



成都信息工程大学
Chengdu University of Information Technology

《金睛财务》 测试报告

[V1.0(版本号)]

[2024 年 4 月 15 日]

目录

第一章简介	1
1.1 目的	1
1.2 范围	1
1.3 名词解释	2
1.4 合约关系结构	2
第2章 测试简介	2
2.1 测试方法	2
2.2 测试环境	3
第3章 测试	3
3.1 功能测试	3
3.1.1 工厂合约功能测试:	3
3.1.2 工厂合约功能测试调用函数接口及参数:	4
3.1.3 工厂合约测试合约:	4
3.1.4 工厂合约测试步骤:	6
3.1.5 多签钱包出入账相关功能测试:	8
3.1.6 多签钱包出入账相关功能测试调用函数接口及参数:	10
3.1.7 多签钱包出入账相关功能测试合约	11
3.1.8 多签钱包出入账相关功能测试步骤	13
3.1.9 多签钱包其他功能测试	24
3.1.10 多签钱包其他功能测试调用函数接口及参数:	24
3.1.11 多签钱包其他功能测试合约	25
3.1.12 多签钱包其他功能测试步骤	26
第4章	31
4.1 总结	31

第一章简介

测试报告为 金睛财务 项目的测试报告,用与测试智能合约相关接口与功能完整性

1.1 目的

目的在于总结测试阶段的测试以及分析测试结果, 描述系统是否符合需求

1.2 范围

此报告包括三个智能合约 holdMoney MultiSigWallet DateTimeContract,测试工厂合约(holdMoney)创建多签钱包(MultiSigWallet),以及多签钱包(MultiSigWallet)实现出账,进账,撤销进账,撤销出账等核心共功能

1.3 名词解释

缩写词或术语	中文解释
最高管理员	创建项目且具有签名权以及赋予其他人签名权的地址
合约所有者	具有签名权的地址

1.4 合约关系结构

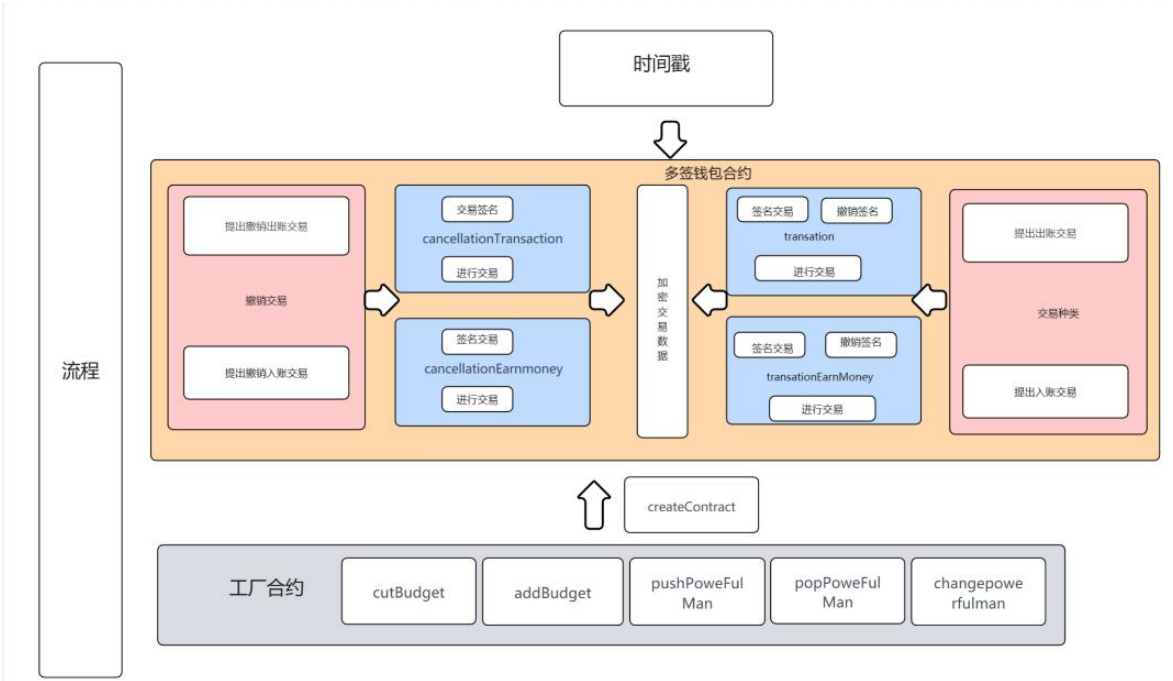


图 1 合约关系图

由工厂合约初始化项目,再由最高所有者创建项目的多签钱包,并实现出账入账等功能,多签钱包的所有交易都会被打上时间戳

第 2 章 测试简介

2.1 测试方法

- 1.本次测试采用手动与自动结合
- 2.采用测试工具合约代替脚本
- 3.主要采用 remix

理由如下：

- 1.测试工具合约允许你编写自动化测试用例，这些用例可以在不人工干预的情况下运行。这可以节省大量的时间，特别是当需要频繁运行测试时。
- 2.测试工具合约通常生成更详细的测试报告，包括测试用例的执行结果、事件触发等信息，帮助你更好地了解合约的行为。
- 3.测试工具合约可以确保测试过程的可重复性。每次运行测试时，它们将执行相同的测试用例，以确保结果一致性。
- 4.测试工具合约可以在与现实最接近的环境测试合约

2.2 测试环境

硬件环境	<i>FISCO BCOS 2.0 以上</i>
软件环境	<i>使用 remix 并且启动 remix 自带的优化功能或者搭载 WaBASE-Front 中间件</i>
系统要求的配置环境	<i>Solidity 环境在 0.6.0 以上</i>

第 3 章 测试

3.1 功能测试

3.1.1 工厂合约功能测试:

测试功能	测试结论
增加合约所有者	正常
减少合约所有者	正常

测试功能	测试结论
替换合约所有者	正常
增加预算	正常
减少预算	正常
创建多签钱包	正常

3.1.2 工厂合约功能测试调用函数接口及参数:

