

Лабораторна робота № 2

Виконали: Тивонюк Володимир та Виграновський Марко ФБ-41мн

Тема: „Реалізація смарт-контракту або анонімної криптовалюти”.

Мета роботи: «Отримання навичок роботи із смарт-контрактами або анонімними криптовалютами»

Для другого типу лабораторних робіт:

1. розгортання та запуск обраної анонімної валюти, протоколювання майнінгу, пошук слідів деанонімізації;

Zcash

```
PS C:\Users\Volodymyr> docker pull zcashfr/zcash
Using default tag: latest
latest: Pulling from zcashfr/zcash
2cae51d1db04: Download complete
1cd62a315822: Download complete
9fe525454c71: Download complete
39a92a2da2b5: Download complete
c1e9a37f19ec: Download complete
e62d08fa1eb1: Download complete
8ba2eca64c62: Download complete
f10e1bddd3f: Download complete
Digest: sha256:7f41072f70808e1457d8a3ca245d7ecef447ea8f90995f6028273269c065ff90
Status: Downloaded newer image for zcashfr/zcash:latest
docker.io/zcashfr/zcash:latest
```

Базовий конфіг zcash



zcash.conf - Notepad

File Edit Format View Help

```
rpcuser=uvo
rpcpassword=51z.14Zf(dzR
rpcallowip=172.17.0.0/16
listen=1
server=1
daemon=0
txindex=1
```

```
view a summary of image vulnerabilities and recommendations
PS C:\Users\Volodymyr> docker volume create zcash-data
zcash-data
PS C:\Users\Volodymyr> docker volume create zcash-params
zcash-params
```

```
docker run -d --name zcash --user root -v zcash-data:/root/.zcash -v
zcash-params:/root/.zcash-params -v
C:\Users\Volodymyr\Desktop\zcash\zcash.conf:/root/.zcash/zcash.conf -p
8232:8232 -p 127.0.0.1:8233:8333 zcashfr/zcash
```

```
PS C:\Users\Volodymyr> docker run -d --name zcash --user root -v zcash-data:/home/zcash/.zcash -v zcash-params:/home/zcash/.zcash-params -v C:\U
sers\Volodymyr\Desktop\zcash:/mnt/config -p 8232:8232 -p 127.0.0.1:8233:8233 zcashfr/zcash
4af4c8791f8efb26e5c35dc8f97050fc8531693de1b83f8b5e4a3653bea1cf92
```

Тепер треба турбовдовго чекати, усе встановлюється

```
32768K ..... 12% 1.71M 4m13s
65536K ..... 19% 1.69M 3m56s
98304K ..... 25% 1.64M 3m40s
131072K ..... 32% 1.70M 3m21s
163840K ..... 38% 1.73M 3m1s
196608K ..... 44% 1.59M 2m44s
229376K ..... 51% 1.63M 2m26s
262144K ..... 57% 1.66M 2m7s
294912K ..... 64% 1.52M 1m49s
327680K ..... 70% 1.68M 89s
360448K ..... 76% 1.67M 70s
393216K ..... 83% 1.69M 51s
PS C:\Users\Volodymyr> █
```

```
PS C:\Users\Volodymyr> docker exec zcash zcash-cli getinfo
{
  "version": 3000050,
  "protocolversion": 170011,
  "walletversion": 60000,
  "balance": 0.00000000,
  "blocks": 0,
  "timeoffset": 0,
  "connections": 0,
  "proxy": "",
  "difficulty": 1,
  "testnet": false,
  "keypoololdest": 1744280606,
  "keypoolsize": 101,
  "paytxfee": 0.00000000,
  "relayfee": 0.00000100,
  "errors": ""
}
```

```
PS C:\Users\Volodymyr> docker exec zcash zcash-cli getblockchaininfo
{
  "chain": "main",
  "blocks": 0,
  "headers": 0,
  "bestblockhash":
"00040fe8ec8471911baa1db1266ea15dd06b4a8a5c453883c000b031973dce08",
  "difficulty": 1,
  "verificationprogress": 1.664597450110196e-08,
  "chainwork":
"00000000000000000000000000000000000000000000000000000000000000002000",
  "pruned": false,
  "size_on_disk": 1700,
  "commitments": 0,
  "valuePools": [
    {
      "id": "sprout",
      "monitored": true,
      "chainValue": 0.00000000,
      "chainValueZat": 0
    },
    {
      "id": "sapling",
      "monitored": true,
      "chainValue": 0.00000000,
      "chainValueZat": 0
    }
  ],
  "softforks": [
    {
      "id": "bip34",
      "version": 2,
      "enforce": {
        "status": false,
        "found": 1,
        "required": 750,
        "window": 4000
      },
      "reject": {
        "status": false,
        "found": 1,
        "required": 950,
        "window": 4000
      }
    },
    {
      "id": "bip66",
      "version": 3,
      "enforce": {
```

```
"status": false,
"found": 1,
"required": 750,
>window": 4000
},
"reject": {
  "status": false,
  "found": 1,
  "required": 950,
  "window": 4000
}
},
{
  "id": "bip65",
  "version": 4,
  "enforce": {
    "status": false,
    "found": 1,
    "required": 750,
    "window": 4000
  },
  "reject": {
    "status": false,
    "found": 1,
    "required": 950,
    "window": 4000
  }
}
],
"upgrades": {
  "5ba81b19": {
    "name": "Overwinter",
    "activationheight": 347500,
    "status": "pending",
    "info": "See https://z.cash/upgrade/overwinter/ for details."
  },
  "76b809bb": {
    "name": "Sapling",
    "activationheight": 419200,
    "status": "pending",
    "info": "See https://z.cash/upgrade/sapling/ for details."
  },
  "2bb40e60": {
    "name": "Blossom",
    "activationheight": 653600,
    "status": "pending",
    "info": "See https://z.cash/upgrade/blossom/ for details."
  },
  "f5b9230b": {
```

```

    "name": "Heartwood",
    "activationheight": 903000,
    "status": "pending",
    "info": "See https://z.cash/upgrade/heartwood/ for details."
  },
  "consensus": {
    "chaintip": "00000000",
    "nextblock": "00000000"
  }
}

