**MATRX 一对多交易设计方案**

**版本: V1.0**

**作者：MATRIX 基金会北京研究院**

**日期：2018.07.19**

任何情况下，与本系统及其衍生产品，以及与之相关的全部文件（包括本文件及其任何附件中的全部信息）相关的全部知识产权（包括但不限于版权、商标和技术秘密）皆属于 MATRIX 基金会所有。

本文件中的信息是保密的，且仅供用户指定的接收人内部使用。未经 MATRIX 基金会事先书面同意，本文件的任何用户不得对本产品和本文件中的信息向任何第三方（包括但不限于除用户指定接收人以外的管理人员、员工和关联公司）进行披露、出借、许可、转让、出售、分发、传播或进行与本产品和本文件相关的任何其他处置，也不得使该等第三方以任何形式使用本产品和本文件中的信息。

未经 MATRIX 基金会事先书面允许，不得为任何目的、以任何形式或任何方式对本文件进行复制、修改或分发。本文件的任何用户不得更改、移除或损害本文件所使用的任何商标。

本文件按“原样”提供，就本文件的正确性、准确性、可靠性或其他方面，MATRIX 基金会并不保证本文件的使用或使用后果。本文件中的全部信息皆可能在没有任何通知的情形下被进一步修改，MATRIX 基金会对本文件中可能出现的任何错误或不准确之处不承担任何责任。

在任何情况下，MATRIX 基金会均不对任何因使用本软件产品和本文件中的信息而引起的任何直接损失、间接损失、附带损失、特别损失或惩罚性损害赔偿（包括但不限于获得替代商品或服务、丧失使用权、数据或利润；或商业中断），责任或侵权（包括过失或其他侵权）承担任何责任，即使 MATRIX 基金会事先获知上述损失可能发生。

MATRIX 基金会保留所有权利

# 目 录

[目 录 1](#_Toc10596)

[变更记录 2](#_Toc23576)

[1 引言 3](#_Toc17888)

[1.1 文档用途 3](#_Toc7501)

[1.2 术语 3](#_Toc18792)

[1.3 使用对象 3](#_Toc7136)

[2 设计说明 4](#_Toc5712)

[2.1 一对一交易 4](#_Toc12005)

[2.1.1 需求描述 4](#_Toc29989)

[2.1.2 设计内容 4](#_Toc15197)

[2.2 一对多交易 4](#_Toc766)

[2.2.1 需求描述 4](#_Toc28078)

[2.2.2 设计内容 5](#_Toc30612)

[2.2.3 流程图 15](#_Toc4083)

[2.2.4 涉及的接口 16](#_Toc10446)

[2.2.5 相关逻辑 16](#_Toc27595)

[2.2.6 参数配置 17](#_Toc8154)

[2.2.7 其他 18](#_Toc23431)

# 变更记录

| 修订日期 | 设计内容 | 作者 | 评审者 | 评审日期 |
| --- | --- | --- | --- | --- |
| 2018-07-19 | 一对多交易 | 叶营 |  |  |
|  |  |  |  |  |

# 引言

## 文档用途

本文档针对一对多交易模块进行的详细设计，主要设计内容包括：

1. 添加1对多交易，以太坊需要能够验证该交易
2. 1对1交易与以太坊结构兼容
3. 添加交易类型，默认交易类型为以太坊交易。

## 术语

| **术语** | **备注** |
| --- | --- |
|  |  |

## 使用对象

阅读对象主要针对熟悉相关业务、技术和应用知识的人员，包括：研究院全体同事

# 设计说明

## 一对一交易

### 需求描述

一对一交易主要延用以太坊的交易，但是为了防止重放攻击需要修改nonce值，具体修改办法在一对多中会有描述，因为此处的nonce和一对多中的nonce是一样的逻辑。