名称	参数 1	参数 2
constructor	10000000	
cutBudget	100000	
addBudget	200000	
pushPoweFulMan	0xe0520f037715baca69dDfa2bD0E621ab707DB751	
popPoweFulMan	2	
changepowerfulman	2	0x6849BD1B078dB932fB32b6eEF1e4675F489fffff
createContract	1	1



3.1.3 工厂合约测试合约:

```
contract check{
    holdMoney public hold;
    constructor(uint budget) public
    {
        hold=new holdMoney(budget);
    }
}
```

```

        require(hold.owner()==address(this), "Owner should be the
deployer.");
        require(hold.viewAccess()==1, "Initial viewAccess should be
1.");
        require(hold.number()==1, "Initial number should be 1.");
    }

    function add() public
    { //测试增加预算的功能
        uint256 Budget=hold.Budget();
        hold.addBudget(200000);
        Budget+=200000;
        require(hold.Budget()==Budget, "wrong add");
    }

    function cut() public
    {
        //测试减少预算的功能
        uint256 Budget=hold.Budget();
        hold.cutBudget(100000);
        Budget-=100000;
        require(hold.Budget()==Budget, "wrong cut");
    }

    function push() public
    { //测试增加合约所有者的功能
        hold.pushPowerFulMan(msg.sender);
        require(hold.isOwner(msg.sender), "wrong push");
    }

    function pop() public
    { //测试减少合约所有者的功能
        hold.popPowerFulMan(2);
        require(!hold.isOwner(msg.sender), "wrong pop");
    }

    function change() public {
        //测试替换所有者的功能
        hold.changepowerfulman(2, msg.sender);
        require(hold.isOwner(msg.sender), "wrong push");//注意合约调用
者
    }

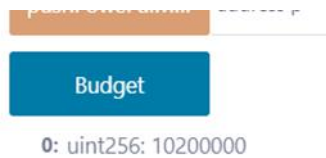
    function create() public {
        //测试创建多签钱包的功能
        hold.createContract(1, 2);
        require(address(hold.mu())!=address(0), "wrong create");
    }
}

```

测试合约地址:0xc8737bd923D95d8031A550f479Be24333530e3d9

3.1.4 工厂合约测试步骤:

1.增加预算(add):触发 add 函数增加预算



效果:预算增加 2000000

2.减少预算(cut)触发 cut 函数减少预算



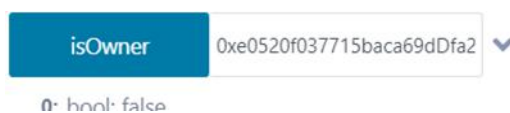
效果:预算减少 100000

3.增加合约所有者(push)触发 push 函数增加合约所有者



效果:该地址成为合约所有者

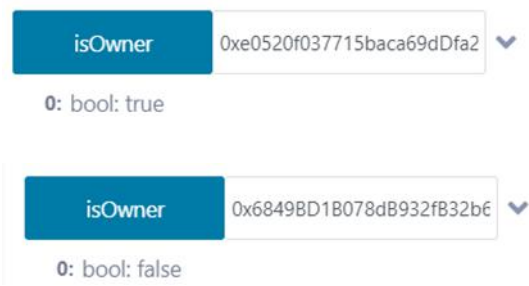
4.减少合约所有者(pop)触发 pop 函数减少合约所有者



效果:该地址不再是合约所有者

5.替换合约所有者(chage)触发 chage 函数替换所有者(先触发一次 push 函数再触发 chage 函数)

Chage 函数触发前:



Change 函数触发后:



效果:合约所有者地址完成替换,旧的地址被新的替代

6.创建多签钱包(create)触发 create



多签钱包地址:

0x3990673609446Da686EcD78cf178f1724CB7546A

效果:多签钱包创建成功

3.1.5 多签钱包出入账相关功能测试:

测试功能	测试结论
提出出账交易	正常
提出入账交易	正常
签名入账交易	正常
签名出账交易	正常
撤销出账交易签名	正常
撤销入账交易签名	正常
完成出账交易	正常
完成入账交易	正常
撤销出账交易签名	正常
撤销入账交易签名	正常
完成撤销入账交易	正常
完成撤销出账交易	正常
查看入账交易数据	正常

测试功能	测试结论
查看出账交易数据	正常
查看出账签名状态	正常
查看入账交易状态	正常

交易数据及其结构

1. 出账:



2.入账:



3.1.6 多签钱包出入账相关功能测试调用函数接口及参数:

出账交易

名称	参数 1	参数 2	参数 3
submitTransaction	0xe0520f037715baca69dDfa2bD0E621ab707DB751	100000	\u63d0\u51fa\u51fa\u8d26\u4ea4\u6613\u6d4b\u8bd5(提出出账交易测试)
confirmTransaction	1		
revokeConfirmation	1		
executeTransaction	1		
revokeAlreadyTransaction	1		
cancellationTransaction	1	\u64a4\u9500\u51fa\u8d26\u4ea4\u6613\u6d4b\u8bd5(撤销出账交易测试)	
isConfirmedtransaction	1	0xe0520f037715baca69dDfa2bD0E621ab707DB751	
getTransaction	1		

入账交易

submitearnMoney	100	\u63d0\u51fa\u5165\u8d26\u4ea4\u6613\u6d4b\u8bd5(提出入账交易)
-----------------	-----	--

confirmearnMoney	1	
revokeearnMoney	1	
earnMoney	1	
revokeAlreadyEarnmoney	1	
cancellationEarnmoney	1	\u64a4\u9500 \u5165\u8d26 \u4ea4\u6613 \u6d4b\u8bd5 (撤销入账交易测试)
isConfirmedtransationEarn	1	0xe0520f037715baca69dDfa2bD0E621ab707DB751
getTransaction	1	

3.1.7 多签钱包出入账相关功能测试合约

```

contract check{
    MultiSigWallet public mu;
    holdMoney public hold;
    constructor(uint budget) public
    {
        hold=new holdMoney(budget);
    }
    function push() public
    { //测试增加合约所有者的功能
        hold.pushPowerFulMan(msg.sender);
        require(hold.isOwner(msg.sender),"wrong push");
    }
    function create() public {
        //测试创建多签钱包的功能
        hold.createContract(1,2);
        require(address(hold.mu())!=address(0),"wrong create");
        mu=hold.mu();
    }
    function tran() public
    {
        //测试提出出账交易的功能
        mu.submitTransaction(msg.sender,100000,"\u63d0\u51fa\u51fa\u8d26\u4ea4\u6613\u6d4b\u8bd5");
    }
}

