# FHE.sol

### NIL8

euint8 NIL8

### NIL16

euint16 NIL16

### NIL32

euint32 NIL32

### isInitialized

function isInitialized(ebool v) internal pure returns (bool)

### isInitialized

function isInitialized(euint8 v) internal pure returns (bool)

### isInitialized

function isInitialized(euint16 v) internal pure returns (bool)

### isInitialized

function isInitialized(euint32 v) internal pure returns (bool)

### mathHelper

function mathHelper(uint8 utype, uint256 lhs, uint256 rhs, function (uint8,bytes,bytes) pure external returns (bytes) impl) internal pure returns (uint256 result)

### add

function add(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the add operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

#### Parameters

Name Type Description lhs euint8 The first input rhs euint8 The second input

#### Return Values

Name Type Description [0] euint8 The result of the operation

### add

function add(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the add operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

#### Parameters

Name Type Description lhs euint16 The first input rhs euint16 The second input

#### Return Values

Name Type Description [0] euint16 The result of the operation

### add

function add(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the add operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## sealoutput

function sealoutput(ebool value, bytes32 publicKey) internal pure returns (bytes) performs the sealoutput function on a ebool ciphertext. This operation returns the plaintext value, sealed for the public key provided

Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description value ebool Ciphertext to decrypt and seal publicKey bytes32 Public Key that will receive the sealed plaintext

**Return Values**

Name Type Description [0] bytes Plaintext input, sealed for the owner ofpublicKey

## sealoutput

function sealoutput(euint8 value, bytes32 publicKey) internal pure returns (bytes) performs the sealoutput function on a euint8 ciphertext. This operation returns the plaintext value, sealed for the public key provided

Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description value euint8 Ciphertext to decrypt and seal publicKey bytes32 Public Key that will receive the sealed plaintext

**Return Values**

Name Type Description [0] bytes Plaintext input, sealed for the owner ofpublicKey

## sealoutput

function sealoutput(euint16 value, bytes32 publicKey) internal pure returns (bytes) performs the sealoutput function on a euint16 ciphertext. This operation returns the plaintext value, sealed for the public key provided

Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description value euint16 Ciphertext to decrypt and seal publicKey bytes32 Public Key that will receive the sealed plaintext

**Return Values**

Name Type Description [0] bytes Plaintext input, sealed for the owner ofpublicKey

## sealoutput

function sealoutput(euint32 value, bytes32 publicKey) internal pure returns (bytes) performs the sealoutput function on a euint32 ciphertext. This operation returns the plaintext value, sealed for the public key provided

Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description value euint32 Ciphertext to decrypt and seal publicKey bytes32 Public Key that will receive the sealed plaintext

**Return Values**

Name Type Description [0] bytes Plaintext input, sealed for the owner ofpublicKey

## decrypt

function decrypt(ebool input1) internal pure returns (bool) Performs the decrypt operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 ebool the input ciphertext

## decrypt

function decrypt(euint8 input1) internal pure returns (uint8) Performs the decrypt operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint8 the input ciphertext

## decrypt

function decrypt(euint16 input1) internal pure returns (uint16) Performs the decrypt operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint16 the input ciphertext

## decrypt

function decrypt(euint32 input1) internal pure returns (uint32) Performs the decrypt operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint32 the input ciphertext

## lte

function lte(euint8 lhs, euint8 rhs) internal pure returns (ebool) This functions performs the lte operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## lte

function lte(euint16 lhs, euint16 rhs) internal pure returns (ebool) This functions performs the lte operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## lte

function lte(euint32 lhs, euint32 rhs) internal pure returns (ebool) This functions performs the lte operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## sub

function sub(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the sub operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## sub

function sub(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the sub operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## sub

function sub(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the sub operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## mul

function mul(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the mul operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## mul

function mul(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the mul operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## mul

function mul(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the mul operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## lt

function lt(euint8 lhs, euint8 rhs) internal pure returns (ebool) This functions performs the lt operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## lt

function lt(euint16 lhs, euint16 rhs) internal pure returns (ebool) This functions performs the lt operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## lt

function lt(euint32 lhs, euint32 rhs) internal pure returns (ebool) This functions performs the lt operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## select

function select(ebool input1, ebool input2, ebool input3) internal pure returns (ebool)

## select

function select(ebool input1, euint8 input2, euint8 input3) internal pure returns (euint8)

## select

function select(ebool input1, euint16 input2, euint16 input3) internal pure returns (euint16)

