Fabirc 账户交易设计

目录

[Fabirc 账户交易设计 1](#_Toc509557932)

[结构定义及说明 3](#_Toc509557933)

[钱包 3](#_Toc509557934)

[钱包交易日志结构体定义 3](#_Toc509557935)

[交易结构体定义 4](#_Toc509557936)

[UTXO结构体定义 4](#_Toc509557937)

[交易流程 5](#_Toc509557938)

[普通交易流程 5](#_Toc509557939)

[模型论证 6](#_Toc509557940)

[交易防双花 6](#_Toc509557941)

[交易隐私性 6](#_Toc509557942)

[交易安全性 6](#_Toc509557943)

# 结构定义及说明

## 钱包

**定义**

每个机构下，都有一个属于自己机构的钱包。钱包的标识由机构的唯一mspId确定。

**字段**

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| MspId | String | 由前缀“WalletAddr”与mspId组成钱包ID |
| UTXOList | String | UTXO地址加密后列表。存储的是该机构UTXO地址加密后的值 |
| Logs | [](log数组) | 钱包日志，日之内信息使用机构自己密钥加密 |
| CryptPubKey | String | 机构私密信息加密公钥，用于己方或者他方机构给己方转账时加密UTXO地址与日志信息 |
| SignKey | String | 用于验证机构交易签名的解密密钥 |

**初始化：**

1. 钱包初始化：
   1. 在前端框架生成两对密钥：Kcrypt，Ksign。

其中Kcrypt密钥对（Kcrypt’，Kcrypt’’）用于加密本机构的私密信息，其中公钥Kcrypt’存于链上（钱包结构体中的cryptPubKey字段），以方便链上对生成的私密信息（UTXOList、Logs字段）进行加密，而私钥Kcrypt’’由机构保存在链下，用于机构对自己的私密信息进行解密。

Ksign密钥对用于对机构与智能合约之间传输信息的签名认证。机构使用私钥Ksign’’对传输信息进行加密，智能合约在接收到机构的加密信息后，使用存于链上的签名验证公钥Ksign’（钱包结构中的SignKey字段）进行解密，若解密成功，则可验证该交易信息来自正确的机构。

* 1. 根据机构mspId在链上创建钱包结构体，并存入生成的CryptPubKey和SignKey。

## 钱包交易日志结构体定义

日志信息在进行一笔交易的时候，需要在交易双方都创建一个交易日志，并且用交易双方各自的

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| TxId | String | 发起的交易的txId |
| TimeStamp | String | 时间戳 |
| RemoteAddr | String | 交易方mspId |
| Amount | Float64 | 转账金额 |
| OpeType | String | 交易类型 |
| Info | String | 交易其他信息 |
| UTXOs | []string | 交易所使用的相关UTXO地址列表 |

## 交易结构体定义

交易是由交易发起方（机构）发送至区块链上的参数，此交易信息并不实际存储在区块链上，且交易参数需要使用机构的签名私钥Ksign’’进行加密签名

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| Amount | Float64 | 转账金额 |
| Timestamp | String | 时间戳 |
| TxType | String | 交易类型 |
| UTXOAddrs | []string | 解密后UTXO列表 |
| Addr1 | String | 待用UTXO地址1 |
| Addr2 | String | 待用UTXO地址2 |

## UTXO结构体定义

UTXO在比特币中称为：未花费输出。在此处借用比特币中的概念，用于记录机构钱包中所获得的其他机构的转账。

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| Addr | String | UTXO地址（UTXO唯一ID） |
| Sign | String | UTXO地址+mspId的哈希值，用于验证UTXO的所属机构 |
| Type | String | 资产类型 |
| Amount | Float64 | UTXO上的金额 |
| hasSpent | String | 是否已被花费 |

# 交易流程

## 普通交易流程



# 模型论证

## 交易防双花

在并发交易的情况下，有可能出现以下情况：机构钱包下拥有a数量余额，用户A在该机构下进行向外转账，转账数量为b的资产，B用户同时也在该机构下进行向外转账，转账c数量资产。当A的操作还未完成，机构余额为a时，B的操作也同时进行，可能发生在余额为a的情况下，同时支付b、c，导致支付后余额d≠a-b-c。

而在此模型的交易流程中：由于在进行UTXO的花费时：

1. 花费UTXO前，检查该UTXO是否所有UTXO的余额是否足够本次支付；
2. 若两笔支付的总和大于余额，则根据鸽巢原理，必有一笔UTXO是需要被两笔交易同时修改，但是根据Fabric平台目前的限制，同时修改一笔记录会导致时间稍后的一笔交易最终失败，所以此情况下不会产生双花。

## 交易隐私性

1. 钱包内UTXO的隐私性：

各个机构内只记录的了所用的UTXO的地址列表，且各个UTXO的地址都是使用该机构公钥加密后的数据，只有本机构才能解密看到所有的UTXO地址，而在链上的所有UTXO虽然公开了改UTXO所拥有的余额，但是由于无法找到该UTXO与某个机构的对应关系，保证了财产的隐私。

1. 发送交易的隐私性：

由于发送的交易信息都是由签名密钥进行加密后发送的，所以通过交易的历史记录，只能看到加密后的信息，而无法直接获取交易信息。

**攻击：**

由于交易签名的解密密钥存储于链上，所以攻击者可以从链上获取交易参数的签名解密密钥，然后通过解密密钥解密链上各个交易信息，所以此模型在交易隐私性上还有待改进。

## 交易安全性

1. UTXO花费验证：在进行UTXO进行花费时，需要验证花费UTXO+mspId的哈希值是否与UTXO上所存储的UTXO+mspId的哈希值是否相等，验证UTXO的归属，保证UTXO不被其他机构花费。
2. 由于Fabric的特性，所以防篡改的特性可通过Fabric保证。