```

```
function tranEarn() public
{
    //测试提出入账交易的功能
    mu.submitearnMoney(100, "\u63d0\u51fa\u5165\u8d26\u4ea4\u6613\u6d4b\u8bd5");
}
function execute(uint index) public
{
    //测试完成出账交易的功能
    require(index>0, "wrong index");
    mu.executeTransaction(index);
    //require(mu.transactions(index-1).executed, "");
}
function executeEarn(uint index) public
{
    //测试完成入账交易的功能
    require(index>0, "wrong index");
    mu.earnMoney(index);
}
function revokAlreadyTran(uint index) public
{
    require(index>0, "wrong index");
    mu.revokeAlreadyTransaction(index);
}
function revokAlreadyEarn(uint index) public
{
    require(index>0, "wrong index");
    mu.revokeAlreadyEarnmoney(index);
}
function cancellTran(uint index) public
{
    require(index>0, "wron d index");
    mu.cancellationTransaction(index, "\u64a4\u9500\u51fa\u8d26\u4ea4\u6613\u6d4b\u8bd5");
}
function cancellEarn(uint index) public
{
    require(index>0, "wrong index");
    mu.cancellationEarnmoney(index, "\u64a4\u9500\u5165\u8d26\u4ea4\u6613\u6d4b\u8bd5");
}
}
```

3.1.8 多签钱包出入账相关功能测试步骤

- 1.调用测试合约 push 增加合约所有者
- 2.调用测试合约 create 创建多签钱包
- 3.调用测试合约 tran 函数 提出出账交易



效果:入账交易已被提出

4.调用测试合约 tranEarn 函数 提出入账交易

EarnMoney

0

▼

0: uint256: value 100

1: string: data 提出入账交易测试

2: bool: executed false

3: uint256: numConfirmations 0

4: address: sender 0xAa4C4EFb07CD74438712c724aa74e6578Feb46EA

5: string: submitearnMoneytime 2023-10-23 20:2:48

6: string: earnMoneytime not

7: uint256: revokeAlreadyEarnmoneycount 0

8: bytes32: signature 0x00

效果:出账交易已被提出

5.切换地址 调用多签钱包的 confirmearnMone 函数 签名入账交易

EarnMoney

0

▼

0: uint256: value 100

1: string: data 提出入账交易测试

2: bool: executed false

3: uint256: numConfirmations 1

4: address: sender 0xAa4C4EFb07CD74438712c724aa74e6578Feb46EA

5: string: submitearnMoneytime 2023-10-23 20:2:48

6: string: earnMoneytime not

7: uint256: revokeAlreadyEarnmoneycount 0

8: bytes32: signature 0x00

效果:入账交易签名数增加为 1

6. 调用多签钱包的 confirmTransaction 函数 签名入账交易

transactions

0

0:

address: destination 0xe0520f037715bac
a69dDfa2bD0E621ab707DB751

1:

address: sender 0xAa4C4EFb07CD744387
12c724aa74e6578Feb46EA

2:

uint256: value 100000

3:

string: data 提出出账交易测试

4:

bool: executed false

5:

uint256: numConfirmations 1

6:

string: submitTransactiontime 2023-10-2
3 19:59:36

7:

string: executeTransactiontime not

8:

uint256: revokeAlreadyTransactioncount
0

9:

bytes32: signature 0x000000000000000000
0000000000000000000000000000000000
0000000000000000

效果:出账交易签名数增加

7. 调用多签钱包的 isConfirmedtransationEarn 函数查看指定地址对指定入账交易的签名情况

isConfirmedtransationEarn

_txId:

1

_owner:

0xe0520f037715baca69dDfa2bD0E6;

Calldata

参数

call

0:

