EIP712StETH

- Source code
- Deployed contract

EIP712StETH serves as a dedicated helper contract forstETH , crucial for complete support ofERC-2612 compliant signed approvals .

Why This Helper Is Needed

The originalLido/StETH contract is implemented in Solidity0.4.24 , while this helper is implemented in Solidity0.8.9 . The newer compiler version enables access to the current network's chain id via the globally available variableblock.chainid . The chain id is mandatory for signature inclusion as perEIP-155 to prevent replay attacks, wherein an attacker intercepts a valid network transmission and then rebroadcasts it on another network fork. Consequently,EIP-155 compliance is critical for securingERC-2612 signed approvals.

View Methods

domainSeparatorV4()

This method returns theEIP712 -compatible hasheddomain separator , which is valid forstETH token permit signatures. The domain separator is essential in preventing a signature intended for one dApp from functioning in another (thereby averting a signature collision in a broader sense).

function

domainSeparatorV4 (address _stETH)

returns

(bytes32) Also, consider theeip712Domain() method that can construct a domain separator fromStETH -specific fields on the client's side, such as within a dApp or a wallet. For instance, Metamask relies oneth_signTypedData_v4 , which requires a non-hashed domain separator being provided.

hashTypedDataV4()

This method returns the hash of a fully encodedEIP712 -compatible message for this domain. The method can validate the input data against the providedv, r, s secp256k1 components.

function

hashTypedDataV4 (address _stETH ,

bytes32 _structHash)

returns

(bytes32)

Parameters

Name Type Description _stETH address Address of the deployedstETH token _structHash bytes32 Hash of the data structure For a specific use case, see theStETHPermit.permit() implementation.

eip712Domain()

This method returns the fields and values necessary to construct a domain separator on the client's side. The method resembles the one proposed inERC-5267 , with the only difference being that it doesn't return unused fields.

function

eip712Domain (address _stETH)

returns

(string

memory name , string

memory version , uint256 chainId , address verifyingContract)

Parameters

Name Type Description _stETH address Address of the deployedstETH token

Returns

Name Type Description name string Name of the token version string Version of the token chainId uint256 Chain identifier verifyingContract address Address of the token contract note Provided the correct_stETH deployed address, it returns:

- ("Liquid staked Ether 2.0", "2", 1, 0xae7ab96520DE3A18E5e111B5EaAb095312D7fE84) for Mainnet.
- ("Liquid staked Ether 2.0", "2", 5, 0x1643E812aE58766192Cf7D2Cf9567dF2C37e9B7F) for Görli. This method facilitates domain separator construction on the client's side, such as in a wallet or widget:

function

makeDomainSeparator (name , version , chainId , verifyingContract)

{ return web3 . utils . keccak256 (web3 . eth . abi . encodeParameters (['bytes32' ,

'bytes32' ,

'bytes32' ,

'uint256' ,

'address'] , [web3 . utils . keccak256 ('EIP712Domain(string name,string version,uint256 chainId,address verifyingContract)') , web3 . utils . keccak256 (name) , web3 . utils . keccak256 (version) , chainId , verifyingContract ,])) }

Useful External Links

- The Magic of Digital Signatures on Ethereum
- ERC-2612: The Ultimate Guide to Gasless ERC-20 Approvals
- Metamask sign-data