Configure cryptographic elliptic curves

By default, the Tesseraenclave uses thejnacl implementation of theNaCl library to encrypt and decrypt private payloads.

TheNaCl primitives provide good security and speed and this is sufficient in most circumstances.

You can configure alternative curves and symmetric ciphers by specifyingencryptor in the Tesseraconfiguration file .

Configure an alternative cryptographic elliptic curve

In theencryptor configuration item, you can provide a compatible JCA provider (for example,SunEC provider).

note The same enclave encryption process is used regardless of whether theNaCl or JCA encryptor is configured. JCA encryptor configuration "encryptor":{ "type":"EC", "properties":{ "symmetricCipher":"AES/GCM/NoPadding", "ellipticCurve":"secp256r1", "nonceLength":"24", "sharedKeyLength":"32" } } Iftype is set toCUSTOM , support is provided for an external encryptor implementation to integrate with Tessera. The kalium support module is configured as a custom encryptor. The pilot third party integration isUnbound Tech's Unbound Key Control (UKC) encryptor (jar available atcom.github.unbound-tech:encryption-ub:).

*[JCA]: Java Cryptography Architecture Edit this page Last updatedonOct 9, 2023 bydependabot[bot]Previous Multiple private states Next Hyperledger Besu support