### 设计内容

详见一对多交易。

## 一对多交易

### 需求描述

在一对一转账的基础上增加一对多交易，即一笔交易可以实现同时给多个人转账，输入包括每个人的地址、转账金额、备注信息。然后将nonce和V值依据转账类型修改成我们自己的格式。按照新的交易格式做交易的hash和签名，并在入池之前进行验证，然后将一对多的交易进行打包并执行。在执行时需要将一对多拆分成独立的交易依次执行一旦某个交易执行失败则表示全部交易执行失败。需要注意的是一对多交易的gas应该是所有的交易gas之和暂不考虑gas折扣情况

一对多转账举例：

转账格式：实现了同时给三个账户转账

eth.sendTransaction({"from": "0x3f52f501f18170a2ff7c5038456ac53024a20b6b","to": "0x4f30afdab0ba93c35d72ac1a767bf11e28fb38f1","value": web3.toWei(0.01),"txType":1,"lockTime":77777,"extra\_to":[{"to": "0x5587bdf512061531460f2df6e425ef6f92c547fa","value":web3.toWei(0.01)},{"to": "0xbcb11482d86f477e068797b22d04f04abe8b8525","value":web3.toWei(0.01)}]})

关于Nonce的描述：

1. 创建交易时如果交易中自带nonce需要验证Nonce是否符合我们规则符合继续，不符合则丢弃该交易
2. 创建交易时如果交易中没有nonce需要去stateDB中获取nonce这时候修改返回的0值为0|0x8000000000000000
3. 广播交易时需要将nonce & 0x7FFFFFFFFFFFFFFF
4. 收到广播交易时需要将Nonce | 0x8000000000000000
5. 验证Nonce是验证其是否大于0x8000000000000000， 如果大于则说明是Matrix交易否则就是其他交易直接丢弃

### 设计内容

修改结构体SendTxArgs，添加三个成员添加后如下所示，标红部分为新增加的成员（原始成员未列出）

type SendTxArgs struct {

TxType Byte `json:"txType"` //交易类型

LockHeight Uint64 `json:"lockHeight"` //区块高度（区间定时）

ExtraTo [ ]\*extraTo\_mx json:"extra\_to"` //扩展的交易

}

如上还需要增加一个结构体用来存放扩展的交易

Type extraTo\_mx struct{

To \*common.Address `json:"to"` //转账地址

Value \*hexutil.Big `json:"value"` //转账金额

Input \*hexutil.Bytes `json:"input"` //输入信息（可以理解为备注信息，主要用于合约）

}

修改交易结构体

type txdata struct {

Extra []Matrix\_Extra ` rlp:"tail"` //扩展的交易（如果是一对一交易则该值为空如果是一对多交易则该值会有一组数据）

}

交易相关的新增的结构体

Type Tx\_to struct{

//转账接受人

Recipient \*common.Address `json:"to" rlp:"nil"` // nil means contract creation

//转账金额

Amount \*big.Int `json:"value" gencodec:"required"`

//输入信息

Payload []byte `json:"input" gencodec:"required"`

}

Type Matrix\_Extra struct{

//交易类型 0为一对一 1为一对多

TxType Byte `json:"txType" gencodec:"required"`

//区块高度

LockHeight Uint32 `json:"lockHeight" gencodec:"required"`

//扩展交易

ExtraTo []Tx\_to `json:"extra\_to" gencodec:"required"`

}

type ExtraTo\_tr struct{

//交易接收人

To\_tr \*common.Address `json:"to"`

//交易金额

Value\_tr \*hexutil.Big `json:"value"`

//输入信息

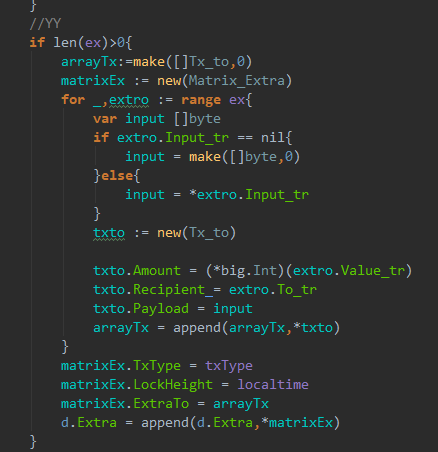
Input\_tr \*hexutil.Bytes `json:"input"`

