ECE 518 Grad Project 2

Alan Palayil Due Date: 12/07/2022

Section I Objective

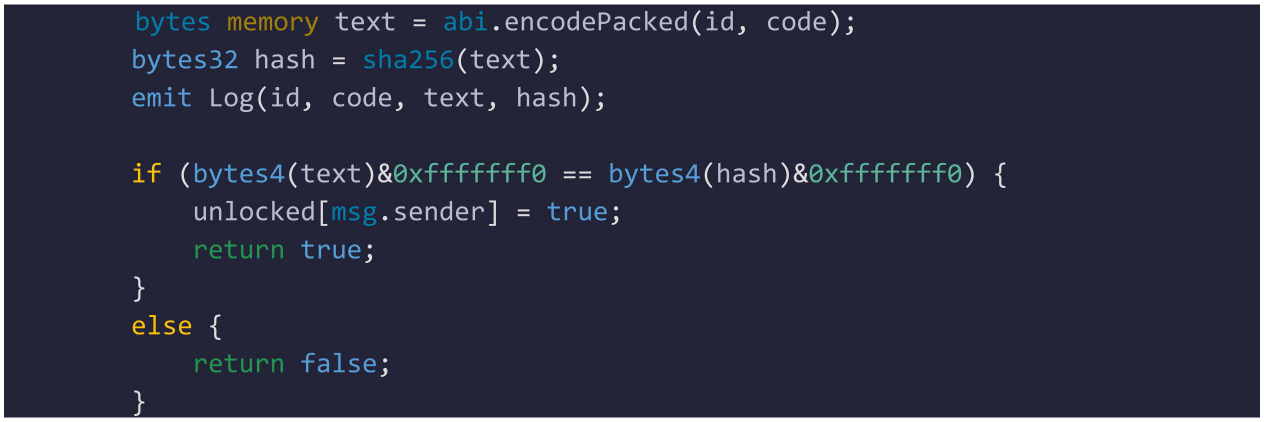
In this project, we have to open a virtual treasure box which is created as a smart contract on the Ethereum blockchain.

Section II The TreasureBox Smart Contract

The smart contract TreasureBox.sol contains two functions 'register' and 'unlock'. The register function registers our CWID with the treasure box and then the function unlock to open the box with a code. This project focuses on finding the code which is tied to our CWID. The smart contract allows to call the two functions multiple times and it will generate an error message if it doesn't follow the protocol. If the code work, it will unlock treasure box and you can't call the unlock function again as an error message will be generating showing you have already unlocked the box.

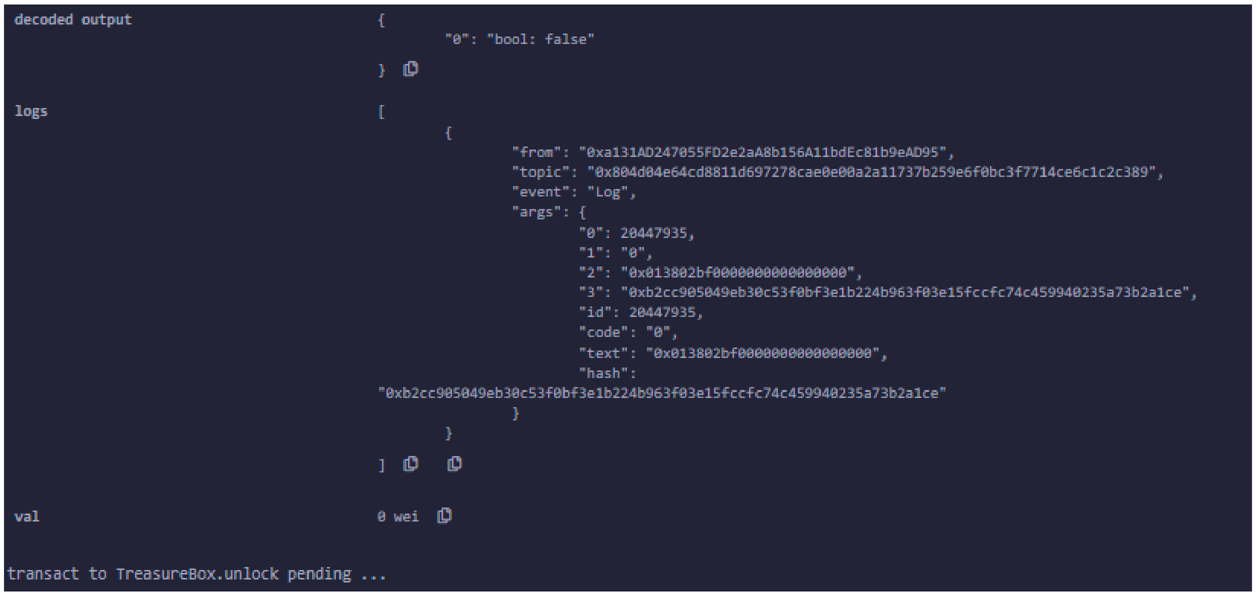
Section III Finding the Code for the CWID

