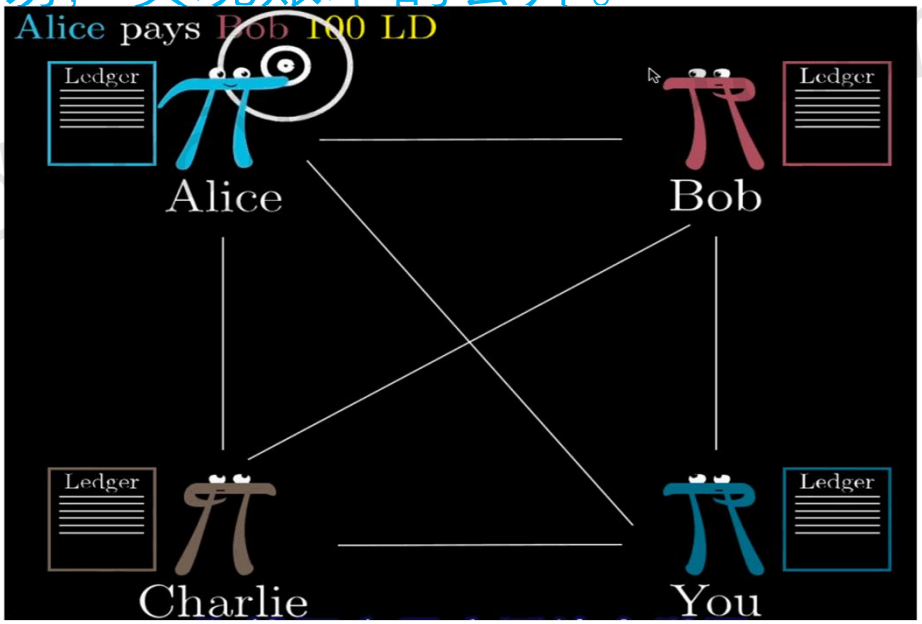
# 区块链隐私保护

网络空间安全学院 代炜琦

# 目录

# 区块链的公开透明与隐私保护

- 比特币：为了保证交易的真实性，比特币广播每一条交易，实现账本的公开。



# 账户模式

张三获得了12.5 枚比特币。过了几天，他把其中 2.5 枚支付给李四。
又过了几天，他和李四各出资 2.5 比特币凑成 5 比特币付给王五。

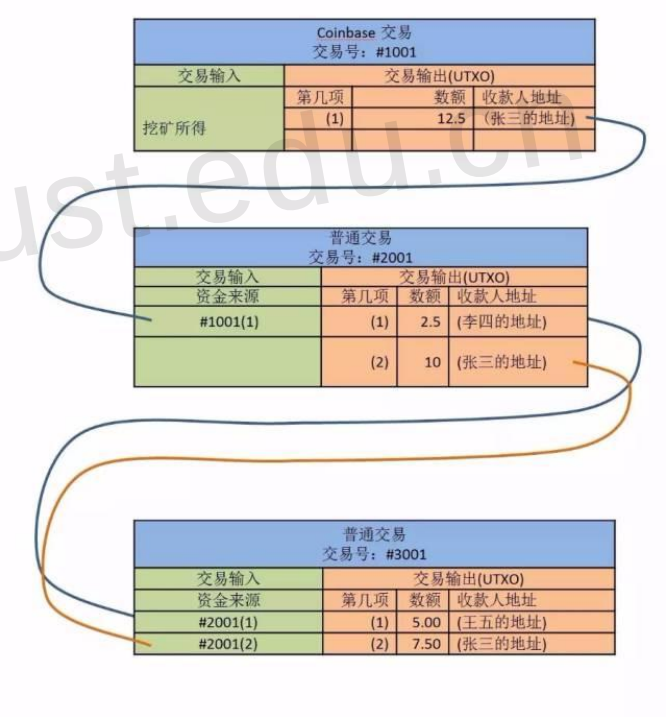

为了避免双花，传统系统往往需要较高的维护成本。

# UTXO

张三挖到12.5 枚比特币。过了几天，他把其中 2.5 枚支付给李四。又过了几天，他和李四各出资 2.5 比特币凑成 5 比特币付给王五。

缺点：**隐私泄露**

- 交易行为（**张三在出2001块的时候交易了2.5比特币！**）
- 账户余额（**张三还有7.5比特币！**）
- 账户流水（**他和李四还有王五进行了交易！**）



# 比特币的隐私保护

比特币系统的匿名设置:




不足：**匿名性有限**

- 大多数用户只有一个地址
- 通过交易分析可以得出同一用户不同地址之间存在的联系

[Reid Martin 11] [Barber Boyen Shi Uzun 12] [Ron Shamir 12] [Meiklejohn PJLMVS 13]

