目前，以太坊生态系统中最大的挑战之一是隐私。
默认情况下，访问公共区块链的所有内容都是公开的。
这不仅意味着资产和交易活动，还意味着ENS域名、POAP、NFT、灵魂令牌等

。 使用一系列以太网APP意味着你的许多活动将向其他任何人公开显示和分析。
我们有必要改善这种状况。
但到目前为止，关于改善隐私的讨论主要围绕着特定的用例

即，ETH和主流的ERC20令牌的隐私保护转移。
本文介绍了各种工具的机制和用例，并在许多其他情况下改善了以太网的隐私状态

即“隐藏地址”(stealth addresses )的概念。 隐藏的地址系统是什么？
假设爱丽丝要把资产转移给鲍勃

可能是一定数量的密码货币，例如1 ETH、500 RAI等，也可能是NFT。
鲍勃收到资产时，他不想让别人知道该资产的受益人就是他。
隐瞒发生转移的事实是不可能的

，特别是在传输了链上只存在一个副本的NFT的情况下，隐藏谁是接收者可能更有可能。 更想要Alice和Bob的，应该是这样的支付流程系统。 即

，Bob向Alice (或支持ENS域名)发送可以接收付款的地址代码。 只有这样，Alice )或其他人才能向他发送资产。 这与当前的支付工作流程大致相同。

必须注意的是，这种隐私性与Tornado Cash提供的隐私完全不同。 Tornado Cash可以隐藏关键替代资产(如ETH和关键ERC-20
)的转移(常用于个人发送给自己)。

但是给鲜为人知的ERC20转账添加隐私非常弱，不能给NFT转账添加隐私。
如上所述的基于密码货币的支付的一般工作流程提高了隐私，也就是说

没有人知道资产受益人是Bob，并且工作流程没有更改。
隐藏地址是Alice或Bob可以生成的地址，但只能由Bob控制

。 Bob生成支出密钥“spending
key” ，将其作为秘密，并使用该密钥生成隐藏元地址“stealth meta-address”。 他将这个元地址传递给Alice (或注册到ENS )。
Alice可以对该元地址执行计算，以生成属于bob的隐藏地址。
然后爱丽丝可以把她想寄的资产送到这个地址

Bob完全管理这些资产。 在迁移过程中，Alice在链上公开了附加的加密数据“临时公钥”，以便Bob发现这个地址是他的。

另一种看法是为隐藏地址提供与Bob相同的隐私属性，并为每笔交易生成新地址，但不需要Bob交互。 隐形地址方案的完整工作流程如下。

1 .鲍勃生成根支出密钥(m )和隐藏元地址(m )。 2.Bob添加了用于注册(m )是bob.eth的隐藏元地址bob.eth的ENS记录。

假设Alice知道鲍勃的地址是鲍勃. eth。 爱丽丝在ENS上寻找鲍勃的隐藏元地址(m )。 4.Alice生成只有她知道的临时密钥

而且，她只能用一次。
5.Alice使用算法将她的临时密钥与bob的元地址相结合来生成隐藏的地址。
她现在可以把资产送到这个地址。

6.Alice还生成了她的临时公钥并将其公开到临时公钥注册表中。
(这可以在与向该隐匿地址发送资产的最初交易相同的交易中进行。 7
.为了让鲍勃发现隐藏的地址

，Bob必须扫描临时公钥注册表，以搜索自上次扫描以来每个人公开的整个临时公钥列表。 8.BOB尝试在每个临时公开密钥上与根支出密钥组合生成隐藏地址

，检查该地址中是否有资产。 如果有，鲍勃计算该地址的支出键，并记住它。
这一切都取决于密码欺诈的两种用途。 首先

需要一对算法来生成共享密钥(shared secret )。 一个算法使用Alice的临时密钥和Bob的元地址，另一个算法使用Bob的根支出密钥和Alice的临时公钥

。 这可以用多种方法来完成； Diffie-
Hellman密钥交换是确立现代密码学领域的成果之一，并将其实现。
但是，仅仅分享秘密是不够的。 如果仅从共享秘密生成秘密密钥

Alice和Bob都可以从这个地址消费。 还增加了密钥盲化机制。 Bob可以组合共享密钥和根成本密钥的一对算法另一方面，Alice可以将共享密钥和Bob的元地址结合起来，Alice可以生成隐藏地址，Bob可以生成该隐藏地址的支出密钥

