Hcash

TECHNICAL YELLOW PAPER

HCASH 技术黄皮书

目录

一、背景

以比特币[1]为代表的基于 UTXO 的区块链和以以太坊[2]为代表的基于账户的区块链向我们打开了新世界的大门。比特币和以太坊的成功，证明了区块链技术的价值和未来的巨大潜力，在这个过程中我们也看到了当前的区块链技术存在的一些不足。

目前的区块链世界已经形成了多链共存的格局，但面临着一个共同性的问题：链间价值无法自由流通，客观上形成了价值孤链，极大地制约了区块链行业生态的发展。

基于哈希算法和 ECDSA 椭圆曲线密码的安全技术给目前的区块链网络提供了安全保护。然而，随着量子计算机技术的进步，未来量子攻击必定会给现有的区块链网络带来极大的威胁。

HCASH 项目在启动之初，就提出了七大愿景：链间价值互通互联、抗量子、PoW+PoS 混合共识、智能合约、隐私保护、区块链+DAG 双重侧链、DAO。实现区块链间价值互通互联和抗量子签名是 HCASH 开发团队第一阶段技术突破的重点。

经过一年的进展，我们的开发团队已经取得了突破性的进展。本文档为 HCASH 黄皮书 V1.01 版本，在本版黄皮书中，我们将展示当前阶段已取得的技术成果，技术开发路线图的相关更新。

随着技术开发的不断深入，我们会持续升级本文档，使其与技术进展保持同步。欲了解 HCASH 的最新信息、技术开发进度，请关注官方网站 https://www.h.cash。

（因为格式和语法的原因，中文版黄皮书出现的所有算法、公式仅供参考，以英文版为准。）

二、HCASH 的演进

在我们的设计蓝图中，HCASH 是一个链接各种区块链技术的新的底层技术平台，让基于信任的价值在不同的区块链系统中自由流通。

在 HCASH 的技术白皮书里，明确定义了跨链、抗量子、PoW+PoS 混合共识、智能合约等具有革命性意义的特性作为技术突破的方向。在 HCASH 的开发和升级演进过程中，社区快速壮大，技术解决方案也在不断成熟。为了更快地实现技术突破，同时保持社区的稳定，我们计划升级后的 HCASH 采用“强耦合、双聚焦”的双链双币机制，在两条并生链的架构上，实现 HCASH 的开发和升级。

2.1 并生双链生态

升级后 HCASH 将实现并生双链的生态模式，由原有的 Hshare 链升级后的主链 HyperCash(简称 HC)和由 HCASH 孵化出的 HyperExchange(简称 HX)主链组成，HyperCash 和 HyperExchange 互相协作，共同支持 HCASH 实现设计愿景。

作为生态中的两个核心枢纽，二者可以深度共享技术特性和生态资源，HyperCash 为 HyperExchange 提供维持系统稳定所需的价值代币，HyperExchange 为 HyperCash 提供更多的价值应用场景；HyperCash 为 HCASH 生态提供区块链底层技术，HyperExchange 则为 HCASH 生态提供区块链间价值互通互联方案，为 HCASH 更快、更多地引进资源，共同壮大 HCASH 生态。

根据规划，原有的 Hshare 链升级为 2.0（即升级为 HyperCash）后，将全面实现抗量子特性，支持 PoW+PoS 的混合共识，专注于区块链底层技术的深度研发；HyperExchange 通过创新的 BMT 协议为 HCASH 生态实现了区块链间的价值互通互联，并将在此基础上实现区块链和非区块链分布式账本（如 DAG）之间的信息与价值自由流通，为构造多资产分布式商业应用生态打造基础。

HyperExchange 和 HyperCash 共同组成了强耦合、双聚焦的双链双币机制，基本实现了 HCASH 的设计愿景。

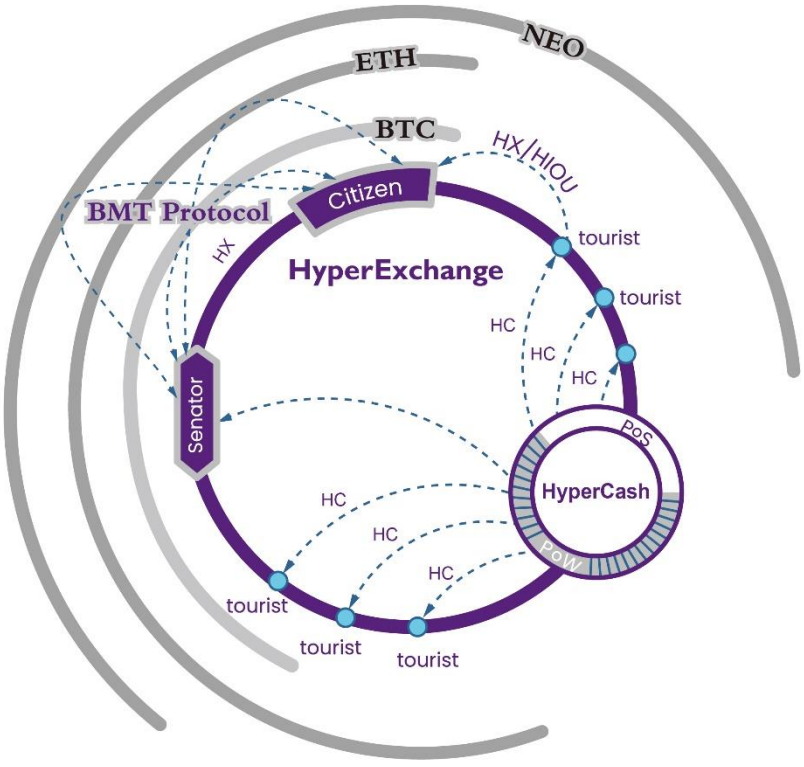


图 1：HCASH 并生双链生态示意图

2.1.1 HyperCash

对于区块链项目来说，研发一条符合 HCASH 发展愿景的链，通常来说，有以下两种路径：

1、开发一条全新的链

2、从已有的开源代码库中分叉出一条链

如何在这两种途径中做出选择，HCASH 将考虑重点放在了两个方面：是否能够保证实现 HCASH 项目设计愿景的重要功能以及两种方式各自需要的时间。

HCASH 团队发现：Decred（以下简称 DCR）所运行的开源链的某些特性，与 HCASH 的项目愿景中的部分目标不谋而和。同时，DCR 作为比特币的一个分叉，在其结构上引入或创建新的流程，将具备良好的兼容性。

采用了 DCR 的开源代码后，HyperCash 具备以下优势：

1、去中心化的自主治理模式

众所周知，区块链的社区治理一直是个令人头疼的难题。类似于比特币的机制，由于矿工权力过大制约了社区的发展，且决策效率低，容易分叉；而类似于 ZCASH 的机制，又会由于过于中心化，降低了社区的参与度。

HyperCash 将采用 DCR 提出的社区治理机制——自主治理模式。

DCR 的技术为全球首个成功的直接链上用户激活投票提供了支持，这种创新型投票模型加强了利益相关者的自主权，并允许从一组规则无缝过渡到另一组规则。去中心化决策制定和自我筹资使得这种模式能够建立一个强大且不断发展的社区，而不受第三方影响。

我们认为，这种社区自主治理的模式，可以照顾到所有利益相关方，可以保证决策的效率，亦可以保障社区永续健康地发展。这将实现 HCASH 愿景中的“自治性”。

2、PoW+PoS 混合共识机制

在公有链领域，一直以来一个突出的矛盾就是去中心化和性能之间的矛盾。

采取 PoW 机制的区块链，在去中心化共识方面做得很好，但性能较低，无法支持大规模的商业化应用。

而采取 PoS 机制的区块链，又会产生过于中心化、权力过于集中的现象。

PoW+PoS 混合共识的混合挖矿模式，可以综合二者的长处。HyperCash 采用 PoW+PoS 混合挖矿的模式，就是基于这个层面的考虑。

3、去中心化矿池

用户以个人名义独自进行 PoS 挖矿，则必须保持桌面钱包运行，如果钱包没有运行，可能导致用户错过网络验证票的时机（在票被添加到票池的时候和票被选为选票的时候），从而无法获得权益奖励的机会。

因此，在无法保持钱包 24 小时运行的情况下，持币者可以选择加入权益矿池。

大多数区块链中的矿池都是中心化的，存在不安全、不透明的风险。

去中心化矿池一个最大的好处就是：用户不会将代币的控制权交给矿池，且矿池是内置在底层协议中的，权益矿池会在协议中预先确定费用，但它无法获取用户资产，也无法领取用户奖励。

升级后的 HC 主链具备去中心化矿池的功能，资产更安全，收益更透明。

4、BLAKE256 算法

BLAKE256 算法是一种非常安全的算法。该算法从 2012 年发布之日至今，没未发现过针对该算法的有效攻击。

众所周知，包括比特币区块链在内的大多数区块链项目采取的都是 SHA2 的算法（SHA2 是一个哈希函数家族，其中包括 SHA256、SHA384 和 SHA512 等。例如比特币采用的就是 SHA256 的算法），SHA2 可以看作是安全系数更高的 SHA1，但它无法抵御哈希长度扩展攻击，可能存在漏洞，且跟 BLAKE256 相比，SHA2 在性能和速度方面明显要处于劣势。

在技术深度上，BLAKE256 可以与 SHA3 比肩。SHA3 跟 BLAKE256 一样安全，但是同样在速度方面处于下风。

HC 主链采取的也是 BLAKE256 算法，可以保障系统的安全和高效。

基于上述情况，HCASH 团队在经过慎重考虑后认为：从 DCR 的开源数据库中分叉出一条链可以在保证项目开发进度的同时实现 HCASH 设计愿景中的重要功能。

事实上，HCASH 一直致力于通过引领开源研发的方式成为区块链领域中一个有建树和连接各方的榜样。HCASH 认为，相比于从零开始构建一条新的主链，选择通过使用开源数据并继续开发，实现更多的优质特性，是一种更高效、更有保障的技术解决方案。

HyperCash 通过技术开发，将具备更多的独有特性：

首先在安全方面，HCASH 在项目启动之初就将抗量子技术作为优先考虑的关键功能之一，尽管目前只有极少数区块链项目注意到了量子计算机的发

展趋势，但随着量子计算机技术的进步，未来量子攻击必定会给现有的区块链网络带来极大的威胁。

经过近一年的技术开发，我们已经基本实现抗量子签名。具体的技术实现细节，将在后文中详述。

安全以外，隐私性也是 HCASH 技术开发的重点。为了实现更好的隐私保护功能，HCASH 有以下两方面的具体实践：一是借鉴 ZCASH 的零知识证明。二是与蒙纳士大学建立区块链技术联合实验室，研发新的抗量子环形签名技术。联合实验室的 Joseph Liu 博士发表了“环形签名”领域的专业论文 ：作为隐私层，环形签名可以通过将交易进行分组，但只在组内随机选取一人进行签名并且让其对签名来进行验证来增加对转账匿名性的额外保护。关于抗量子环形签名的进一步研发还在进行中。有关抗量子环形签名和零知识证明的具体技术实现细节，将在后文中详述。

作为 HCASH 生态双枢纽之一的 HC 主链，将会持续进行底层链技术的深度研发，并对相应技术特性进行针对性的升级，抗 ASIC、智能闪电网络协议 HAILP 等已列入下一步的开发计划，而 HCASH 设计愿景中的跨链的功能将由另一个核心枢纽 HyperExchange 完成。

2.1.2 HyperExchange

HyperExchange 作为 HCASH 并生双链生态的另一个核心枢纽，它是由 HCASH 孵化出的一条主链，它率先通过创新的 Blockchain Multi Tunnel(BMT)协议、HyperExchange Axis、Indicator、智能合约等区块链技术成果，实现了区块链间的价值互通互联。

HyperExchange 不但实现了跨链，还具备图灵完备的智能合约功能，为实现复杂的分布式商业应用打下了基础。

HyperExchange 还具备以下特性：

1、采取 RPPOM 共识，拥有去中心化矿池

HyperExchange 使用 RPPOM(Random Pool Proof of Multi-properties)的去中心化共识算法，即随机多资产权益质押共识。

在 HyperExchange 生态中有四种角色：Tourist、Candidate of Citizen、Citizen、Senator。

Tourist：包括 HyperExchange 链上的所有用户以及与 HyperExchange 互通的其他链上的所有用户。

Candidate of Citizen ：普通用户可以交纳 1 万个 HX 成为 Candidate of Citizen（此笔费用将会销毁） 。Candidate of Citizen 可以接受链上其他用户的资产质押。

Citizen：Citizen 是 HyperExchange 链上的生产者和社区治理者，负责记账和产块，它是链上的去中心化矿池。Candidate of Citizen 成为 Citizen 需要提供链上多资产质押（HSR/HC 会获得较高的权重），可以自己提供，也可以由其他用户支持提供。当 Candidate of Citizen 的质押资产不足的时候，几乎不可能成为 Citizen。

Senator：Senator 是 HyperExchange 生态中的资产管理者和社区治理者。Senator 负责共识管理其他链上的多重签名冷热地址里的资产，并且跨链流通必须通过 Senator 的共识来达成。成为 Senator 必须交纳 50 万 HSR/HC 作为责任保证金，Senator 的更替要通过 Citizen 的投票来确定。

Citizen 通过质押资产得到独立出块的机会并获得奖励。而 Tourist 也可以通过质押资产到 Citizen 获得挖矿奖励。在产出新块之后，系统会根据内置的底层协议按照质押金的权重给 Citizen 及其支持者自动分配奖励。这个质押分配的过程是通过 RPPOM 共识的方式来实现的。

这种质押机制有以下明显的优势：

（1）质押挖矿是由全网节点共同验证的，接受质押资产的 Citizen 在获得区块奖励之后，系统内置的底层协议会自动按照质押金权重进行分配，不会像中心化的矿池一样，存在很多不确定因素。

（2）类似的 PoS 矿池实际上用户需要把资金转账给矿池，而 HyperExchange 链上只是质押资产，私钥仍然由自己保管，无需转账给 Citizen，不会有安全问题，用户随时可以撤回质押资产。

2、百分百准备金，安全可靠

为了确保 HCASH 生态系统的稳定和安全，HyperExchange 的准备金比率为百分百，每个 HIOU 都有一个真实的原链资产（如 BTC、ETH）质押在原链上由 RPPOM 共识管理的多重签名冷热地址里。这确保 HyperExchange 的所有资产都不会凭空增加或销毁，每一个资产的增加或减少都一一对应着用户在原链上资产的充值或提现。

