Privacy Pitfalls

Similar to any blockchain-based implementation, while smart contracts on Fhenix provide a secure and decentralized way of executing transactions on encrypted data, there exist potential security risks that developers should be mindful of. Although the computations themselves are encrypted, offering privacy, metadata can still inadvertently disclose confidential information.

For all blockchain privacy platforms, there exist a different set of potential issues that developers need to be mindful of. The specific scenarios depend on the implementation details of the chain, and the tools that the infrastructure provides to mitigate such issues. Fhenix is at a very early stage and constantly changing, so instead we'll keep this section light and give a simple example of metadata leakage - gas usage.

For example, suppose a smart contract written in Solidity contains a conditional statement. In such cases, the path taken by the conditional, though encrypted, can leak information. A typical scenario is a conditional branch based on the value of a private variable, where gas usage, events, or other metadata could reveal the branch taken.

function

performActionBasedOnBalance (uint256 amount)

public

{ if

(balance [msg . sender]

= amount)

{ // perform some operation }

else

{ // perform another operation } } In the above Solidity example, an observer could infer from the gas usage, emitted events, or other metadata, the branch that was executed, indirectly revealing whether the sender's balance was greater than or equal to the specified amount.

While this example seems simple, it is important to remember that often transactions can be cheaply simulated with different input parameters. In the above example, performing logarithmic search would reveal the exact balance fairly quickly.

Also, it's essential to mention the importance of adding access control to functions that handle sensitive data. For instance, a function that displays the balance of a user should only be accessible by the specific user. We expand on this in our sections about access control.

Developers on Fhenix (and any chain that provides privacy or encryption) should analyse their contract's privacy model. Recognizing the type of information that should remain confidential and understanding what information, if leaked, won't affect the operation of the contract and its users, is critical. This helps in structuring your smart contracts in a way that safeguards the private aspects, while still operating efficiently. Despite the encryption provided by FHE, it is essential to understand and address these potential issues. Edit this page

Previous Permits Next FHERC20