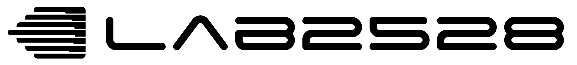
2017-04

|  |
| --- |
| 基于Onetime-address和环签名的隐私交易方案  Version 0.8.0 |



**目录**

[基于Onetime-address和环签名的隐私交易方案 1](#_Toc479012320)

[1 用户账户系统 3](#_Toc479012321)

[1.1 账户系统构成 3](#_Toc479012322)

[1.2 用户密钥的生成算法 3](#_Toc479012323)

[1.2.1 主账户生成 4](#_Toc479012324)

[1.2.2 子账户Onetime-address生成 4](#_Toc479012325)

[1.2.3 子账户的验证和对应私钥 4](#_Toc479012326)

[1.2.4 监管和view-key 5](#_Toc479012327)

[1.2.5 交易验证者Validator 5](#_Toc479012328)

[2 环签名方案 6](#_Toc479012329)

[2.1 环签名简介 6](#_Toc479012330)

[2.2 环签名方案 6](#_Toc479012331)

[3 交易协议 8](#_Toc479012332)

[3.1 交易场景 8](#_Toc479012333)

[3.2 交易发起 8](#_Toc479012334)

[3.3 交易验证 9](#_Toc479012335)

[3.4 交易确认 9](#_Toc479012336)

[3.5 交易监管 10](#_Toc479012337)

[4 隐私效果及创新点 10](#_Toc479012338)

[4.1 隐私效果 10](#_Toc479012339)

[4.2 创新点 10](#_Toc479012340)

[5 下一步工作 11](#_Toc479012341)

# 用户账户系统

## 账户系统构成

账户体系是整个隐私交易的基础，我们采用Onetime-address，每个用户拥有唯一主账户和多个子账户，子账户也可以看作智能合约中的账户，而子账户通常不是用户自己产生，而是由交易对方为用户生成，如下图所示。



Figure ‑ 用户账户系统

Alice拥有唯一主账户公钥，以及若干子账户，，，…。Bob为Alice转账时，通过主账户公钥生成了和，Carol为Alice转账时，生成了。可以看到为Alice转账的每一笔交易都会为Alice生成一个子账号，即是Onetime-address，而只有Alice拥有这些子账号的管理和使用权限。

## 用户密钥的生成算法

是椭圆曲线的阶子群，是的生成元。通常选取比特币或以太坊的曲线。

### 主账户生成

Alice产生两个随机数，以作为Alice的私钥，并计算和，以作为 Alice公钥。最终Alice拥有spend key ，而公开作为Alice的主账户地址。

### 子账户Onetime-address生成

当Bob要为Alice转账时，需要使用Alice的主账户为Alice生成子账号。

Bob产生随机数，计算

，

这里\_p是将椭圆曲线的点映射到的函数。

则Alice生成子账号为

(。

由于是随机数，可以知道(是Onetime-address，随机数并没有公开，只是公开了含有信息的，而并不能计算出，由椭圆曲线离散对数问题保证。是子账户的公钥分量，是随机因素分量。