3、高效

依照 RPPOM 共识，HyperExchange 主链每 5 秒产出一个块，相对于 BTC 每 10 分钟一个块和 LTC 每 2.5 分钟一个块，交易确认速度有了显著的提升。在 HyperExchange 主链上进行 BTC 或 LTC 资产转移或者交易时的性能分别约为 BTC 主链的 120 倍，LTC 的 30 倍。

HyperExchange 的理论 TPS（每秒处理交易数）值达到 1 万,足以承载多条链上的高负荷交易。

具体对比如表 1 所示：

区块链	产块间隔	区块大小	理论 TPS
HyperExchange	5 秒	20M	10000
BTC	10 分钟	1M	7
ETH	17 秒	无上限（800万gas)	22
LTC	2.5分钟	1M	28
NEO	20 秒	无上限	1000
EOS	1.5 秒	无上限	百万

表 1：主流公链性能对比

从上述对比中我们可以看出，HyperExchange 在产块速度、区块大小和理论 TPS 上，其性能较 BTC、LTC、ETH、NEO 等链都有明显的提升，HyperExchange 的产块间隔为 5 秒，理论 TPS 为 1 万，足以应付高频率、高容量的业务需求。

尽管跟 EOS 理论上百万级的 TPS 有些差距，但需要指出的是：EOS 的超级节点要求服务器之间具有非常稳定的网络连接，并且其对服务器的性能要求很高。EOS 的产块机制是 21 个超级节点按顺序产块，规律明显，它存在被攻击的风险，诸如 DNS 欺诈、DDOS 等。

相比之下，HyperExchange 链对服务器和网络的性能要求则没有那么高，适应性更强，更容易接入其他链上的资源。并且，HyperExchange 的产块是从所有满足质押金的候选 Citizen 中随机选出的。这意味着 HyperExchange 的产块节点具备很大的不确定性，很难在网络中被发现。因此可以在机制上最大程度地规避网络攻击的风险，并且部分节点被攻击完全不会影响全网的稳定。

详细内容参看 HyperExchange 白皮书。

2.2 双币

HyperCash 和 HyperExchange 的链上资产分别为 HC 和 HX。为了整个生态的代币平衡，在适当的时机，基金会会提供一个一次性的交换方案：HC 和 HX 的兑换比例为 1:100（兑换细则以 HCASH 基金会官方公告为准）。

2.2.1 HC

HC 是 HyperCash 的链上资产，和现有的 HSR 1:1 兑换（兑换细则以 HCASH 基金会官方公告为准）。HC 是维持整个 HCASH 生态稳定的价值代币，它既可以作为 HC 主链的链上资产，参与 HC 链上的治理，又可以作为质押资产参于 HyperExchange 链上代币 HX 的产生。

2.2.2 HX

HX 是 HyperExchange 链上的功能代币，用于维持和使用 HCASH 生态内的相关功能。HX 先以 ERC20 的形式存在，等待 HyperExchange 上线后，兑换成 HX(兑换细则以 HCASH 基金会官方公告为准）。

HC 和 HX 的兑换比例为 HC:HX=1:100(兑换比例和兑换时间以 HCASH 基金会官方公告为准）

HX 的产出机制请详见 HyperExchange 白皮书。

三、HCASH 技术实施进展

3.1 跨链

HCASH 的核心技术愿景是建立一个全新的底层技术平台用以链接各种不同的区块链技术，从而让基于信任的价值在不同的区块链系统中自由流通。在我们的设计中，价值流通不仅限于各区块链之间，也包括基于区块链的分布式账本和基于非区块链的分布式账本（如 DAG）系统之间。

在近一年的时间里，我们的开发团队经过不懈的努力，已经通过 HCASH 双链生态之一的 HyperExchange 实现了各区块链间的价值互通互联。我们的开发团队会继续按照技术开发路线图的设定，推进研发进程，在目前链间跨链技术的基础上，实现基于区块链的分布式账本和不基于区块链的分布式账本（如 DAG）系统之间的价值流通。

在本黄皮书中，我们所说的跨链，是指各区块链间的跨链。

3.1.1 跨链的实现

HyperExchange 率先通过创新的 Blockchain Multi Tunnel(BMT)协议、HyperExchange Axis、Indicator、智能合约等区块链技术成果和创新，实现了区块链间的价值互通互联，为实现打通链间复杂的分布式商业应用打下了基础。

3.1.1.1 Blockchain Multi Tunnel Protocol

Blockchain Multi Tunnel Protocol（区块链多隧道协议）是 HyperExchange 独创的用于区块链间点对点信息传输的通讯协议。

当用户在 HyperExchange 中创建账户时，HyperExchange 会根据 Blockchain Multi Tunnel Protocol 生成对应的隧道账户，并与 HyperExchange 账户绑定。当发生跨链交易时，通过 Blockchain Multi Tunnel Protocol 将相应数据进行封装打包安全传递。

3.1.1.2 HyperExchange Axis

HyperExchange Axis 是 HyperExchange 中实现跨链的重要组成部分。作为一个多资产的去中心化账本，它负责记录、验证、广播跨链数据以及生成和销毁对应的 HIOU 代币（HIOU：指 HyperExchange 上对应锚定的其他资产链数字资产，例如：HXBTC 指 HyperExchange 上对应锚定的比特币资产），并完成跨链资产的转移。在 HyperExchange Axis 中，通过 Senator 的共识来保证网络安全性，HyperExchange 账户和对应的隧道账户的绑定关系由 Senator 在链上进行共识验证，且每一笔资产的跨链交易也要由 Senator 进行共识验证，每一条资产链和 HyperExchange Axis 的状态保持一致。

3.1.1.3 Indicator

Indicator 是 HyperExchange 的用户端，Indicator 负责生成 HyperExchange 生态内的相应数据，用户通过 Indicator 实现

HyperExchange 账户注册、成为生态角色、参与生态建设，例如挖矿、跨链交易等，获得相应的生态收益。

3.1.1.4 智能合约

在 HyperExchange 跨链生态里，还可以通过智能合约来实现多资产挂单求购等交易行为。包括以下几点：

挂单求购规则

- 挂单求购使用智能合约实现，一个挂单求购就是一个新智能合约。
- 挂单求购允许部分匹配，部分成交。
- 挂单求购允许撤单。
- 挂单求购的匹配成交和撤单都是调用智能合约的交易。
- 求购挂单交易，匹配挂单交易，撤销求购交易都需要手续费。

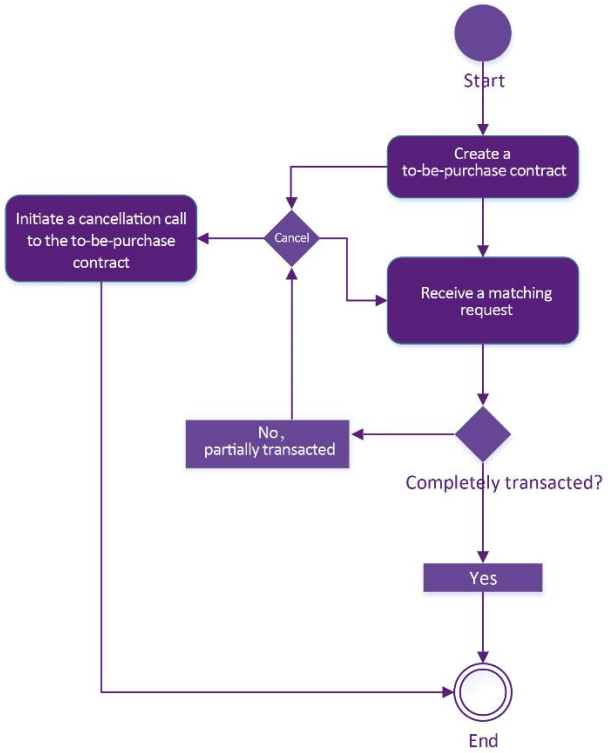


图 2：智能合约挂单求购

3.1.2 跨链的操作流程

3.1.2.1 初始化流程

1.初始化 HyperExchange，进行初始资产分配，基本参数配置。

2.HyperExchange 上 Senator 通过共识在 BTC、LTC 等资产链上分别创建多重签名账户，并把多重签名账户的地址由所有 Senator 签名后广播到 HyperExchange 上。

3、每个 Senator 交纳 HCASH 基金会规定的责任保证金，用于维护链的稳定。

3.1.2.2 账户的创建

为了完成跨链转账，Tourist 需要在 Indicator 上创建 HyperExchange 账户，Indicator 会提供隧道账户创立选项，并与 HyperExchange 账户进行绑定。

当资产链（比如 BTC）往多重签名账户充值的时候，HyperExchange 在确认后会将等量的 HIOU，发放到绑定的 HyperExchange 账户上。

在整个 HyperExchange 初始化或者是有新的 Senator 加入 HyperExchange 网络的时候，需要在资产链(BTC,LTC 等)创建或更新多重签名账户。

账户类型包括：

1、HyperExchange 账户

用户首先需要创建一个 HyperExchange 账户，用于存储、交易 HyperExchange 上的多种资产，包括 HIOU、HX 等。

2、隧道账户

当用户在 HyperExchange 中创建账户时，HyperExchange 会根据隧道协议生成对应的隧道账户，并与 HyperExchange 账户绑定。隧道账户绑定 HyperExchange 账户每日免费限额 1 万笔，超出限额需 Citizen 审核。（1 万可共识修改）

3、多重签名冷热账户

跨链资产将会保存在由 Senator 共识创建和管理的各个资产链上的多重签名冷热账户里。

3.1.2.3 跨链充值流程

用户在 HyperExchange 中除了有 HyperExchange 账户外，还拥有若干个绑定的隧道账户，通过 HyperExchange 包含的其他资产链的轻钱包组件，用户可以从其他资产链地址或从中心化交易所中充值到 HyperExchange 账户绑定的隧道账户地址里。

HyperExchange 上的 Citizen 检测到关联的多重签名的充值地址收到转账后，等待块数到达一定确认高度后， 根据充值的来源地址，找到关联的 HyperExchange 账户，并在 HyperExchange 上共识给这个账户地址相应的 HIOU（例如 HXBTC/HXLTC 等）。此时 HIOU（例如 HXBTC/HXLTC 等）默认为冻结资产(m 块)，等待 HyperExchange 块数到一定高度（比如 m+17）后共识生成的 HIOU（HXBTC/HXLTC 等）资产从冻结状态转为可用状态。

在 HyperExchange 上充值分为以下环节：

1、用户充值到 HyperExchange 账户绑定的隧道账户里，这个交易确认依赖于原资产链上确认时间，如 BTC 延迟 6 块确认，LTC 延迟 12 块确认。

2、从隧道账户转账到由 Senator 共识管理的多重签名地址里。这个交易确认同样依赖于原资产链上确认时间，如 BTC 延迟 6 块确认，LTC 延迟 12 块确认。

3、HyperExchange 上 Citizen 通过共识在 HyperExchange 上给用户生成对应的 HIOU，并等待 HyperExchange 上区块延迟 17 个区块后确认。

跨链充值的底层操作流程图如下：

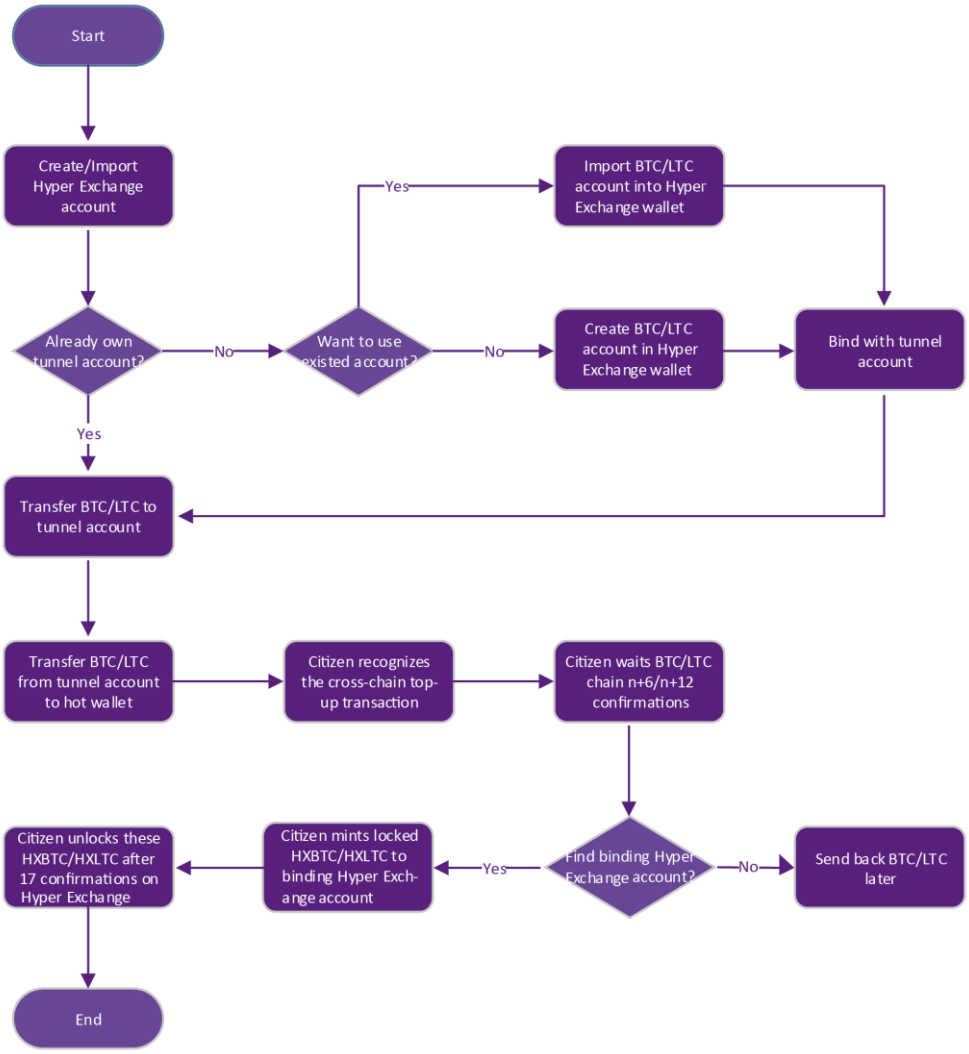


图 3：跨链充值操作流程

在实际操作中，Indicator 会在人机交互上包装简化大部分流程，用户只需要比较简单的步骤即可完成跨链充值。

3.1.2.4 跨链提现流程

用户发起提现交易，交易中包含其他资产链的提现地址。

Senator 收到提现交易后，进行签名，并在网络中广播。当轮到某个 Citizen 要出块时，Citizen 收集 Senator 对该交易的签名，判断当前是否收集到不少于 2/3 的 Senator 的签名，如符合条件则将该交易及所有收集到的签名打包进块，否则该交易将不会打包进块，由后面的 Citizen 负责处理。

