区块链上的交易并没有直接关联到你的身份，这看起来似乎是可以具有隐私保护的。每个人都可以匿名地生成钱包，并进行交易。

然而，事实远没有这么简单。

区块链技术的巨大前景之一是假名（pseudonymity）的使用：交易被记录在公共账本里，但是它们又与由数字和字母组成的地址保持关联。因为无需将真实世界的身份信息关联到地址上，交易的发起者似乎是不可能被追踪的。

然而，这种想法是错误的。没有将假名关联到个人信息，这确实可以保护隐私。然而只要有人建立了链接，则隐私就不再是秘密。一个例子是执法机构坦言他们在调查时，可以识别比特币用户，对他们进行反匿名（deanonymizing）。

这是怎么发生的呢？

商业网站的 Web Tracker 和 cookies 会轻易泄露与交易相关的信息。任何人，包括政府、执法机构和恶意用户都可以利用这些信息。

此外，区块链平台（如以太坊）的用户与智能合约进行着复杂交互。智能合约的所有细节，包括发送者和接受者、交易数据、执行的代码和合约内部存储的状态，都是公开。

大部分公司都不会考虑把重要的商业数据上传区块链中，因为黑客、竞争者和其他非授权组织都可以轻易看到这些信息。思考一下：

* 电子医疗记录是十分隐私和敏感的信息。
* 身份识别数据如身份证不能在智能合约上公开。
* 凭证管理如密码和密钥都不能放在公开和不安全的智能合约中。
* 金融文件如股权结构表或员工薪资都不能公开。
* 这样的例子不胜枚举

隐私保护对于个人、组织和企业来说，都是一个本质挑战。许多人为区块链和数字货币着迷，是因为这个去信任和抵抗审查的系统能带来金融上的变革。矛盾的是，我们在使用的是一个公开且容易被追踪的账本。

**隐私保护解决方案**

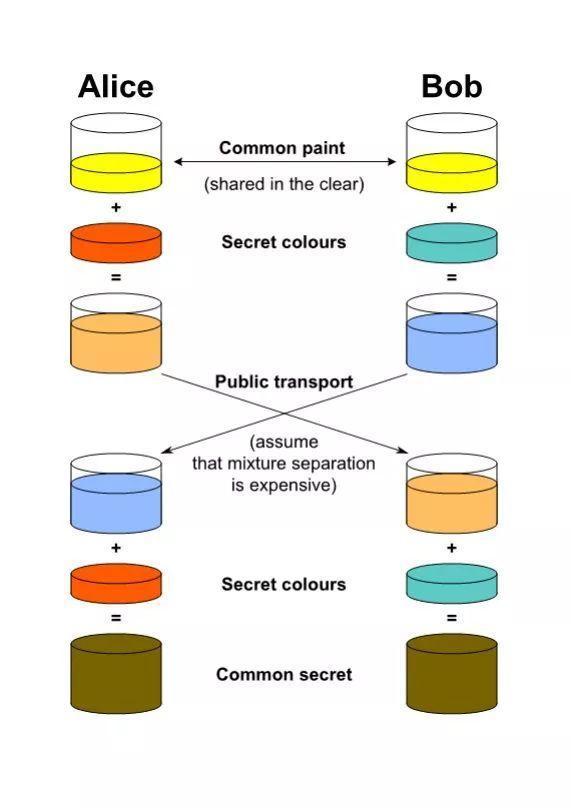
下面是一些不同开发团队正在努力实现的方案。

Elliptic Curve Diffie-Hellman-Merkle (ECDHM) addresses

理解 ECDHM 地址之前，你需要理解 Diffie-Hellman 密钥交换，其背后的思想在于双方之间建立一个共享的秘密。在公开网络里，这可以被用于交换秘密信息。

**这是如何做到?**

发送方和接收方共享 ECDHM 地址，然后通过共享的秘密将其转化成比特币地址。该比特币地址只会被拥有该秘密的人知道。唯一公开的东西只有可重复使用的 ECDHM 地址。因此，用户不用担心交易会被追踪。