bool: true

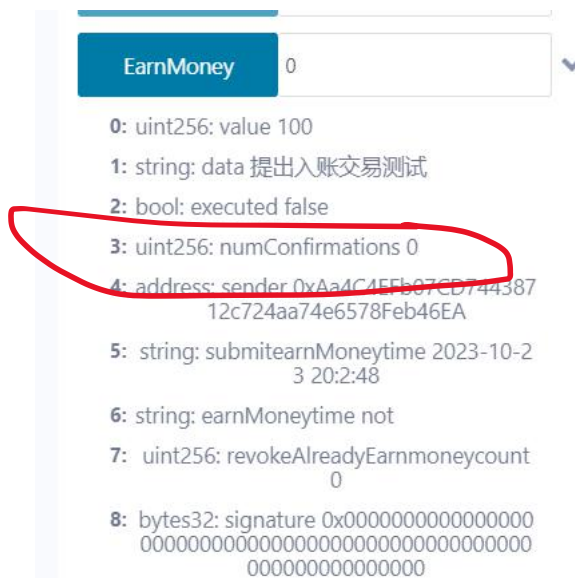
效果:查看指定地址对指定交易完成签名

8. 调用多签钱包的 isConfirmedtransation 函数查看指定地址对指定入账交易的签名情况



效果:查看指定地址对指定交易完成签名

9.调用多签钱包 revokeearnMoney 函数撤销入账签名



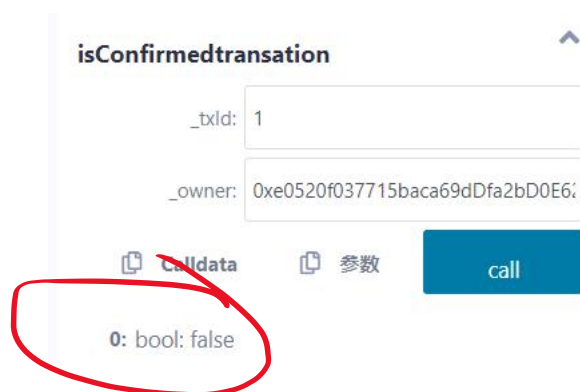
效果:入账交易签名数减少为 0

10.调用多签钱包 revokeConfirmation 函数撤销出账交易的签名



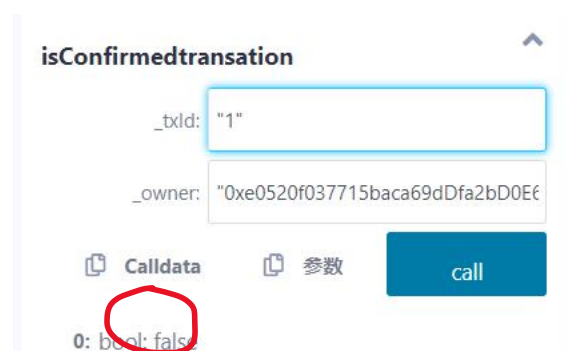
效果:出账交易签名数减少

11.调用多签钱包的 isConfirmedtransationEarn 函数查看指定地址对指定入账交易的签名情况



效果：查看指定地址对指定交易没有签名

12 调用多签钱包的 isConfirmedtransation 函数查看指定地址对指定入账交易的签名情况



效果：查看指定地址对指定交易没有签名

13.调用 confirmTransaction 重新签名出账交易

transactions

0

▼

0:

address: destination 0xe0520f037715bac
a69dDfa2bD0E621ab707DB751

1:

address: sender 0xAa4C4EFb07CD744387
12c724aa74e6578Feb46EA

2:

uint256: value 100000

3:

string: data 提出出账交易测试

4:

bool: executed false

5:

uint256: numConfirmations 1

6:

string: submitTransactiontime 2023-10-2
3 19:59:36

7:

string: executeTransactiontime not

8:

uint256: revokeAlreadyTransactioncount
0

9:

bytes32: signature 0x0000000000000000
00000000000000000000000000000000
0000000000000000

效果:出账交易的签名数增加

14.调用 confirmearnMone 重新签名入账交易

EarnMoney

0

▼

0:

uint256: value 100

1:

string: data 提出入账交易测试

2:

bool: executed false

3:

uint256: numConfirmations 0

4:

address: sender 0xAa4C4EFb07CD744387
12c724aa74e6578Feb46EA

5:

string: submitearnMoneytime 2023-10-2
3 20:2:48

6:

string: earnMoneytime not

7:

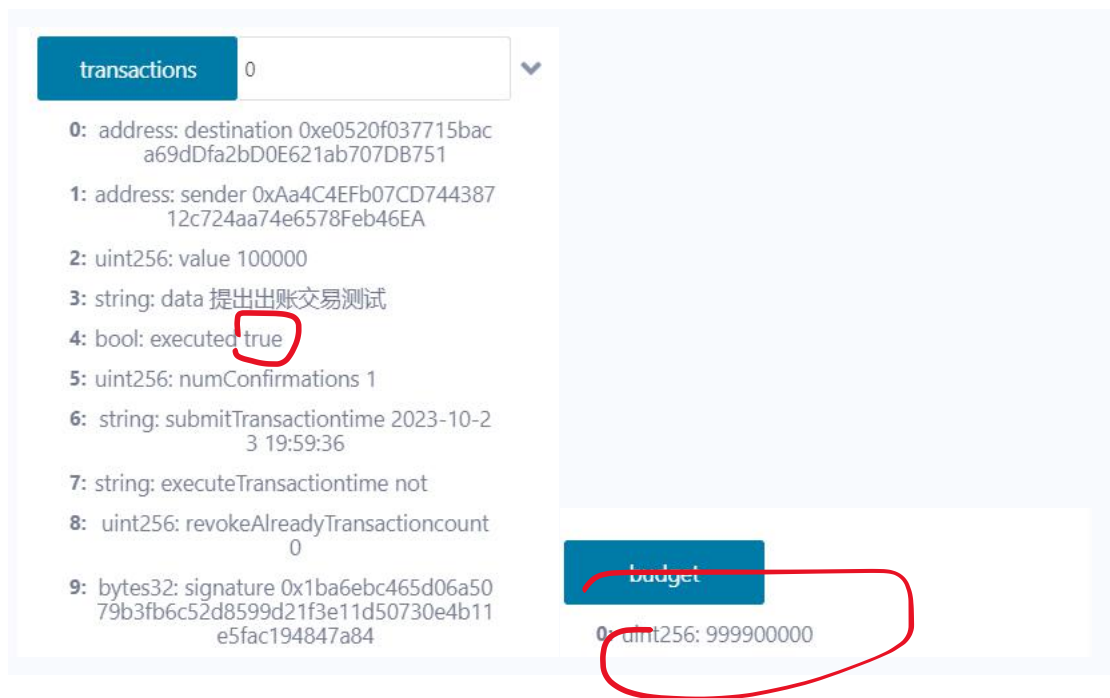
uint256: revokeAlreadyEarnmoneycount
0

8:

bytes32: signature 0x0000000000000000
00000000000000000000000000000000
0000000000000000

