# Zcash技术分析

 Zcash是bitcoin的分支，保留了bitcoin原有的模式，基于比特币0.11.2版代码修改的。Zcash钱包资金分2种：透明资金、私有资金，透明资金类似比特币资金；私有资金加强了隐私性，涉及到私有资金的交易是保密不可查的，透明资金与透明资金的交易是公开可查的。

         Zcash被称为零币，与‘灵’同音，不好听，且有归零的感觉，统一称为Z币。Zcash主链的币叫ZEC，等同于BTC，测试链的币称为TAZ。

         目前zcash仅支持linux系统（推荐ubuntu），不支持Windows、MAC，且只支持命令行，没有图形界面（zcash团队没人擅长GUI）。

**地址**

Zcash中含有2种地址，实现bitcoin和zcash的兼容。

1、  透明资金地址taddr

类bitcoin地址，保存在UTXOs中，长度为34位。

命令./src/zcash-cli getnewaddress可以生成新的taddr地址。

例如：msFgKsVAia4aFB1qFGX7iJ3XjMZv6c26Ab

2、  私有资金地址zaddr

保存在notes对象中，长度为96位，前2位为标记，主链地址前2位是”zc”，spendingkey的前2位是”SK”，测试链的地址前2位是”tn”。

notes对象是加密的，解密才能查看，用的是chacha20-poly1305算法。

命令./src/zcash-cli z\_getnewaddress可以生成新的zaddr地址。

例如：tngBSug9xvpqudziMc3QbfReHG6BeufHDbv4vEZDmnYqSpRbqu5USZqBTXENQGdUbYMkxqHbb9j3RmEr7Lza8q7hbTQxZnM

具体生成过程看z\_getnewaddress函数。

命令z\_listaddresses获取节点钱包中的zaddr列表。

命令./src/zcash-cli zcrawkeygen可以查看地址、key等。

地址taddr实现了原有的bitcoin的功能，zaddr实现了zcash的功能。

taddr可以发送交易给taddr、zaddr，zaddr可以发送交易给taddr、zaddr。taddr与taddr的交易是透明可查的，涉及到zddr的交易就保密了。

**零知识证明**

零知识证明是在不泄漏信息的情况下，生成证明，验证者通过验证证明确定是否正确。

Zcash使用了SCIPR实验室的零知识证明库libsnark，在此基础上做了修改。Zcash团队成员包含多个libsnark开发人员（不是全部）。修改如下：

1、  libsnark支持4种语言："R1CS"、"BACS"、"USCS"、"TBCS"，zcash使用了"R1CS"。

2、  libsnark支持3种椭圆曲线："edwards"、"bn128"、"alt\_bn128"，zcash使用了"alt\_bn128"。

3、  libsnark支持2个库：gadgetlib1、gadgetlib2，zcash使用了gadgetlib1，gadgetlib1是底层库，支持所有的R1CS预处理特征。

SCIPR实验室的libsnark：https://github.com/scipr-lab/libsnark。

Zcash修改后的libsnark：https://github.com/zcash/libsnark。

当zcash交易涉及到私有资金时，使用零知识证明生成证明（函数r1cs\_ppzksnark\_prover），保存到类JSDescription的proof中，校验交易时校验证明（函数r1cs\_ppzksnark\_verifier\_strong\_IC）。

libsnark看了看，知道怎么用的，但详细的也没看懂。

零知识证明白皮书：https://eprint.iacr.org/2013/507.pdf

**交易**

透明资金发送到透明资金地址，可以调用bitcoin的RPC命令发送交易，涉及到私有资金、地址，只能用zcash的RPC命令了，目前是z\_sendmany命令。

格式：

z\_sendmany "fromaddress" [{"address":... ,"amount":...},...]