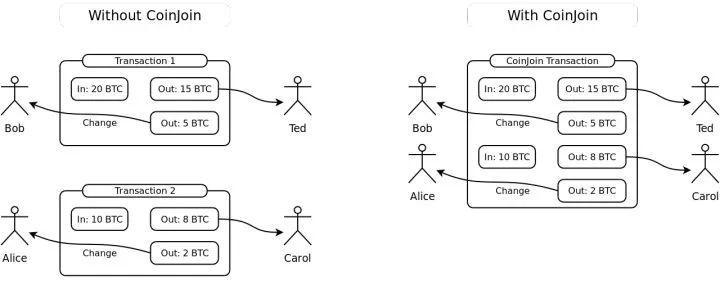
[[](http://upload-images.jianshu.io/upload_images/10818463-6fcca4b0002310d1?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)](http://upload-images.jianshu.io/upload_images/10818463-6fcca4b0002310d1?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)

Conceptual diagram that illustrates the general idea of the key exchange by using colors instead of very large numbers (Source: <https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>)

ECDHM 地址方案包括 Peter Todd 提出的 Stealth Addresses，Justus Ranvier 提出的 BIP47 Reusable Payment Codes 和 Justin Newton 提出的 BIP75 Out of Band Address Exchange。然而，没有一个方案得到实际应用。

**混合器**（Mixer）

混合器的思想是把交易混到一个池里，收支平衡由池中的私有账本来记录。当池中的资金被花费后，原始交易就变得难以追踪了。观察区块链的人可以看到池的支付金额和收款人，但是无法追踪交易的发起人。混合器服务的一个例子是 CoinJoin。

[[](http://upload-images.jianshu.io/upload_images/10818463-c711b9e28f38de89?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)](http://upload-images.jianshu.io/upload_images/10818463-c711b9e28f38de89?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)

Source: <https://en.wikipedia.org/wiki/CoinJoin>

不幸的是，混合器不是一个可靠的解决方案。研究者可以确定 CoinJoin 里的交易，他们证明，攻击者只要花费 $32,000，就能以 90% 的成功率识别交易。并且，研究者还证明混合器几乎无法抵御女巫攻击（Sybil attacks）和拒绝服务攻击（Denial-of-Service attacks）。

另一个令人不安的地方是，需要通过一个相对中心化的实体来维护混合器的私有账本，这意味着需要一个可信第三方来“混合”交易。

CoinJoin 不是默认交易方法，因此很少人参与到进混合池里，这导致匿名集合十分小。在人数少的集合里，可以很容易确认交易的来源。

另一种混合器方案是 CoinShuffle，这是德国萨尔布吕肯大学研究团队设计的去中心化混合协议。CoinShuffle 尝试解决 CoinJoin 里需要可信第三方来混合交易的问题。

**门罗币**（Menoro）

不同于其他的山寨币，门罗币不是比特币的分叉，而是基于另一种协议 CryptoNote。门罗币的最大特色是环签名（Ring Signature）方案。

环签名是一种群签名，群里的每个签名者都拥有一对私公钥对。不像传统的加密签名证明交易是单个签名者用私钥签的，群签名证明交易是群里的某个人签名的，但不会暴露具体是谁签的。

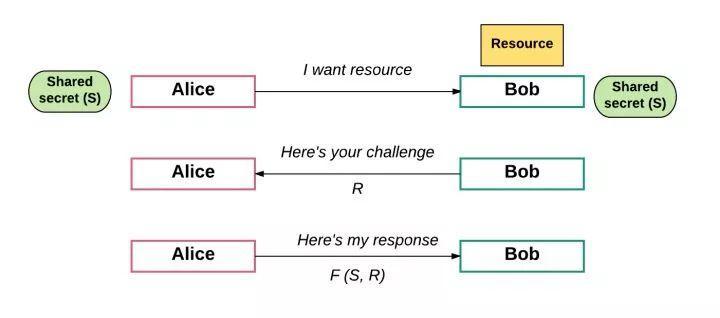
**零知识证明**（Zero-knowledge proofs）

零知识证明是指，在不揭露特定知识的情况下，证明者（prover）可以说服验证者（verifier）他们知道该知识。换句话说，零知识的输入是秘密的，证明者不会向验证者揭露任何知识。零知识证明可以被用在隐私保护方案里。例子包括：

