1.

Players needed to pick an integer n to represent their choices of rock paper scissors within 2500 blocks approximately 12 hours after the contract deployed. The modulo 3 of n pointed to their choice; n mod 3 = 0 = rock; n mod 3 = 1 =paper; n mod 3 = 2=scissors.

After both players have sent their hashes to the contract, they could reveal by sending the integer they pick within 1 day. The contract will calculate `sha256(abi.encode(rps_choice)`to check if the hashes match their claimed hashes.

When the second player successfully revealed their choice, the contract would send 2 ETH to the winner or 1 ETH if they draw.

Both parties were unable to change their choices or verify the other parties' choice before both parties had made their choice. Players needed to calculate `sha256(abi.encode(rps_choice,nounce))` locally to get the byte32 format of their hashes and sent to the contracts, therefore, their choice would be protected. The only function to edit the hashes is the `SetChoiceHash(bytes32 my_rps_hash)`. There were several players counter to prevent any parties tampering with the hashes. The method of transferring money was safe as well since there was a counter to prevent a fallback function to double charge the money. So, neither party could cheat, tamper, or verify others' choices.

A day after the contract is deployed, if no player joins the match or one of the players refuses to reveal their choices, the other player can take all the contract balance by calling `function reveal_cheating()`.

2.1

After the auction was deployed. Bidder should join the auction with `sha256(abi.encode(bidding_price,uint nounce)` , to protect their bidding price. After 1 day, bidder could reveal their bidding prices. Bidder should pay the amount of their claimed bidding price to get back their guaranteed fee, while revealing their bidding prices. If the bidder won, the contract would hold their claimed bidding price and set their address as the winner. The previous winner of the bid would get back their money which was held by the contract.

2.2

To ensure the person with the highest bid reveals their choice, the owner of the contract could add a guaranteed fee for each bidding, each call of the `function join_Bidding(bytes32 bid_hash)` requires a guaranteed fee, which would be returned after revealing bidding price. If any bidder refuses to reveal and pay, the remaining guarantee fee could be sent to the contract owner address after two days the auction was deployed.

2.3

Join auction needs 53446 gas.

Revealing requires 172661 gas.

Each participant requires 226107 gas to join and reveal.