}

修改stateDB中getnonce方法返回0的地方和所有获取Nonce是返回0的地方，改为“0 | 0x8000000000000000”但是如果交易中自带Nonce则不需要修改,只需要验证是否符合我们的格式如果不符合则丢弃。修改Nonce时需要特别仔细因为涉及的地方会有很多，改起来较为繁琐。

将收到的转账数据转换为交易的结构体，这时候需要根据

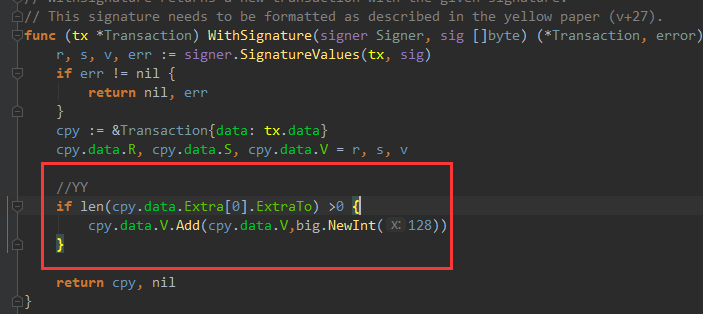
txtype==0 && localheight==0&& len(Extra)>0来判断需要走哪个方法，也就是走原来以太坊一对一的逻辑，否则 txdata.Extra 需要添加内容。原来以太坊调用的是toTransaction()方法，该方法在最后调用NewTransaction（），而我们需要重新写方法实现这个流程。（为了方便区分下面的方法都是以一对多为原则）重写对外方法NewTransactions，然后其中调用重写的新方法newTransactions,与原来方法比较多了如下图所示的的内容



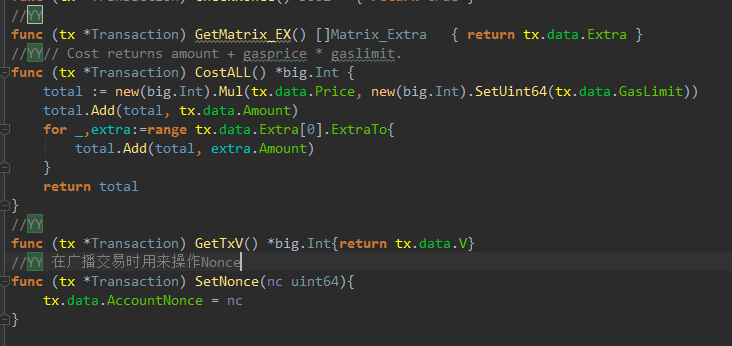
对交易数据进行签名，在调用Hash()方法时需要判断如果是一对多交易则把扩展交易也进行签名。如下图所示



修改交易中的V值，如果是一对多则需要给V值加上128，如下图所示

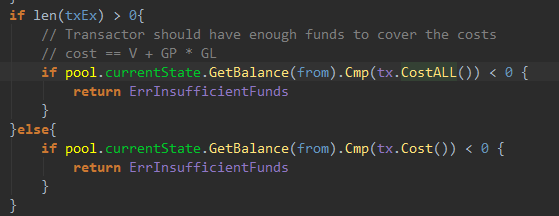


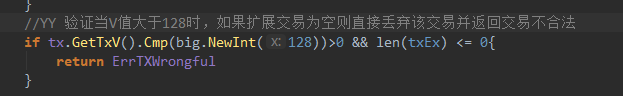
在将交易添加到交易池之前需要验证交易，包括金额是否小于0，账户余额是否充足等。在验证过程中首先需要在transaction.go中新增四个方法如下图所示



