Keep Network：公链的隐私层

最近更新：2017 年 10 月 15 日

作者：

Matt Luongo：mhluongo@gmail.com

Corbin Pon： corbin.pon@gmail.com

译者：Erica（erica@dappchaser.com)、Frank

摘要

本文介绍了 Keep，一种新的隐私元语（primitive，译者注：程序设计术语，程序执行中不可被中断的基本操作），用于在公有链上开发智能合约，实现私密信息的安全存储和使用，并且支持基础设施，包括 keep 市场和代币。

我们对隐私基础设施采取渐进式处理，在以太坊公共网络上推向市场，并进行迭代，适应其他公有链和跨链的使用。

1. 动机

1.1 公有链的讽刺性

公有链为金融技术带来了前所未有的透明度和可审计性。记录不可更改，可验证，并且抗审查。

幸的是，对于许多潜在用户而言，这些优势也是弱势。

对于公有链启动的每个金融用例，伴随其公共身份而来的是其他限制。比特币被誉为比传统金融系统更私密的支付方式，但是熟悉这项技术的人都知道，虽然它可能具有抗审查功能，但默认情况下，比特币网络当然不是私密的[1]。被引

入以太坊的开发者迅速学会调整自己的期望[2]——将所有合约状态发布到区块链，并且可以被利益竞争者轻松读取。

比特币和以太坊项目的开发者已经认识到这些问题。

保密交易（Confidential Transactions, CT）[3]是一项仍在进行的努力，它旨在通过侧链为比特币网络带来更好的隐私性，当然也会带来可互换性（Fungibility）[4]。Zerocash 项目[5]将零知识证明应用于比特币，创造出 Zcash[6]，一种使用 zk-SNARKs 确保交易隐私性的加密货币。

早在 2014 年 12 月，以太坊的创始人之一，Vitalik Buterin，就曾探索用安全多方计算（sMPC）[7]来解决这个问题。在最近的著述中，Buterin 分享说，“当（他）和其他人与公司讨论在区块链上构建应用程序时，总会出现两个主要问题：可扩展性和隐私性” [8]。

公有链的可扩展性是其被广泛采用的一个障碍。加密货币领域的一些顶尖人才[9][10][11]正在处理多个数量级的改进。然而，隐私性并未引起同等关注，尤其是在智能合约中。

在今天的公有链上，智能合约的基本用例（包括满足某些条件后发布私密，评估借款者的贷款风险，以及签署信息和交易）都举步维艰。

1.2 现行方法

在实践中，开发者已找到许多方法，以构建使用私有数据的去中心化应用。

1.2.1 “哈希暴露”模式（hash-reveal pattern）

公链的一种常见模式是保留应用程序用户的私人数据。合约可以接收和操纵私有数据的哈希，通常称为承诺（commitments）[12]，而用户可以保留原始数据，直到私有数据在链下暴露。 我们称之为“哈希暴露”模式。

对于许多应用程序来说，这种方法是令人满意的。较于典型的 Web 应用程序，存在一个显著的隐私优势——没有中心化的第三方数据库处于风险之中。在许多用户之间分散存储，对于攻击者来说，意味着分布式、多样化的目标（译者注：攻击难度更高）。

然而，也存在一个值得重视的缺点。 “哈希暴露”模式要求交易方的所有用户都在线，监视系统，在必要时提供私有数据，并针对其他用户提供的私有数据验证哈希。

这项要求使“哈希暴露”模式对于复杂协议来说并不灵活，不适用于那些不完全围绕活跃参与者运行的系统，比如去中心化自治组织（DAO）等。

1.2.2 私有链

对隐私限制的另一种回应（主要来自金融业）是建立私有链，即所谓的“许可账本”（Permissioned Ledger） 。

这些系统以可信或半可信的方式运行。它们可以使用 RAFT 之类的系统达成共识，而不使用工作量证明，或者为对抗性网络而设计的其他共识机制。

摩根大通（旗下的区块链项目）Quorum 就是这样的系统之一[13]，它是以太坊的分叉，支持私有合约状态和网络参与者之间的消息传递。另一个同类系统是微软最近宣布的 Coco Framework[14]，它在现有的私有链上提供数据隐私。