**从用户交易的角度上看，比特币的匿名性是不能接受的**

# 目录

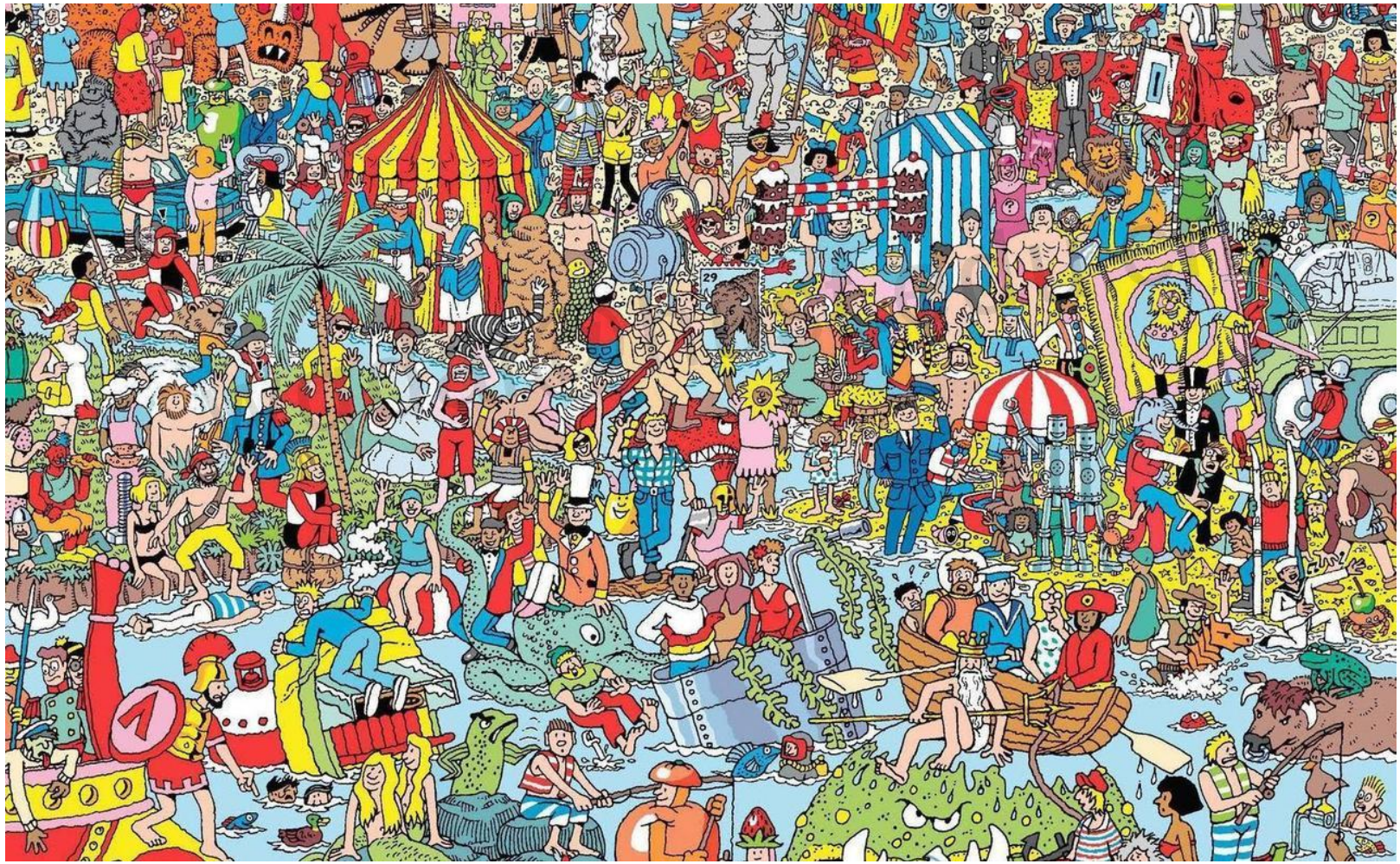
- 账户模式与UTXO
- **零知识证明**
- Zcash
- 资产恢复方案

# 零知识证明--简介

- 寻找瓦尔多

怎么在**不暴露瓦尔多具体位置**的情况下向别人证明你找到了瓦尔多?

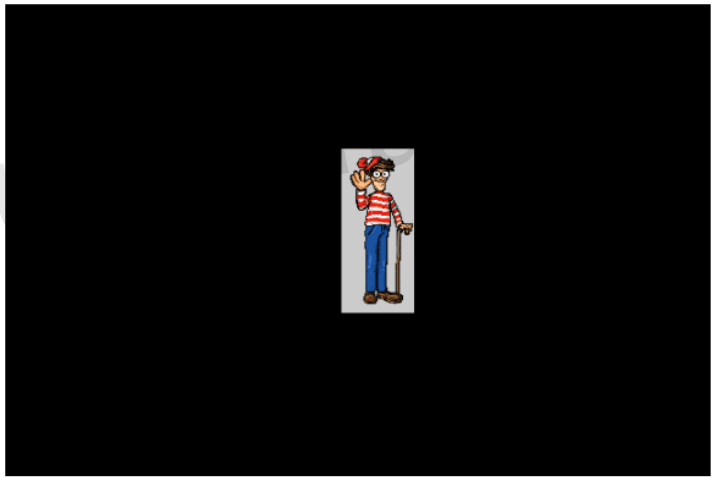
# 零知识证明--简介



# 零知识证明--简介

- 简易解决方案：

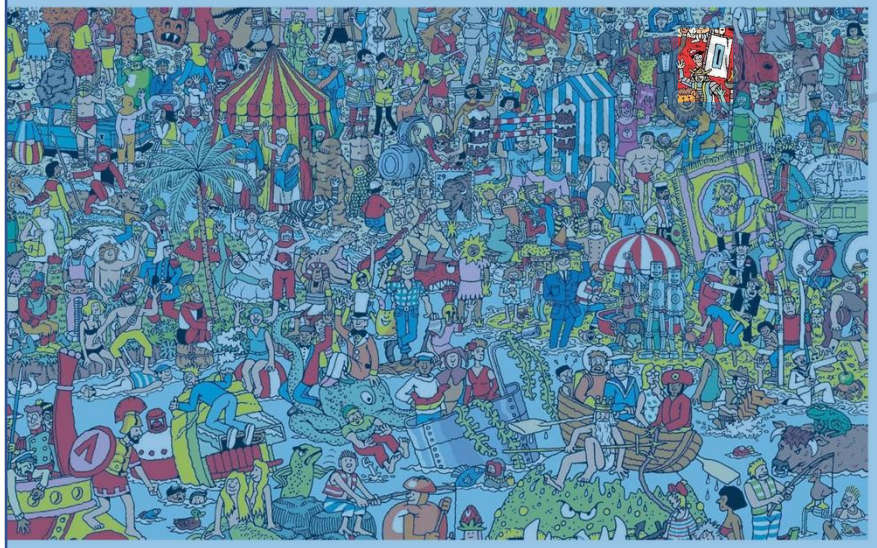
首先准备来一个大纸板，大约是游戏图片的两倍大小，然后在纸板上剪出一个**矩形小窗口**。接着，可以在确保 其他人没有偷看的时候，在游戏图上移动纸板，使得瓦尔多的图案正好出现在矩形窗口中。