然后在pool.validateTx（）修改如下代码

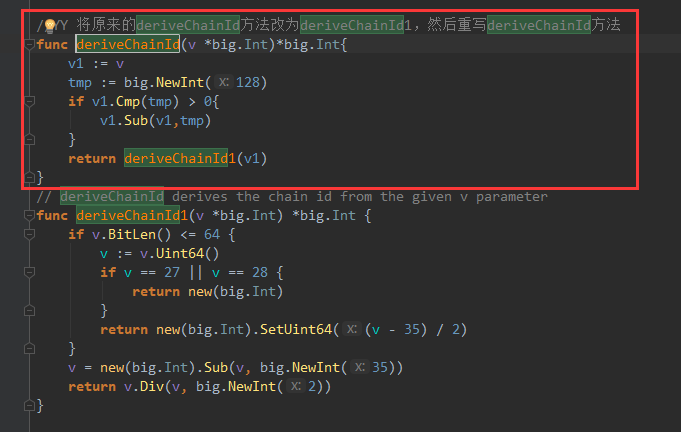




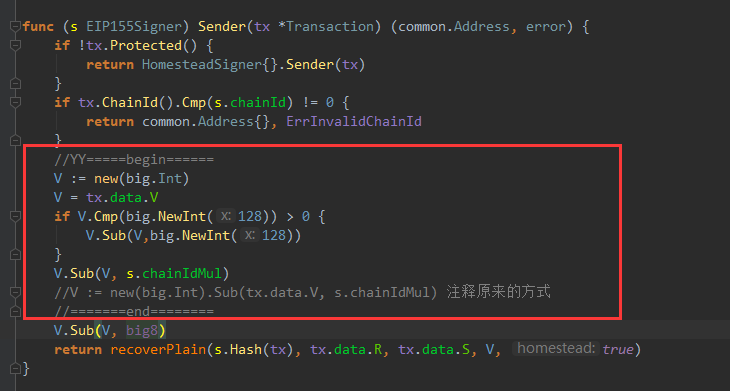


需要注意的是pool.validateTx（）验证时会调用types.Sender（）方法，该方法再调用signer.Sender（）方法，后面这个Sender方法是一个结构体的成员，调用的是transaction\_signing.go中的func (s EIP155Signer) Sender方法，在该方法中添加判断，如果是一对多交易需要将V值减去128之后再使用，同时子交易总数量不能超过1000.

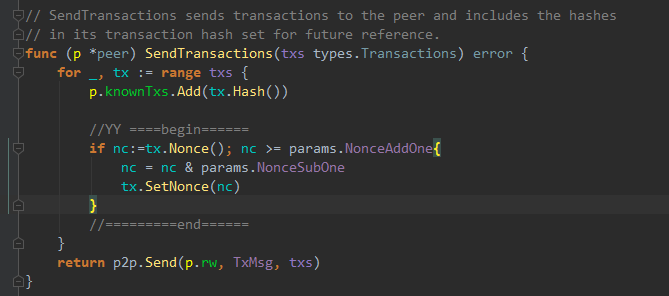
具体的去掉v 值的方式：重命名deriveChainId方法为deriveChainId1，然后修改deriveChainId方法如下