效果:入账交易签名增加

15.调用测试合约 execute 函数 完成出账交易



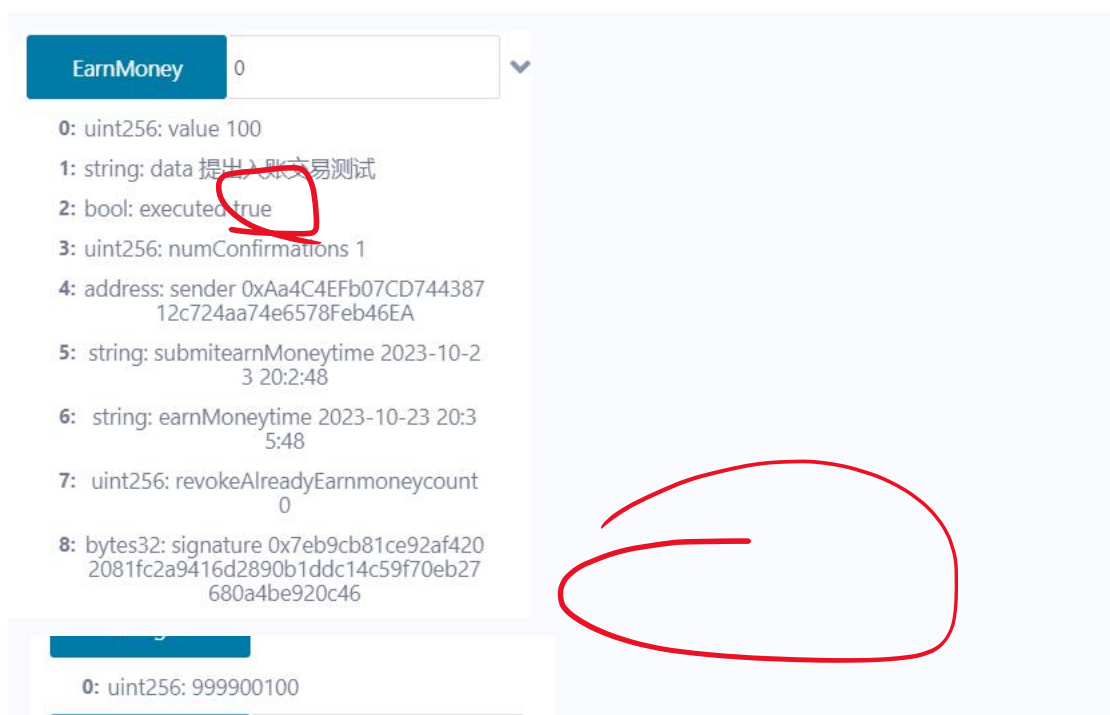
transactions 0

- 0: address: destination 0xe0520f037715bac a69dDfa2bD0E621ab707DB751
- 1: address: sender 0xAa4C4EFb07CD744387 12c724aa74e6578Feb46EA
- 2: uint256: value 100000
- 3: string: data 提出出账交易测试
- 4: bool: executed true
- 5: uint256: numConfirmations 1
- 6: string: submitTransactiontime 2023-10-23 19:59:36
- 7: string: executeTransactiontime not
- 8: uint256: revokeAlreadyTransactioncount 0
- 9: bytes32: signature 0x1ba6ebc465d06a50 79b3fb6c52d8599d21f3e11d50730e4b11 e5fac194847a84

budget 0 uint256: 999900000

效果:出账交易的状态为 true 交易完成,预算减少

16. 调用测试合约 executeEarn 函数 完成进账交易



EarnMoney 0

- 0: uint256: value 100
- 1: string: data 提出入账交易测试
- 2: bool: executed true
- 3: uint256: numConfirmations 1
- 4: address: sender 0xAa4C4EFb07CD744387 12c724aa74e6578Feb46EA
- 5: string: submitearnMoneytime 2023-10-23 20:2:48
- 6: string: earnMoneytime 2023-10-23 20:35:48
- 7: uint256: revokeAlreadyEarnmoneycount 0
- 8: bytes32: signature 0x7eb9cb81ce92af420 2081fc2a9416d2890b1ddc14c59f70eb27 680a4be920c46

0: uint256: 999900100

效果:入账交易的状态为 true 交易完成,预算增加

17.调用测试合约的 revokeAlreadyEarn 签名撤销已完成入账交易

revokealreadyEarnmoney

0: 0

1: 0xAa4C4EFb07CD74438712c724aa74

Calldata 参数 call

0: bool: true

EarnMoney 0 EarnMoney - call

- 0: uint256: value 100
- 1: string: data 提出入账交易测试
- 2: bool: executed true
- 3: uint256: numConfirmations 1
- 4: address: sender 0xAa4C4EFb07CD74438712c724aa74e6578Feb46EA
- 5: string: submitearnMoneytime 2023-10-23 20:2:48
- 6: string: earnMoneytime 2023-10-23 20:35:48
- 7: uint256: revokeAlreadyEarnmoneycount
- 8: bytes32: signature 0x7eb9cb81ce92af4202081fc2a9416d2890b1ddc14c59f70eb27680a4be920c46

