Overview

In this guide, we'll be creating a shielded ERC20 token using Solidity. Our token will be unique in that it will offer encrypted token balances, thereby enhancing privacy for token holders.

We'll be making use of the FHE library and Fhenix Helium Testnet to enable this functionality - it allows us to perform computations on encrypted data without first having to decrypt it, which is vital for preserving privacy.

You can find all the completed code in ourexample project repository . You can just skip there if you just want to see the final code.

What We'll Be Buildingâ

We'll be building a contract for a new token thatextends the standard ERC20 token functionality. Our contract will introduce an additional layer of privacy by encrypting token balances. This means that even though transactions are public on the blockchain, it will be impossible to know the balance of a user's account without having the corresponding decryption key.

Our token will offer the ability to 'wrap' and 'unwrap' tokens, where wrapping refers to the conversion of regular tokens into their encrypted form and unwrapping refers to the conversion back into regular tokens.

In addition, our contract will also support the transfer of encrypted tokens from one account to another. The balance of encrypted tokens can also be queried by the token holder, keeping their balance private from others on the network.

Why Is This Useful?â

Traditional ERC20 tokens operate transparently, meaning that balances and transactions are publicly visible on the blockchain. This transparency can lead to issues around privacy. For example, once an address is linked to an individual, anyone can view their token balance and see all incoming and outgoing transactions.

By using encryption, we can offer the same functionality while greatly enhancing user privacy. Encrypted balances ensure that no one can determine a user's token balance without the appropriate decryption key. This type of token can be beneficial for users who want the benefits of transacting on the blockchain but with an additional layer of privacy.

This could be particularly useful in a range of applications, from privacy-preserving DeFi applications to personal tokens where the individual does not want their total supply public.

The shielded ERC20 token we will build offers the right balance between transparency, needed for the operation of the blockchain, and privacy, providing individuals the discretion they need over their own finances.

By just extending (and not replacing) the basic functionality of the ERC20 standard we can also maintain compatibility with applications that support the ERC20 token, such as wallets and DeFi applications. Edit this page

Previous Shielded ERC20 Token Next Setting Up