这些系统解决了隐私问题，但牺牲了公有链的许多好处——去可信任化、公共责任、抗审查和无需许可的创新。

1.2.3 零知识证明

零知识证明已被用于维护公有链上的隐私，其中最著名的是 Zcash[6]项目。

零知识证明允许一方（证明方）向另一方（验证方）证明陈述，而无需透露用于证明该陈述的知识。例如，证明者可以通过加密验证者选择的消息来表明他

们有权访问私钥。验证者可以通过使用公共密钥解密密文来简单地检查证明。但是，私钥仍然是秘密的。

与该领域更相关的是，零知识证明可用于一方证明他们有权使用资金，或者在 Zcash 的案例中，可用于一方向矿工证明，根据网络共识规则，交易是有效的。

零知识证明可用于在公链上构建私有金融系统。但是，它们本身就没有允许将私有数据从一方安全地委派给另一方，并且也需要始终在线，因为“哈希暴露”模式的要求。

零知识证明是功能强大的加密工具，可以与其他技术结合使用，以安全地委派秘密访问和计算（请参阅第 3.1 节）。

2 介绍 KEEP

公有链的透明性与许多自治智能合约对私有数据的需求之间存在不匹配，为了解决这一点，我们引入了 keep。

Keep 是私有数据的链下容器。Keep 允许合约管理和使用私有数据，而无需让数据暴漏于公有链上。

2.1 Keep 的运行

尽管 Keep 维持链下状态，但它们是由链上的合约布建和传达信息。我们将根据这些链上操作介绍 Keep。第 3 节和第 4 节介绍了 Keep 的实际运用，包括如何保证安全。

Keep 的运行

创建（Create）： $Contract_{owner}$ 发布创建请求，包括初始保证金和公钥 $K_{Contract_{owner}}$。

接受（Accept）： 作为 $Keep_{accepted}$ 的 Keep，将发布一个或多个公钥接受 $K_{Keep_{accepted_i}}$，表示准备就绪。

填充（Populate）： $Contract_{owner}$在链上发布初始机密，并由一个或多个 $K_{Keep_{accepted_i}}$总体或分份加密，或发布一个机密规范，以待生成。

授权（Grant）： $Contract_{owner}$发布另一个合约地址，$Contract_{delegate}$，以及权限级别 P_{read}或 P_{admin}。

计算（Compute）： $Contract_{owner}$或 $Contract_{delegate}$发布一个函数来计算机密 $F(S,...)$以及其他参数到 F。最初是 $F \in \{f_{identity}, f_{rsa}, f_{ecdsa}\}$，尽管已计划另外的功能。

结果（Results）： $Keep_{accepted}$会通过一次或多次调用，以整体或者部分的形式发布其计算结果。

关闭退出（Shutdown）：具有 P_{admin}权限的 $Contract_{owner}$或 $Contract_{delegate}$发布关闭退出请求。

2.1.1 创建和填充

合约 $Contract_{owner}$通过将请求发布到区块链，请求 Keep。一旦作为 $Keep_{accepted}$的 Keep 接受请求和完成链下初始化，它将使用一组公钥响应请求，调用合约可以使用公钥与 Keep 进行私密通信。

一旦创建 Keep，可以通过多种方式进行填充。去中心化应用（dApps）可以将私密数据发布到区块链，通过 Keep 的公钥进行加密，也可以将数据发送到 Keep 的链下。或者，Keep 可以使用伪随机数据自行填充。

2.1.2 在链上发布数据

Keep 旨在根据其机密计算函数，并将结果发布到区块链。

最初，keep 将在链上发布其机密，不经修改或使用 $Contract_{owner}$提供的公钥进行加密。

这能够实现当今公共智能合约中难以实现的功能，例如机密暴露的失能开关（dead man switch，译者注：即操作者因死亡等原因失去行为能力，能自动起作用的开关），可在各种去中心化市场计划中使用。

Keep 可以扩展到通过其他多种方式使用机密，包括用作对称加密和签名的关键材料。

2.1.3 访问管理

Keep 合约所有者 Contractowner 可以将对 Keep 的访问权委派给其他合约。

可以分别授予读取权限和管理员访问权限，从而允许另一个合约 i（$Contract_{delegate}$）请求发布 Keep 的内容（读取权限，P_{read}），或进一步委托给其他合约的访问权限（管理员权限，P_{admin}）。 所有者（$Contract_{owner}$）也可以撤消自己的访问权限。

访问管理可实现了多方机密托管和机密访问的可审计性。

2.1.4 销毁

根据不同的用例，Keep 的生命周期可长可短。合约可以请求关闭一个 Keep，也应处理意外终止的 Keep，这些情况将在 5.2 节中详细介绍。

（reviewing）

（To be continued）