We go over the code in TreasureBox.sol, and after examination the two main functions within this file are register and unlock. The register function is called when an ID (CWID number for the treasure chest) in input into (uint 32). The unlock function is what we focus on, with the code as input (uint 64) and returns a boolean which will define if the treasure box was unlocked. The following is the condition we need to fulfill in order to unlock the treasure.



```
        bytes memory text = abi.encodePacked(id, code);
        bytes32 hash = sha256(text);
        emit Log(id, code, text, hash);

        if (bytes4(text)&0xfffffff0 == bytes4(hash)&0xfffffff0) {
            unlocked[msg.sender] = true;
            return true;
        }
        else {
            return false;
        }
```

The abi.encodePacked(id,code) function seems to append the id and code into a hexadecimal string. This can be viewed if I input my CWID for register and 0 for unlock. The CWID is converted to hexadecimal number: 013802bf, appended with 0000000000000000, and is stored as a uint64 in hexadecimal. So, with this we found the value of text.

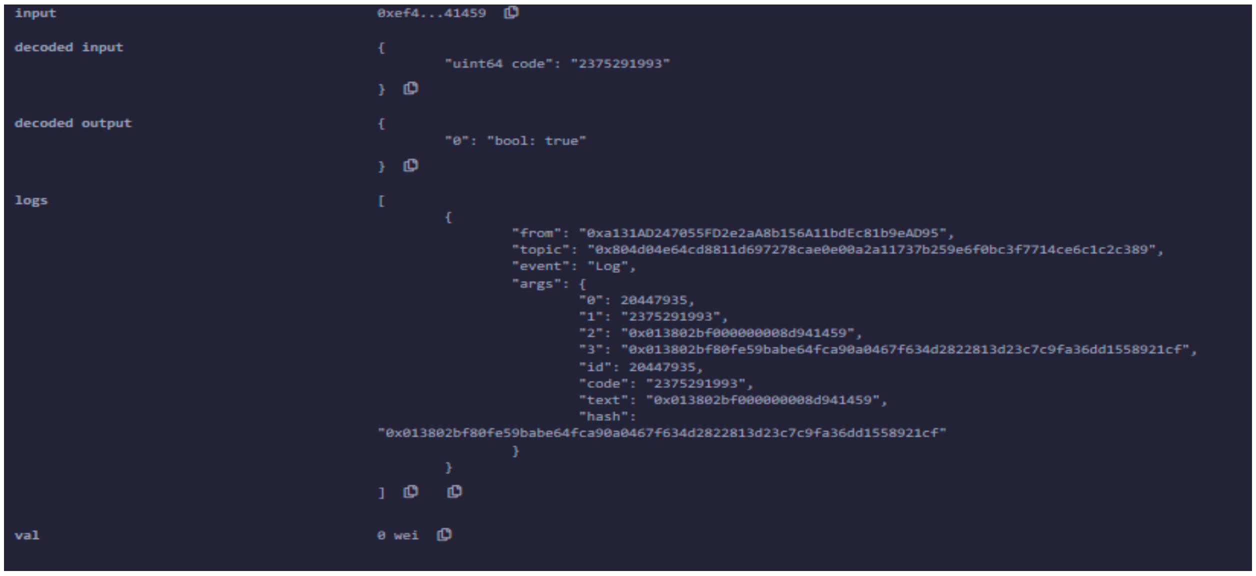


```
decoded output                  {
                                        "0": "bool: false"
                                }

logs                            [
                                        {
                                                "from": "0xa131AD247055FD2e2aA8b156A11bdEc81b9eAD95",
                                                "topic": "0x804d04e64cd8811d697278cae0e00a2a11737b259e6f0bc3f7714ce6c1c2c389",
                                                "event": "Log",
                                                "args": {
                                                        "0": 20447935,
                                                        "1": "0",
                                                        "2": "0x013802bf0000000000000000",
                                                        "3": "0xb2cc905049eb30c53f0bf3e1b224b963f03e15fccfc74c459940235a73b2a1ce",
                                                        "id": 20447935,
                                                        "code": "0",
                                                        "text": "0x013802bf0000000000000000",
                                                        "hash":
"0xb2cc905049eb30c53f0bf3e1b224b963f03e15fccfc74c459940235a73b2a1ce"
                                                }
                                        }
                                ]

val                             0 wei

transact to TreasureBox.unlock pending ...
```