* 例子1：质询/响应比赛

在计算机安全里，质询-响应认证（challenge-response authentication）是一个协议族。在协议里，一方进行提问（“质询“），另一方给出正确的答案（”响应”）以获得授权。在区块链里，这个“比赛”能被用于验证交易。如果某一交易是非法的，其他节点会注意到其非法性。这就需要提供可验证的证明（verifiable proof），来证实交易是非法的。如果验证失败，则会产生一个“质询”，要求交易的发起人生成一个“响应”，来证明交易是合法的。

这里有一个例子：假设只有 Bob 可以访问某些资源（如他的车）。Alice 现在也想访问它们（如开这辆车去杂货店）。Bob 发起一个质询，假设为“52w72y“。Alice 必须用一个字符串来响应 Bob 发起的质询。使用一个只有 Bob 和 Alice 知道的算法，这是找到答案的唯一方式。此外，Bob 每次发起的质询都会不一样。知道先前正确的响应，并不能给 Alice 带来任何的优势。

[[](http://upload-images.jianshu.io/upload_images/10818463-ab99f3251f5271a9?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)](http://upload-images.jianshu.io/upload_images/10818463-ab99f3251f5271a9?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)

质询/响应比赛已经被使用在区块链，如以太坊里了。然而，我们需要相应的函数库和工具，来使这一类的认证方案更容易被使用。

* 例子2：zkSNARKs

zkSNARKs 到底是什么？让我们来逐步分解其定义：

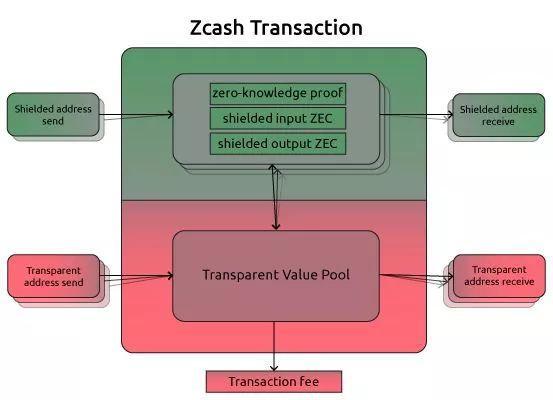
zk=zero-knowledge（零知识）。不需要信息本身的条件下，可以证明该信息存在。

* SNARK：Succinct Non-interactive Adaptive ARgument of Knowledge
* 简洁（Succinct）表示证明简洁，可以被快速验证。
* 非交互（Non-interactive）表示验证者不需要和证明者进行交互。取而代之，证明者会预先公开它的证明，而验证者可以检查它的正确性。
* 适应性知识论证（Adaptive argument of knowledge）表示某些计算的证明。

尽管我希望有一天可以写一篇文章介绍 zkSNARKs，但这里我会跳过技术细节。zkSNARKs是一个的构建隐私保护的组件，它令人振奋且具有远大前景，但有几点需要注意：

* SNARKs 是资源密集型的
* SNARKs 能让用户证明他们拥有访问某个秘密的权限。但用户有职责维护秘密，让它在需要的时候能被访问到。
* SNARKs 需要一个启动阶段，来准备需要证明的电路或运算。该阶段由一组可信团体预先进行。这不仅意味着你需要信任进行该启动阶段的团体，还意味着不适合用 SNARKs 进行任意的运算，因为总需要一个准备阶段。
* 例子3：zkSNARKs + Zcash

Zcash 是基于 zk-SNARKs，具有隐私保护特点的加密货币。在 Zcash 所谓的“私密交易（shielded transactions）”里，每一个被使用的币都带有一组匿名集合。私密交易使用“私密地址（shielded addresses）”，它要求发送方或接收方生成一个零知识证明，以在不泄露交易信息的情况下，允许其他人验证交易。