交易被打包进块后，用户所拥有的对应 HIOU（比如 HXBTC、HXLTC 等）即销毁。Senator 验证后先判断多重签名热钱包的余额是否充足，如果充足则在此资产链上发出从 Senator 管理的多重签名冷热钱包到提现地址转账的签名，满足多重签名条件后则提现完成。如果余额不足，即从多重签名冷钱包中提取资产到多重签名热钱包，等热钱包里的资产充足后 Citizen 再发起转账流程。

跨链提现分为两步：

1.用户在 HyperExchange 上发起提现申请，等待 Citizen 打包后（平均 2.5 秒），等待收集不低于 2/3 的 Senator 对该提现请求的签名。若有超过 1/3 的 Senator 不认可该请求，则该交易作废。

2.当被打包的交易收集到不低于 2/3 的 Senator 的签名后，将会生成一笔对应资产链的出账交易，由 Citizen 广播到对应资产链确认后（需等待相应的块数），HyperExchange 上 Citizen 打包提现完成交易，当该交易被 HyperExchange 上认可时，即算完成提现交易。

具体技术实现过程为：用户发起提现申请后，Citizen 发起对应的从多重签名地址转账到用户目标地址的原始交易，将原始交易包装到 src_trx 中广播到 HyperExchange 上。Senator 同步到 src_trx 然后验证后，将对

src_trx 的签名包装到 sig_trx 广播并由 Citizen 记账到 HyperExchange 上。当全网收集到足够的 Senator 签名后，Citizen 把收集的签名打包后生成完整的提现交易，然后在 Indicator 内置的对应的资产链轻钱包组件中调用相应资产链的交易广播接口，广播到相应资产链的网络中去。

跨链提现的底层操作流程图如下：

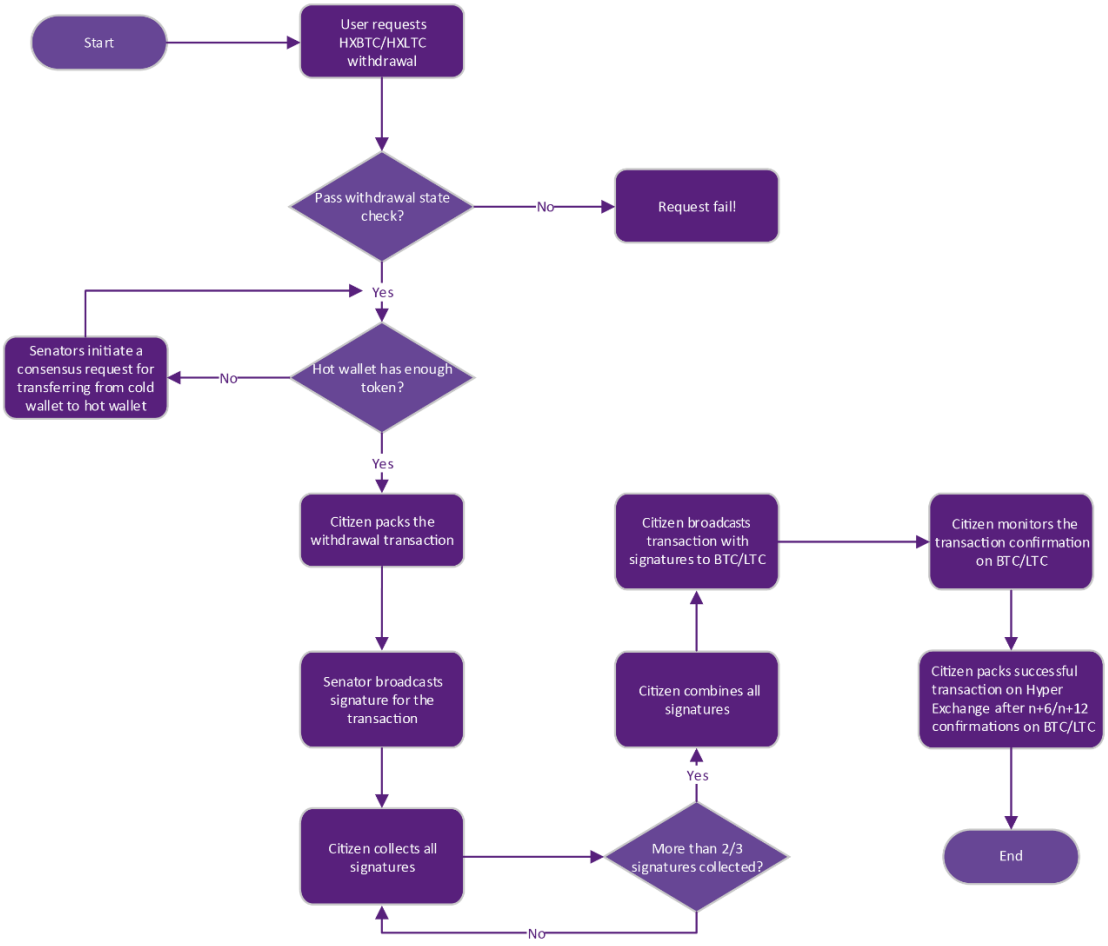


图 4：跨链提现操作流程

3.1.3 侧链资产的安全性

3.1.3.1 资金动态平衡策略

如果多重签名热钱包账户资产超过 3 倍限额则发起资金动态平衡流程，将多重签名热钱包地址内的资产转到多重签名冷钱包地址。

具体的技术实现方案为：由一个 Senator 发起资金动态平衡流程，链上打包后，其他 Senator 会进行验证，验证成功后会将签名交易广播，然后由 Citizen 打包到 HyperExchange 上。当 Citizen 收集到不少于 2/3 的 Senator 的签名后，合成 HIOU 对应的原链的交易，最后广播到原链上完成资金动态平衡流程。

3.1.3.2 资金动态平衡期限

每 10000 个区块执行一次资金动态平衡流程，根据多重签名热钱包的资产多寡，将多重签名热钱包的资产总额调整至设定范围内。

上述资金动态平衡交易由该块的出块 Citizen 创建基础交易广播后由不低于 2/3 的 Senator 追加签名。

3.2 抗量子

目前，在以比特币为代表的区块链系统中，SHA-256 哈希算法和 ECDSA 椭圆曲线密码给比特币网络提供最基本的安全保护。

需要强调的是，随着量子计算机的出现，作为区块链底层安全支撑技术之一的传统公钥密码的安全性将受到严峻的挑战，它将对已有区块链系统的安全性造成极大的杀伤力。特别是用于求解离散对数的 Shor 算法[Sho99]将对比特币使用的 ECDSA 签名方案产生很大的安全威胁。

当发生这种情况时，人们可以很容易地从比特币交易所呈现的公钥中计算得出密钥。因此，当交易向网络广播时，在被纳入区块链之前交易都处会于危险之中。特别是在攻击者可能会拦截交易，获得公钥并计算相应的密钥。然后，攻击者可以修改交易内容并生成修改交易的有效签名。如果由攻击者产生的新交易在原始交易之前被纳入区块链，那么攻击者可以从原始输出地址盗取比特币。

HCASH 项目旨在建立一个安全，高效，强力且可靠的去中心化系统。在 HCASH 设计的所有功能中，抗量子是最有吸引力的特性之一。抗量子密码学也称为后量子密码学，能够抵抗量子计算机的攻击。HCASH 整合了两个最流行的抗量子签名方案，BLISS [DDLL13] 和 MSS / LMS [BDH11,1LM95]。BLISS 是目前效率最高的抗量子签名方案，拥有最小的公钥和最短的签名大小，而 MSS / LMS 则是基于哈希签名方案中最高效的，MSS / LMS 的安全假设很弱（即安全性很强），其安全性仅依赖于底层哈希算法的安全性，在该安全假设下 MSS/LMS 是可证明安全的。 [ABL + 17]。到目前为止，密码学者已经对两种方案的性能、效率、安全性等进行了较为充分的分析论证。

在实施 BLISS 和 MSS / LMS 签名时有两个需要解决的问题：

一是 Bliss 算法中的离散高斯采样（DGS）模块在实现中存在旁路攻击的风险[PBY17]。需构建安全高效的抗旁路攻击方案。

二是虽然 BLISS 和 MSS / LMS 签名相较于其他抗量子签名方案，都比较简短，但其公钥、签名长度远大于传统数字签名算法 ECDSA 的公钥、签名长度，在密码货币或区块链系统中实现这些签名算法将带来交易大小显著增加，从而引发系统吞吐量明显下降。

HCASH 团队经过近一年的技术开发，针对第一个问题已经取得了突破性的进展。 我们已经找到了针对 BLISS 旁路攻击的对策，并在实施中整合了许多旁路攻击的防御技术。针对第二个问题，为了减轻长签名带来的负面影响，我们从隔离见证中得到灵感，创新性地提出了一种新型隔离见证方案，该方案能很好地解决抗量子签名算法中签名较长带来的吞吐量明显下降的问题。相关成果将在下文中详细描述。

我们将从以下五个方面来展示我们的方案：在第 1 部分，我们将简要地回顾一下各种抗量子签名方案，并解释为 HCASH 选择 BLISS 和 MSS / LMS 的原因。第 2 部分展示 BLISS 方案，并提出了我们针对 BLISS 的旁路攻击的对策。第 3 部分描述了 MSS / LMS 方案。第 4 部分简要介绍区块传输协议。最后，第 5 部分总结 HCASH 的抗量子方案。

3.2.1 抗量子签名方案的选择

许多公钥签名方案已经在其他公开文献中提出，并且被推测在量子环境下是安全的。到目前为止，已有四类主流的抗量子密码技术：

1）Hash-based cryptography（基于 Hash 函数的后量子密码），其安全性依赖于抗碰撞的 Hash 函数；包括 MSS [Mer89，DOTV08]，LMS [LM95，MCF17]，XMSS [BDH11]，SPHINCS [BHH + 15]和 NSW [NSW05]；

2）Multivariate-quadratic-equations cryptography（基于多变量二次方程的后量子密码），其安全性依赖于有限域上的多变量二次多项式映射（陷门单项函数）；包括 RAINBOW [DS05]。

3）Code-based cryptography（基于编码理论的后量子密码），其安全性依赖于纠错码理论；包括 CFS [CFS01]和 QUARTZ [PCG01];

4）Lattice-based cryptography（基于格理论的后量子密码），其安全性基于格上的困难问题； 包括 GVP [GPV08]，LYU [Lyu12]，GLP [GLP12]，BLISS [DDLL13]，DILITHIUM [DLL + 17]和 NTRU [MBDG14];

表 2 给出了这些方案的签名大小和公钥大小[ABL + 17，BDH11，dOL15] 。

类型	名称	安全性(bits)	PK 大小(kb)	Sig. 大小(kb)
I	LMS	128	0.448	22.624
	MSS	128	0.256	30.752
	XMSS	100	13.568	15.384
	SPHINCS	128	8	328
	NSW	128	0.256	36
II	GPV	100	300	240
	LYU	100	65	103
	GLP	100	12	9
	BLISS	128	7	5
	DILITHIUM	138	11.8	21.6
III	CFS	83	9216	0.1
	QUARTZ	80	568	0.128
IV	RAINBOW	160	305	0.244

表 2：量子后签名方案的签名和公钥大小

考虑了公钥和签名大小等因素后，我们发现实用的选择是基于格以及基于哈希的签名方案。 哈希签名方案最重要的优势是具有可证明的安全性，至少在随机预言模型中成立 。目前， 针对基于哈希的方案最有效的量子攻击是 Grover 的搜索算法[Gro96]。正由于 Grover 算法的存在，基于哈希的方案的量子安全级别只是传统安全级别的一半。因此，在相同的量子安全级别中，基于格的方案就签名方面和公钥大小等方面具有优势。

MSS 由 Ralph Merkle 在 20 世纪 70 年代后期开发，此算法是基于一次性签名产生的，如 Winternitz（WOTS）。MSS 的安全性仅依赖于哈希函数的正确性，即抗碰撞性和第二前像耐受性。 LMS 是 MSS 的多种变体之一，它改善了 MSS 的安全性并提高了效率。 LMS 是由 Leighton 和 Mechali 于 1995 年提出的，此算法在 Katz [Kat16，Kat]和 Fluhrer [Flu17] 分析成果的基础上进行过多次修改。我们决定在 HCASH 的基于哈希的签名中使用最新的 LMS 算法[MCF17]，并使用 SHA3 作为基础哈希函数，这是因为 SHA3 拥有优秀的抗碰撞性和第二前像耐受性。

对比现有的抗量子签名方案，BLISS 算法拥有最短的签名长度和公钥，其安全性依赖于两点：一，NTRU 问题的困难程度，二，解决此问题就等同于找到 NTRU 格中的短矢量这一假设。 BLISS 的缺点是难以通过安全的方式来实施，因为它容易受到旁路攻击[PBY17]。

DILITHIUM 是另一种基于格的签名方案，尽管这种签名方案有着较长的公钥和较大的签名大小，但是它具有更高的性能，在安全级别上，目前 DILITHIUM 没有任何缺陷，这使它看起来比 BLISS 更安全。但是由于 DILITHIUM 仅在最近才被提出，其执行的安全性并没有像 BLISS 一样的经过详细论证。考虑到我们已经已经有抵抗 BLISS 旁路攻击的对策，并且 BLISS 的公钥和签名大小对区块链数字货币性能有显着的正面影响，因此我们认为 BLISS 是比 DILITHIUM 更好的选择。

HCASH 使用了 MSS / LMS 和 BLISS 两种方案，兼顾了 BLISS 的效率和 MSS / LMS 的安全性。为了解决 BLISS 旁路攻击问题，我们在具体的实施方案中整合了许多旁路攻击对策。

3.2.2 BLISS Algorithm BLISS 算法

BLISS 签名方案是对 LYU [Lyu12]方案的改进，即更改签名的概率分布方式，例如用双峰高斯分布替换离散高斯分布。这种改进显著降低了拒绝采样率，该拒绝采样率会制约签名分布服从固定的高斯分布，从而消除签名分布造成的任何信息泄漏。 图 5 解释了这种改进。