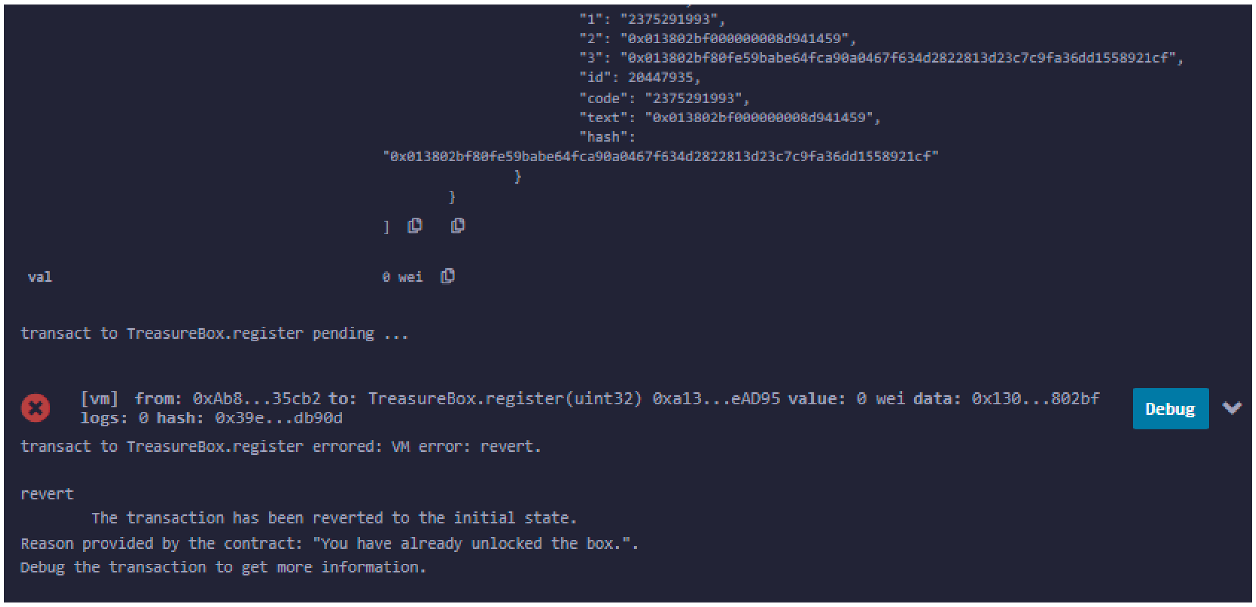
Now in order to full fill the condition, the SHA256 hash of the 'text' first 8 characters (4bytes) need to be equal to the first 8 characters of 'text' i.e., '013802bf' which should unlock the treasure with a true boolean return.

The basic idea of the code is to start at the value 013802bf0000000000000000, calculate the sha256 hash of the value. It then compares the first 8 hex values of the hash and compares it to the first 8 hex values of the value "013802bf". The loop continues to increment the value till the compare was true and return an output of the value when the loop breaks. After running the code for around 2 hours, the loop broke at the value 013802bf000000008d941459. I took "8d941459" and converted it from hexadecimal to decimal, which gave the code value as "2375291993".

I input 2375291993 to unlock and the boolean returned true which indicated that the TreasureBox was unlocked. If I ran the unlock again a message stating that the box is already unlocked is generated.



```
input                           0xef4...41459

decoded input                   {
                                        "uint64 code": "2375291993"
                                }

decoded output                  {
                                        "0": "bool: true"
                                }

logs                            [
                                        {
                                                "from": "0xa131AD247055FD2e2aA8b156A11bdEc81b9eAD95",
                                                "topic": "0x804d04e64cd8811d697278cae0e00a2a11737b259e6f0bc3f7714ce6c1c2c389",
                                                "event": "Log",
                                                "args": {
                                                        "0": 20447935,
                                                        "1": "2375291993",
                                                        "2": "0x013802bf000000008d941459",
                                                        "3": "0x013802bf80fe59babe64fca90a0467f634d2822813d23c7c9fa36dd1558921cf",
                                                        "id": 20447935,
                                                        "code": "2375291993",
                                                        "text": "0x013802bf000000008d941459",
                                                        "hash":
"0x013802bf80fe59babe64fca90a0467f634d2822813d23c7c9fa36dd1558921cf"
                                                }
                                        }
                                ]

val                             0 wei
```



```
                                                "1": "2375291993",
                                                "2": "0x013802bf000000008d941459",
                                                "3": "0x013802bf80fe59babe64fca90a0467f634d2822813d23c7c9fa36dd1558921cf",
                                                "id": 20447935,
                                                "code": "2375291993",
                                                "text": "0x013802bf000000008d941459",
                                                "hash":
                                "0x013802bf80fe59babe64fca90a0467f634d2822813d23c7c9fa36dd1558921cf"
                                          }
                                    }
                                ]

 val                            0 wei

transact to TreasureBox.register pending ...

[vm]  from: 0xAb8...35cb2 to: TreasureBox.register(uint32) 0xa13...eAD95 value: 0 wei data: 0x130...802bf
logs: 0 hash: 0x39e...db90d
transact to TreasureBox.register errored: VM error: revert.

revert
        The transaction has been reverted to the initial state.
Reason provided by the contract: "You have already unlocked the box.".
Debug the transaction to get more information.
```