```

Повний вузол Zcash успішно запущений, параметри завантажені, конфігурація прочитана, і він почав процес синхронізації з мережею Zcash. На даний момент він ще не завантажив жодного блоку крім першого.

```

PS C:\Users\Volodymyr> docker exec zcash zcash-cli z_gettotalbalance
{
  "transparent": "0.00",
  "private": "0.00",
  "total": "0.00"
}
PS C:\Users\Volodymyr> docker exec zcash zcash-cli getwalletinfo
{
  "walletversion": 60000,
  "balance": 0.00000000,
  "unconfirmed_balance": 0.00000000,
  "immature_balance": 0.00000000,
  "txcount": 0,
  "keypoololdest": 1744280606,
  "keypoolsize": 101,
  "paytxfee": 0.00000000,
  "seedfp": "6bcfeabd9a98fa15412d6ec60b85f43653cb1509096e89f174484770870a38af"
}

```

Маємо дефолтний порожній гаманець

Створимо нові прозорі та екрановані адреси:

```

PS C:\Users\Volodymyr> docker exec zcash zcash-cli getnewaddress # Створить T-адресу
t1eAKqnfg7SNmTkLmZEnhDvQ7AXrtgYWrVt
PS C:\Users\Volodymyr> docker exec zcash zcash-cli getnewaddress # Створить T-адресу
t1YQVmlzSyrq1pnrJm7M9P2a3UbtuXdvQ76
PS C:\Users\Volodymyr> docker exec zcash zcash-cli z_getnewaddress # Створить Z-адресу
zs1yafpn320rzlywz86yg9ehsy820hvzz5s2tavjeukkrzq9w0vmgpldc9hznpyzc8my56vceazgf
PS C:\Users\Volodymyr> docker exec zcash zcash-cli z_getnewaddress # Створить Z-адресу
zs1d33pldm922yrswrcgey053ay66ld9ftzqrzddj23pkvyx4urv6aaht8zhmagqxs5l5edvufstau
PS C:\Users\Volodymyr> 

```

Транзакцію провести не можу, адже коштів то не маю

Далі вмикаю майнинг:

```
gen=1
genproclimit=-1
```

```
PS C:\Users\Volodymyr> docker exec zcash zcash-cli getmininginfo
{
  "blocks": 0,
  "currentblocksize": 0,
  "currentblocktx": 0,
  "difficulty": 1,
  "errors": "",
  "genproclimit": -1,
  "localsolps": 0,
  "networksolps": 0,
  "networkhashps": 0,
  "pooledtx": 0,
  "testnet": false,
  "chain": "main",
  "generate": true
}
```

майнинг ввімкнувся але, без синхронізації та потужностей ПК я нічого не замайню

name	usage	cpu	memory	disk	network	cpu	cpu usage
Vmmem		0.7%	3,566.3 MB	0 MB/s	0 Mbps	0%	