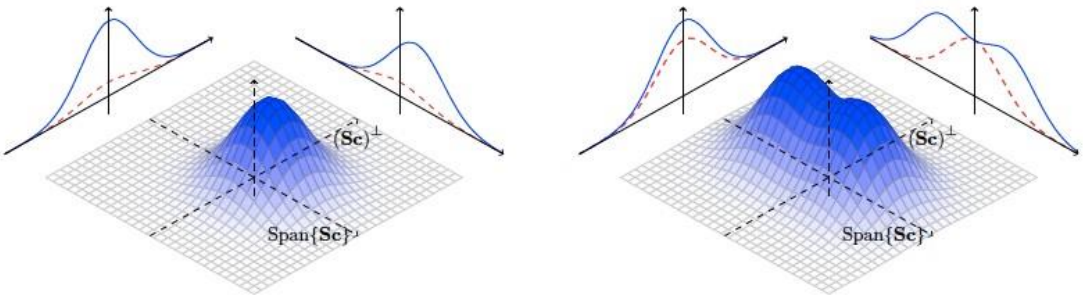
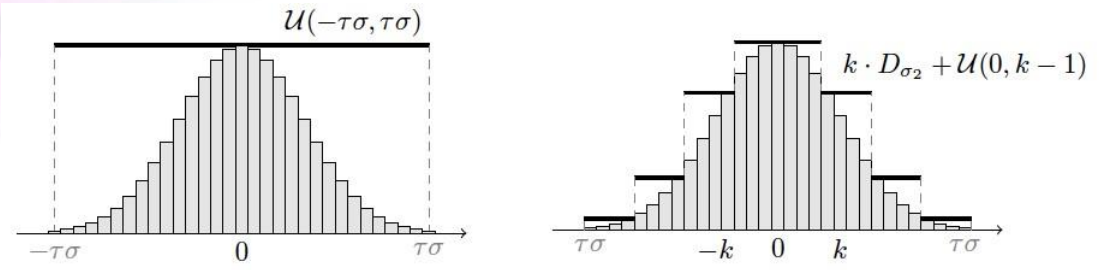


图 5：用双峰高斯分布改进拒绝采样

接下来我们简要介绍 BLISS 算法的基本思想。

BLISS 提出了一种高效的离散高斯抽样算法。在 BLISS 出现之前，所有已知的在格上与离散高斯分布统计上相近的采样算法，都需要在某点处进行长整数算术运算或大存储器存储。而 BLISS 所提出的高斯抽样算法则可以高效地对离散高斯进行采样，而且既不需要通过大型预先计算表格，也不需要对超越函数进行评估。首先，算法是根据具有 exp（x / f）和 1 / cosh（x / f）形式的偏差伯努利分布进行采样的，不需要实际计算超越函数。 其

次，BLISS 基于二进制离散高斯分布构建有效分布，并将拒绝采样应用于该分布，以获得目标离散高斯分布。 该过程将由图 6 解释。



(a) from uniform distribution (repetition rate ≈ 10)

(b) from our adapted distribution (repetition rate ≈ 1.47)

图 6：采用排斥采样的离散高斯样本

在算法 1 中给出了具有 exp（x / f）形式的偏差伯努利分布的采样算法。具有 1/ cosh（x / f）形式的偏差伯努利分布是基于以下事实进行采样的：B1/cosh(x) = Bexp(-|x|) ⊘ (B1/2 ∨Bexp(-|x|))，其中 Ba⊘Bb 被定义为 Ba/(1— (1—a)b)。 对二进制离散高斯分布进行采样的方式在算法 2 中。在算法 3 中给出了最终的离散高斯分布采样算法。

算法1 具有exp（x / f）形式的偏差伯努利分布的采样算法

输入： x ∈ [0, 2^{ℓ}) 二进制形式的整数x = $x_{\ell-1} \cdots x_0$

1: 计算 c_i = exp(- 2^i/f) ， 0 ⩽i ⩽ℓ — 1

2: 对于 i = ℓ - 1 到 0

3: 如果x_i = 1 那么

4: 根据ci 概率的伯努利分布对A_i 进行采样

5: 如果A_i = 0 那么回到0

6: 结束步骤3

7: 结束步骤2

算法2 对二进制离散高斯分布进行采样

输出：根据二进制离散高斯分布的整数$x \in Z^+$

1: 生成一个bit，$b \leftarrow B_{1/2}$

2: 如果 $b = 0$ 那么回到0

3: 对于 $i = 1$ 到 ∞

4: 绘制随机bits $b_1 \cdots ; b_k , k = 2i - 1$

5: 如果 $b_1 \cdots b_k \neq 0 \cdots 0$，那么重新开始

6: 如果 $b_k = 0$ 那么回到i

7: 结束步骤 3

算法3 离散高斯分布采样

输入：一个整数$k \in Z$

输出：根据离散高斯分布的偏差$k\sigma_2$的整数$z \in Z^+$

1：通过算法2采样x

2：在{0; … ; k - 1}中统一采样y

3：计算$z \leftarrow kx+y$

4：采样$b \leftarrow B\exp(-y(y+2kx)/(2\sigma^2))$

5：如果不是b，则重新开始

6：如果$z = 0$，那么以概率1/2重新启动

7：生成一个bit $b \leftarrow B_{1/2,}$ 并返回$(-1)^b z$

BLISS 也用 NTRU 环 $R_q = Z_q[X]/(x^n + 1)$取代了从中取得矩阵元素的环 $R = Z$。 公钥矩阵 A 和私钥矩阵 S 的大小分别减少到 1×2 和 2×1。 BLISS 的私钥由两个来自 Rq 的多项式 s_1和 s_2组成，并且公钥由一个多项式 a_1组成（矩阵 A 的另一个元素是常数 q-2）。 签名由一对多项式 z_1和 z_2以及一个挑战(challenge) c 组成，它是一个系数来自{0, 1}的稀疏多项式。

BLISS 通过压缩多项式 z_2来降低签名大小，并且不影响安全性。 特别是，BLISS 用等值于$\lfloor u \rceil_d - \lfloor u - z_2 \rceil_d$ 的$z_2^{\dagger}$取代了的 z_2，其中$\lfloor \cdot \rceil_d$ 表示从多项式的每个系数中删除 d bits。 与 z_2相比，$z_2^{\dagger}$系数更小，因此更容易被霍夫曼编码压缩。

BLISS 方案的效率得到了进一步优化，其改进版本被称为 BLISS-B [Duc14]。 BLISS-B 用一个程序 GreedySC（S,c）代替多项式 S·c 的乘积，该程序有效地计算 S·c'，其中 c'与非零系数子集的符号 c 不同。 GreedySC（S，c）有效地导出了一个 c'，使得$||S \cdot c'|| 2$ 被最小化，从而为拒绝采样率降低了一个相当大的因数。 算法 4 给出了计算 GreedySC（S，c）的算法。

算法4 GreedySC

输入：矩阵$S = [s_1; \ldots; s_n]$和二进制向量c

输出：$c' = c \bmod 2$的$v = Sc'$

1：$v \leftarrow 0 \in Z^m$

2：对于$i \in I_c$

3：$\zeta_i \leftarrow sgn(v, s_i)$

4：$v \leftarrow v - \zeta_i \cdot s_i$

5：结束步骤2

BLISS 方案的最终算法在算法 5，算法 6 和算法 7 中给出。

3.2.2.1 BLISS 抵抗旁路攻击

BLISS 已被证明容易遭受旁路攻击[EFGT17，PBY17]。 这种脆弱性的主要来自离散高斯采样，它在格（lattice）密码学中扮演重要角色[MW17]。以下给出了改进后的、能有效抵抗旁路攻击的 Bliss 算法结构。

算法5 BLISS密钥生成

输出：密钥对（A, S），使得AS = q mod 2q

1：选择f, g作为均匀的多项式，使d_1恰好落入{±1}且d_2在{±2}中。

2：S = $(s_1, s_2)^T$ ← $(f, 2g + 1)^T$

3：如果$N_k(S) \geqq C^2 \cdot 5 \cdot (\delta_1 n + 4 \delta_2 n) \cdot k$，那么

4：重新开始

5：结束步骤3

6：a_q = (2g+1)/f mod q（如果f不可逆，则重新开始）

7：输出（A, S）其中 A = $(2a_q, q - 2)$ mod 2q

算法6 BLISS签名算法

输入：信息μ，公钥，A = $(a_1, q - 2) \in R_{2q}^{1\times2}$，密钥 S = $(s_1, s_2)^T \in R_{2q}^{1\times2}$

输出：信息μ的签名$(z_1, z_2^{\dagger}, c)$

1: y_1, y_2 ← DZn, σ

2: μ = ζ • a_1 • $y_1 + y_2$ mod 2q

3: c ← H(u d mod ρ , μ)

4: (v_1, v_2) GreedySC(s, c)

5：选择一个随机bit, b

6：$(z_1, z_2) \leftarrow (y_1, y_2) + (-1)^b \cdot (v_1, v_2)$

7：继续以$1/(M \exp(-\frac{||v||^2}{2\sigma^2}) \cosh(\frac{\langle z,v\rangle}{\sigma^2}))$为概率，否则重新开始

8：$z_2^{\dagger} \leftarrow (u_d - u - z_{2\,d}) \bmod p$

9：输出(z_1, z_2, c)

算法7 BLISS 验证算法

输入：信息μ，公钥$A = (a_1, q - 2) \in R_{2q}^{1\times 2}$，签名为$(z_1, z_2^{\dagger}, c)$

输出：接受或拒绝签名

1：如果是$||(z_1 \mid 2^d \cdot z_2^{\dagger})||_2 > B_2$，则拒绝

2：如果是$||(z_1 \mid 2^d \cdot z_2^{\dagger})||_\infty > B_\infty$，则拒绝

3：如果是$c = H(\zeta \cdot a_1 \cdot z_1 + \zeta \cdot q \cdot c_d + z_2^{\dagger} \bmod p, \mu)$，则接受

为了设计具有高效率和安全性的离散高斯采样算法，人们已经付出了很多努力。信息泄漏的另一个来源是拒绝采样，这会导致程序执行和内存访问取决于随机数据。

首先，我们通过常量方式以概率为e^x的形式实现伯努利采样器。伯努利采样器用于BLISS [DDLL13]所使用的离散高斯采样算法中，并且执行过程取决于x的bit。具体来说，采样器调用表格在x中进行查找，查找内容为每个值为1的bit. 我们通过强制程序来执行查找，用以消除这种潜在的泄漏源，无论bit值是多少。

其次，我们通过将y分解为两个独立采样的y_1和y_2的和来防止攻击者得出被采样的y。然后我们首先计算Ay_1和Ay_2，并以此相加计算Ay。我们仔细选择y_1和y_2的标准差和其他参数，使得统计距离Δ（y，$y_1 + y_2$）根据[Pei10]中的定理3.1可以忽略不计。由于保护技术(protection

techniques)带来的成本，签名生成过程比未受保护的 BLISS 慢三倍，但速度仍然很快。

因此，我们提出的上述方案能很好地兼顾安全性与可用性。

3.2.3 The MSS and LMS Algorithms MSS 和 LMS 算法

基于 hash 的签名方案是从一次性签名方案（OTS）开始的，一次性签名方案意味着一个签名方案中每个密钥对不能用于签署多个信息。 目前最有效的 OTS 方案是 Winternitz 方案（WOTS）[BBD09]。 WOTS 的核心思想是迭代地将函数应用于秘密输入中，而迭代次数取决于要签名的信息。 这些函数是以下函数族的成员

$$F(n) = \{f_k := \{0,1\}^n \rightarrow \{0,1\}^n\} \mid k \in \{0,1\}^n$$

Let $f_k^0(x) := x, f_k^1(x) := f_k(x)$ and $f_k^i(x) := f_{fk}\text{-}1(x)(x)$.

WOTS被参数化，哈希大小为n，压缩级别$w \in Z^+$，$w>1$。

WOTS密钥生成。 若要为WOTS生成一对密钥，请选择一个随机值$x \in \{0,1\}^n$。 ℓ的计算如下

$$\ell_1 = \frac{n}{\log(w)}, \quad \ell_2 = \frac{\log(\ell_1(w-1))}{\log(w)} + 1, \quad \ell = \ell_1 + \ell_2$$

密钥 sk 由 l 个均匀随机选择长度为 n 的位串组成。

公钥 pk 从 sk 中的计算如下：$pk_0 = x$， $pk_i = f_{sk_i}^{w-1}(x)$， $1 \leqslant i \leqslant \ell$.

WOTS签名生成。 为了对以base-w展示给出的n位消息$M = (M_1, \cdots, M_{\ell_1})$进行签名，如$0 \leqslant M_i \leqslant w-1$，首先计算校验和

$$c = \sum_{i=0}^{\ell_1} (w - 1 - M_i)$$

并且在base-w中表示C为（C_1; ...; $C\ell_2$）。 然后我们设定B =（b_1; ...;$b\ell$）= M||C。 签名计算如下

$$\sigma = (\sigma_1, \cdots, \sigma\ell) = (f_{sk_1}^{b_1}(x), \cdots, f_{sk_\ell}^{b_\ell}(x))$$

WOTS签名验证。 若要验证签名，首先如上所述计算base-w字符串B，然后检查

$$f_{\sigma_1}^{w-1-b_1}(x), \cdots, f_{\sigma_\ell}^{w-1-b_\ell}(x) = (\mathrm{pk}_1, \cdots, \mathrm{pk}\ell)$$

Merkle 签名方案（MSS）适用于任何加密哈希函数和任何一次性签名方案。MSS 使用 Merkle 树将 N = 2^H OTS 公钥连接在一起，其中 H 是树的高度。Merkle 树根被用作 MSS 的公钥。 用来计算 Merkle 根的算法被称为 Treehash，参见算法 8。

为了生成信息M的签名，需要选取下一个未使用的OTS公钥的密钥，并生成OTS的签名。 MSS的签名包括OTS签名，OTS公钥和用来证明Merkle树中存在OTS公钥的验证路径。要验证签名，首先要检查OTS签名对OTS公钥的有效性，然后验证OTS公钥是具有验证路径的Merkle树的有效叶。

算法8 Treehash

输入：高度H> 2

输出：Merkle树的根

1：对于j = 0;…; 2^H - 1做

2：计算第j片叶子：NODE1 ← LEAFCALC（j）

3：虽然$NODE_1$与STACK顶层节点具有相同的高度

4：从栈中弹出顶层节点：$NODE_2$ ← STACK.pop()

5：计算它们的父节点：$NODE_1$ ← g($NODE_2$ || $NODE_1$)

6：结束时

7：推动栈上的父节点：STACK.push（$NODE_1$）

8：结束步骤1

9：设R是存储在栈上的单个节点：R ← STACK.pop()

10：返回 R

MSS私钥由2^H OTS密钥组成。对于大多数实际应用来说，存储如此大量的数据是不可行的。通过使用确定性伪随机数发生器（PRNG）并仅存储该PRNG的种子可以节省空间。然后，每个一次性签名密钥必须被生成两次，一次用于MSS公钥生成，一次在签名阶段生成。除了私钥大小以外，使用PRNG进行一次性签名密钥生成还有另一个好处：只要PRNG具有前向安全性，它就会使MSS变得安全。