         交易分3种情况：

1、    透明资金发送到透明地址

延用bitcoin的交易方式，仅仅打签名即可。

2、    透明资金发送到透明地址、私有地址

透明资金发送到透明地址，依然延用bitcoin的方式，发送到私有地址的需要构造证明，对整个交易签名发送。

3、    私有资金发送到透明地址、私有地址

每笔交易需要生成证明，涉及到找零、矿工费的处理，最后再签名发送。

       交易中的透明地址、私有地址的输入、输出是不同的结构，涉及到私有地址的采用了新增的结构（JSInput、JSOutput），涉及到的私有地址的数据需要先签名解密解析数据。

详细过程参考函数z\_sendmany：

https://github.com/zcash/zcash/blob/master/src/wallet/asyncrpcoperation\_sendmany.cpp#L126

         生成证明参考函数perform\_joinsplit：

https://github.com/zcash/zcash/blob/master/src/wallet/asyncrpcoperation\_sendmany.cpp#L804

**交易签名**

         Zcash私有资金交易时，不仅仅依靠零知识证明，而且对交易打签名，采用了ed25519算法。

zcash交易的零知识证明、签名需要的公钥、密钥使用ed25519算法生成的，详细参考函数crypto\_sign\_keypair。

签名参考函数sign\_send\_raw\_transaction。

https://github.com/zcash/zcash/blob/master/src/wallet/asyncrpcoperation\_sendmany.cpp#L660

**挖矿算法**

         Zcash的挖矿算法是EquiHash，理论依据是广义生日悖论，目前采用的基本算法BasicSolve，理论上存在最优算法OptimisedSolve，正在实现。

         EquiHash算法需要高内存，最优算法每个挖矿线程需要1G内存，短时间内很难出现低成本的矿机。

         Zcash使用EquiHash计算解决方案，保存到区块头的nSolution，接收区块时校验。

BasicSolve算法：

https://github.com/zcash/zcash/blob/master/src/crypto/equihash.cpp#L323

**块奖励**

zcash总量是2100万个币，每隔2.5分钟出一个块，每天576个块。

         区块奖励分2个阶段：

1、    慢慢开始挖矿阶段

为了照顾在开始阶段不懂挖矿的，zcash采用了慢慢开始挖矿机制，前20000个块的奖励比较少，线程递增到12.5个币。

慢慢开始挖矿阶段也分2个阶段：

(1)       前10000个块，

块1的奖励是0.000625币，随着高度线性递增，增加幅度是0.000625币，块9999的奖励是6.249375币。

(2)       块10000～19999

块10000的奖励是6.250625，随着高度线性递增，增加幅度是0.000625币，块19999的奖励是12.5币。

         前20000个块的奖励总量是125000币，相当于10000个块奖励是12.5个币的总量。产生10000个块需要的时间大约是17.3天，缓慢挖矿的时间周期（20000个块）大约是34.7天。

2、    正常块奖励

从块20000开始，奖励是12.5个币，奖励按照每840000个块（每4年）减半，计算块奖励时块高度减少10000，即940000块是第一次减半，每隔840000个块减半，以此类推。

         不懂挖矿的矿工也不必着急，因为即使懂了，前20000个块的奖励很少，有1个月的时间搞定挖矿。

**创始人奖励**

前4年（区块0～840000）产生的区块奖励的20%归zcash公司（总量的10%），80%归矿工。4年后的区块奖励全部归矿工。

创建区块时，构造2个输出，分别给矿工、创始人，当广播区块后，其他节点接受区块时，检验区块有效性，当高度小于840000时，检验是否有对创始人的20%的输出。

创始人奖励地址采用2/3多重签名FOUNDERS\_REWARD\_SCRIPT。

**难度调整**

每产生一个区块都需要重新计算难度，难度计算基于前17个块（42.5分钟），前17个区块的难度是固定的（powLimit），从第18个区块开始调整难度，最大值是powLimit。

详细计算过程参考GetNextWorkRequired函数。

Bitcoin是每隔2016个块调整难度，关于zcash的每个块都要重新调整难度，有个讨论，参考：https://github.com/zcash/zcash/issues/147