因为不能直接操作交易中的V值，所以还需要添加下面的内容

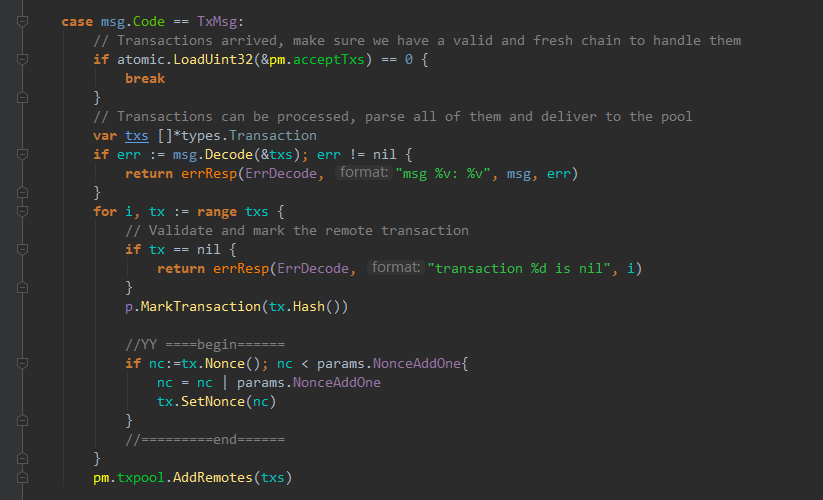


验证通过后会把交易添加到交易池并广播交易在广播时需要修改交易的Nonce如下图



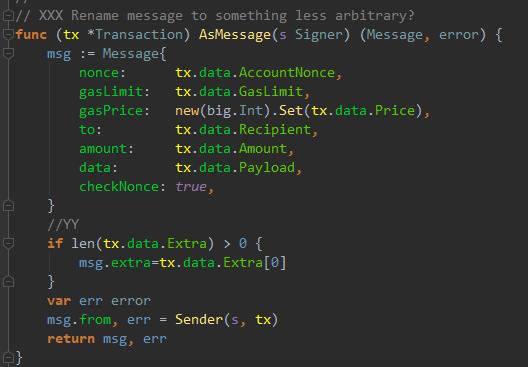
至此交易从产生到入池在到广播的过程就结束了，标志着这条交易可以被打包执行了，下面将开始打包执行的操作。

首先是接受广播的交易，接收的过程是在eth/handler.go文件中的handleMsg（）方法，如下图



接下来修改Message结构体，在结构体中添加一个成员Extra Matrix\_Extra //扩展交易

然后AsMessage方法修改如下

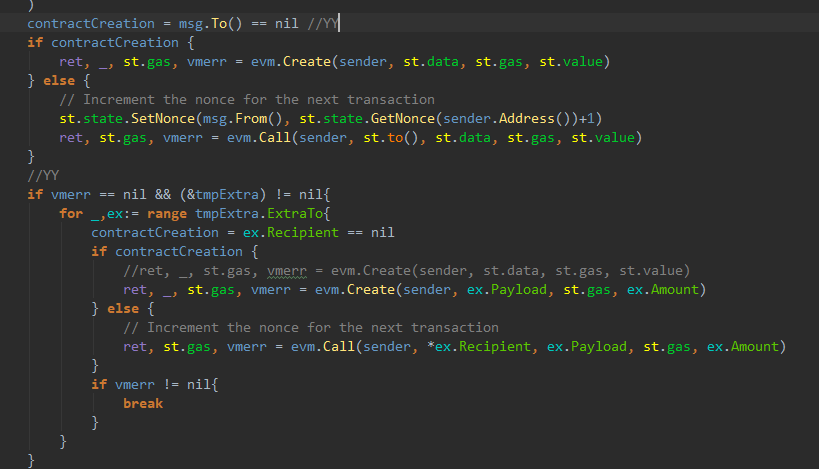


添加Message的扩展方法：func (m Message) Extra() Matrix\_Extra { return m.extra } 并在state\_transcation.go中的Message的接口中添加该方法

TransitionDb方法，再扣除gas之前计算总共需要的gas（不考虑折扣）添加代码如下



修改调用call方法的地方，修改为循环调用call方法执行交易，添加代码如下



### 流程图

具体流程图如下所示：



### 涉及的接口

无

### 相关逻辑

无

### 参数配置

在此只列出新增加的结构体和新增加的成员

type ExtraTo\_Mx struct{

To2 \*common.Address `json:"to"`

Value2 \*hexutil.Big `json:"value"`

Input2 \*hexutil.Bytes `json:"input"`

}

type SendTxArgs struct {

TxType byte `json:"txType"`

LockHeight uint64 `json:"lockHeight"`

ExtraTo []\*ExtraTo\_Mx `json:"extra\_to"`

}

type Tx\_to struct{

Recipient \*common.Address `json:"to" rlp:"nil"` // nil means contract creation

Amount \*big.Int `json:"value" gencodec:"required"`

Payload []byte `json:"input" gencodec:"required"`

}

type Matrix\_Extra struct{

TxType byte `json:"txType" gencodec:"required"`

LockHeight uint64 `json:"lockHeight" gencodec:"required"`

ExtraTo []Tx\_to `json:"extra\_to" gencodec:"required"`

}

type txdata struct {

Extra []Matrix\_Extra ` rlp:"tail"`

}

### 其他

应该考虑交易的区间定时功能，有相关的逻辑和流程需要一并考虑。