计算Merkle根和PRNG种子的认证路径是一项十分需要计算量的工作。同时，将整个Merkle树保存在内存中虽然效率很高，但会造成较大的内存使用量。为了降低存储和计算成本，人们提出了各种Merkle树遍历算法(Merkle tree traversing algorithms)。 Merkle树遍历的主要目的是依次输出所有节点的叶节点值以及相关的认证路径。目前最有效的Merkle树遍历算法是由Szydlo在[Szy04]中提出的算法。详情请参阅算法9。

由Leighton Micali提出的LMS是对MSS在安全性和效率方面的一种改进。这里我们简要介绍一下对密钥、公钥和签名的改进。

对于一次性密钥，LMS将包括typecode，I和q在内的安全字段作为MSS中普通密钥x = (x_0,…, $x_{\ell-1}$)的前缀，最终密钥编码为

$$Typecode||I||q||x_0||\cdots||x_{\ell-1}$$

算法9 对数Merkle树遍历

1：设置s = 0

2：对于每个h ∈ [0; H − 1]，输出$AUTH_h$

3：对于所有满足2^h除以s + 1的h

4：将$AUTH_h$设置为$STACK_h$中的唯一节点值

5：设置startnode =（s + 1 + 2^h）⊕2^h

6：$STACK_h$：初始化（startnode, h）

7：结束步骤3

8：对于i从1到2H − 1

9：设ℓ_{min}为$STACK_h$.low

10：让focus成为最低h，使$STACK_h$.low = ℓ_{min}

11：$STACK_h$.update

12：结束步骤8

13：设定s = s + 1

14：如果s <2^H，则转到第2行

因此，LMS中的密钥比MSS多消耗24个字节。当涉及公钥（即Merkle根）时，采用类似的预填充，一个LMS的公钥被编码为

$$sigtype||otstype||I||v$$

其中sigtype和otstype都是typecode，而v是MSS的原始公钥。

LMS签名σ被格式化为

$$q||\ \sigma_{ost}||sigtype||A_s$$

其中 σ_{ost} = otstype$||C||\ \sigma_0||\cdots||\ \sigma_\ell$ 是一次性签名，且 $A_s = \{a_i\}_{i=0}^{l-1}$

在MSS中，一次性公钥是y =（y_0; y_1;…; y_ℓ），并且如果M上的σ_{ost}由y的配对密钥签名，则一次性公钥可以来源于σ_{ost}和信息M中。在LMS中，用来验证一次性签名σ_{ost}的一次性公钥y =（y_0; y_1;…; y_ℓ）被压缩为

$$y = typecode||\ I//q//H(I//q//D\_PBLC//y_0//y_1//\ \cdots\ //y_\ell)$$

而MSS的对应项则公开存储每个y_i。 每次需要验证时，一次公钥候选z =（z_0; z_1; … ; z_ℓ）可以从σ_{ost} =（σ_0; σ_1 ;…; σ_ℓ)中导出，然后检查z的扩展摘要为$H(I//q//D\ PBLC//z_0//z_1//\ \cdots\ //z_\ell)$与公钥中的对应分量$H(I//q//D\ PBLC//y_0//y_1//\ \cdots\ //y_\ell)$进行比较。

LMS 和 MSS 都基于 Merkle 树。 其他中间例程，例如通过遍历树更新认证路径 A_s =（a_0; a_1; ...;a_ℓ），是相同的。 主要区别在于组件 σ_{ost}中的一次性公钥 y。

3.2.4 Block Transmission Protocol 区块传输协议

尽管 BLISS 和 MSS / LMS 签名在所有后量子签名方案中相当小，但它们仍然比传统的 ECDSA 签名更长。长签名将使交易规模显著增加，从而引发系统吞吐量明显下降。为了减轻长签名带来的负面影响，我们设计了一种基于隔离见证思想的新的数据区块传输协议。

简而言之，隔离见证的概念是将签名与交易主体分开。具体而言，交易签名不被计入交易 ID 的计算。我们通过允许用户在传递区块时仅传输交易

ID 来进一步改进这一方案。这样可以使传输区块的通信成本显着降低。当接收到新区块时，节点会检查节点中所有的交易记录，节点将检查他/她是否已经获得（并保存到内存池中）该区块包含的所有的交易 ID，如果发现有遗漏，他会向周围的节点请求以填补缺失的交易记录信息。图 7（见下文）是新生成的区块的传输过程的例子。

3.2.5 结论

HCASH 支持多个抗量子签名方案，并且与传统的 ECDSA 签名兼容。同时，我们创新地提出了基于隔离见证思想的新型通信协议，该协议减轻了长签名对网络的压力，实现了抗量子签名，将在未来的量子环境下为 HCASH 用户提供了高等级的安全保护方案。

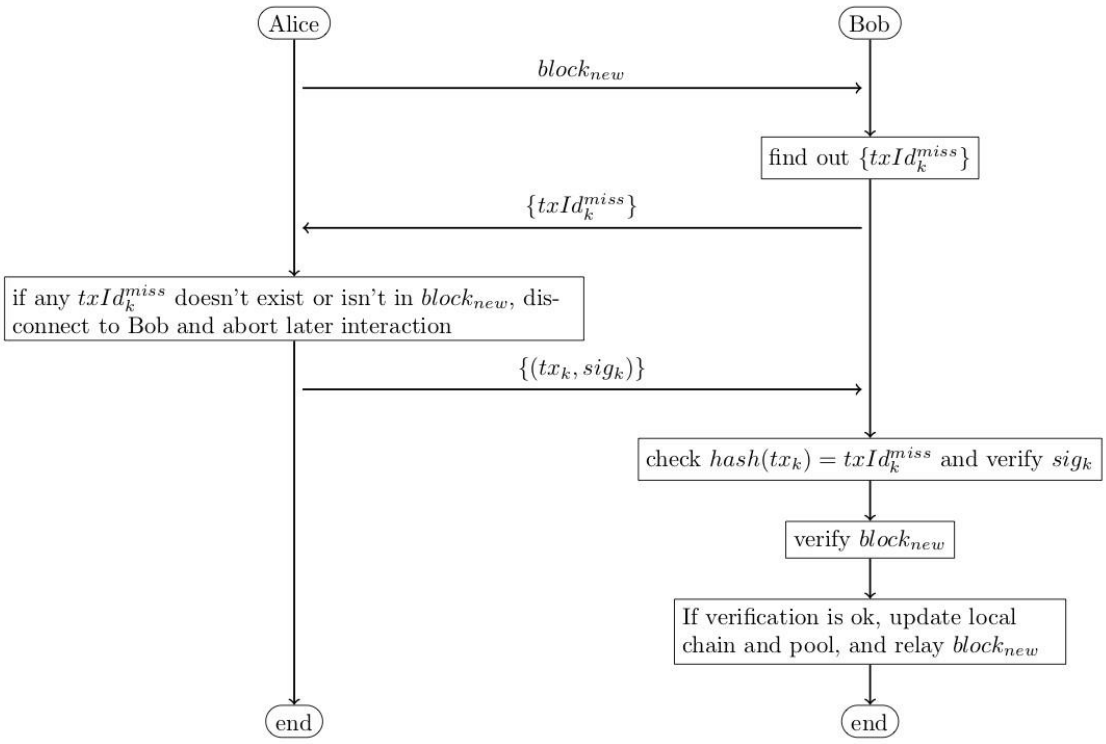


图 7：传播一个新的区块

基于隔离见证的区块传播协议将大大减少大量抗量子签名带来的通信成本，并能够被推广并移植到其他区块链和加密货币系统中。

抗量子签名已经在升级后的 HC 主链上实现，并将通过接口在 HyperExchange 上实现。届时 HCASH 的两条主链都将具备抗量子特性，成为区块链领域屈指可数的拥有极高安全系数的跨链生态。

3. 3 智能合约

HCASH 生态通过 HyperExchange 链上图灵完备的智能合约帮助用户实现复杂的跨链交易、资产通证化，为开发者提供 DAPP 开发的便捷支持，为实现跨链分布式商业应用打下基础。

为了能吸引更多的生态建设者到 HyperExchange 上开发 DAPP，共同建设 HCASH 生态，HyperExchange 系统致力于运用区块链技术为搭建去中心化应用(Decentralized Applications)提供基础架构，通过针对各类开发语言的兼容引入，使得各种技术栈的开发者都可以迅速对接，便捷高效地开发 DAPP。

3.3.1 全面兼容的开发语言

- HCASH 使用一种图灵完备并为区块链智能合约定制设计的字节码规范作为智能合约虚拟机的实现规范。提供静态类型的高级编程语言比如 C#，Java，TypeScript 等的编译器实现从高级语言生成智能合约字节码。
- HCASH 也将不断丰富各种语言的 SDK 方便开发者移植到不同的平台，从而不断吸引各行各业的 DAPP 开发者。
- 通过完善的 API 的设计和丰富的原生智能合约，简化开发者的准备工作，使开发者可以快速上手相应的开发工作。
- HCASH 提供一些常用数值操作，字符串操作等的基本库，以及一些链上查询，交易等的内置函数库，在智能合约中可以调用内置库。
- 智能合约部署到链上后，除了可以被用户直接调用或者存取资产，还可以调用其他智能合约/内置原生合约，或被其他智能合约调用。
- 面向不同行业的 DAPP 应用，拥有不同的技术需求和侧重点。例如去中心化的社交、去中心化的存储和去中心化的游戏、去中心化的金融服务等，所需要的技术侧重点会有所不同。我们通过图灵完备

的智能合约，对行业DAPP深入探索后，慢慢会形成适用于该行业的DAPP 标准，HCASH 会不断将这些标准收录变为原生智能合约，方便开发者更加快速地迭代 DAPP。随着 DAPP 的产品化进程不断推进，我们希望普通互联网用户也可以真正进入区块链应用并感受到区块链技术带来的价值。

- 后续 HCASH 将推出专属的带调试功能的 IDE，使得开发者既可以享受 IDE 带来的便捷，也可以在调试错误时更快地定位问题，从而大大提高 DAPP 的开发效率。

3.3.2 轻节点、轻存储

目前主流主链在接入智能合约时，一个明显的弊端就是无法很好地做到资源隔离，随着生态的不断扩大，将不可避免地造成拥堵，例如以太坊就出现某个热门众筹导致整个系统拥堵的现象。

HyperExchange 链上每次调用执行智能合约时，都会先初始化一个独立的轻量级执行环境：仅需要在链上查找到合约字节码，然后执行合约字节码；需要访问链上数据时，通过 native API 来调用，即不需要调用所有数据。在后续的开发中，HCASH 将继续完善相关技术，最终实现各 DAPP 的运行环境独立，互不影响。

每个智能合约有各自的独立状态存储区，称作 storage。合约交易的执行导致某个智能合约的状态存储区发生数据变化时（storage 改变），不会保留所有历史的 storage 的全量备份，而是只保存 storage 的当前状态和 storage 每次变化的变化量。

通过这样的设置，用户想要得到智能合约的执行结果时，可以很轻易地获取 storage 的当前值，而无需读取所有的数据，这样大大降低用户的工作量，也减少了节点的数据存储要求，节省了系统资源，提升了系统的处理效

率，还方便按 storage 的实际变化按需计算 gas。同时，用户也可以通过历史变化量还原或者回滚得到 storage 的每次历史的值。

3.3.3 并发模型提升效率

HCASH 通过把智能合约交易派发到并行的执行管道同时执行，能够提高大量合约交易并发执行的能力，缩短整体交易执行时间。

- 把一个出块时间内需要执行的智能合约，划分为到不同管道中执行。管道这里特指合约处理器，意味着不同的管道可以在不同的线程中同时进行合约执行。
- 管道分为串行管道和并行管道，不同并行管道可以并行执行，串行管道不能和其他管道并行执行，最多只有一个串行管道。
- 一个管道启动后，其中的智能合约交易串行执行。
- 先执行串行管道中的智能合约，再并行执行多个并行管道中的智能合约。
- 读取或者修改的账户或合约地址列表称作本交易的依赖地址列表。智能合约交易可以在客户端预先执行获取依赖地址列表。普通交易也有依赖地址列表。依赖地址列表有冲突的交易不能放入不同的管道。
- 交易中不预先附加依赖地址列表的交易，进入串行管道，但是提高手续费。
- 合约声明的依赖地址列表和实际执行的依赖地址列表有冲突时，执行失败，惩罚性收手续费并打包空白交易。
- 通过管道的并行执行，可以大幅提高（最少提高 4 倍）大部分场景下的合约交易 TPS。

3.4 PoW+PoS 混合共识

3.4.1 常见共识机制的缺陷

一个强大的加密货币协议应该努力提供一个更合理的激励结构，在这个结构下维护系统中不同参与者的利益，以维持系统健康运行。

在区块链世界里，社区达成共识一直是一个难以解决的问题。

例如，在过去的两三年中，基于 PoW 的比特币扩容共识一直严重影响着比特币社区的发展；Zcash 等采用中心化的管理模式又相当程度地阻碍了社区成员的参与热度。目前，大多数区块链社区所使用的单一决策共识，难以应对新技术快速迭代的挑战，也不易在兼容利益相关者之间的意识形态差异的同时，又不损害区块链的关键价值主张，即分散化的完整性。

单一的 PoW 机制对能源的消耗很大，同时存在被攻击的风险，随着大型矿池的出现，算力越来越集中，理论上存在发动 51%攻击的可能。攻击者通过获得大量高算力硬件设备（ASIC），获得超过全网 51%的 PoW 算力，逆转最近的分布式帐本的记录来进行“双花”(double-spend)，或者通过拒绝在其产生的区块中纳入交易记录并以此执行 PoW 拒绝服务（PoW-DoS）攻击。如果攻击者希望破坏或损害比特币网络，则很可能实现其目标，因为双花或 PoW-DoS 可能会导致用户对比特币协议失去信心。攻击者可以通过双花直接获得经济收益，或者通过执行 PoW-DoS 来勒索他人并要求更高的交易费用。

此外，矿工以盈利为第一目标，对自身利益的考量往往会超过社区整体的利益。还以比特币为例，网络日渐拥堵，手续费设置较少的交易长时间得不到打包而停留在 mempool 里，导致手续费越来越高，矿工费已经高达千分之 1.5 左右，拥堵时刻甚至达到千分之 5，对于小型交易来说这个费率是过高的，严重影响了比特币的商业应用。

基于 PoS 的机制虽然具有相应的优势，但也不能有效地解决加密货币所面临的主要风险。PoS 的主要风险之一是过于中心化，这是因为，考虑到规模经济，大型节点的计算和验证能力将会胜过业余 PoS 矿工。通过 PoS 系统，PoW 数据中心所受到的特殊风险确实有所减少，但引入了新的中心化风险（大型权益持有人可能试图对系统进行控制）。另一个主要风险是由于黑名单“受污染”币而导致的可替代性侵蚀，这种风险与 PoS 正交(orthogonal)，并且可以通过混合[3,4]或者 SNARKs [5,6]来减轻。