Alice的每一个子账户都是基于主账户的随机账户。

### 子账户的验证和对应私钥

Alice对链中所有Onetime-address做scan，取链中的分量和自己私钥参与计算

若，则可以确定(是自己的子账号；

若，则继续scan。

这是由于，于是

由于仅有Alice知道私钥，则只有Alice能验证属于自己的子账号(。

Alice需要花子账号(的资产，需要计算(对应的私钥，

。

于是有子账户的公钥分量。对于每个子账户，Alice都会计算其对应的私钥。

### 监管和view-key

由上述用户系统可以知，除Alice和Bob以外，并无第三方知道Alice的子账号(。

监管的功能应满足如下2点：

1. 监管方可以知道(属于Alice；
2. 监管方不能消费(的资产。

于是定义用户view-key，并以安全方式传送给监管方。则：

1. 监管方知道和，可以计算，并验证若，则(属于Alice；
2. 监管方不知道，无法计算私钥，所以不能消费(的资产。

所以view-key能够满足监管的功能。

### 交易验证者Validator

Validator负责验证每笔交易并写入区块链中。所有Validator共用一套公私钥对, 将全网公布，而将安全保存在本地。

# 环签名方案

## 环签名简介

2001年，Rivest等人在如何匿名泄露秘密的背景下提出了一种新型签名技术，成为环签名（Ring Signature）。环签名可以被视为一种特殊的群签名（Group Signature），由于群签名需要可信中心和安全的建立过程，往往在匿名保护上存在漏洞（签名者对于可信中心是可追溯的），而环签名在群签名基础上去除了可信中心和安全建立过程，对于验证者来说，签名者是完全匿名的，所以环签名更具实用价值。

自环签名提出后，大量学者发现其重要的价值，基于椭圆曲线、门限等多种环签名被大量设计开发，总体概括可分为四类：

1. 门限环签名
2. 关联环签名
3. 可撤销匿名性环签名
4. 可否认环签名

由于本文档中设计目标为区块链上智能合约的代币隐私性，我们设计一种关联环签名，以实现隐私性的同时防止双花问题。

## 环签名方案

环签名可分为四个部分：**GEN，SIG，VER，LNK**。

**GEN**：采集公共参数，以及在world state中随机选取个地址；对用户公私钥对，，为点的阶，生成公钥镜像。

**SIG**：针对所需签名消息，利用公钥地址集，其中为用户真实公钥，计算输出签名。

**VER**：基于消息，公钥地址集和签名，验证签名合法性，输出“True”或“False”。

**LNK**：利用集合，判断签名是否已使用。

具体过程介绍如下：

**GEN**：签名者选取其所需使用的Onetime-address的公私钥对，有，计算，其中为hash函数，输出椭圆曲线上一个随机的点。在world state中随机选取个地址合同构成公钥地址集，其中。

**SIG**：签名者选择随机数和，做如下计算：

计算：

，为hash函数

计算：

最后生成签名：

**VER**：验证者验证签名时，基于消息、公共参数和、，计算：

然后验证等式是否成立，若成立，则签名有效，执行LNK。

说明：验证过程发现等式成立即需要，，以为例说明如下：

时，有，显然成立，而时：，综上有，同理成立，则说明VER过程是正确的。

**LNK**：对于区块链上已经出现过的所有构建集合，若当前在集合中出现，则说明该地址金额已经使用，认为签名交易不合法；若没有出现过，则认为签名交易合法，并将加入集合。

# 交易协议

## 交易场景

用户1主账户私钥为，主账户公钥为；用户1主账户私钥为，主账户公钥为。用户1在智能合约SC中账户为Onetime-address1，想要从这个账户中为用户2转移Value的代币。此为交易的场景。

## 交易发起

用户1要发起这笔交易，需先构造一个合法的交易，然后通过P2P网络将这笔交易传播出去。一笔交易包含四个字段：Trans From，Trans To，Data，For Validator。构造过程如下：

1. 从区块链历史交易中抓取个账户地址，包括自己账户地址在内，构成一个含有个账户的账户集合，构成整笔交易字段Trans From。这是为环签名做准备，隐藏交易发起者身份。
2. 交易字段Trans To为调用的智能合约SC的地址。
3. 构造交易字段Data过程为：首先为用户2构造Onetime-address2（构造过程见1.2.2）；其次，构造智能合约中代币的交易，交易输入账户SC\_TransFrom为用户1控制的Onetime-address1，交易目标地址SC\_TransTo为刚为用户2构造的Onetime-address 2，交易代币值SC\_Value为Value，并且用户1为这笔交易进行签名SC\_Sig，它与Onetime-address1匹配；再次，生成加密密钥, 用它将SC\_TransFrom、SC\_TransTo、SC\_Value、SC\_Sig加密，最终形成交易的Data字段。
4. 交易的For Validator字段的构造：首先使用Validator的公钥对进行加密；其次，使用环签名方案，对交易的Trans From、Trans To、Data以及加密的进行环签名，得到签名Ring Signature。将加密的和Ring Signature作为For Validator字段。