[[](http://upload-images.jianshu.io/upload_images/10818463-cde3f8f3a4ed93f2?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)](http://upload-images.jianshu.io/upload_images/10818463-cde3f8f3a4ed93f2?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)

Zcash transaction diagram

Zcash 无疑是一个值得关注的有趣项目。

* 例子4：zkSNARKs + Ethereum

在以太坊下一个要升级的协议 Metropolis 里，开发者将可以在链上高效地验证 zk-SNARKs。

我们可以在支持 SNARKs 的以太坊里做什么呢？可以把某些合约变量被设为不可见。秘密信息可以与那些遵守使用 SNARKs 的合约规则的用户存储在一起，而不是直接存储在链上。每一组用户群自身都需要一个可信的启动阶段，这会增加一些用于准备阶段的开销。但一旦电路被设置好，它就可以被任意数量的交易使用。

在支持 SNARKs 的以太坊里，你无法把隐私与用户分开，即做不到自治性隐私（autonomous privacy）。由于以太坊的 SNARKs 依赖用户在链下维护密钥，因此如果没有这些用户，就没有地方可以找到这些秘密。

* 例子5：zkSTARKs

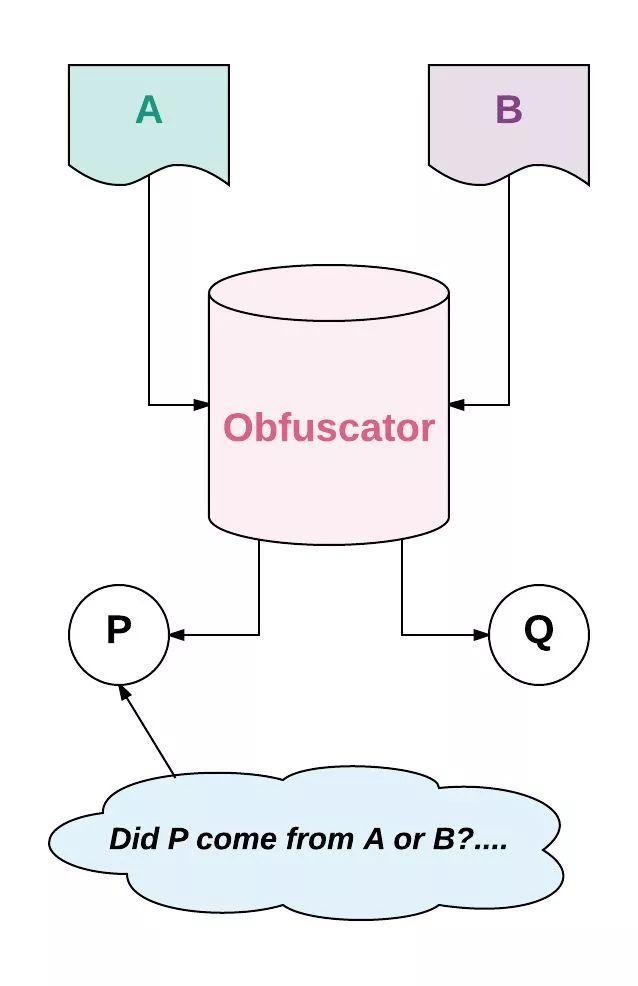
ZK-SNARKs 有一个更年轻更闪耀的同胞兄弟：ZK-STARKs，其中“T“表示”透明（transparent）“。ZK-STARKs 解决了 ZK-SNARKs 的一个主要缺陷：需要依赖一个可信的启动阶段。由于 ZK-STARKs 只依赖哈希和信息论，因此它更简易；由于不再使用椭圆曲线和指数假设，面对量子计算机时它更加安全。

总而言之，尽管在上述的零知识证明隐私保护方案的前沿研究中，我们取得了惊人的进步，但仍然有许多工作需要去做。我们需要对零知识证明的函数库进行实证研究和实践检测，使其成熟。我们需要在不同公链上对 zkSNARKs 和 zkSTARKs 进行实验。在真实世界的场景里，Zcash 则需要在扩展性上给出令人满意的使用案例。我们离这些仍有很长的路要走。