The approach is written in Python and generates the two inputs required to unlock the TreasureBox. The register (CWID) and the Unlock Code in decimal.

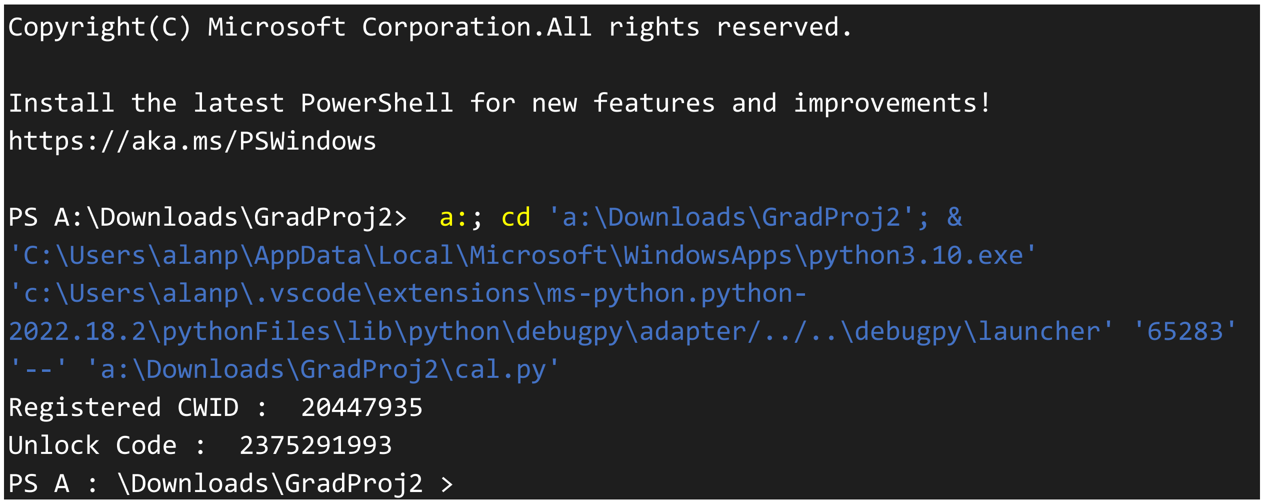
Section IV Appendix

Python Code:

```
from hashlib import sha256
try:
    text = "013802bf0000000000000000"
    cwid = text[:8]
    hash = sha256(bytes.fromhex(text))
    hash_id = hash.hexdigest()[:8]
    while hash_id != cwid:
        text = int(text, 16)
        text += 1
        text = format(text,'x')
        text = text.zfill(24)
        hash = sha256(bytes.fromhex(text))
        hash_id = hash.hexdigest()[:8]
    cwid_id=int(cwid,base=16)
    code_id = int(text[8:24], base=16)
    print('Registered CWID : ', str(cwid_id))
    print('Unlock Code : ',str(code_id))
except KeyboardInterrupt:
    print(text)
```

Output:

```
Windows PowerShell
```



```
Copyright(C) Microsoft Corporation.All rights reserved.

Install the latest PowerShell for new features and improvements!
https://aka.ms/PSWindows

PS A:\Downloads\GradProj2>  a:; cd 'a:\Downloads\GradProj2'; &
'C:\Users\alanp\AppData\Local\Microsoft\WindowsApps\python3.10.exe'
'c:\Users\alanp\.vscode\extensions\ms-python.python-
2022.18.2\pythonFiles\lib\python\debugpy\adapter/../..\debugpy\launcher' '65283'
'--' 'a:\Downloads\GradProj2\cal.py'
Registered CWID :  20447935
Unlock Code :  2375291993
PS A : \Downloads\GradProj2 >
```