这样我们就在**没有暴露具体位置**的情况下向他人证明我们知道这个问题的答案。

# 零知识证明--简介

# 零知识证明--简介



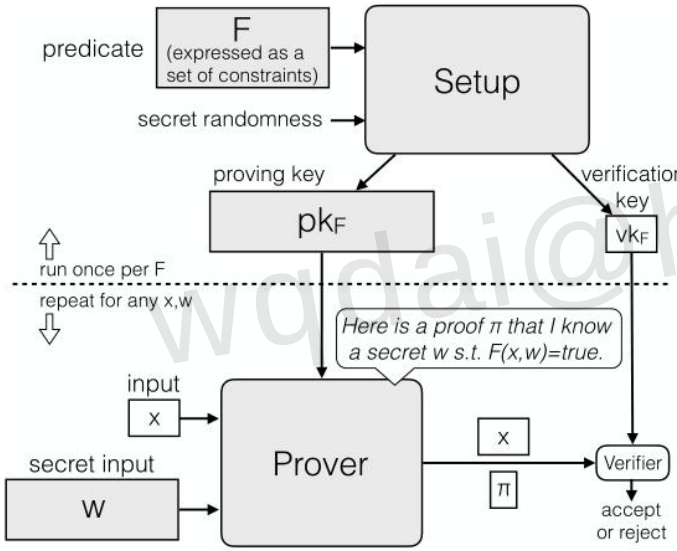
# 零知识证明--非正式定义

零知识证明: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。常被用来证明/验证如下NP语句(NP statement):

“对于某一**公开论断（predicate）**F和**公开输入**x，我知道**某一秘密输入**w,使得w是论断F关于x的**正确解**。”

“对于**寻找瓦尔多**这个问题以及**公开的图片**，我知道**瓦尔多的位置信息**，使得其他人在图片中**能够找到**瓦尔多””

# 零知识证明—zk-snark协议




zk-snark是" zero knowledge Succinct Non-interactive ARgument of Knowledge"的简写，是非交互式零知识证明（Non-interactive zero-knowledge proofs）中的一种，zk-snark将$F$和某一秘密随机数通过$setup$算法生成$pk, vk$ 。

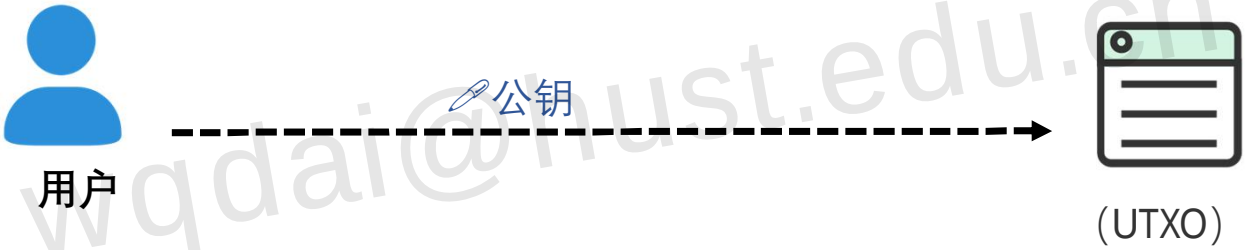
证明者，也就是知道$F$答案$x$的人，通过证明秘钥$pk$、 $x$和秘密输入$w$生成零知识证明$\pi$（ $\pi$长度固定），可以被任意验证者在有限时间内使用$vk$校验，但是证明者无法通过$\pi$的值获取w的有效信息。