Ресурси системи воно використовує

```
"verificationprogress": 1.664586126764972e-08,
```

Наскільки я розумію верифікація має дійти до 1, і цього ніколи не станеться((

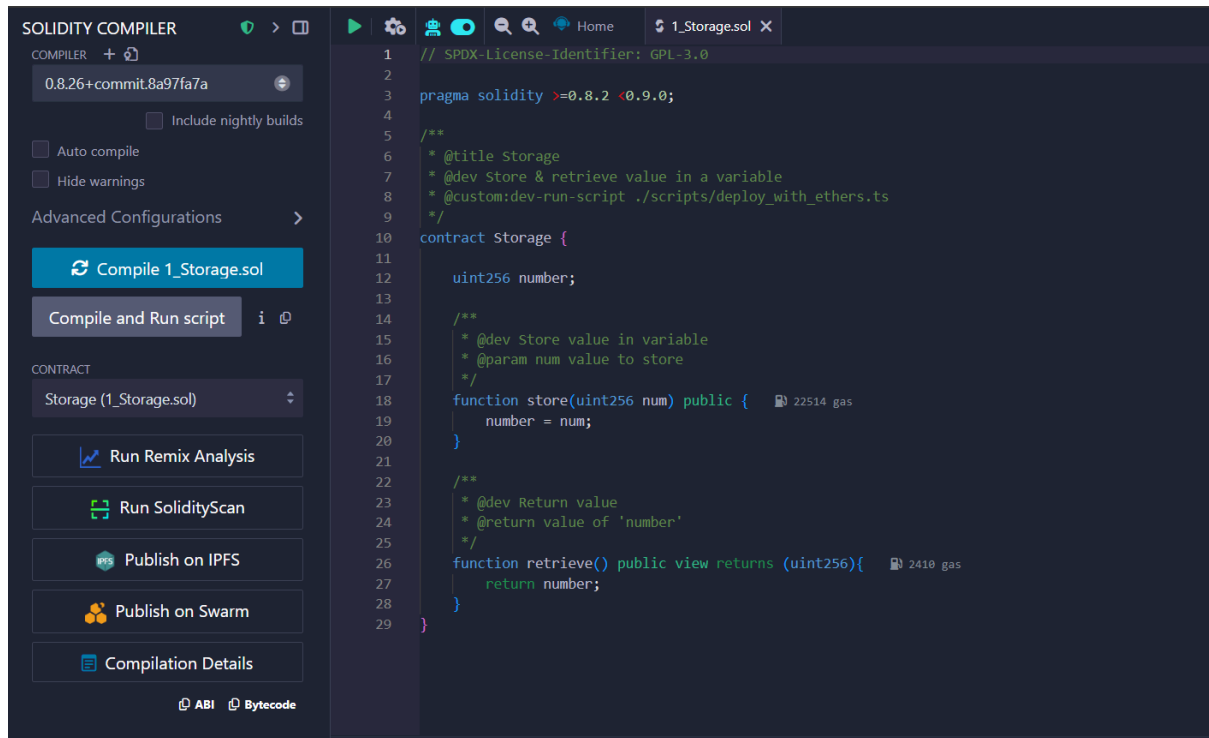
Plan B: Dash - спробував, не вийшло (аналогічні проблеми)

Моніторинг: був би через лог файли, бачили б повідомлення про знайдений блок (generated, proof-of-work found) із зазначенням його хешу та винагороди. Однак, через відсутність повної синхронізації та низьку ефективність CPU-майнінгу, практично ці логи відображали б лише марні спроби роботи на застарілому стані ланцюжка.

Деанонімізація: відбувалася за рахунок аналізу взаємозв'язків між прозорими (T-адреси) та екранованими (Z-адреси) транзакціями. Вивчення транзакцій, де кошти входять (T->Z) або виходять (Z->T) з екранованого пулу, оскільки саме тут прозорий та анонімний світи перетинаються.

Смарт контракти будуть розроблятись у RemixIDE адже не знадобиться встановлення нічого локально (Ethereum)

2. розгортання та запуск обраного смарт-контракту, підвищення ефективності роботи смарт-контракту з точки зору витрати гасу;



Storage – це дуже простий смарт-контракт, зазвичай для платформ на базі Ethereum, що демонструє базове збереження даних. Користувач взаємодіє з ним, викликаючи функцію store, якій передається одне значення – ціле число, що записується у змінну стану контракту. Потім іншою функцією, retrieve, можна прочитати це збережене число.