效果:撤销已完成入账交易签名数增加,指定地址对指定入账交易撤销签名状态为 true

18. 调用测试合约的 revokAlreadyTran 签名撤销已完成出账交易

revokealreadyTransaction

0: 0

1: 0xAa4C4EFb07CD74438712c724aa74

Calldata 参数 call

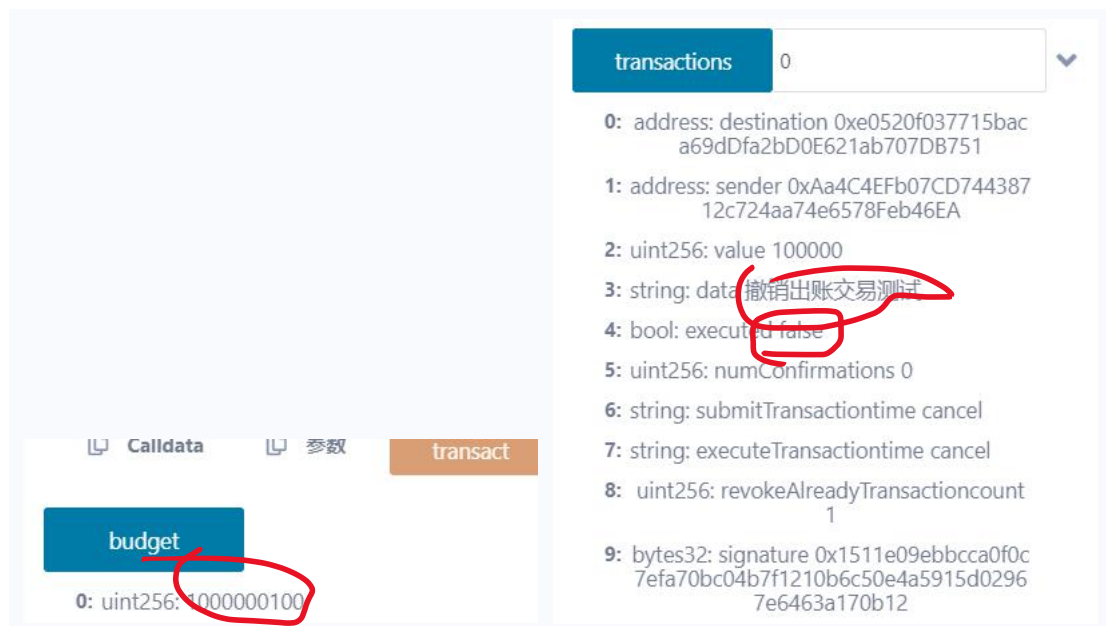
0: bool: true

transactions 0

- 0: address: destination 0xe0520f037715bac a69dDfa2bD0E621ab707DB751
- 1: address: sender 0xAa4C4EFb07CD74438712c724aa74e6578Feb46EA
- 2: uint256: value 100000
- 3: string: data 提出出账交易测试
- 4: bool: executed true
- 5: uint256: numConfirmations 1
- 6: string: submitTransactiontime 2023-10-23 19:59:36
- 7: string: executeTransactiontime not
- 8: uint256: revokeAlreadyTransactioncount
- 9: bytes32: signature 0x1b5ebc465d06a5079b3fb6c52d8599d21f3e11d50730e4b11e5fac194847a84

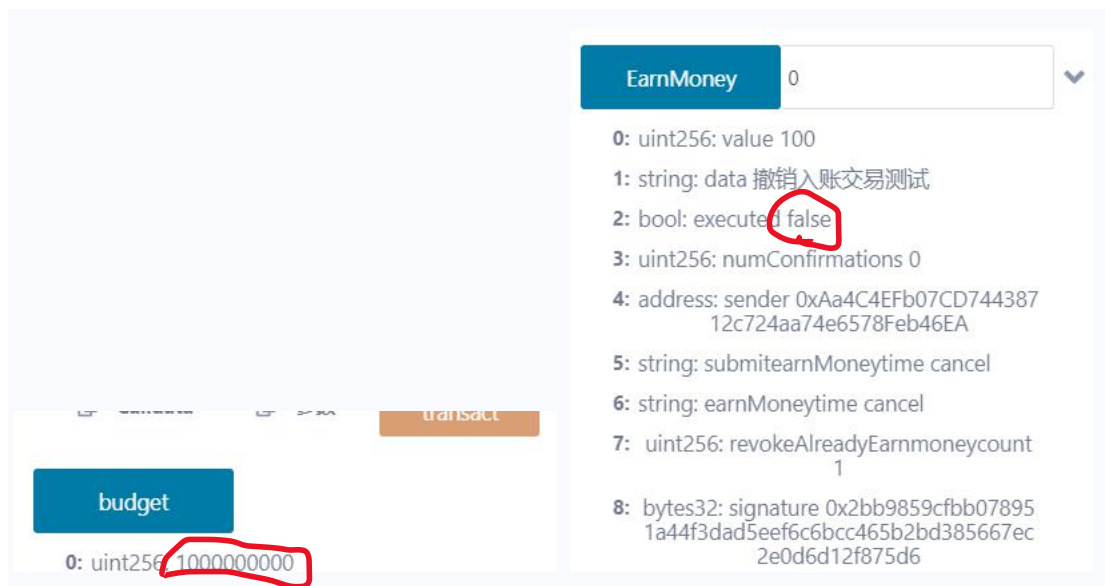
效果:撤销已完成出账交易签名数增加,指定地址对指定出账交易撤销签名状态为 true