**CoinBase保护**

矿工挖矿的奖励不能直接发送给透明资金地址，只能发送给私有资金地址，然后才可以发送给透明资金地址。

币成熟期是100。

**创建新块**

         Zcash每个挖矿线程都有自己的地址、计数器，每个线程挖出来的新块的块奖励的地址是不同的，做矿池的需要注意，改成一个地址。

         参考函数BitcoinMiner（https://github.com/zcash/zcash/blob/master/src/miner.cpp#L442）。

**区块大小是2M。**

**挖矿工具**

         Zcash目前仅支持CPU挖矿，GPU的挖矿工具在研发中，尚未有人公开。

zcash支持2种挖矿方式：solo、矿池。

1、    Solo挖矿

在zcash.conf设置gen=1，启动zcashd即可挖矿。

2、    矿池

Zcash开发者str4d做了个矿池和挖矿工具，矿池地址：http://zmine.io/，挖矿工具地址：https://github.com/str4d/zcash/tree/standalone-miner。

                  zcash-miner是zcash的分支，主要添加了stratum协议、cpu挖矿。

(1)       stratum协议

采用了cpp-ethereum的stratum的实现，V2版，做了些修改，移除了eth协议中的命令。

参考commit：https://github.com/str4d/zcash/commit/c5ea2cfb190944dfa23566561d1cb37dcb116833

(2)       cpu挖矿

把equihash算法做了些修改，接受矿池下发的难度来计算。

参考文件standaloneminer.cpp。

https://github.com/str4d/zcash/blob/standalone-miner/src/standaloneminer.cpp

**参数文件**

         Zcash有2个参数文件，包含了证明、验证key，分别是：z9-proving.key、z9-verifying.key，在~/.zcash-params目录下。z9-proving.key大概900M，z9-verifying.key大约14K。使用./zcutil/fetch-params.sh可以下载这2个文件。

         程序启动初始化时（AppInit2），读取这2个文件中的证明、验证key，生成证明时需要证明key，验证证明时需要验证key。

**API和RPC**

         Zcash保留了bitcoin的API、RPC，涉及到透明资金、地址的可以使用bitcoin的接口，涉及到私有资金、地址的只能用zcash新增的接口。

         Zcash新增的接口包括查询余额、生成地址、发送交易等，如下：

* Accounting: z\_getbalance, z\_gettotalbalance
* Addresses : z\_getnewaddress, z\_listaddresses
* Keys : z\_exportkey, z\_importkey, z\_exportwallet, z\_importwallet
* Operation: z\_getoperationresult, z\_getoperationstatus, z\_listoperationids
* Payment : z\_listreceivedbyaddress, z\_sendmany

详细信息参考文档https://github.com/zcash/zcash/blob/master/doc/payment-api.md。

**挖矿硬件**

EquiHash白皮书讲了2点：

1、  GPU比CPU快4倍，

2、  ASIC是否可能实现

详细信息参考白皮书第10页。

https://www.internetsociety.org/sites/default/files/blogs-media/equihash-asymmetric-proof-of-work-based-generalized-birthday-problem.pdf

**加密库**

         Zcash使用了开源加密库libsodium（https://github.com/jedisct1/libsodium），zcash使用了其中的chacha20-poly1305、blake2b、ed25519算法。

chacha20-poly1305算法：

https://github.com/jedisct1/libsodium/blob/master/src/libsodium/include/sodium/crypto\_aead\_chacha20poly1305.h

blake2b算法：

https://github.com/jedisct1/libsodium/blob/master/src/libsodium/include/sodium/crypto\_generichash\_blake2b.h

ed25519算法：

https://github.com/jedisct1/libsodium/blob/master/src/libsodium/include/sodium/crypto\_sign.h

**保护隐私技术比较**

         目前市场上有3种保护隐私的币，分别是达世币DASH、门罗币XMR、Z币Zcash。

1、    达世币（暗黑币改名叫达世币）

DASH采用的混币CoinJoin的方案，把一些交易混合在一起，增加追踪的难度。

CoinJoin参考DASH币白皮书http://www.8btc.com/dash-whitepaper

2、    门罗币XMR

XMR采用了环签名，签名者利用自己的私钥以及环成员的公钥进行签名，验证者只能确定该签名来自环中的某个成员，但无法确定签名者的真实身份。

参考文档http://www.doc88.com/p-1731949359328.html

3、    Z币

Zcash的交易采用了零知识证明，隐藏交易信息。

         这3种技术相比较来看，zcash的隐私保护性更好。

**风险**

1、  zcash目前是由公司开发的，公司具有绝对的话语权，后续公司会成立基金会，zcash的开发交给基金会推进。如果zcash社区与公司、基金会出现意见分歧，而代码是开源的，可以硬分叉。