**代码混淆**（Code Obfuscation）

另一种隐私保护机制是代码混淆。该方案要找到一种方式来混淆程序 P，混淆器（obfuscator）会产生第二个程序 O(P)=Q，使得在给 P 和 Q 相同的输入时，产生相同的输出。但是 Q 不会揭露与 P 内部构造相关的任何信息。这使得我们可以在 Q 内部隐藏如密码和身份证等的私密信息，但同时在程序里使用这些信息。

虽然研究者已经证明完全的黑盒混淆器是不可能实现的，但不可区分混淆器（indistinguishability obfuscation）是可以实现的，这是一种概念上弱化的混淆器。不可区分混淆器 O 的定义是，如果你使用两个同等的程序 A 和 B（如把相同值输入到 A 或 B 里去产生相同的输入）计算得到 O(A)=P 和 O(B)=Q，则在无法进入程序 A 或 B 的情况下，则在计算上分辨 P 来自于 A 还是 B 是不可行的。

[[](http://upload-images.jianshu.io/upload_images/10818463-9b5f75027c31e1b2?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)](http://upload-images.jianshu.io/upload_images/10818463-9b5f75027c31e1b2?imageMogr2/auto-orient/strip%7CimageView2/2/w/1240)

最近，研究者 Craig Gentry, Amit Sahai 等人完成了不可区分代码混淆器。然而，该算法的计算开销十分高昂。

如果开销问题可以得到改善，则能带来巨大的潜在好处。

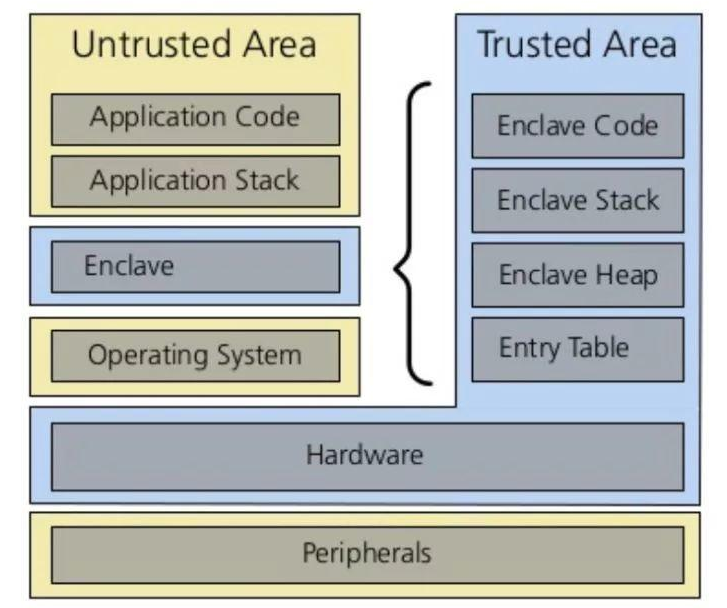
举个例子，假设以太坊的智能合约里有  Coinbase 的密码。则我们可以写出这样一个程序：当智能合约满足了特定条件后，合约通过中间节点初始化与 Coinbase 的 HTTPS 会话，使用密码进行登录，然后执行交易。由于合约里的信息被混淆了，因此中间节点或区块链的其他参与者都没法修改发出的请求和获取用户密码。

**预言机**（Oracle）

在区块链世界里，预言机是指在智能合约和外部数据源之间传递消息的角色。它在链上智能合约和链下外部数据源之间充当数据的运输者。因此，一种保护信息隐私性的方法是使用预言机从外部数据源中取出隐私数据。

**可信任执行环境**（Trusted Execution Environments）

可信任执行环境（TEE）是位于主处理器里的一个安全区域。在 TEE 里加载运行的代码和数据会得到隐私性和完整性的保护。TEE 可以与面向用户的操作系统并行运行，但比后者具有更好的隐私性和安全性。



Source: <https://www.slideshare.net/JavierGonzlez49/operating-system-support-for-runtime-security-with-a-trusted-execution-environment-phd-thesis>