中选择所需的族。 所有这些都不需要在隐藏地址和bob的元地址之间创建公共链接(或一个隐藏地址和另一个隐藏地址之间)。 使用椭圆曲线密码学隐藏地址

使用椭圆曲线密码学隐藏地址最初是Peter
Todd于2014年在比特币的背景下引入的。 这项技术的结构如下。
Bob生成密钥(m )，计算为M=G * m

其中，g是椭圆曲线的公认生成点。 隐藏源地址为(m )。 Alice生成临时密钥(r
)，发行临时公开密钥R=G * r。

Alice可以计算一个共享密钥S=M * r，并且Bob也可以计算相同的共享密钥S=M *
r。 一般来说

另外，在比特币和以太网(包括正确设计的ERC-4337账户)中，地址是包含用于验证来自该地址的交易的公开密钥的散列。 因此，只要计算公钥就可以计算地址。
为了计算公钥

，Alice或Bob可以计算p=mg*hash(s )。
要计算该地址的私钥，Bob可以计算p=mhash ) s。 这满足了上述所有要求

而且，很简单。 也有EIP试图在以太网上定义隐藏地址标准，在支持此方法的同时，为用户开发其他方法提供了空间。 例如，它支持Bob具有单独的支出和视图键

也可以使用不同的密码学来实现量子安全)。 现在，你可能觉得藏身之处并不太难，但理论知识已经很扎实，招聘只是一个实施细节。
但是，问题是，真正有效的实现需要一些重要的实现细节。

隐性地址和交易费用的支付假设有人给你发了NFT。
如果你想确保隐私，他们会把它送到你管理的隐藏地址。
扫描链上的临时公钥时，钱包会自动找到该地址

。 现在可以自由证明NFT的所有权，也可以转让给别人。
但一个问题是，由于该账户的ETH余额为0，也无法支付交易费用。
即使是ERC-4337令牌付款人也无效

因为它只适用于可替代的ERC20令牌。 而且，不能从主要的钱包里发送ETH。
因为这样的话，就会创建公开可见的链接，就没有隐私了。

有一个简单的方法可以解决这个问题。 只需使用ZK-
SNARKs转移资金并支付费用即可。
但是，这样会消耗大量的Gas，仅一次转账就额外消耗数十万的Gas。

另一种聪明的方法是信任专用的事务聚合器(MEV术语搜索者searchers )。 通过这些聚合器，用户可以一次进行支付，并购买一组可用于支付链上交易的“tickets”

。 如果用户需要将NFT放在不包含其他内容的隐藏地址上，他们会为聚合器提供其中一个ticket，然后使用Chaumian盲方法进行编码

。 这是20世纪80年代和90年代提出的集中式隐私保护电子现金计划中使用的原始协议。 搜索者接受ticket，并重复免费将交易包括在包中，直到交易在一个块中被成功接受。

假设隐藏地址、隔离支出和视图键Bob想要另一个路径支出键和视图键，而不是可以执行所有操作的一个主“路径支出键”。
这个视图键可以看到鲍勃所有的隐藏地址

但是，不能支出。 在椭圆曲线的世界里，这可以用非常简单的密码技术来解决。
Bob的元地址(m )当前格式为(k，v )，对G * k和G * v进行编码

其中k是支出键，v是视图键。 共享密钥当前为S=V * r=v * R。
其中r保留Alice的临时密钥，r保留Alice发布的临时公钥。

隐藏地址的公钥为p=kg*hash(s )，私钥为p=k hash(S ) s )。
第一步(生成共享秘密)使用视图密钥

另外，第二步骤(Alice和Bob并行算法生成隐藏地址和它的私钥)使用根支出密钥。
这有很多用例。 例如，在Bob希望接收POAP的情况下

，Bob可以在他的POAP钱包或不太安全的Web界面上显示密钥，扫描链条，显示他的所有POAP。 您不需要在这个接口上花费那些POAP的权限。

为了便于扫描整个隐藏地址和易扫描临时公钥集，一种技术是在每个临时公钥中添加视图标签。
使用上述机制执行此操作的一种方法是将视图标签设置为共享密钥的一个字节()

、S modulo 256的x坐标或hash(s )的第一个字节。 这样，bob只需对每个临时公钥执行一次椭圆曲线乘法运算来计算共享密钥，并且由于有视图标签，扫描也变得容易。

虽然源地址和抗量子安全上的方案依赖椭圆曲线，但该方案效果很好，但很遗憾容易受到量子计算机的攻击。 需要切换到抗量子算法

。 有两个自然候选：椭圆曲线同源和格(lattices )。 椭圆曲线同源是基于非常不同的椭圆曲线的数学结构，具有线性特性，可以使用与上述相同的加密技术

但巧妙地避免了量子计算机易受离散对数攻击的循环群的建立。 基于同源密码学的主要弱点是高度复杂的基础数学及其复杂性下隐藏攻击可能性的风险

。 一些基于智人(密码学)的协议去年遭到攻击，但其他协议仍然安全。
同源的主要优点是比较小的密钥大小和直接移植基于椭圆曲线的方法的能力。 A 3-isogeny in CSIDH