比如：有人对创始人奖励不满意，已经提议去掉创始人奖励。

2、  zcash的顾问包括：前任比特币核心开发人员Gavin、eth创始人Vitalik Buterin，zcash的发展会受到这2个人的影响。

比如：区块大小2M源于bitcoin classic。

参考Github: https://github.com/zcash/zcash/issues/765

**注：**

         zcash的有些技术点，我也一知半解，如果有不对的地方请指出来，对zcash技术有兴趣的可以加入QQ群**377078828**一起讨论。

作者：**龙少**

微博：**007龙少**

# 匿名数字货币——Zcash

比特币有多隐私呢？ 一方面，他是完全匿名的。另一方面，他是完全透明可追踪的。我们理解的比特币网络的匿名性是指用户能持有一个比特币地址而不公开任何身份信息。到这里为止，看起来不错，但是，在比特币世界里发生的所有事情都是可追踪的，每一个基于比特币的交易都保存在区块链里。但毕竟，如果你的名字不是精确的和比特币地址联系起来，那么人们无法知道你进行了什么交易，是么？错。很多数据挖掘的大机构已经很擅长从大的网络动作中推理信息：谷歌，facebook，linkedin，只要积累下大量数据，就能用他们的模型去提炼尽可能多的信息。国际刑警根据比特币交易流水追踪到暗网人员。直接把全球最大的暗网silkroad一锅断掉。这更加剧了人们对比特币匿名性的怀疑。比特币不能提供强大的隐私保证，付款交易记录在公共分散账本中，从中可以得到很多信息。

有没有办法能既实现区块链的去中心化，同时保护隐私，隐匿交易信息呢？

混币器就是一种匿名交易的解决方案。通俗点说，区块链上有多个搅拌器，要转账的每个币都要经过搅拌器和许多别的币混合，这样，从混币器出来后，除了混币器节点，别人谁都不晓得这个币来自哪里。用户把一组币委托给混合器，然后从混合器中取出等价值的币。达世币（Dash）就是采用了混币器方案。

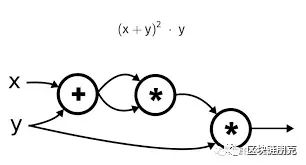


但这种方案有几个明显的缺点：

1. 要求参与混币的人员在线进行混币。如果双方就混币的数额不能达成一致的话则必须推迟。为了使得混合充分，交易确认时间滞后大。
2. 混合器是中心化部署的，混合器节点能获取交易的所有信息，能盗币。而且，相对于去中心化的网络，很容易破坏混币器运行，并延迟其他参与者完成混币交易。
3. 即使经过多轮混币器混合，如果用户的钱包在进行支付时不清除浏览器cookie的话，可以通过技术手段识别用户的钱包，因为混币只会掩盖地址之间的交易链接，但不会完全破坏它们

相对于混币器，Zerocoin提出了一个全新的方案——零知识证明。简单地说，零知识证明是你做某件事或知道某件事的证明，而不泄露任何其他信息。例如，怎么证明您知道保险柜密码，而不需要实际显示密码？打开保险柜就好。在使用零知识证明的情况下，交易间的联系被完全打破。  
Zerocoin的工作原理是，熔铸掉你自己的加密货币(也就是所谓的零币熔铸)，然后再兑换等量的做记号的新币(被称为零币取回)。这些币在没有任何交易历史的情况下出现，与新挖矿而得到币相似。零知识证明用来证明你确实烧过了加密货币，而没有透露你烧过的具体加密货币信息，因此你有权赎回等量的新币。与只是掩盖交易环节的混币不同，这种机制完全打破币的交易环节。转账交易接受方得到的币看起来完全是全新的币。