# 目录

- 账户模式与UTXO
- 零知识证明
- Zcash
- 资产恢复方案

# ZCash

**比特币模型**




$$UTXO_1 = (PK_1), UTXO_2 = (PK_2), UTXO_3 = (PK_3)$$

# ZCash

Zcash 简要模型

$UTXO_1 = (PK_1), UTXO_2 = (PK_2), UTXO_3 = PK_3$

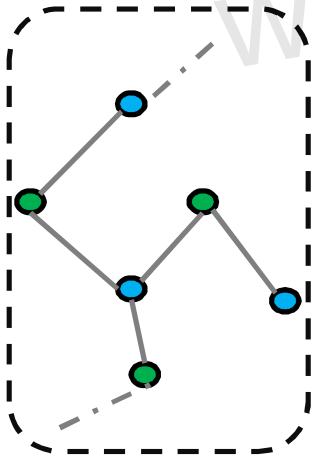
加入序列数r对UTXO进行混淆

$UTXO_1 = (PK_1, r_1), UTXO_2 = (PK_2, r2), UTXO_3 = (PK_3, r_3)$

链上不再记录UTXO，而是他们的摘要表

hash计入区块链网络

$H_1 = HASH(UTXO_1), H_2 = HASH(UTXO_2), H_3 = HASH(UTXO_3)$

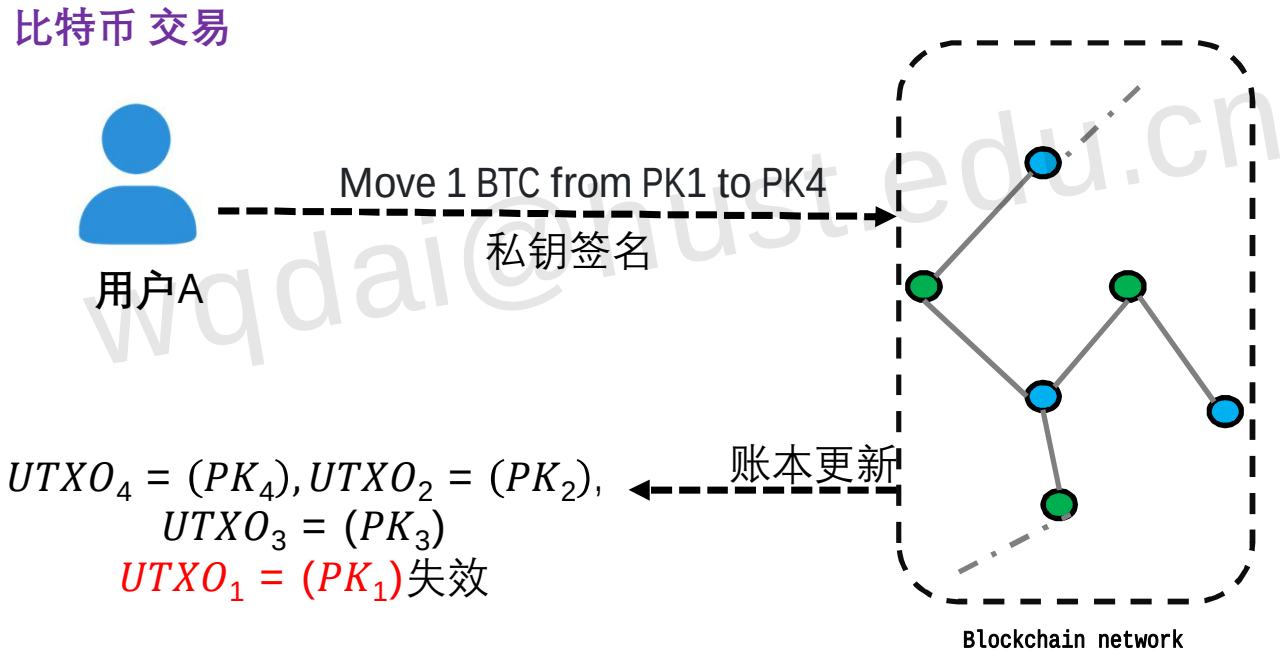


Blockchain network

记录已经被消费的UTXO的随机数

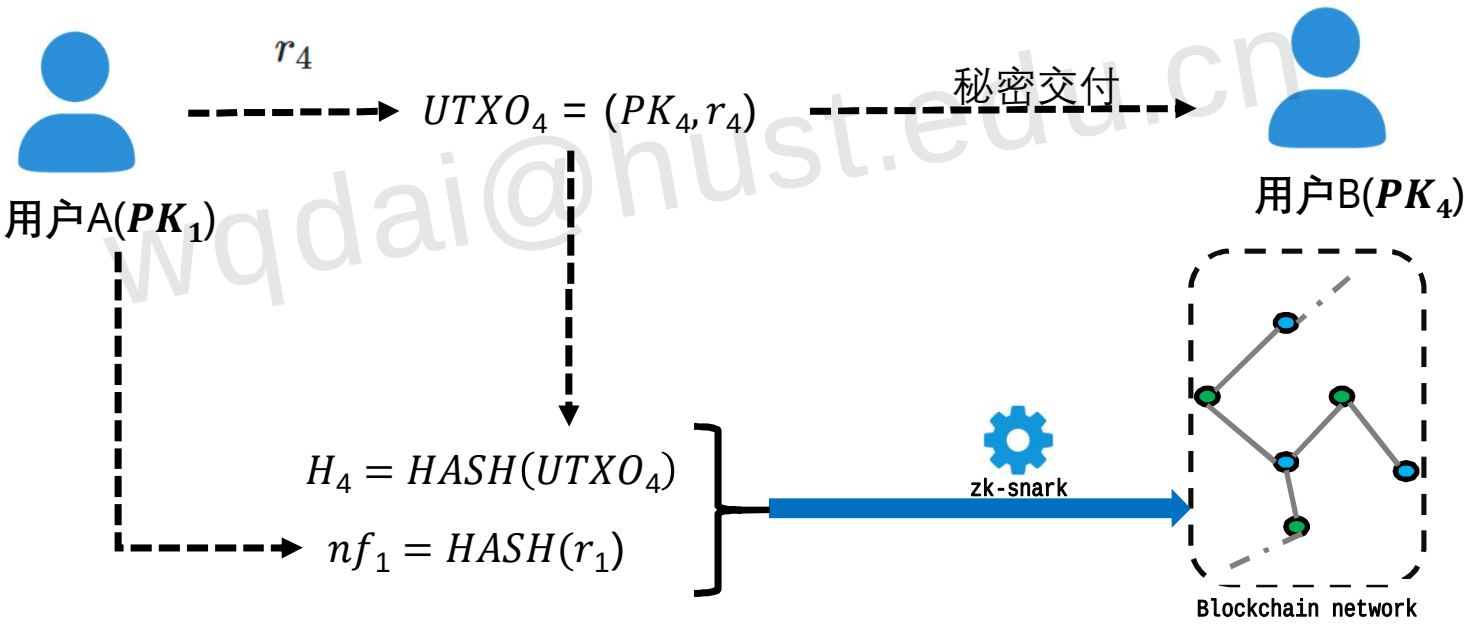
| Hashed UTXO | Nullifier set |
|---|---|
| $H_1 = HASH(UTXO_1),$ | $nf_2 = HASH(r_2)$ |
| $H_2 = HASH(UTXO_2)$ | |
| $H_3 = HASH(UTXO_3)$ | |

# ZCash



# ZCash

Zcash 交易



# ZCash


**零知识证明内容**

用户A知道公共输入 $UTXO_4 = (PK_4, v)$ 以及秘密输入 $SK_1$和$r_1$,需要向区块链网络证明以下几点:

1、 $UTXO_1 = (PK_1, v)$的hash值存在于区块链网络的hash表中
2、用户A知道$PK_1$对应的私钥$SK_1$
3、 $nf_1 = HASH(r_1)$是关于$r_1$的hash，如果$nf_1$没有出现在nullifier set中，说明对应的$UTXO$还没被消费