格(lattices )是非常不同的密码结构，依赖于比椭圆曲线同形的简单数学，可以非常强大，例如完全同态加密。 隐藏的地址方案可以建立在格上

尽管设计的最佳方案是悬案。 但是，基于格的结构往往密钥大小较大。
同态加密，格的应用

。 FHE还可以用于以各种方式帮助隐形地址协议。 Bob外包可帮助计算检查整个链中是否包含资产的隐藏地址，而无需公开视图密钥。
第三个方法是从通用黑匣子原语中构建隐藏住所

。 该方案的共享密钥生成部分直接映射到密钥交换，这是公钥密码系统的重要组成部分。 更难的部分是让Alice只生成隐藏地址而不是支出键，让Bob生成支出键的并行算法。 不幸的是

不能使用构建公钥加密系统所需的更简单的组件来构建隐藏地址。
一个简单的证明是可以用隐藏地址方式构建公钥加密系统。 如果爱丽丝想给鲍勃加密消息，她可以发送n笔交易每个交易都被发送到bob的隐藏地址，或者被发送到她自己的隐藏地址，bob可以看到他接收的交易来读取消息。 这是很重要的。
数学证明不能只使用散列进行公钥加密

因为只用散列就能证明知识为零，所以隐藏的地址不能只用散列。
这确实是使用比较简单成分的方法。
零知识证明可以由散列和[密钥隐藏]公钥密码构成

。 鲍勃的元地址是公共加密密钥加上散列h=hash(x )，他的支出密钥是对应的解密密钥加上x。
要创建隐藏地址，Alice将生成值c将Bob可读C加密作为临时公钥公开。 此地址本身是ERC-4337帐户，其代码通过向交易请求零知识证书来验证交易并证明值为x和c的所有权

，使其成为k=hash(hash(x )，c )。 其中k是帐户代码的一部分。
如果知道了x和c，鲍勃就可以自己重建地址和代码。

) c )的加密不会告诉鲍勃以外的任何人任何信息。 另外，(k
)是散列，关于c的事情几乎不公开。 钱包代码本身只包含(k )

，c )私有意味着k )不能追溯到h )。 但是，这需要STARK。
最终，我认为后量子以太体的世界很可能与使用了很多STARK的应用相关联

因此，此处描述的聚合协议提倡将所有这些STARK组合为一个递归STARK以节省空间。 隐藏的地址和社交恢复，以及多L2钱包已经很久了

，我一直对社交恢复钱包感兴趣。 社交恢复钱包有多个签名机制，可以由机构、您的其他设备或朋友的某种组合共享密钥。
如果主密钥丢失，大多数密钥将允许恢复帐户访问。 但是

社交恢复钱包不能很好地与隐藏地址结合。 如果必须恢复帐户(更改控制帐户的私钥)，则还必须执行步骤以更改n个隐藏wallet的帐户验证逻辑。 这将需要n笔交易

以高昂的费用、便利性和隐私成本为代价。

对社交恢复与多种L2协议的交互也存在同样的担忧。 如果您在Optimism、Arbitrum、StarkNet、Scroll和Polygon上拥有帐户，则可以有10个以上的并行实例进行扩展

且每个实例都有一个帐户，则更改密钥可能是一项非常复杂的操作。
更改多个链中多个帐户的密钥是一项很大的工作。

也许可以使用一些自动化软件，在两周内以随机间隔将资产移动到新的隐藏地址，以降低基于时间的关联的有效性。 但是，这并不完美。
另一种方法是在家长之间秘密共享根密钥，而不是使用智能约定进行恢复。 但是

中选择所需的族。 这将消除家长禁用恢复帐户的权限的能力，存在长期风险。
一个更复杂的方法是零知识的证明。 这样可以跨越许多帐户，甚至许多L2协议

中选择所需的族。 在基础链上或某些L2上的某个地方，它由单个k值控制。
更改此值将更改所有帐户的所有权。 所有这些都不会失去多个帐户之间的联系。
结论现有基本隐性住所可快速实施

并且，可以显著提高以太网上用户的隐私。
出于其他隐私相关的原因，我认为钱包应该开始转向更原生的多地址模式。
例如，为与之交互的每个APP应用程序创建新地址可能是一种选择。 但是

、隐性地址会带来长期的可用性问题，例如社会恢复困难。 从长远来看，这些问题是可以解决的，但藏身之处的生态系统看起来确实很大程度上依赖于零知识的证明。