运用零知识证明实现匿名性比混币更牛叉吧~~然而Zero coin自身也有缺陷，比如，只能处理固定面值的交易、一笔匿名交易占据数据大小是25kb。



Zerocash同样采用了零知识证明，同时提出了zkSnarks,改进Zerocoin。有了Zerocash和zkSNARKs技术，匿名数据大小现在只有1 kb，可以快速验证。而且，所有交易金额都是隐藏的，在进行熔铸时不需要使用固定的面额。zk-Snarks是零知识证明的一个变体,它包含4部分：

1.用一种特定的方法，编一个谜题

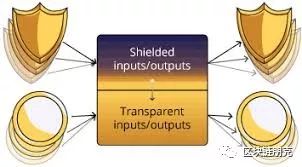
2.证明者解谜

3.用同一种方法再编译谜题

4.证明者解谜。如此往复多次。

这样验证者就可以在不知道真实的编码值的情况下验证交易合法了。

Zcash转账交易的交易分3种情况：



1.透明资金发送到透明地址

延用bitcoin的交易方式，仅仅打签名即可。

2.透明资金发送到透明地址、私有地址

透明资金发送到透明地址，依然延用bitcoin的方式，发送到私有地址的需要构造证明，对整个交易签名发送。

3.私有资金发送到透明地址、私有地址

每笔交易需要生成证明，涉及到找零、矿工费的处理，最后再签名发送。

Zerocash也有不足。

首先，Zerocash难以审核。和Zerocoin一样，Zerocash需要一个信任机制，但Zerocash的设置要复杂得多。Zcash采用了一场涉及6人的多方仪式来完成设置，其参数泄露的唯一方式是，所有的6人相互串通，并保留相关参数。换句话说，你必须信任这6个人中的任何一位，相信他们摧毁了初始参数并且相信这个仪式被正确执行。如果代码中存在bug，或者密码缺陷或多方信任设置的问题，攻击者可以生成无限的Zcash。

其次Zerocash使用一种zkSNARKs的新加密算法。但是，相对RSA技术，其并未经过多年的使用和审查，也并未被广泛使用。许多密码学专家认为zkSNARKs的保密性相对较弱。  
由于需要进行复杂的数学计算，因此，在一台功能强大的计算机上，匿名交易的生成时间要比之前任何一种匿名方式要长得多，特别是对配置较低的计算机来说，这一过程太长。这使得人们无法广泛使用它的匿名功能，也可能会不适用一些功能不太强大的系统，比如移动设备。

块奖励

Zcash出块和奖励机制类似比特币。Zcash总量是2100万个币，最初的 20,000 个块的奖励很少。这样做的目的是为防止早期的快速挖矿对Zcash 系统造成伤害。 Zcash 系统每 2.5 分钟一个块， 20,000 个块之后，每个块奖励 12.5 个ZEC。Zcash 总量 2100 万枚，每 4 年减半一次。

比特币Zcash总量2100万2100万出块时间10分钟2.5分钟块大小1M2M产量产量递减。每挖出21万个区块，产量减半。第1个区块奖励50btc第 1~ 20,000 块，奖励随着高度线性递增，增加幅度是0.000625币个块。块1的奖励是0.000625币，在第 20,000 个块处奖励 12.5 个ZEC。从块20000开始，奖励按照每84000块减半一次创世块2009年1月3日2016年10月29日第一次产量减半2012年11月28日预计创世4年后，高度940000块时奖励分配全部归矿工前4年，20%归Zcash,80%归矿工。第四年以后全部归矿工

风险

Zcash整个挖矿过程基于创始密钥（Genesiskeys）或“主私钥”（master private key），而这个密钥是由6个人共同持有的。密钥一旦外泄，不怀好意的持有者就能够创造无数个Zcash。而这种高强度的生成方式很可能导致网络崩溃。问题在于，这6个有权获取主密钥的人都来自同一家企业。假设该企业经营不善，或者密钥被盗用，整个Zcash网络很容易遭到交易攻击。

# Zcash核心：零知识证明（一）

[吴博聪](https://www.zhihu.com/people/warriorwu)

[**吴博聪**](https://www.zhihu.com/people/warriorwu)

