Introduction

This section presents key management suggestions for operators. Within the context of EigenLayer, operators ought to be acquainted with proper practices concerning key loading, particularly signing keys.

Keys

Central to every Proof of Stake mechanism lies a signature scheme. Signatures serve to authenticate the identity of every validator, enabling the attribution of their actions, whether positive or negative, to them. Focusing on Ethereum validator keys exemplifies keys necessitating optimal security and accessibility measures, as they may maintain a robust connection with Nodes, further underlining their importance.

A validator's honesty can be confirmed by examining their signed messages, while malicious behavior can be demonstrated through messages that contravene consensus rules.

Indeed, in Ethereum, a validator's identity is synonymous with their public key. To be precise, each validator possesses two sets: a signing and withdrawal keys.

Signing keys

Asigning key is the key a validator needs to sign attestations and propose blocks. Because a validator needs to sign a message at least once per epoch, the client software must have custody of the key.

Withdrawal keys

Because the client software is always connected to the internet, there is a chance that one’s signing key is compromised. To reduce the impact of such a breach, the actions a validator can perform are split between two keys.

The signing key, as explained above, is used for the validator to perform their duties. On the other hand, thewithdrawal key has the power to control a validator's funds (transferring and withdrawing ETH).

A validator should only need to use their withdrawal keys a few times over the lifetime of being a validator. This means they can be put into cold storage and stored with high security (offline). Previous Troubleshooting Next Key Management Best Practices for Node Operators