## 交易验证

Validator在拿到一笔交易后，进行如下验证以及计算：

1. 验证交易中Ring Signature是否与Trans From的账户集合匹配，如果验证通过，则表明交易合法，否则拒绝。
2. 使用私钥对加密的进行解密，得到。
3. 使用对Data字段进行解密，获得明文。
4. 以解密后的Data字段作为输入，执行Trans To中地址对应的智能合约。智能合约会验证 SC\_Sig是否与SC\_TransFrom匹配，如果匹配则在SC\_TransFrom账户中减掉SC\_Value的代币，同时创建SC\_TransTo账户，并添加金额SC\_Value。

## 交易确认

用户2在需要确认用户1的代币转移是否成功，过程如下：

1. 对智能合约维护的Onetime-address进行Scan，过程为1.2.3所示。
2. 计算得到用户1为自己生成的Onetime-address2对应的私钥，过程为1.2.3所示。

至此，用户2已经清楚转移代币是否成功，以及可以对新的Onetime-address操作。

## 交易监管

监管者有所有被监管成员的view key。监管过程如下：

1. 先用用户1的view key ，生成加密密钥。然后利用对区块链上Data字段进行解密，如果解密成功，则为用户1发送的交易。解密出的SC\_TransFrom即为用户1在智能合约中控制的账户。
2. 用其他用户view key进行试计算，计算过程为1.2.3所示，如果可以匹配，那么就是用户1转移代币的目标用户。
3. 转移代币的量即为SC\_Value。

通过上述方式，监管者就可以看到监管对象之间的每一笔代币交易以及代币数量。

# 隐私效果及创新点

## 隐私效果

基于Onetime-address和环签名的隐私交易方案能够做到一下隐私效果：

1. 利用环签名方案保证交易发送者匿名全网匿名。
2. 利用Onetime-address保证智能合约代币账户与主账户隔离。
3. 由于上述两条性质的存在，智能合约中账户币值的隐藏并非必须。

## 创新点

基于Onetime-address和环签名的隐私交易方案有如下创新点：

1. 全网匿名。

任何隐私方案，都要同时兼顾隐私性、交易正常验证和智能合约正常执行。在此前的方案中，为保证这一点，需要给Validator很大的权限，匿名性是针对普通用户而言，对Validator而言并匿名，因为它可以知道交易的双方以及交易值。而这套方案中，即使是交易者，也并不能拿到任何关于交易双方的信息。因此，Validator的选择就灵活了很多，可以适应不同的应用场景。

1. 可控监管

在众多应用场景中，监管者并不是唯一的，而且很有可能不同监管者有不同的监管用户。之前的监管方案中，如果用户A被监管，其他用户与A的交易信息均会被A的监管者看到，即使A的监管者并没有监管这些用户的权限。而这套方案中，监管者对于自己监管对象之间的交易能够掌握完全的信息，对于自己监管对象与非监管对象之间的交易，仅能看到交易数额，并不能够得到非监管对象的身份信息。这种可控的机制，使得监管设计更加游刃有余，适应复杂应用场景。

1. 简捷的密钥管理

在这套方案中，用户密钥只生成一次，即主账户的公私钥。只需要安全保存主账户的私钥，不需要维护智能合约中代币账户的私钥。

# 下一步工作

1. 根据应用场景要求，研究并设计更加高效的环签名方案，在安全和效率之间达到最优效果。
2. 研究佩德森承诺技术（Pedersen Commitment），思考交易值Value的隐私性问题。
3. 精化View Key的设计，争取更加便捷可控。