STORAGE AT 0XD8B...33FA8

Balance: 0 ETH

store uint256 num

retrieve

Low level interactions

CALLDATA

Transact

Можемо взаємодіяти з контрактом

Balance: 0 ETH

store 5

retrieve

0: uint256: 5

Як бачимо зберігаємо дані в контракті


gas	30611 gas	
transaction cost	26618 gas	
execution cost	5414 gas	




для запису

execution cost	2410 gas (Cost only applies when called by a contract)	
----------------	--	--

та для читання

Для такого простого контракту оптимізація мінімальна, додати if statement щоб не вставляти значення що воно таке саме


```
function store(uint256 num) public {  24636 gas
    if (num != number) {
        number = num;
    }
}
```

gas	27287 gas	
transaction cost	23727 gas	
execution cost	2523 gas	

Значно впає execution cost

3. розробка власного смарт-контракту

Мій контракт - формат вгадування ключового числа:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract NumberGuesser {

    address public immutable owner;
    uint256 private immutable secretNumber;

    uint256 public guessCount;
    bool public guessedCorrectly;
    address public winner;

    event GuessAttempt(address indexed guesser, uint256 guess);
    event CorrectGuess(address indexed winner, uint256 number);

    modifier notYetGuessed() {
        require(!guessedCorrectly, "Number already guessed");
        _;
    }

    constructor() {
        owner = msg.sender;
        secretNumber = uint8( // Cast to uint8 for range 0-255
            uint256( // Convert hash to number
```

```

        // Combine block difficulty and timestamp for more
variance
        keccak256(abi.encodePacked(block.prevrando,
block.timestamp, msg.sender))
    )
    ) % 100; // Modulo 100 for range 0-99
}

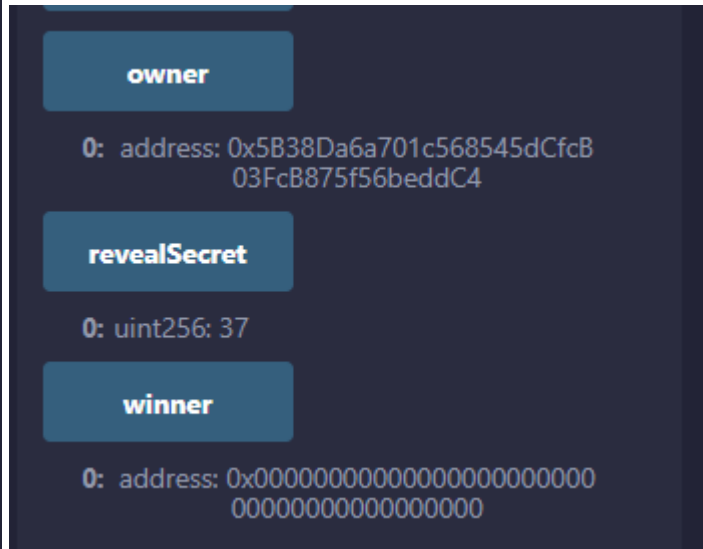
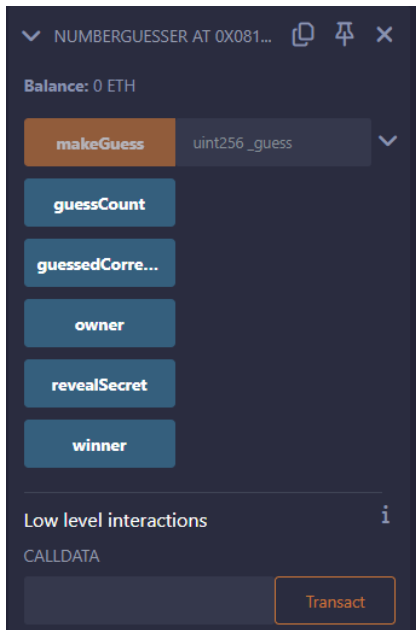
function makeGuess(uint256 _guess) external notYetGuessed {
    require(_guess < 100, "Should be between 0 and 99");
    guessCount++;
    emit GuessAttempt(msg.sender, _guess);

    if (_guess == secretNumber) {
        guessedCorrectly = true;
        winner = msg.sender;
        emit CorrectGuess(msg.sender, secretNumber);
    }
}