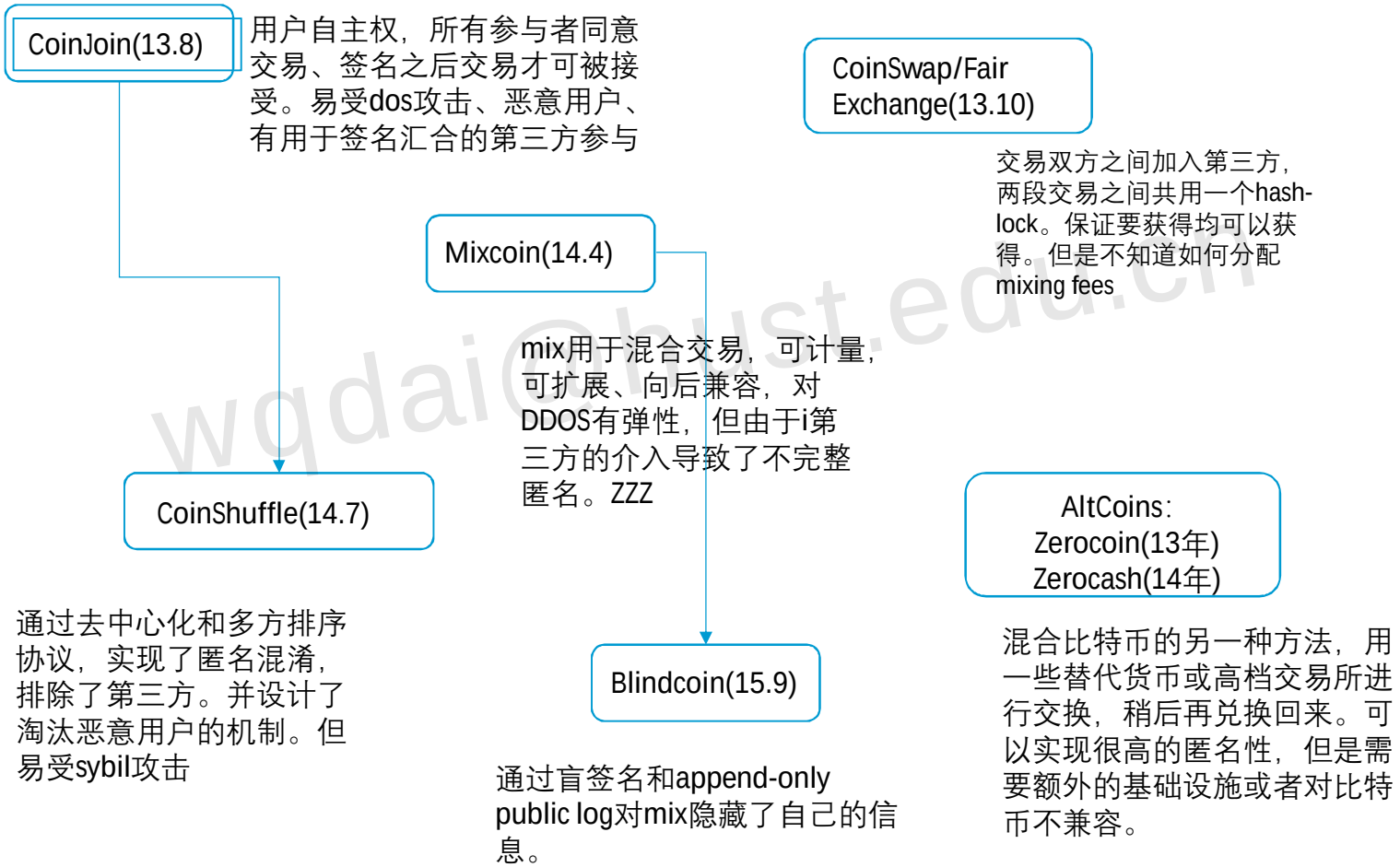
对应证明要点

- 证明pk1上拥有某一UTXO的hash
- 证明用户有权使用该UTXO进行消费
- 证明该UTXO还没有被用户消费

# 隐私保护技术

| 技术名称 | 定义 | 不足 | 创新点 |
| --- | --- | --- | --- |
| 链下方式（off-chain）（zeroCoin侧链） | 这一分类包括链下信息传送、侧链（sidechain）以及区块链中隔离数据的交易通道。侧链和交易通道使用者可以在私有化控制下的链中进行交易，其中的资产在公链中也是有效的。 | 不同计算机中需数据复制，因此会破坏网络回弹力 | 多个隔离的交易通道，公私数据带并存。传播阶段 |
| 环签名（以太坊crytoNote环签名） | 发送的交易通常捆绑了多个发送方的公钥，仅从环签名难以识别出交易发送方以及最终的签署方信息。实质为基于群签名的改进 | 部分观察值能够提高攻击者定位地址的准确度，三角分配的方法定位虚假地址 | 交易携带多个公钥。隐藏发送方私钥。（人群中隐藏） |
| 零知识证明（ZeroCoin） | 用于加密数据的密码学验证，可以不公开发送方以及交易金额，但同时又能做到证明这笔交易的合理性。 | 速度很慢（48秒），不适合大流量交易。且加密货币用此技术，还需其他一些密码学要素 | 验证方面的隐私性保护 |
| 隐身地址（bitcoinj\BIP） | 是发选择 一个大随机数。这个数字用一系列公式（公式中因含着数据种子）进行计算后得到新公钥，其对应的私钥只能由收款人计算出来，而且与原来的数据种子不相关。<br>款人 | 不提供“100％匿名” 如果您了解一个交易或其中的一方，您可以推断出那些硬币来自哪里或去哪里。 | 收款方从多个不同的地址收入这些钱，所有交易都看起来毫无联系。同时付款方也只知道自己支付的这笔交易，无法知道其他人跟收款方的交易 |

