@reuven realized an interesting potential attack vector, which is captured in this issue:

https://github.com/enigmampc/EnigmaBlockchain/issues/133.

The gist of it is - when reading a value from the state, one has to leave the enclave and fetch the encrypted data from disk (outside of the enclave). This includes executing untrusted code, which means a malicious host can tamper with the data in certain ways, including using the enclave as a decryption oracle.

The solution is to authenticate the data being encrypted + the key pointing to that memory location in the state.

authenticated encryption can provide security against chosen ciphertext attack. In these attacks, an adversary attempts to gain an advantage against a cryptosystem (e.g., information about the secret decryption key) by submitting carefully chosen ciphertexts to some “decryption oracle” and analyzing the decrypted results. Authenticated encryption schemes can recognize improperly-constructed ciphertexts and refuse to decrypt them. This, in turn, prevents the attacker from requesting the decryption of any ciphertext unless it was generated correctly using the encryption algorithm, thus implying that the plaintext is already known. Implemented correctly, authenticated encryption removes the usefulness of the decryption oracle, by preventing an attacker from gaining useful information that the attacker does not already possess.