## select

function select(ebool input1, euint32 input2, euint32 input3) internal pure returns (euint32)

## req

function req(ebool input1) internal pure Performs the req operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 ebool the input ciphertext

## req

function req(euint8 input1) internal pure Performs the req operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint8 the input ciphertext

## req

function req(euint16 input1) internal pure Performs the req operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint16 the input ciphertext

## req

function req(euint32 input1) internal pure Performs the req operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint32 the input ciphertext

## div

function div(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the div operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## div

function div(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the div operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## div

function div(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the div operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## gt

function gt(euint8 lhs, euint8 rhs) internal pure returns (ebool) This functions performs the gt operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## gt

function gt(euint16 lhs, euint16 rhs) internal pure returns (ebool) This functions performs the gt operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## gt

function gt(euint32 lhs, euint32 rhs) internal pure returns (ebool) This functions performs the gt operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## gte

function gte(euint8 lhs, euint8 rhs) internal pure returns (ebool) This functions performs the gte operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## gte

function gte(euint16 lhs, euint16 rhs) internal pure returns (ebool) This functions performs the gte operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## gte

function gte(euint32 lhs, euint32 rhs) internal pure returns (ebool) This functions performs the gte operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## rem

function rem(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the rem operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## rem

function rem(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the rem operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## rem

function rem(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the rem operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## and

function and(ebool lhs, ebool rhs) internal pure returns (ebool) This functions performs the and operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs ebool The first input rhs ebool The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## and

function and(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the and operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## and

function and(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the and operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## and

function and(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the and operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## or

function or(ebool lhs, ebool rhs) internal pure returns (ebool) This functions performs the or operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs ebool The first input rhs ebool The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

### or

function or(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the or operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

### or

function or(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the or operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

### or

function or(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the or operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

### xor

function xor(ebool lhs, ebool rhs) internal pure returns (ebool) This functions performs the xor operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs ebool The first input rhs ebool The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

### xor

function xor(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the xor operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

### xor

function xor(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the xor operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

### xor

function xor(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the xor operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

### eq

function eq(ebool lhs, ebool rhs) internal pure returns (ebool) This functions performs the eq operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs ebool The first input rhs ebool The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

### eq

function eq(euint8 lhs, euint8 rhs) internal pure returns (ebool) This functions performs the eq operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## eq

function eq(euint16 lhs, euint16 rhs) internal pure returns (ebool) This functions performs the eq operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## eq

function eq(euint32 lhs, euint32 rhs) internal pure returns (ebool) This functions performs the eq operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## ne

function ne(ebool lhs, ebool rhs) internal pure returns (ebool) This functions performs the ne operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs ebool The first input rhs ebool The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## ne

function ne(euint8 lhs, euint8 rhs) internal pure returns (ebool) This functions performs the ne operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## ne

function ne(euint16 lhs, euint16 rhs) internal pure returns (ebool) This functions performs the ne operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## ne

function ne(euint32 lhs, euint32 rhs) internal pure returns (ebool) This functions performs the ne operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] ebool The result of the operation

## min

function min(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the min operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## min

function min(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the min operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## min

function min(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the min operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## max

function max(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the max operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## max

function max(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the max operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## max

function max(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the max operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## shl

function shl(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the shl operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## shl

function shl(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the shl operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## shl

function shl(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the shl operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## shr

function shr(euint8 lhs, euint8 rhs) internal pure returns (euint8) This functions performs the shr operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint8 The first input rhs euint8 The second input

**Return Values**

Name Type Description [0] euint8 The result of the operation

## shr

function shr(euint16 lhs, euint16 rhs) internal pure returns (euint16) This functions performs the shr operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint16 The first input rhs euint16 The second input

**Return Values**

Name Type Description [0] euint16 The result of the operation

## shr

function shr(euint32 lhs, euint32 rhs) internal pure returns (euint32) This functions performs the shr operation

If any of the inputs are expected to be a ciphertext, it verifies that the value matches a valid ciphertext Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description lhs euint32 The first input rhs euint32 The second input

**Return Values**

Name Type Description [0] euint32 The result of the operation

## not

function not(ebool value) internal pure returns (ebool) Performs the "not" for the ebool type

Implemented by a workaround due to ebool being a euint8 type behind the scenes, therefore xor is needed to assure that not(true) = false and vise-versa