因此，在区块链领域，大家一直都在探索尝试更为优化的共识机制。

3.4.2 PoW+PoS 混合共识

HCASH 通过汲取和融入 Decred 的理念和经验，决定在 HC 主链上采用 PoW + PoS 混合共识机制。我们认为，这种混合共识可以整合 PoW 和 PoS 两种共识的优势，避免单一 PoW 或单一 PoS 机制的不足，形成安全效率兼顾、参与者利益平衡、环保持续的共识机制。

PoW + PoS 混合共识机制可以有效提高社区成员的参与度。这种创新型投票模型加强了利益相关者的自主权，允许从一组规则无缝过渡到另一组规则，所有权益持有人都能够参与决策，共同决定是否实施某些技术或协议，以及开发团队是否应该应用某些功能。更重要的是，它比传统的投票方法更有效率，使得社区决策能够更流畅地实施，一旦投票通过，所有决策将被记录在区块链中并立即执行，从而避免矿工、矿池、交易所和钱包服务提供商之间的共识分歧。为了确保公平，投票权重基于持有的代币数量和挖矿工作的权益，任何代币持有人都可以参与投票。

相比纯 PoW 机制，PoW+PoS 混合共识机制以通过激励（incentivize）PoS 节点，维持比纯 PoW 机制下更多的持续在线节点（continual online presence notes），以此更有效地维持网络拓扑构结构的稳定性，从而一定程度上帮助抵御网络攻击。PoW + PoS 混合共识赋予了权益持有人生成块的

权力，如果攻击者希望在向网络广播之前实施双花，攻击者需要事先控制大量的权益。此外，通过相关机制选出的权益持有者来确定哪些交易应该包含在该区块中的规则能够提供一定安全性，以抵御试图通过拒绝交易来敲诈或破坏网络的攻击者。

PoW + PoS 混合共识机制下，所有 PoW 产出的块都必须经过 PoS 的验证才能成为合法的块，即 PoW 负责出块 ，PoS 负责投票决定区块的有效性。矿工和持币者共同参与产块，二者互相制衡，这样就几乎杜绝了算力垄断的现象，确保了网络的安全。并且，PoW+PoS 混合共识机制，可以通过投票机制来有效控制硬分叉。

另外，PoW + PoS 的混合共识机制可以保护新用户的利益，防止早期投资者从 PoS 系统获得过多的奖励。

3.4.3 PoW+PoS 混合共识机制的实现

PoW+PoS 混合共识机制的基本实现逻辑是：PoW 矿工创建区块；PoS 矿工确认这些区块的合法性。

将 PoW+PoS 混合共识机制的挖矿跟 PoW 挖矿做一个比较：

在 PoW+PoS 混合共识机制下，同一区块高度，一定时间内，不同的 PoW 矿工都可以生成区块，由 PoS 投票选出合法区块，因此，优先算出的区块不一定被采用。并且，如果 PoW 矿工违背社区内其他用户的利益，那么他们的区块奖励也会被剥夺。

而单一 PoW 机制的挖矿，在区块高度为 H 时，矿工 A 只要率先计算出了正确哈希值，在链上广播，其他矿工验证他的哈希值，如果共识哈希值正确，矿工 A 就可以获得这个区块的奖励和记账权，其他矿工的劳动是无意义的。

我们可以看到，PoW+PoS 混合共识机制可以解决矿工权力过大的问题，让区块链生态得到平衡发展，同时保障了网络的安全性。在实际操作中，PoW+PoS 混合共识并非只是简单的混合，其背后有着复杂的技术实现路径和投票逻辑。

3.4.3.1 PoW+PoS 的实现过程

HCASH 的开发团队在 HC 主链上实现了 Decred（和 Bitcoin-NG）的挖矿算法 Blake256 R14 投票机制（实用且灵活的 PoS 方案），并优化了相关执行细则。

具体的实现过程如下：

1.挖矿过程将从 PoW 开始，矿工们竞相解决数学问题。根据哈希计算出块，被开采的块不包含任何交易。

2.此时，系统将切换到 PoS，基于包含在该文件头中的信息，选择一组随机验证器来签署新区块。即通过 PoS 投票来决定 PoW 产块的合法性。我们将参与 PoS 投票的用户称之为验证者。在这里，验证者被选择的机会取决于验证者持有的代币数量。拥有的代币越多，该验证者被选中的概率就越高。验证者会选择将某个交易放入该块内。（与权益系统相关的交易被插入到 UTXO 套件中。）

具体的投票过程涉及到以下几个环节：

投票池

为了使投票权相对公平，我们借鉴 DCR 的机制使用了投票池的概念。所有的票都被置于投票池中，投票池中的总票数是固定的，系统随机选择，选中者成功参与投票后，返还购买选票的费用。获得的出块奖励由参与的 PoW 矿工和 PoS 矿工共同获得。

通过每隔 288 个区块进行票价调整，投票池里面总票数控制为 40960。

购买票

投票池中的票是需要 HC 代币的持有者购买的，可以通过钱包直接购买票。总购票成本为选票价（Ticket price）加上 选票费（Ticket fee）,选票费是支付给 PoW 矿工的，是将票放入新挖区块的手续费。

内存池

用于购买票的 HC 会被系统锁定，并且在投票完成前不可以撤回。刚买的票，需要被矿工打包记录在区块里才能生效，存放未被打包票的地方叫内存池（mempool）。内存池中，选票费高的更容易被矿工选中，可以更快地进入选票池（Ticket pool），因为每个新区块最多能记录的票数是有上限值的，所以内存池中的票存在竞争关系。

准选票

矿工打包完成，票在区块链中一经记录，这张票就成了准选票（immature ticket），也称之为未成熟选票。准选票可以理解为：它还不在选票池中，不具有被选中资格，同时选票费也无法退回，只能等待一定的时间进入选票池。

如果内存池（mempool）中的票过多，经过一定时间没有被矿工打包记录，那么选票价（Ticket price）和选票费（Ticket fee）会被退回。这个和比特币交易打包很像，矿工费过低的数据包最终可能会被退回。

选票

成功进入选票池后，等待被系统选中，成为真正的选票，参与投票，拿到区块奖励。

系统随机决定，被选中的可能性服从泊松分布函数。简单来说，28 天内被选中的概率是 50%，142 天内被选中概率是 99.5%，如果 142 天仍然没能被选中，选票价将会退还，选票费已用于支付矿工验证的手续费，不予退还。

3.一旦这些选定的验证者完成区块签名，区块将成为一个完整的区块。如果某些验证程序不能用于签名块，它们将被选中用于签署下一个区块，并

且将选择一组新的验证程序，直到当前块得到正确数量的签名为止。交易费用将分配给参与该区块签名的矿工和验证者。

4.验证之后，将会有一个标志来指示前一个常规交易树（包含 coinbase 和 non-stake 相关的交易）是否有效。然后系统切换到第二个 PoW，将 PoS 的结果插入到新的 PoW 区块中，并设置标志（基于大部分验证）以确定前一个区块（特别是常规交易树）的有效性。如果前一个区块的常规交易树已经过验证，则将这些交易插入到 UTXO 集合中，然后重新启动整个过程。

一般来说，PoW 和 PoS 的组合更像是一个彩票系统：验证者购买（锁定）一些 HC 作为票。票的持有者将有机会参与投票，决定是否在现有区块链上添加一个新区块，或者决定前一个常规交易树是否有效（包含 coinbase 和 non-stake 相关的交易）。作为回报，奖励将发给 PoW 矿工和验证者（购买选票的 HC 也将返还）。如果验证失败，也就是说区块无法获得足够的投票，该区块将被丢弃，并且将区块内的交易信息返回到另一个区块中。

3.4.3.2 去中心化 PoS 权益池

中心化矿池潜在的问题是权益持有人将把他们的资产放在在中心化 PoS 权益池中，没有经验的用户可能会认为在线钱包能够比他们更好地保护资产。事实上，权益持有人将他们的资产委托给中心化 PoS 权益池的风险远大于 PoW 矿工将他们的算力委托给矿池时所承担的风险。权益持有人丢失币是中心化 PoS 权益池难以避免的风险。

中心化 PoS 权益池的另一个风险是，无法保证实际控制人按照实现约定的规则进行权益分配。

HCASH 拥有去中心化 PoS 权益池，通过内置在底层协议中的合约，系统会在扣除权益池管理费后自动对用户进行收益分配，权益池无法获得用户的

HC 代币，也无法获得用户的收益，即在不涉及信任的情况下在参与者之间分配奖励。

由于管理费用高昂，这种去中心化的权益池可能不适合小型权益持有人。 这些权益持有人可能会将其信任扩展到中心化权益池上，而中心化权益池中的控制人则会参与到更大的去中心化权益池中。

3.4.3.3 PoW+PoS 混合共识机制下的分配比例

HC 的总量将保持不变，在创世区块产生后，每个区块奖励总量约为 6.4 个 HC，之后每隔 12288 个区块奖励减产为 999/1000，延续 99 年，出块时间平均为 2.5 分钟。

综合考虑到 HCASH 的现状和未来公有链运行的安全性，在 HCASH 主链 2.0（HypeCash）启动后，PoW / PoS 挖矿比例将调整为 2：1，HC 主链出块奖励的分配比例为：PoW 占比为 60%、PoS 占比为 30%、开发团队及社区建设占比为 10%（占总量的 5%）。

完成测试，主网正式上线前，HCASH 团队会有其他文档详细阐述代码和经济模型。

3.5 隐私保护

高等级的隐私保护是 HCASH 最初的设计愿景之一。

在开发过程中，为了符合政策和监管的要求，我们提出了最终通过抗量子环形保密交易和零知识证明来实现隐私保护的方案。

通过技术手段，我们不单单在资产转移的过程中实现双向加密，还可以应用到很多其他对交易隐私要求极高的领域，同时还将实现区块链与非区块链分布式账本（如 DAG）之间的加密通信，如：能够跨越平台实现诸如从 HCASH 客户端到 Byteball 客户端的加密通信。

3.5.1 环形保密交易

环形保密交易 Ring Confidential Transaction（RingCT）是先进的隐私保护算法。 目前，HCASH 已经提出了更先进的抗量子环形保密交易 (Quantum Resistant RingCT)，它同时具备量子抗性，能够在资产转移过程中实现双向加密，并且实现对交易隐私的高要求。

RingCT 可以在两个方面提供用户隐私保护：

1）匿名发送者的身份

2）隐藏交易名称

RingCT 协议由三个技术部分组成，即一次性可链接环签名，同态承诺方案和范围证明。

一次性可链接的环签名允许支出者使用他的一次性密钥（对应于他的一次性公钥）签署交易，并通过从区块链中挑选其他 n-1 个公钥来匿名交易。区块链中的任何人都只能知道交易是由 n 个用户（其公钥包含在此交易中）中的一个生成的，没有人能够找出这个用户是谁的确切位置（发现真正的发

件人的概率是相等于随机猜测）。HCASH 使用此技术来保护发送者的隐私。由于发件人是隐藏的，因此还需要另一种方法来检测发送者是否双花。在这里，环签名的可链接性可以提供帮助，它不仅具有匿名性，还将允许任何人检测发送者是否双花（通过使用相同的密钥来生成两个或更多的交易）。该性质为拒绝双花交易提供了必要的安全保障。

第二部分是同态承诺方案。尽管可链接的环签名可以匿名发件人的身份，但交易金额仍然向公众开放。 为了隐藏交易金额，HCASH 部署了同态承诺方案。 类似于加密方案，承诺方案允许消息隐藏。 另一方面，并没有解密算法。 换句话说，只有做出承诺的人才能通过揭示承诺过程中使用的信息和随机性来揭示承诺的价值。任何人都可以验证承诺的正确性。如果 m1±m2 承诺可以在不知道 m1 和 m2 的值(只知道 m1 和 m2 的承诺)的情况下获得，承诺方案即为同态。

如果输入金额的总和等于输出金额的总和，交易则一致。若要隐藏交易金额，同时仍允许矿工验证交易一致性，可以使用同态承诺方案。具体而言，所有交易金额都存储在承诺中。我们假设承诺方案中使用的随机子使用安全通道从发送方传输到接收方[1]。 矿工可以使用输入和输出承诺的同态属性来验证交易一致性。

剩下的任务是结合可链接的环签名和同态承诺来隐藏发送者地址和交易金额。HCASH 团队计划以创新的方式实现这一目标。未来，如果对 0 的承诺是可链接环签名方案的有效公钥，且可链接环签名方案的私钥可以从承诺所使用的随机子中导出，则实现这一任务是可能的；同时，想要从任何非零值的承诺中导出私钥是不可能的。

首先，扩展可链接环签名扩展，以支持如下形式的环：$\{(pk_{11}, pk_{12}), (pk_{21}, pk_{22}), \cdots, (pk_{n1}, pk_{n2})\}$(为简单起见，我们假设每个用户只有 2 个

[1] 这可以通过在接收者的公钥下将随机子的加密附加到交易中来完成

包，因此有 2 个公钥)，其中对于某个 j∈{1，...，n}，签名者需要同时使用 sk_{j1},sk_{j2}用来签名。

其次，我们将同态承诺方案与扩展后的可链接环签名方案相结合。 为了便于讨论，我们假定每个钱包的形式为（pk，cn），其中 pk 是该钱包的公钥(地址),cn 是该钱包数额 v 的同态承诺。 假设实际的支出者(spender)是（pk_{π}，cn_{π}），且支出者选择 n-1 个其他地址，并将它们排列为 (pk_1, cn_1)，(pk_2，cn_2)，…，($pk_{\pi-1}$，$cn_{\pi-1}$)，($pk_{\pi+1}$，$cn_{\pi+1}$)，(pk_n，cn_n)。 假设输出是(pk_{out}，cn_{out})。

第三，计算新的“公钥”pk’i = eval（cn_i，cn_{out}，-），其中 eval 是同态承诺方案的评估函数，pk_i是 $v_i - v_{out}$的承诺。 现在，由于底层同态承诺方案和上面讨论的扩展可链接环签名方案的特殊属性，如果 v_i-v_{out} =0，则 pk_i是有效的公钥。

最后，支出者基于{(pk_{11}，pk_{12}，pk’$_1$)，(pk_{21}，pk_{22}，pk’$_2$)，…，(pk_{n1}，pk_{n2}，pk’$_n$)}计算扩展的可链接环签名。 其理论基础是，支出者知道(pk_{i1}，pk_{i2}，pk’$_i$)的密钥，因为如果交易一致，则 pk’$_i$是承诺 0。 否则，用户将无法计算 pk’$_i$的密钥，从而无法计算所需的扩展可链接环签名。