19. 调用测试合约的 cancelTran 撤销已完成出账交易



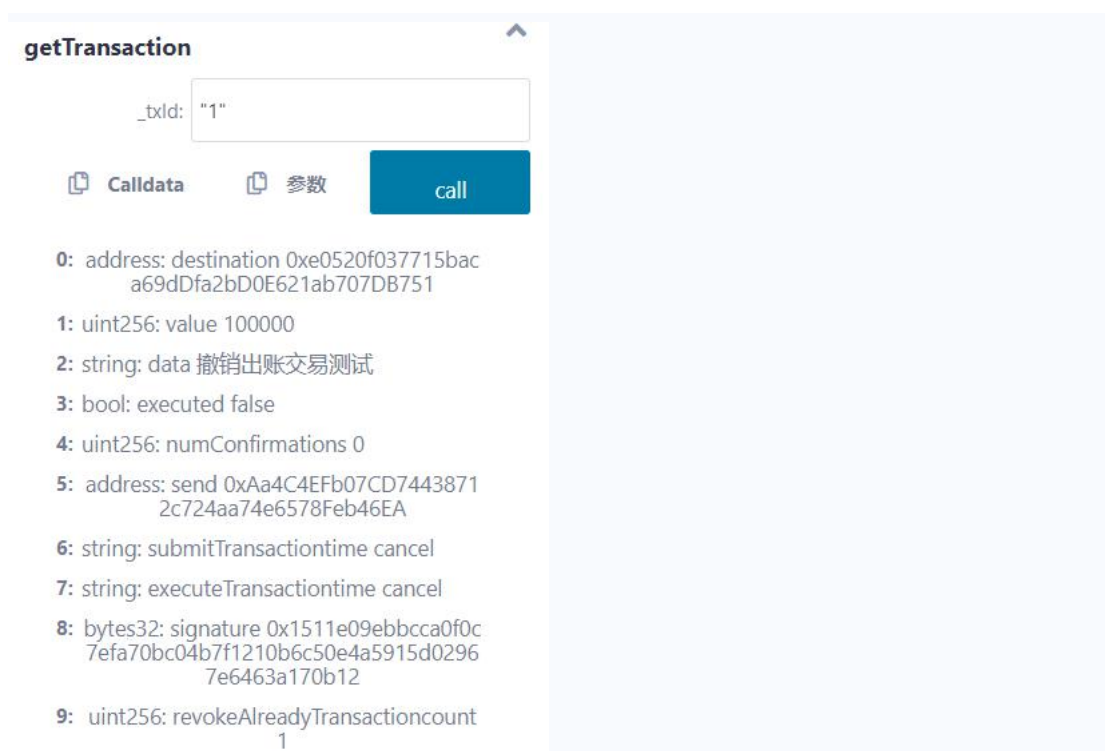
效果:撤销出账交易,预算增加 100000,交易状态改为 false

20 调用测试合约的 `cancelEarn` 撤销已完成入账交易

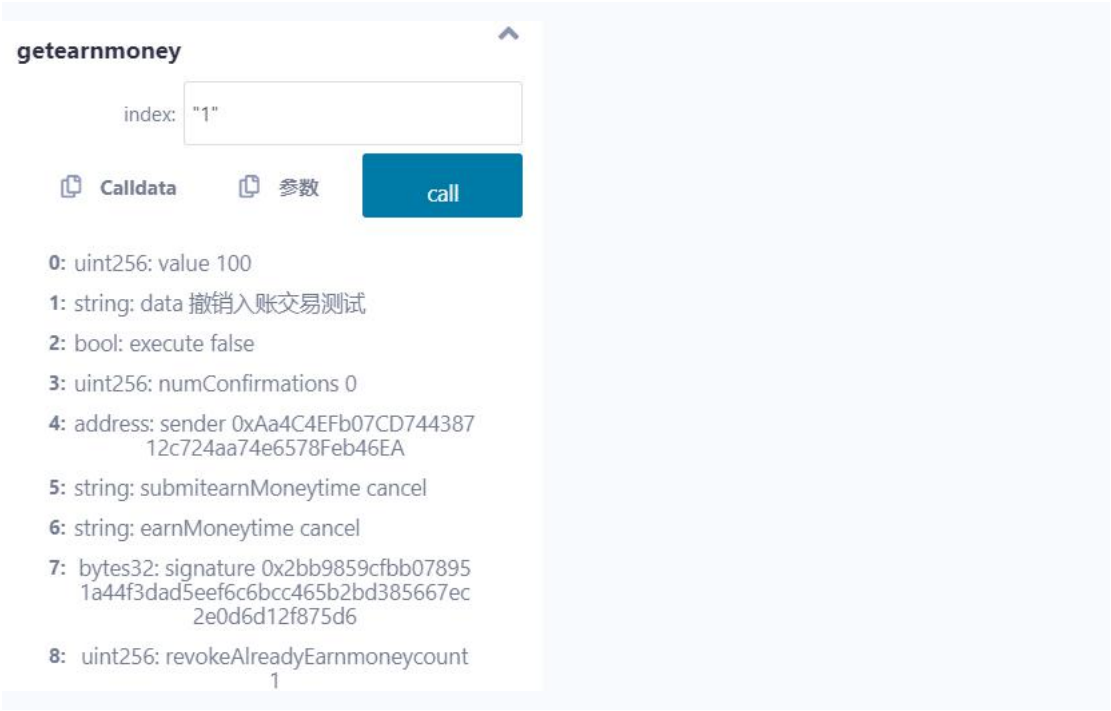


效果:撤销入账交易,预算减少 100, ,交易状态改为 false

21.调用多签钱包的 getTransaction 函数查看交易详情



22.调用多签钱包的 getearnmoney 函数查看交易详情



3.1.9 多签钱包其他功能测试

测试功能	测试结论
改变查看等级	正常
改变签名所需人数	正常
替换合约最高管理员	正常

3.1.10 多签钱包其他功能测试调用函数接口及参数:

名称	参数
changerequiredSignatures	2
chageContractowner	0xe0520f037715baca69dDfa2bD0E621ab707DB751

changeViewRight

1-3

3.1.11 多签钱包其他功能测试合约

```

contract check{
    MultiSigWallet public mu;
    holdMoney public hold;
    constructor(uint budget) public
    {
        hold=new holdMoney(budget);
    }
    function push() public
    { //测试增加合约所有者的功能
        hold.pushPowerFulMan(msg.sender);
        require(hold.isOwner(msg.sender), "wrong push");
    }
    function create() public {
        //测试创建多签钱包的功能
        hold.createContract(1,2);
        require(address(hold.mu())!=address(0), "wrong create");
        mu=hold.mu();
    }
    function tran() public
    {
        //测试提出出账交易的功能
        mu.submitTransaction(msg.sender,100000,"\u63d0\u51fa\u51fa\u8d26\u4ea4\u6613\u6d4b\u8bd5");
    }
    function tranEarn() public
    { //测试提出入账交易的功能
        mu.submitearnMoney(100,"\u63d0\u51fa\u5165\u8d26\u4ea4\u6613\u6d4b\u8bd5");
    }

    function changeViewLevel(uint level) public
    {
        require(level>=1&&level<=3,"wrong level");
        mu.changeViewRight(level);
        require(mu.level()==level,"wrong Level");
    }
}

```

```
    }  
    function changerequiredSignatures() public  
    {  
        mu.changerequiredSignatures(2);  
        require(mu.requiredSignatures()==2,"wrong  
requiredSignatures");  
    }  
    function chageContractowner() public  
    {  
        mu.chageContractowner(msg.sender);  
        require(mu.owners(0)==msg.sender,"wrong chage");  
    }  
}
```

测试合约地址：

0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C

3.1.12 多签钱包其他功能测试步骤

- 1.调用测试合约 push 函数两次(注意两次调用的合约调用者应该不同)
- 2.调用测试合约 create 函数创建多签钱包
- 3.调用测试合约的 tranEarn 函数提出入账交易
- 4.调用测试合约 tran 函数提出出账交易
- 5.调用多签钱包的 changeViewLevel 函数改变查看等级 2 级查看等级(记得切换地址):