**Parameters**

Name Type Description value ebool input ebool ciphertext

**Return Values**

Name Type Description [0] ebool Result of the not operation onvalue

## not

function not(euint8 input1) internal pure returns (euint8) Performs the not operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint8 the input ciphertext

## not

function not(euint16 input1) internal pure returns (euint16) Performs the not operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint16 the input ciphertext

## not

function not(euint32 input1) internal pure returns (euint32) Performs the not operation on a ciphertext

Verifies that the input value matches a valid ciphertext. Pure in this function is marked as a hack/workaround - note that this function is NOT pure as fetches of ciphertexts require state access

**Parameters**

Name Type Description input1 euint32 the input ciphertext

## asEbool

function asEbool(struct inEbool value) internal pure returns (ebool) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an ebool

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

**Return Values**

Name Type Description [0] ebool a ciphertext representation of the input

### asEuint8

function asEuint8(ebool value) internal pure returns (euint8) Converts a ebool to an euint8

### asEuint16

function asEuint16(ebool value) internal pure returns (euint16) Converts a ebool to an euint16

### asEuint32

function asEuint32(ebool value) internal pure returns (euint32) Converts a ebool to an euint32

### asEbool

function asEbool(euint8 value) internal pure returns (ebool) Converts a euint8 to an ebool

### asEuint8

function asEuint8(struct inEuint8 value) internal pure returns (euint8) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an euint8

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

**Return Values**

Name Type Description [0] euint8 a ciphertext representation of the input

### asEuint16

function asEuint16(euint8 value) internal pure returns (euint16) Converts a euint8 to an euint16

### asEuint32

function asEuint32(euint8 value) internal pure returns (euint32) Converts a euint8 to an euint32

### asEbool

function asEbool(euint16 value) internal pure returns (ebool) Converts a euint16 to an ebool

### asEuint8

function asEuint8(euint16 value) internal pure returns (euint8) Converts a euint16 to an euint8

### asEuint16

function asEuint16(struct inEuint16 value) internal pure returns (euint16) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an euint16

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

**Return Values**

Name Type Description [0] euint16 a ciphertext representation of the input

### asEuint32

function asEuint32(euint16 value) internal pure returns (euint32) Converts a euint16 to an euint32

### asEbool

function asEbool(euint32 value) internal pure returns (ebool) Converts a euint32 to an ebool

### asEuint8

function asEuint8(euint32 value) internal pure returns (euint8) Converts a euint32 to an euint8

## asEuint16

function asEuint16(euint32 value) internal pure returns (euint16) Converts a euint32 to an euint16

## asEuint32

function asEuint32(struct inEuint32 value) internal pure returns (euint32) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an euint32

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

#### Return Values

Name Type Description [0] euint32 a ciphertext representation of the input

## asEbool

function asEbool(uint256 value) internal pure returns (ebool) Converts a uint256 to an ebool

## asEuint8

function asEuint8(uint256 value) internal pure returns (euint8) Converts a uint256 to an euint8

## asEuint16

function asEuint16(uint256 value) internal pure returns (euint16) Converts a uint256 to an euint16

## asEuint32

function asEuint32(uint256 value) internal pure returns (euint32) Converts a uint256 to an euint32

## asEbool

function asEbool(bytes value) internal pure returns (ebool) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an ebool

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

#### Return Values

Name Type Description [0] ebool a ciphertext representation of the input

## asEuint8

function asEuint8(bytes value) internal pure returns (euint8) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an euint8

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

#### Return Values

Name Type Description [0] euint8 a ciphertext representation of the input

## asEuint16

function asEuint16(bytes value) internal pure returns (euint16) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an euint16

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

#### Return Values

Name Type Description [0] euint16 a ciphertext representation of the input

## asEuint32

function asEuint32(bytes value) internal pure returns (euint32) Parses input ciphertexts from the user. Converts from encrypted raw bytes to an euint32

Also performs validation that the ciphertext is valid and has been encrypted using the network encryption key

**Return Values**

Name Type Description [0] euint32 a ciphertext representation of the input

## asEbool

function asEbool(bool value) internal pure returns (ebool) Converts a plaintext boolean value to a ciphertext ebool

Privacy: The input value is public, therefore the ciphertext should be considered public and should be used only for mathematical operations, not to represent data that should be private

**Return Values**

Name Type Description [0] ebool A ciphertext representation of the inputEdit this page

Previous eAddress Next Permissioned.Sol