当方案扩展到支持多个输入和输出时，上述设计容易受到整数溢出/下溢攻击。 具体而言，上述协议仅确保承诺输入值的总和等于承诺输出值的总和。 例如，恶意的支出者可以创建 100 和-99 的输出承诺，如果他使用的输入值为 1，那么交易是有效的。此外，在密码学中，我们在有限域（通常特征是素数 p），因此-99 被认为是 p-99，这是一个很大的值。

为了防止这种类型的攻击，支出者还需要提供证据，来证明每个输出承诺在特定范围内，例如 0，2^{64} -1 之间。 因此，保密交易的最后一部分是承诺价值在给定范围内的范围证明。

3.5.2 抗量子环形保密交易(Quantum Resistant RingCT)

RingCT 协议不具备抗量子安全性。 因此，在量子计算普及后，人们可以很容易地伪造签名，系统也将轻易被破解。 为了提供抗量子安全性，我们使用基于格(Lattice)的密码学来构造整个 RingCT 协议（我们称之为 Lattice RingCT）。HCASH 采用的这种设计可以增强交易中的安全性并保护抗量子时代的隐私。

我们在前面的段落和[7]中采用了这个方案作为基础。 它提供了一次性可链接环签名和同态承诺方案的构造。 然而，它只能支持 1 个钱包作为输入和 1 个钱包输出（因此不需要范围证明）。 我们首先在这里概述细节：

令$R_q = Z_q[x]/f(x),$，其中 f（x）是为 n 次数的多项式。 公参数是$A_0^{'} \in R_q^{1\times(m-1)}$，用于提交标量消息$m \subseteq Dom_m \subseteq R_q$，其中$Dom_m$是该消息的域，如下所示：$Com_{A_0^{'}}(m,S_0) = A_0^{'} \cdot S_0 + m.$ 同态操作的属性也被定义为：

$$Com_{A_0^{'}}(m_1,S_0)\, Com_{A_0^{'}}(m_2,S_0^{'}) \equiv Com_{A_0^{'}}(m_1,S_0) + Com_{A_0^{'}}(m_2,S_0^{'})\ mod\ q$$

$$= Com_{A_0^{'}}(m_1+m_2,S_0+S_0^{'})mod\ q$$

$$Com_{A_0^{'}}(m_1,S_0) \ominus Com_{A_0^{'}}(m_2,S_0^{'}) \equiv Com_{A_0^{'}}(m_1,S_0) - Com_{A_0^{'}}(m_2,S_0^{'})\ mod\ q$$

$$= Com_{A_0^{'}}(m_1-m_2,S_0-S_0^{'})mod\ q$$

其中$m_1, m_2 \in R_q$为环元素，$S_0, S_0^{'} \in R_q^{(m-1)\times 1}$为环元素的（m-1）维向量。 整数$m_1, m_2 \in Z$以二进制编码为系数向量$m_1 = (m_{j,0}, \dots, m_{j,l-1}, 0, \dots 0) \in \{0,1\}^n$，其中$m_j = \sum_{i=0}^{l-1}(m_{j,i} \cdot 2^i)$，且$m_{j,i} \in \{0,1\}$，$j \in \{0,1\}$.

注意：我们的 RingCT v1.0 遵循[7]中的可链接环签名方案而构造，使用了一个称为$L2RS.Lift$的附加函数来将环元素从$R_q^{1\times m}$提升到$R_{2q}^{1\times m}$。 以下对我们 RingCT 的描述中，我们将根据需要，利用该函数做出一些提升。

Lattice RingCT v1.0 算法由以下五种算法（Setup，KeyGen，Mint，Spent，Verify）组成：

1. $PubParams \leftarrow Setup(\lambda)$: 输入安全参数λ，该算法输出公共参数$A_0^{'}$，和$H_0^{'}$，其中$H_0^{'} = (\hbar_{0,1}, \dots, \hbar_{0,m-1}) \leftarrow R_q^{1\times(m-1)}$是$m-1$个环元素的向量（注意，散列函数$H_0^{'}$用于在步骤 4 中创建环签名。更多细节参见[7]

2. $(A_{in}, S_{in}) \leftarrow KeyGen(PubParams)$: 在输入公共参数时，输出输入钱包（IW）密钥对$(A_{in}, S_{in})$，其中$A_{in} \in R_q^{1\times m}$是公钥（或 一次性地址）和$S_{in} = (S_0, 1) \in Dom_{S_0} \subseteq R_{2q}^{m\times 1}$是私钥。 零的承诺被定义为：$a'_{1(in)} = A_0^{'} \cdot S_0 \bmod q \in R_q = Com_{A_0^{'}}(0, S_{0(in)})$.

3. $(cn^{'}, ck^{'}) \leftarrow Mint(A_{in}, a_{in})$：输入一个有效的一次性地址$A_{in}$以及一个输入数量$a_{in} \in \{0,1\}^n$来创建一个币$cn_{in}^{'}$，该算法从$S_0$的域中选择一个币密钥$k_{in}^{'} \in Dom_{S_0}$，其中每个分量均以$(-2^\gamma, 2^\gamma)$为系数被均匀而独立地被选择。 它计算对输入量$a_{in}$的随机性$ck_{in}^{'}$作为$cn_{in}^{'} = Com_{A_0^{'}}(a_{in}, ck_{in}^{'})$的承诺，并返回（$(cn_{in}^{'}, ck_{in}^{'})$. 一个账户构成$(a_{1(in)}^{'}, cn_{in}^{'}) \in R_q \times R_q$.

4. $\left(TX, \sigma_{L^{'}}(\mu)\right) \leftarrow Spend(\mu, OW)$：该算法由采用[7]中的环签名方案的以下步骤组成：

a. 输出钱包（OW）的新币由支出者创建。 它生成$ck_{out}^{'} \in Dom_{S_0}$，

其中每个分量都被均匀且独立地选择，系数为$(-2^{\gamma}, 2^{\gamma})$，则计算$cn_{out}^{'} = Com_{A_0}^{'}(a_{out}, ck_{out}^{'})$，其中$a_{out}$表示输出量。新的OW被设置为$(a_{1(out)}^{'}, cn_{out}^{'}) \in R_q \times R_q$.

b. 转账字符串$\mu \in \{0,1\}^*$定义了环签名消息。

c. 基础环签名的公钥列表被构造为$L^{'} = \{(\hat{a}_{1(in),i}, cn_{in,i}^{'})\} \in R_q \times R_q$ for $1 \le i \le w$是环签名的大小。其分量的被如下方式产生：

i. $\hat{a}_{1(in),i}^{'} = a_{1(in),i}^{'} + cn_{in,i}^{'} - cn_{out}^{'} = Com_{A_0}^{'}(a_{in,i} - a_{out}, S_{0(in),i} + ck_{in,i}^{'} - ck_{out}^{'})$.

ii. $cn_{in,i}^{'} = Com_{A_0}^{'}(a_{in,i}, ck_{in,i}^{'})$.

d. 我们称[7]中的$L2RS.Lift$函数将$L^{'}$从$L^{'}$ from $R_q^{1\times m}$提升到$R_{2q}^{1\times m}$：

i. $L^{'} = \{(L2RS.Lift(A_0^{'}, \hat{a}_{1(in),i}^{'}), L2RS.Lift(A_0^{'}, cn_{in,i}^{'}))\} = \{(\hat{A}_{1(in),i}, CN_{in,i})\} \in R_{2q}^{1\times m} \times R_{2q}^{1\times m}$, for $1 \le i \le w$.

ii. 用户π的私钥为 $S_{in,\pi}^{''} = (S_{in,\pi}, CK_{in,\pi}) \in R_q^{m\times 1} \times R_{2q}^{m\times 1}$, where:

- $S_{in,\pi} = (S_{0(in),\pi} + ck_{in,\pi}^{'} - ck_{out}^{'}) \in R_{2q}^{m\times 1}$.
- $CK_{in,\pi} = (ck_{in,\pi}^{'}, 1) \in R_{2q}^{m\times 1}$.
- e. 该算法创建环签名$\sigma_{L^{'}}(\mu)$.（更多细节见[7]，算法5）。
- f. 我们设置事务$TX = (\mu, L^{'}, OW)$.
- g. 该算法最终输出TX和$\sigma_{L^{'}}(\mu)$.
-

5. $(Accept/Reject) \leftarrow Verify(TX, \sigma_{L^{'}}(\mu))$: 该算法验证签名（根据[7]中的算法6）并返回$Accept$或$Reject$。

这种结构支持从一个IW到一个OW，因此在这种情况下，范围证明[8]是不需要的。

下一步是将其扩展到多个输入/输出钱包。 我们正在基于格的环境中努力实现这一部分。

最后一步是给出所有承诺值的基于格的范围证明。 有几种方法可以实现这个目标。 原始方法是构建一个 2 比 1 的环签名来进行范围证明。 更有效的方法是部署子弹证明技术(the bullet-proof technique)[9]并将其转换为基于格的设置。目前我们还在研究关于这部分的解决方案。

一次性公钥生成

交易接收者的一次性公钥可以由接收者给予发送者。 但是，这意味着发送者必须为每次交易而联系接收方，这可能并不是我们希望的。 另一方面，公开一次性公钥可能会危及接收者的隐私。为了解决这个问题，我们将设计一种新的隐形地址生成机制。

更具体地说，接收者发布一组长期公钥。每次发送者希望向该接收者发送交易时，他/她选择该组长期公钥的随机子集，并使用它们来生成新的伪随机公钥。该过程中使用的随机性由发送方和接收方之间的非交互式密钥交换确定，因此不需要传送。最后，接收者可以恢复这种随机性来重新计算随机子集，并计算伪随机公钥的私钥。 这样就隐藏了接收者的身份，而不需要发送者和接收者之间的额外通信。

3.5.3 零知识证明

零知识简易非交互式论证（zk-snark），通常被称为零知识证明，是另一种提供交易隐私的有效技术。 具体来说，zk-snark 允许证明者提供一个简短的证明来证明自己知道某个秘密而不透露该秘密的事实。被认为提供非常高级隐私保护的加密货币零币（Zcash），是 zk-snark 的第一个成功应用的例子。

然而，目前 zk-snarks 技术仍然有很多限制。因此，应用 zk-snark 的 zcash 有以下问题。虽然 zk-snark 在证明大小（proof size）和验证者计算量（verifier’s computation）方面是高效的，但在证明者计算量（prover’s computation）和参数大小（parameter size）方面它是低效的。 具体来说，Zcash 中的公共参数大约为 1 GB，即使是强大的计算机，仍需要几分钟才能生成一次证明。 对算力和记忆体的高要求使得 zcash 实现安全的独立移动钱包变得非常困难。

随着移动设备的普及，移动支付已被广泛接受，如果要在加密货币中使用 zk-snark，则迫切需要解决上述问题。我们在改进效率方面已取得了令人鼓舞的进展。我们估计这些对效率的改进应该足以实现基于 zk-snarks 的加密货币的移动支付。下面我们简要介绍一下我们的发现，其细节可以在[10]中找到。提高效率是基于以下改进。首先，zk-snark 要求需要证明的语句以算术门的形式表示。在高层次上，这些语句的陈述与代币的构建块相关，包括散列函数，承诺方案和伪随机函数。在 zcash 中，这种构建块的选择是散列函数 SHA256。然而，这种构建块的电路由近百万个布尔门组成，也导致参数和证明者的结果由数百万个元素和步骤组成。

首先，我们建议选择能以更少算术门表示的构建模块。这样，证明者的计算和参数大小要比现在的系统小得多。

在第二个改进中，我们在两方面优化了架构。一个是使用三元搜索树而不是二叉搜索树；第二，使用窗口大小=3 的基于窗口的模幂运算。虽然通常二叉搜索树更简单，但在一个配备有双线性映射的组中，可以用较少的算术门来表示涉及 3 个元素的操作，因此，整个架构可以用更小的算术门表示，从而进一步提高效率。

结合这两种技术，我们基于 zk-snarks 的加密货币参数大小将减少大约 90%，为实现在移动客户端的钱包提供了必要的基础。

3.5.4 结论

HCASH 基于格（lattice）的密码学构造的 RingCT 协议（我们称之为 lattice RingCT 协议），实现了量子抗性，可以在未来的量子环境下保护用户的隐私，增强交易的安全性。HCASH 对 zk-snark 的改进和优化，也为安全的移动端支付提供了理论和技术基础。

HCASH 开发团队对 RingCT 协议和 zk-snark 所做的这些创新升级，将在未来给 HCASH 用户在高等级隐私保护方面提供更多的选择方案。

3.6 区块链和 DAG 的双重侧链

在 HCASH 的白皮书中，明确地将打造区块链和 DAG 的双重侧链作为我们的重要技术愿景之一， HCASH 是一个链接各种区块链技术的新的底层技术平台，并实现基于区块链和基于非区块链的分布式系统信息与价值的互联互通。

在技术开发的第一阶段，DAG 并非是 HCASH 开发的重点，此阶段技术团队聚焦于对 DAG 技术的未来路线的探索和研究上，随着技术开发的深入，后续我们会有更多的成果公布。

DAG 以其独特的技术特性和发展潜力，对于未来壮大 HCASH 生态有着不可或缺的作用。HCASH 技术团队力图在双主链生态中开发基于 DAG 结构的分布式账本，为用户提供去中心化、去许可化、去信任化、具有公平访问权限和可加密协议的分层基础设施网络架构，致力于推动商业经济和分布式账本系统的结合，支持行业与企业链上资产流通与交换，实现更广泛的数字资产商业应用需求。

3.6.1 DAG 技术的潜力和优势

DAG（有向无环图），它是一种非块状的数据存储结构，“有向”指的是有方向，准确的说应该是同一个方向，“无环”则指够不成闭环，任何一个方向都无法回到原点。在大数据领域中，DAG 通常用于大数据框架比如 Hadoop、Storm、Spark 的执行引擎。如图 8：

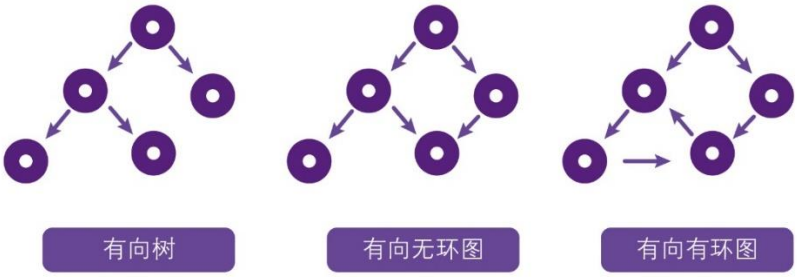


图 8：数据结构图

在 DAG 中，没有区块的概念，DAG 的组成单元是一笔笔的交易，无需矿工进行打包。交易依赖于后一笔交易对前一笔交易的验证，换句话说，你要想进行一笔交易，就必须要验证前面的交易，具体验证几个交易，根据不同的规则来进行。这种验证手段，使得 DAG 可以异步并发的写入很多交易，并最终构成一种拓扑的树状结构，能够极大地提高扩展性。

