I wish to discuss a kind of security situation to ensure 3rd parties can trust SGX.

1. Suppose we task a SGX enclave to run a program with a secret value known only to that enclave (of the same measurement). Usually this is not an issue if the same party develops the SGX enclave and also uses it: this party knows the expected measurement, and knows how the secret is generated:

Alice —(deploy and use) → enclave(secret)

For example, the secret could simply be randomly generated for the first time, then is sealed in a way only the same enclave can decrypt it, and is saved externally for later use. Since Alice knows the expected enclave measurement which is verified through attestation, she trusts that the secret value is safe.

1. What if 2nd party have to connect to and use the enclave, and must be assured that their data submitted to the enclave will not be leaked or the secret value is not known to Alice?

Alice ----deploys—> enclave(secret)

Bob ----connect to and use —> enclave(secret)

In this situation, Bob has to ensure: a) He connects the right version of enclave (with right measurement which is linked to a known version of source code), b) that nobody knows the secret.

What's in my mind is to have the source code of enclave open, audited, and have a certified version. The secret value has to be sealed to that enclave (rather than to enclave signer, as an upgraded enclave can easily leak the secret value). And since it is generated by the enclave, nobody knows its value.

I am wondering if there is a more "web3" way to do this, for example, saving the enclave measurement to Ethereum. Or, the secret value could be generated somehow MPC. Or letting 3rd parties to run the enclave, and those hosting enclave with incorrect measurements will be punished in some way. Any comments are welcome. Thanks in advance.