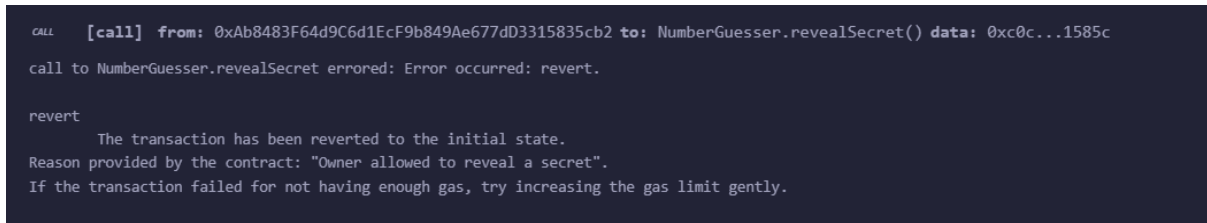
function revealSecret() external view returns (uint256) {
    require(msg.sender == owner, "Owner allowed to reveal a
secret");
    return secretNumber;
}
}

```

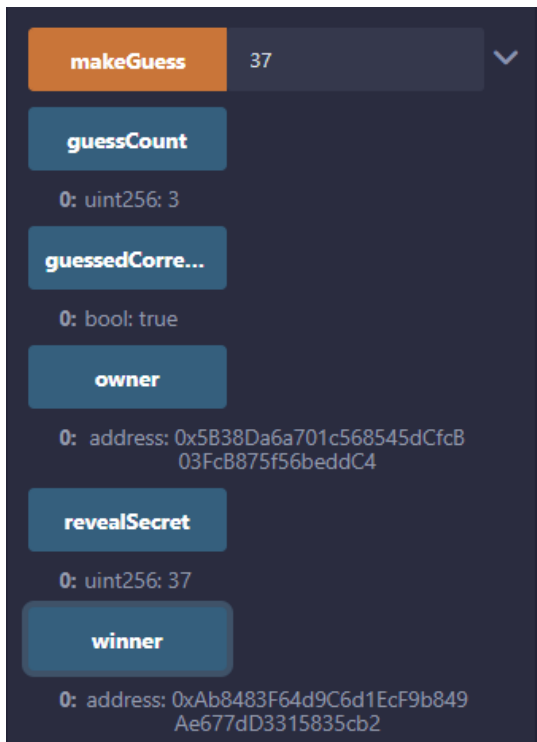
1. Конструктор генерить число
2. Підключенні до контракту можуть вгадувати число доки його не відгадають
3. Ім'я переможця записати до змінної в контракті та зупинити приймання вгадування
4. Власник може переглянути число яке треба вгадати
5. Кількість вгадувань може переглядати будь хто, та хто є власником



Отримуємо значення яке треба вгадати та переходимо на іншого юзера



Спроба ревілу під іншим юзером



Як бачимо, після декількох вгадувань, адреса того хто вгадав збереглася у вкладинці winner

```
✖ [vm] from: 0xAb8...35cb2 to: NumberGuesser.makeGuess(uint256) 0x081...a0f85 value: 0 wei data: 0x226...00027 logs: 0 hash: 0xed4...13e34
transact to NumberGuesser.makeGuess errored: Error occurred: revert.

revert
  The transaction has been reverted to the initial state.
Reason provided by the contract: "Number already guessed".
If the transaction failed for not having enough gas, try increasing the gas limit gently.
```

Подальше вгадування не буде дозволено

```
✖ [vm] from: 0xAb8...35cb2 to: NumberGuesser.makeGuess(uint256) 0xB30...Ea098 value: 0 wei data: 0x226...003e7 logs: 0 hash: 0xdeb...11725
transact to NumberGuesser.makeGuess errored: Error occurred: revert.

revert
  The transaction has been reverted to the initial state.
Reason provided by the contract: "Should be between 0 and 99".
If the transaction failed for not having enough gas, try increasing the gas limit gently.
```

Помилка якщо не те число

gas	60440 gas	📄
transaction cost	52556 gas	📄
execution cost	31352 gas	📄

Ціна вгадування, всі інші операції мінімальна ціна.

Оптимізація контракту:

gas	33071 gas	📄
transaction cost	28757 gas	📄
execution cost	7553 gas	📄

gas	52736 gas	📄
transaction cost	45857 gas	📄
execution cost	24653 gas	📄

ціна в середньому впала, але незначно - це за рахунок пакування змінних в ініціалізації функцій. Також immutable для змінних які не будуть змінюватись - легше читання.

```
address public immutable owner;  
uint256 private immutable secretNumber;  
  
uint96 public guessCount;  
bool public guessedCorrectly;  
address public winner;
```

Висновки:

Отримано практичні навички розгортання вузла криптовалюти, базової взаємодії з ним, а також розробки, розгортання, тестування та базової оптимізації смарт-контрактів на Solidity в середовищі Remix IDE. Виявлено ключові операційні вимоги та обмеження роботи з повним вузлом Zcash.