# 解决方案发展脉络

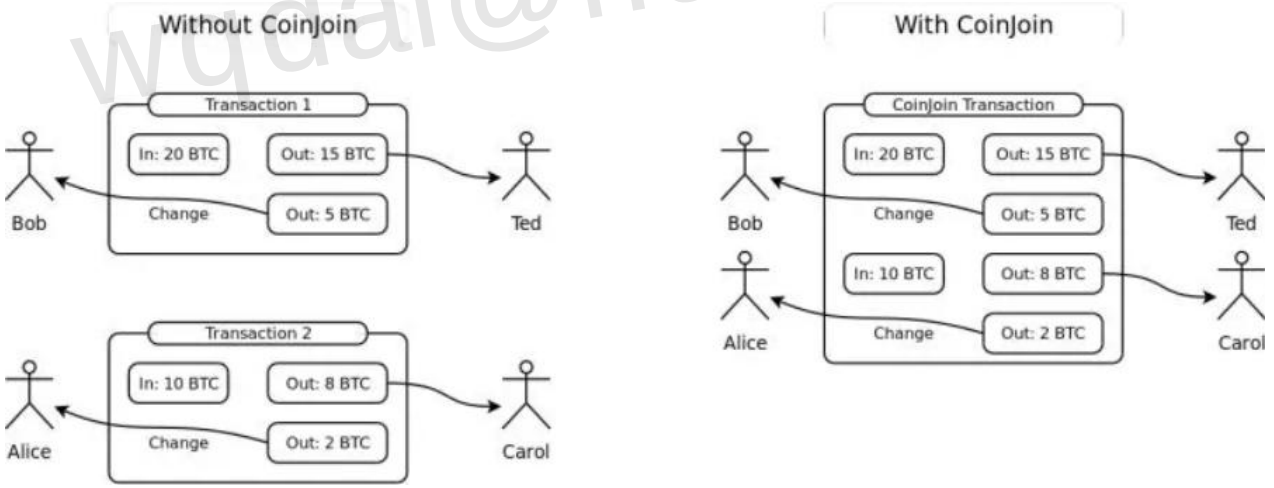


# 隐私保护方案--coinJoin

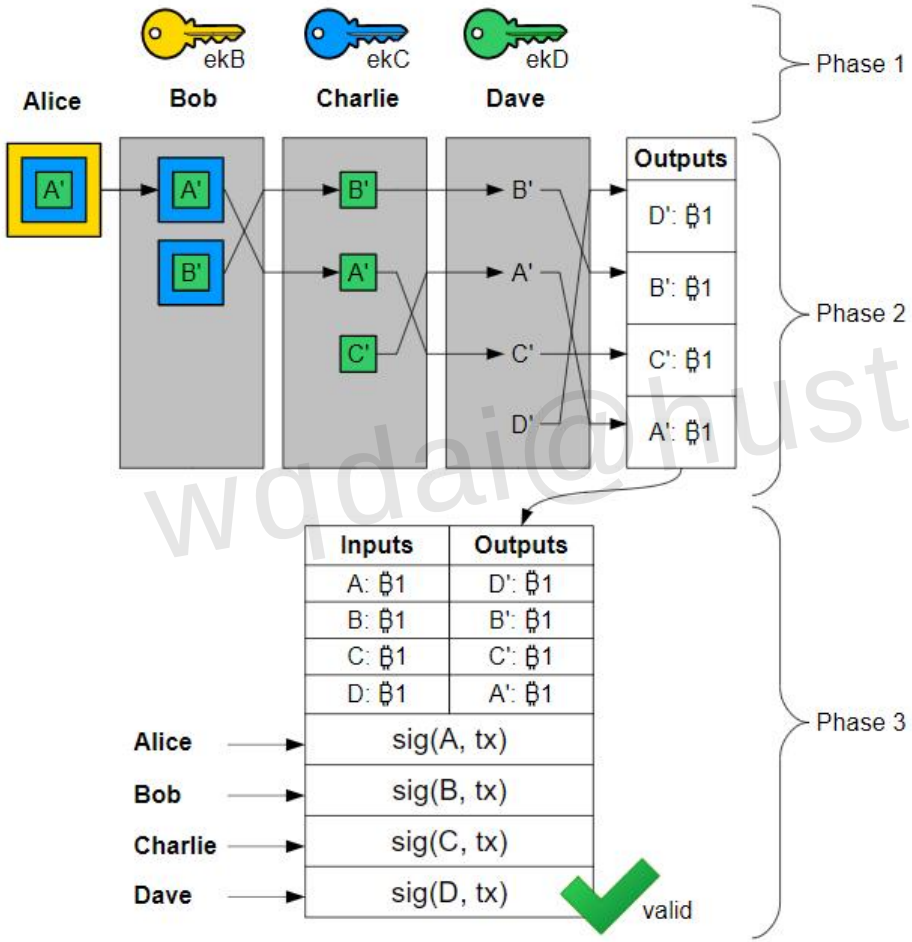
coinJoin:
背景：传统上，一个交易就只是一次转帐，不管交易双方是一对一、一对多还是多对多的方式，都很容易从区块链上得出交易双方的转账信息

而coinJoin提出的改进方案是**多个转账人**之间合作组成**一个群体**，将所有的转账打包到**一个交易**中，这样就无法知道输入和输出之间的关系了。





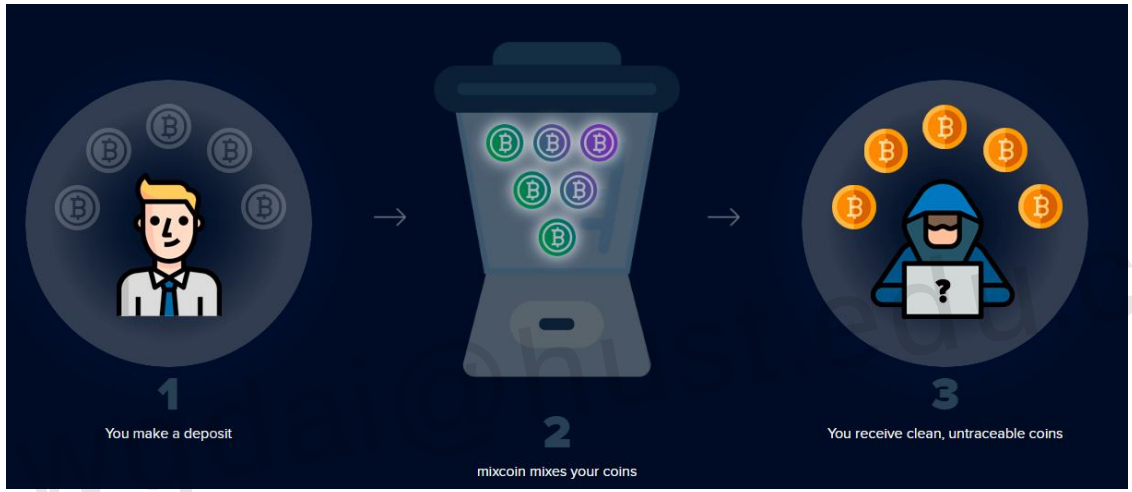
# 隐私保护方案--CoinShuffle




An overview over a successful run of CoinShuffle. [svg] [png]

