Arithmetic Operations

FHE.sol supports encrypted arithmetic operations like addition and subtraction. These operations can be performed directly oneuint types, enabling encrypted computations.

Comparison Operations

- Purpose
- : Perform comparisons between encrypted values (e.g., greater than, less than).
- Usage Example
- : Make decisions based on encrypted values without revealing their contents.

Example Use Cases

Encrypting a Value

To encrypt a value, convert a plaintextuint32 into aneuint32 :

uint32 plaintextValue = 123; euint32 encryptedValue = FHE.asEuint32(plaintextValue);

Decrypting a Value

To decrypt an encrypted value back to plaintext:

uint32 decryptedValue = FHE.decrypt(encryptedValue); warning Decryption of data should be done with caution. Decrypted data should be handled with care and should not be exposed to unauthorized parties.

Performing Encrypted Arithmetic

You can perform arithmetic operations directly on encrypted values. For example, adding two encrypted values:

euint32 sum = encryptedValue1 + encryptedValue2;

Conditional Logic with Encrypted Values

Use comparison operations to implement logic based on encrypted values:

euint32 result = FHE.select(encryptedValue1.gt(encryptedValue2), encryptedValue1, encryptedValue2); This example chooses betweenencryptedValue1 andencryptedValue2 based on their encrypted comparison.

Integrating FHE into Smart Contracts

When incorporatingFHE.sol into your smart contracts, consider the following:

- Privacy vs. Gas Cost
- : While FHE provides strong privacy guarantees, it's computationally intensive and can lead to higher gas costs. Balance the need for privacy with the cost implications.
- Data Types
- : Ensure that your use cases are compatible with the data types and operations supported byFHE.sol
- .
- Security
- : Understand the security model of FHE, including its limitations and how it fits into the overall security posture of your application. Edit this page

Previous FHE Contracts Next Inputs