getearnmoney1getearnmoney - call

0: uint256: value 0

1: string: data

2: bool: execute false

3: uint256: numConfirmations 0

4: address: sender 0x00

5: string: submitearnMoneytime 2023-10-23 22:0:12

6: string: earnMoneytime not

7: bytes32: signature 0x00

8: uint256: revokeAlreadyEarnmoneycount 0

getTransaction1getTransaction - call

0: address: destination 0x00

1: uint256: value 0

2: string: data

3: bool: executed false

4: uint256: numConfirmations 0

5: address: send 0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C

6: string: submitTransactiontime 2023-10-23 21:59:48

7: string: executeTransactiontime not

8: bytes32: signature 0x00

9: uint256: revokeAlreadyTransactioncount 0

6.调用多签钱包的 changeViewLevel 函数改变查看等级

getearnmoney1getearnmoney - call

0: uint256: value 100

1: string: data 提出入账交易测试

2: bool: execute false

3: uint256: numConfirmations 0

4: address: sender 0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C

5: string: submitearnMoneytime 2023-10-23 22:0:12

6: string: earnMoneytime not

7: bytes32: signature 0x00

8: uint256: revokeAlreadyEarnmoneycount 0

getTransaction1

0: address: destination 0xe0520f037715bac a69dDfa2bD0E621ab707DB751

1: uint256: value 100000

2: string: data 提出出账交易测试

3: bool: executed false

4: uint256: numConfirmations 0

5: address: send 0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C

6: string: submitTransactiontime 2023-10-23 21:59:48

7: string: executeTransactiontime not

8: bytes32: signature 0x00

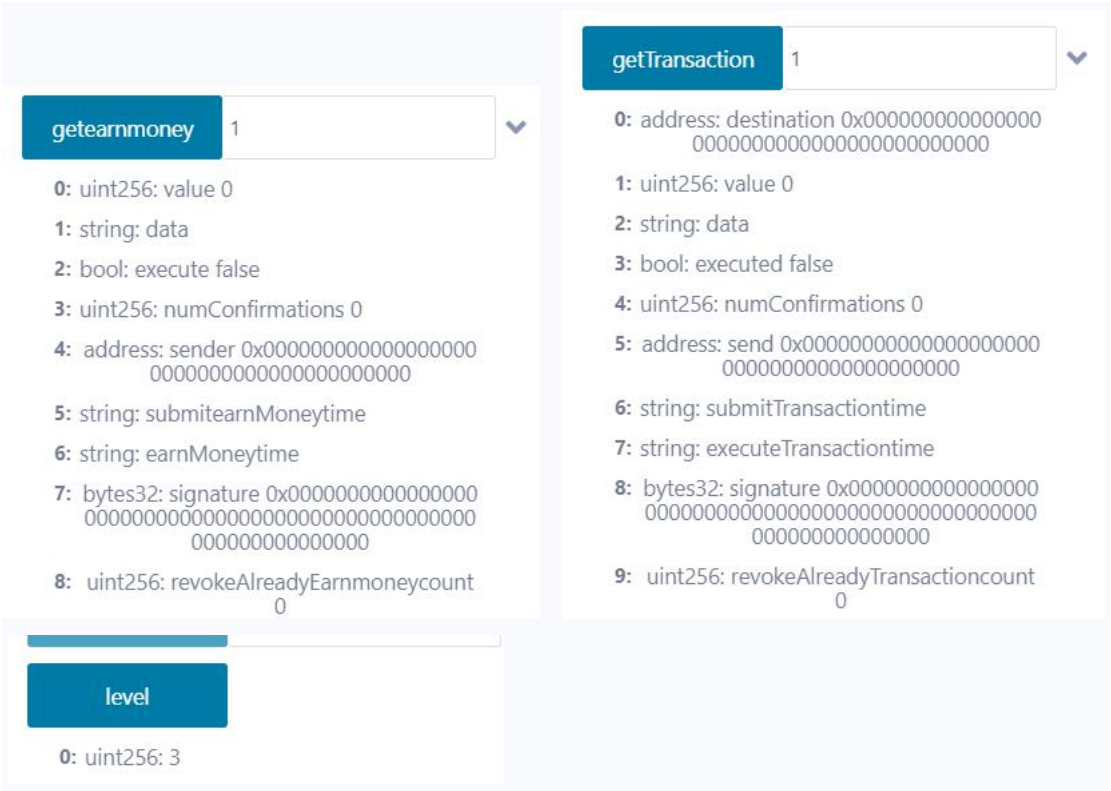
9: uint256: revokeAlreadyTransactioncount 0

level

0: uint256: 1

效果:查看信息改变,查看等级改为 1

7.调用多签钱包的 changeViewLevel 函数改变查看等级



效果: 查看信息改变,查看等级改为 3

8. 调用多签钱包的 changerequiredSignatures 函数改变所需签名数



效果:所需签名数改变为 2

9.触发多签钱包的 confirmearnMoney 函数签名入账交易

```
getearnmoney 1 getearnmoney - call ✓
0: uint256: value 100
1: string: data 提出入账交易测试
2: bool: execute false
3: uint256: numConfirmation 1
4: address: sender 0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C
5: string: submitearnMoneytime 2023-10-23 22:0:12
6: string: earnMoneytime not
7: bytes32: signature 0x0000000000000000000000000000000000000000000000000000000000000000
8: uint256: revokeAlreadyEarnmoneycount 0
```

效果:签名数增加 1

10. 触发多签钱包 earnMoney 函数试图完成交易

[illegible]

效果:出现错误

11.再次触发 confirmearnMoney 函数签名入账交易函数

getearnmoney

1

▼

0: uint256: value 100

1: string: data 提出入账交易测试

2: bool: execute false

3: uint256: numConfirmations 2

4: address: sender 0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C

5: string: submitearnMoneytime 2023-10-23 22:0:12

6: string: earnMoneytime not

7: bytes32: signature 0x00

8: uint256: revokeAlreadyEarnmoneycount 0

效果:签名数增加

12.再次触发 earnMoney 函数试图完成交易

getearnmoney1

0: uint256: value 100

1: string: data 提出入账交易测试

2: bool: execute true

3: uint256: numConfirmations 2

4: address: sender 0x19f86BBa5f7BF71c50F83cD64e976EfcA911210C

5: string: submitearnMoneytime 2023-10-23 22:0:12

6: string: earnMoneytime 2023-10-23 22:37:0

7: bytes32: signature 0x3fbf3bb7ab43d28af716a8a58cbd12f52608295609e12248131dd0a495bdd3fb

8: uint256: revokeAlreadyEarnmoneycount 0

budget

0: uint256: 10000000100

效果:完成交易,说明所需签名数已被改变

13.触发测试合约的 chageContractowner 函数改变最高所有者

owners

: 0

Calldata参数call

0: address: 0xe0520f037715baca69dDfa2bD0E621ab707DB751

效果:owners[0]被改变

第 4 章

4.1 总结

测试内容	测试结果简述	测试结论
功能性	工厂合约功能,多签钱包出入账等功能,多签钱包其他功能均正常。	与设计文档要求相符
性能	成功率在 99%以上	与设计文档要求相符