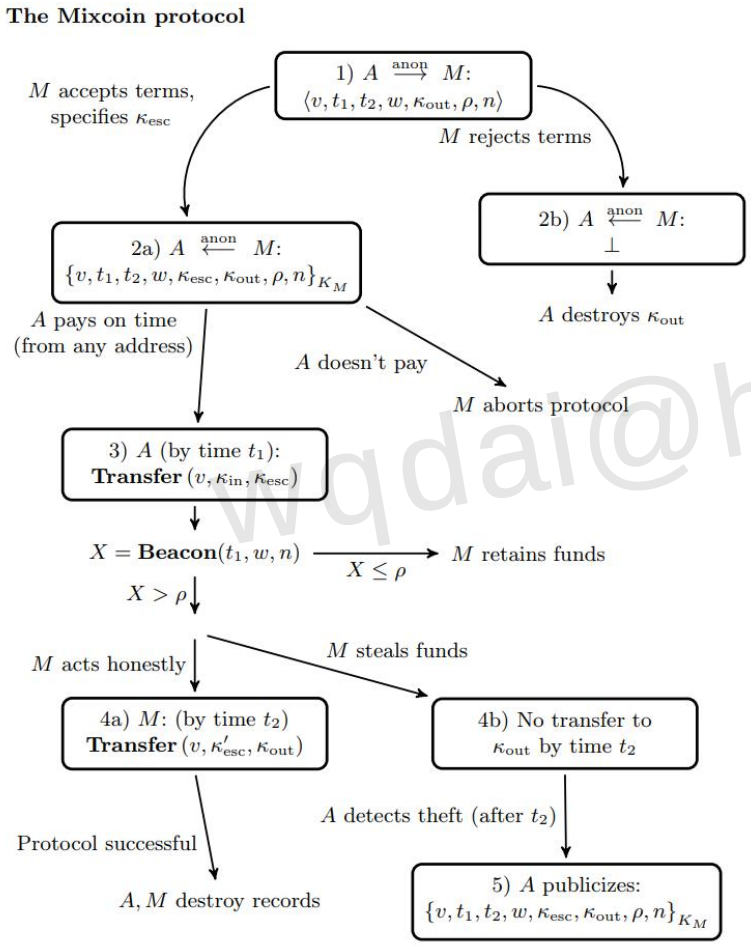
- 为了创建一个混合协议，同时确保输入地址不能与新的输出地址相连，参与者以一种遗忘的方式洗牌他们的输出地址，类似于解密混合网络

# 隐私保护方案--mixcoin



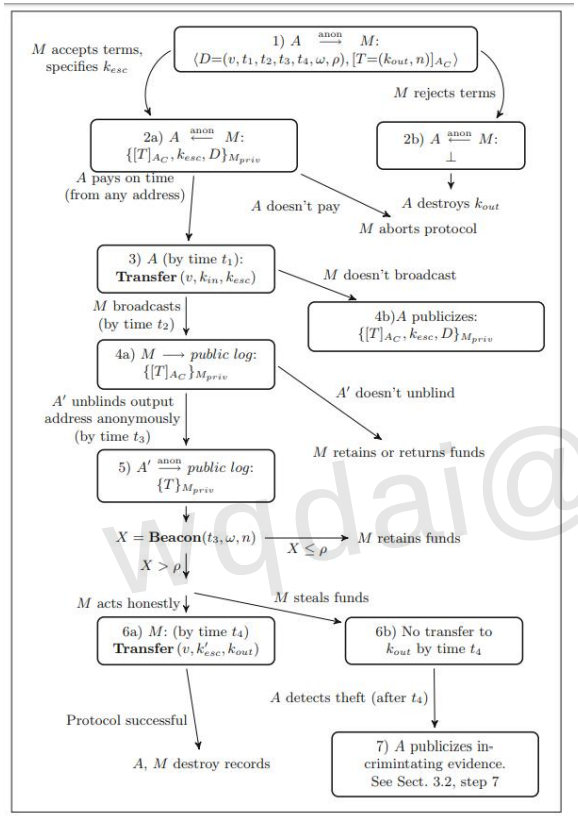

- 当用户向mix系统缴纳费用之后，他们就会受到保护，同时被提供一个保证，当mix系统作恶时，用户可以提供证据。
- 该保证包括一个签署的协议，如果用户在一段时间内将资金支付给组合指定的托管地址，那么组合将在达成协议的最后期限之前将等量的资金转移给用户指定的输出地址
- 匿名性的实现：收集混合费会激励mix行为诚实行事。如果mix费用足够高，那么寻求最大化利润的理性mix系统不会冒险作弊，以免被抓住。

# 隐私保护方案--mixcoin




| | |
|---|---|
| $v$ | the value (chunk size) to be mixed |
| $t_1$ | the deadline[9] by which Alice must send funds to the mix |
| $t_2$ | the deadline by which the mix must return funds to Alice |
| $\kappa_{\text{out}}$ | the address where Alice wishes to transfer her funds |
| $\rho$ | the mixing fee rate Alice will pay |
| $n$ | a nonce, used to determine payment of randomized mixing fees |
| $w$ | the number of blocks the mix requires to confirm Alice's payment |