公号【丧心病狂的社会攻城狮】微博 @Warrior比特币玩家

​关注他

4 人赞了该文章

[首发于微信号『丧心病狂的社会攻城狮』](https://link.zhihu.com/?target=http%3A//mp.weixin.qq.com/s/svt3dVguManBjApK6Fewtg)

微信&知乎『**丧心病狂的社会攻城狮**』，[金色财经专栏](https://link.zhihu.com/?target=http%3A//www.jinse.com/member/25398)&[微博](https://link.zhihu.com/?target=https%3A//weibo.com/btcwar)@**Warrior比特币玩家**。

在**Zcash：黑夜中潜行**一文中曾提到Zcash的突破之处在于使用了零知识证明（zero knowledge proof）实现了私密交易与去中心化共存，那么，零知识证明究竟是什么？

它指的是证明者能够在不向验证者提供任何有用的信息的情况下，使验证者相信某个论断是正确的。比如证明者向验证者证明并使其相信自己知道或拥有某一消息，但证明过程不能向验证者泄漏任何关于被证明消息的信息。 其中又分为交互式零知识证明和非交互式零知识证明。

## 交互式证明

举个例子，如何向色盲证明两个球颜色不同。



假设A有一个红球和黄球，她的一个色盲朋友B不相信A的球颜色不同，A如何才能让B相信这是真的呢？一个简单的办法如下：

1. A把两个球给B，B每只手拿一个球
2. A转过身背对B
3. B抛一枚硬币，如果正面朝上，则保持不动，否则交换左右手的球
4. A转过身，B问A是否交换过球
5. 如果A回答错误，则B不相信；否则，重复100次实验，如果A都回答正确，则B相信这两个球是不同颜色的

为什么？如果两个球颜色不一样，那么A每次都应该能分辨；如果两个球颜色一样，则不管B有没有交换过，A都无法分辨，只能乱猜。这样每次猜对的概率只有1/2，重复100次，都猜对的概率将小到可以忽略，所以A将必然猜错。

这就是交互式零知识证明的一个例子，上述证明有三个特点：1）交互过程，整个证明需要A和B进行交互才能完成；2）具有随机性，即B需要抛一枚硬币，来决定是否交换球；3）零知识，虽然B最终相信了这两个球是不同颜色的，但B还是不知道这两个球是什么颜色的。

# Zcash核心：零知识证明（二）

[吴博聪](https://www.zhihu.com/people/warriorwu)

[**吴博聪**](https://www.zhihu.com/people/warriorwu)

公号【丧心病狂的社会攻城狮】微博 @Warrior比特币玩家

​关注他

2 人赞了该文章

[首发于微信号『丧心病狂的社会攻城狮』](https://link.zhihu.com/?target=http%3A//mp.weixin.qq.com/s/Z0_IjVao13kT7txa5mjtMA)

微信&知乎『**丧心病狂的社会攻城狮**』，[金色财经专栏](https://link.zhihu.com/?target=http%3A//www.jinse.com/member/25398)&[微博](https://link.zhihu.com/?target=https%3A//weibo.com/btcwar)@**Warrior比特币玩家**。

上一篇中我们提到了交互式零知识证明，这一次，我们真正来讲讲Zcash所使用的非交互式零知识证明。

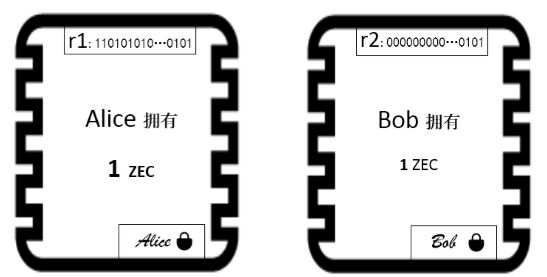
## 非交互式零知识证明

顾名思义，非交互式即证明过程不需要反复**本文由币乎社区（**[bihu.com](https://link.zhihu.com/?target=https%3A//bihu.com/)**）内容支持计划赞助**

