**5 Experimental Evaluation**

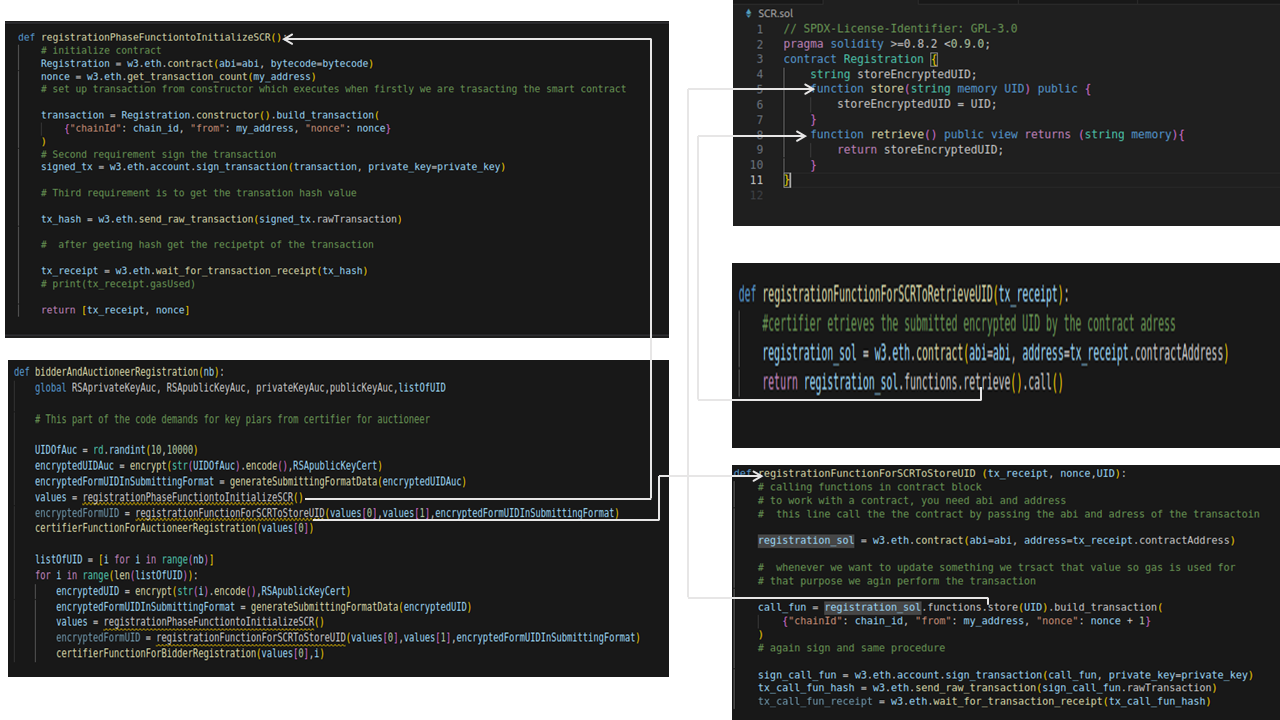
The experimental evaluation is categorized in two subgroups in first group we look at the different function with their input, output and descriptions and in other groups we look at the implementation of different algorithms with smart contracts. The implementation and testing of the proposed hybrid privacy preserving auction scheme has been performed over the intel i7-12700 12 gen processor with 32GB ram and with max clock speed of 4900MHz. The software suite that we have used is Ganache[reference] for simulating Ethereum blockchain on local node together with web3.py[reference] library to call and run smart contracts inside the python code. The off-chain computation is simulated in python programming language on local node.

**Registration phase on-chain and off-chain data handling**

On registration of any new bidder a new call to the smart contract is sent with the of the bidder encrypted with the certifier public key. This section shows the working of the code and its way of execution and handling of the smart contract, how the off-chain and on-chain data is computed and transferred.

From the function bidder and auctioneer submit the encrypted and this encryption is performed offchain in their own node, and after that initialize the contract by the function , later the tx\_recipet and nonce value is passed as the parameter to the and this function further calls the smart contract and stress the encrypted UID as a form of input to smart contract. One key thing t notice is that the encryption of on the node is generating the ciphertext of the size given as the key length of the encryption and this value can’t be stores directly to the byte variable for that purpose the preprocessing needs to be done of the ciphertext and is converted to the string format and that passed to smart contract otherwise the size limit for byte datatype is only of 32 bytes which is too far short than the length to the ciphertext as in our case it is 1024 bits(128 bytes).

Certifier in returns provides the public and private key value pairs and makes the public key value pairs of all the participants public on the blockchain.



Following figure represents the functions of the certifier to generate the public and private key value pairs for and.



**Bid submission phase off-chain and on-chain data handling**

In this section we look at the calling, computations and data transfer related to the smart contract .