# 隐私保护方案--Blindcoin

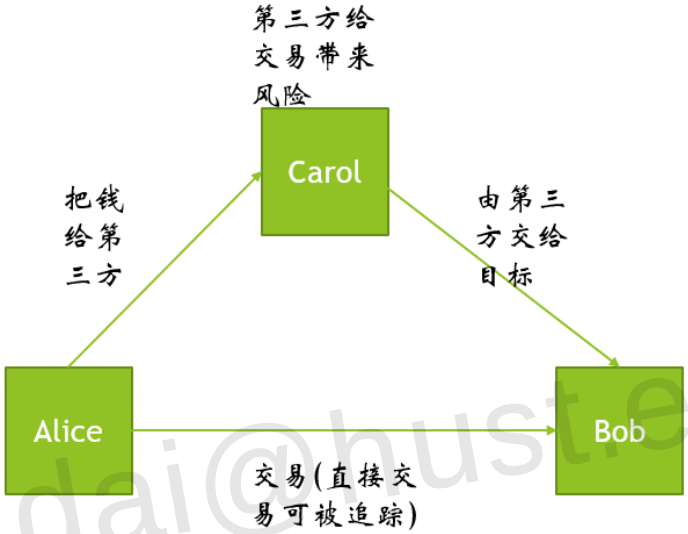



| | |
|---|---|
| $k_{in}$ | the address from which the $A$ pays, possibly linked to $A$'s true identity |
| $k_{out}$ | the address to which the user wishes funds transferred |
| $k_{esc}$ | an escrow address, unique for each user, that $M$ provides for $A$ to pay |
| $k'_{esc}$ | an escrow address that $M$ uses to pay to $k_{out}$ |
| $A'$ | an anonymous identity that $A$ can use to post to the public log |
| $M_{pub}$ | the public key of $M$ |
| $M_{priv}$ | the private signing key of $M$ |
| $A_C$ | a secret commitment/encryption function of $A$ |
| $A_{C'}$ | the inverse of $A_C$ |
| $\omega$ | the number of blocks $M$ requires to confirm $A$'s payment |
| $n$ | a per-user nonce, used to determine payment of randomized mixing fees |
| $v$ | the value (chunk size) to be mixed |
| $\rho$ | the mixing fee rate $A$ will pay |
| $T$ | the token, which is the triple $(k_{out},n)$ |
| $t_1$ | the time by which $A$ must $v$ BTC to $k_{esc}$ in order to participate in the mix |
| $t_2$ | the time by which $M$ must post the token $T$ to the public log |
| $t_3$ | the time by which $A'$ must unblind the output address via the public log |
| $t_4$ | the time by which the mix must transfer $v$ BTC to $k_{out}$ |
| $D$ | the mix parameters, a tuple $\{t_1,t_2,t_3,t_4,v,\omega,\rho\}$ |



在Mixcoin协议中，从用户输入到输出地址的映射对mix服务器是可见的。Blindcoin修改Mixcoin协议，以确保对mix服务器隐藏任何用户的输入/输出地址映射。为了实现这一点，我们使用了一个盲签名方案和一个只进行追加操作的公共日志。该方案与比特币完全兼容，强制混合必须负责任，甚至在恶意混合的情况下也能保持用户匿名，对拒绝服务攻击很有弹性，而且很容易扩展到许多用户。

# 隐私保护方案--CoinSwap

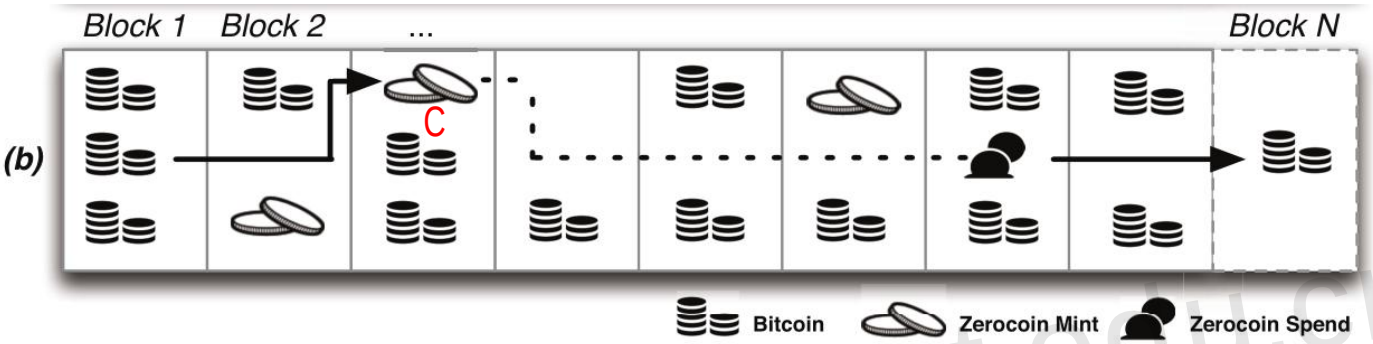





CoinSwap [Ma13b]是格雷戈里•马克斯韦尔(Gregory Maxwell)提出的另一项通过第三方进行交易的建议。爱丽丝没有直接把硬币交给鲍勃，而是把硬币交给卡罗尔，卡罗尔又把硬币交给鲍勃。Alice和Carol以及Carol和Bob之间的事务是托管事务，可以使用受散列锁(hashlock)保护的赎回事务来使用这些事务。这就保证了爱丽丝和卡罗尔都不能偷币。

如今，CoinSwap在比特币上是可用的。它甚至可以用于跨不同链执行事务。然而，匿名性确实依赖于所有2of2e

# 隐私保护方案--zerocoin




假设Alice需要隐藏相关的交易，那么她可以选取一个**序列数（serial number）$S$**和**随意数$r$**，通过**$Hash$得到$C$**,同时花费$1来锻造一个ZeroCoin，网络中所有节点共同维护一个Clist.

当Alice需要花费这个ZeroCoin的时候，她就把（$S$，$\pi$）提交到区块链网络中，**π是非交互式零知识证明**，证明两点：

1）她知道一个$C$ 属于$(c1, \ldots Cn)$，

2）她知道一个r可以把S通过$hash$成$C$。

如果所有节点的零知识验证通过且之前没有任何一笔交易包含$S$,那么Alice可以以此花费$1.