微博上总有读者私信询问关于**区块链出海**的问题，所以建了个**芥末圈**子（**非知识星球**）『区块之殇』，方便大家交流，内容也会优先在里面发布。**原创不易**，因为知乎和微信的限制，目前只能在微信中扫码进入，欢迎[点击加入](https://link.zhihu.com/?target=http%3A//mp.weixin.qq.com/s/SHZ4OZ0a2jY_IfDHndEGMA)。**本文由币乎社区（**[bihu.com](https://link.zhihu.com/?target=https%3A//bihu.com/)**）内容支持计划赞助**

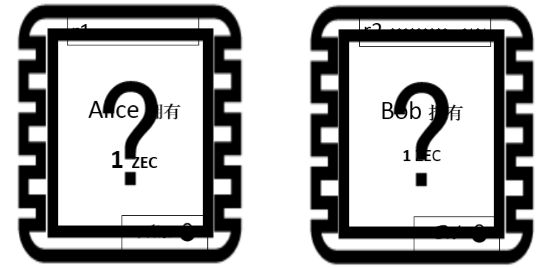
的问题，所以建了个**芥末圈**子（**非知识星球**）『区块之殇』，方便大家交流，内容也会优先在里面发布。**原创不易**，因为知乎和微信的限制，目前只能在微信中扫码进入，欢迎[点击加入](https://link.zhihu.com/?target=http%3A//mp.weixin.qq.com/s/SHZ4OZ0a2jY_IfDHndEGMA)。在双方间传递信息，这个如何实现？我们以Zcash为例：

Alice要向Bob转一个ZEC，此时Alice拥有一张1ZEC的支票，要转账给Bob时，先给Bob新建一张1ZEC的支票，这时Alice和Bob都有了一张支票。

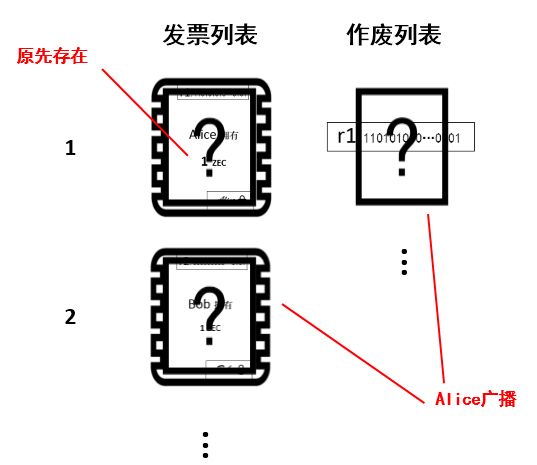


这两张“支票”都是有效的。Alice的支票开始就存在于整个Zcash网络，Bob的支票在生成后也会被广播到全网。

为了隐藏交易者信息，要对两张支票进行加密处理，在全网中存在的“支票”其实是这样子的。



同时，因为资产只能有一份，所有矿工手里还有一个作废列表。Alice要同时广播自己的“发票代号”，录入作废列表中。发票代号也是加密的。所以矿工们能看到的信息其实是这样的。其中Alice的支票是原先存在的，Alice的支票代号r1和Bob的支票是在交易过程中被Alice广播的。



矿工们能获取的信息相当有限，但是这并不影响对矿工对交易有效性的判断。

判断的逻辑相当简单：矿工拿到Alice给的支票代号r1，去作废列表中检索，假如作废列表中已经存在r1，则证明r1所对应的的支票早已失效;若作废列表中并不存在r1，则证明r1对应的支票仍旧有效，此时矿工把r1录入作废列表中，把新生成的支票录入支票列表中。所以记账的过程就是对原有支票登记失效，并存入现有支票的过程。

在这个过程中，我们不难发现，每笔交易矿工能接收到的东西只有一个发票代号，和一张新的发票，而且这两样东西都是被加密的。所以矿工并不知道转账双方是谁，也不知道转账金额是多少。Bob也不知道是谁转的账，但只要Alice把解锁支票的查看私钥通过私信给Bob，Bob就可以知道这笔交易的所有信息。

这种方法就属于非交互式零知识证明。整个交易过程中，Bob并没有见过Alice的支票，但是还是实现了资产所有权的转移。同时，Alice也可以通过发送查看密钥的方式向Bob证明自己身份。