人们最早注意到 DAG，也是出于区块链扩展方面的考虑。DAG 在某些方面的确有着巨大的优势。包括：

1.交易速度快

由于 DAG 摒弃了区块概念，交易直接进入全网中，所以交易速度预期比基于 PoW 和 PoS 的区块链会快出很多。

2.无需挖矿

DAG 把交易确认的环境直接下放给交易本身，无需由矿工打包成区块后同意交易顺序。所以 DAG 网络中没有矿工的角色，从而可以大大节省能源。

3.极低手续费

在 DAG 网络中，不会出现类似比特币和以太坊因为矿工的激励机制带来的价格竞争，手续费极低，因此特别适合小额高频交易。

DAG 技术的高处理速度、极低手续费等特点，非常适合用于一些高频、小额的交易，特别是在推动区块链技术与商业的结合上，有着独特的优势。HCASH 在自己的生态中打造一条 DAG 侧链，不仅可以大大提升主链的扩展能力，还可以引进更多的资源，扩大 HCASH 生态。目前市面上的 DAG 项目大都集中于某一个行业或者某一个单一领域，我们认为，这一领域还有很多可挖掘的潜力。

3.6.2 现有 DAG 技术的不足之处

尽管 DAG 有着自身的优势，但要实现 DAG 侧链并非那么容易的事情。其中一个关键原因就是，当前 DAG 也有其不可回避的缺点，包括：

1、交易时长不可控

DAG 网络里的验证规则是后面的交易验证前面的交易，这就很容易出现最后的交易迟迟无法被验证的情况，尤其是在网络发展初期节点数量比较少的情况下，交易时长无法预测。尽管当前 Byteball 等项目采用了见证人等手段来解决这个问题，但一定程度上违背了去中心化。

2、异步机制导致一致性的不可控

DAG 作为一种异步通讯机制，其在提高了扩展性的同时也带来了一致性的不可控问题。区块链是同步操作的验证机制，能够保证较高的一致性。DAG 的异步操作机制不存在一个全局的排序，在运行智能合约时，这就很可能会出现节点间所存储的数据在运行一段时间以后出现偏差的情况。

在区块链中，首先需要注意的因素即为一致性，分布式账本需要一致，上链后的交易信息要保持一致，创建的新区块，它里面包含的信息必须要和以往的历史信息保持一致，整个系统都要设计为尽力维持一致性。而 DAG 则更为注重可用性，保证每个用户都能在这个网络上面完成交易，但是，DAG 的异步机制会引发各数据单元无序生成，导致交易冲突。

3、安全性存疑

DAG 技术应用到去中心化账本领域是近几年才发生的，它没有像比特币经历过长达 10 年的安全验证，其安全性还没有得到大规模验证。

3.6.3 HCASH 在 DAG 领域的开发方向

鉴于 DAG 的这些不足，我们发现，要实现更优化、更安全、性能更佳的 DAG 分布式账本，必须解决三大问题：一是资源隔离问题：一旦 DAG 和商业结合落地，必然会有大量的数据和资产上链，尤其是当 HCASH 链接大量的 DAG 账本时，如何对这些不同领域和功能的侧链进行资源隔离的同时，也能保持生态内的价值互通互联，这是一个巨大的挑战；二是安全性问题：DAG 技术仍需提升系统安全保障。三是一致性问题：DAG 的异步通讯机制在提高了扩展性的同时也带来了一致性的不可控问题。区块链是同步操作的验证机制，能够保证较高的一致性。但是 DAG 作为异步操作，它不存在一个全局的排序机制，在运行智能合约时，这就很可能会出现节点间所存储的数据在运行一段时间以后出现偏差的情况。

基于上述考虑，HCASH 团队计划针对 DAG 技术的高并发能力、资源隔离、高效共识系统进行开发，并将在量子安全和跨链领域取得的技术成果融合应用，打造新一代 DAG 分布式账本。

1、高并发能力

相对于区块链分布式账本，DAG 本身已经具备高并发优势。我们试图引入新的技术手段，进一步提升 DAG 系统的高并发能力。

我们通过选择合理的 SQL 数据库，并对现有的 SQL 数据进行改造，有效提升了 DAG 系统地高速高并发能力。

绝大多数关系型数据库系统的执行计划均为解释性语言（一条语句在 SQL Server 中执行,现有进行绑定,语义分析,基于成本的优化等一系列过程，然后生成相应的解释性语言执行计划）而引擎在执行相应的执行计划时会调用相应的数据库函数,运行每一个运算符，系统在执行复杂的解释性语言时，会带来高时间消耗并降低 CPU 的效率，降低 TPS。针对这个问题，我们将做出两点优化：一是选择了相应的内存数据库解决方案，设计了一个基于内存优化的高性能的数据库引擎；二是通过对解释性语言的优化来提高并发能力，例如避免一些会引起全盘扫描的字段。其他手段包括数据分页处理（如数据库分页）、减少交互次数、优化业务逻辑等。

我们将基于这些技术方向进一步实现大幅度提升 DAG 系统高并发能力的目标。

2、分层

由于当前区块链技术体系中的每个全节点都拥有全网所有数据，因此无法满足高容量存储和快速处理的需求。

HCASH 开发团队决定在 DAG 底层数据结构上，通过分层架构体系，将不同的资源进行隔离，从而大幅度地提高主链的性能。未来在 HCASH 的 DAG 分布式账本上，将由不同功能层次的数据逻辑组成一个分层的价值交换网络，分别为基础共识层、合约计算层、数据存储层、跨链交换层、应用支持层。

基础共识层

HCASH 将向用户提供一种去中心化、去许可化、去信任化、具有公平访问权限和可加密协议的 DAG 基础设施网络分布式账本，对接互联网与数字资产;现有互联网应用设备及终端都能作为节点加入到 HCASH 网络中，实现网络、应用开发及价值流通。

任何接入 HCASH 网络的设备，无论是移动终端，还是物联网设备，都能通过共识协议，进入 HCASH 价值交换网络生态。

合约计算层

由逻辑上独立的 Node Network 去中心化计算节点网络组成，每一个 Node 实际都是由 HDVM 安全容器虚拟机+HCASH DAG 侧链扩展智能合约接口组成。

HDVM 安全容器虚拟机用于资源管理隔离、封装虚拟机的 API 通讯等;在之上运行 HCASH DAG 侧链扩展智能核心，HCASH DAG 侧链将具备图灵完备的智能合约模块，用于开发 DAPP，其核心逻辑使用 Node.js 开发，前端则可以使用任意技术，前后端之间通过 JSON RPC 协议进行通讯。

数据存储层

HCASH DAG 侧链将通过独立的多中心化/分布式存储层解决链下数据存储:

1、分布式非结构化存储，这部分利用 IPFS 项目进行分布式储存。主要的原理是，并不需要每个节点来存储所有的信息，有一系列的存储节点在他们之间来分散存储信息。

2、分布式结构化存储， 这部分 HCASH DAG 侧链通过全球化多中心/分布式数据库来实现，并为运行、存储的节点提供奖励。

链下数据的锚定通过 Hash 和默克尔树接口 API 进行，保障校验和数据不可篡改性。

跨链交换层

HCASH DAG 侧链通过未来升级后的 HyperExchange BMT 协议，实现基于区块链的分布式账本和基于 DAG 的分布式账本系统间的价值交换。

应用支持层

HCASH DAG 侧链将拥有独立的 DAPP 支持组件，为企业用户提供方便快捷的 DAPP 开发支持。

3、共识及一致性问题

市场现有的一些基于 DAG 技术的产品，并不支持智能合约，这是因为就 DAG 本身而言，它存在一个较大的隐患——不能完全保证交易状态的原子性和统一性。从时间上来讲，可能存在特定节点(比如远程节点) 确认某笔交易的时间无法估计。从节点上来讲，全网络节点中的某个节点可能无法更新某一时刻的交易信息，即该节点没有被广播到某一时刻的交易信息。这些情况对于很多商业场景来说是一个极大隐患。

所有 DAG 架构中的交易单元，称之为普通交易单元。普通单元中包含签名、交易、父辈哈希值，而每一个交易都是延迟确认的，由于架构实现中存在的不确定性，即交易确认的不确定性，节点同步的不确定性，由于网络延迟所带来的顺序不一致性等， 每一个节点都是由后续节点来确认，但是由于后续节点的加入在不同的网络节点上的插入顺序不能保证，使智能合约的实现成为了一个很大的困难。

HCASH 设计引入基于账户多链结构的 DAG 架构，使得网络中参与者的基本单位独立出来，同时引入了一个验证链，采用专用的算法选举，来解决了上述不确定性，实现智能合约，并且将智能合约和普通交易的解决方案保持了一致。

4、跨链

早期的跨链技术更多关注资产转移 ，以 Ripple 和 BTC Relay 为代表;第二代跨链技术更多关注跨链基础设施 ，代表有 Polkadot 和 Cosmos。HCASH 生态双枢纽之一的 HyperExchange 创新地实现了多资产跨链交易，其图灵完备的智能合约系统支持链上多资产金融衍生交易。

我们的开发团队会继续按照技术开发路线图的设定，推进研发进程，在目前链间跨链技术的基础上，实现基于区块链的分布式账本和不基于区块链的分布式账本（如 DAG）系统之间的价值流通。（参考上文 HyperExchange 跨链原理及流程）

5、抗量子

HCASH 开发团队创新地提出了基于隔离见证思想的新型通信协议，该协议减轻了长签名对网络的压力，实现了抗量子签名，此外，HCASH 基于格（lattice）的密码学构造的 RingCT 协议（我们称之为 lattice RingCT 协议），也实现了量子抗性。这些创新的研究成果未来将会寻求在 HCASH DAG 侧链上的应用，实现量子环境下高等级的安全保护方案。

3.7 去中心化自治

去中心化自治组织(DAO)是密码学技术革命的最理想的产物。DAO 的 源头可以追溯到 Ori Brafman 在《海星和蜘蛛》(2007 年)[11]中描述的组织的去中心化，和 Yochai Benkler 在《网络财富》(2006 年)[12]描述的“对等生产”(peer production)。但是这两个概念被与密码学货币相关的技术所连接起来，Dan Larimer 提出了 DAC 的概念，他将比特币看作一个 DAC。

关于 DAC 为了对 DAC 有一个明晰的定义，我们总结了 DAC 所必需的七点特征:

- 公开性，DAC 系统的设计公开透明，公开透明性是整个 DAC 系统的基石，一个暗箱操作的组织不能作为 DAC，现在的软件开源精神成为公开性的一个典型范例。
- 去中心化性，没有中心化个人和组织能控制整个 DAC，这条特性决定了自相似性，去中心化特性保证了 DAC 系统的生命力。
- 自治性，DAC 系统人人可以参与，参与者都是 DAC 系统的子公司或者子单元，并从自身角度促进 DAC 的发展。参与者的自发行为保障了 DAC 的运行。
- 价值性，DAC 系统必须是具有使用价值的，比如比特币系统的国际支付网络、匿名交易、避税、价值储存、不可冻结、不可监管的特性，这条特性决定了比特币 DAC 系统的盈利性。
- 盈利性，DAC 的参与者会获得 DAC 系统发展的奖励，盈利性由 DAC 本身的价值性确定。
- 自相似性，即使在只有部分 DAC 节点的情况下，DAC 系统仍能正常运作并发展，部分单元节点的摧毁不会影响 DAC 的发展，由去中心化性保证。
- 民主性，DAC 系统核心协议的改变需要绝大多数单元的投票才能完

成，去中心化特性和自治性决定了 DAC 必须是一个能够民主投票的系统。

Vitalik 将 DAC 概念进行扩展，提出了更为普遍的 DAO 概念(分布式自治 组织)，不受监管的众筹和服务拆分是 DAO 的构成要素，还有密码学技术管 理层和基于信任的自动化，这使得 DAO 能够运行起来，正如 Stan Larimer 所 说“在一组商业规则的控制下，不需要人类的参与”。然而这种理想状态下的自 治组织，如果在系统设计阶段不进行严格的把控，也会造成非常严重的后果。2016 年 6 月，史上最大的以太坊众筹项目 The DAO，这个众筹超过 1.5 亿美元的分布式自治组织，因为代码漏洞，遭受黑客攻击，在当时损失超过 360 万以太币，当时的价值超过 6000 万美元。并引发 ETH 社区分裂，造成现有的 ETC 与 ETH 双链共存局面。

在 HyperCash 的系统内，有 5%的代币会发送到一个 DAO，由 HC 的全体持有者通过即时动态投票来决定资金的用途，例如，开发钱包等基础设施建设，或者进行公开推广等商务公关活动，DAO 的形态为 HCASH 社区提供了源源不断的活力与积极向前发展的动力，同时，HCASH DAO 的代码会进过严格的审核并在初期加入必要的人工干预(由基金会邀请第三方进行代码安全审核),以保障 DAO 在早期的资金运用过程中不出现重大失误。

四、技术开发路线图

HCASH 白皮书发布后，我们组建了包括世界著名高校区块链联合实验室在内的开发团队，按照白皮书设计的七大愿景积极推进各项开发工作。由于我们着眼于在区块链领域建立新的技术标准并重新定义价值，因此，所面临的挑战也是空前的。近一年来，在各开发团队成员的共同努力下，经过不断创新，HCASH 在各方面都取得了不同程度的进展，特别是在抗量子签名和链间价值互通互联领域取得了突破性的成果，概括来说：基本实现了区块链间的价值互通互联；基本实现了抗量子签名；创新地将抗量子技术与环形签名相结合，完成了抗量子环形签名的可行性开发路线方案；改进了零知识证明，为实现基于零知识证明的移动支付打下了基础。

作为一项可能会改变世界的新型技术，区块链拥有无限可能，HCASH 对区块链技术的深入探索是永无止境的，我们将会在目前取得的成果的基础上，按照总体规划，加快我们的开发进程，实现我们的技术愿景，同时始终对各种新的区块链技术保持关注并积极实践。

我们在第一阶段开发的基础上，对下一阶段开发工作做出了新的规划，作为全面实现 HCASH 技术愿景的重要阶段，接下来的技术开发路线图如下：


HyperCash
已开发完成
量子抗性签名
PoW+PoS混合共识
社区投票模块功能完备
7月
HC主链性能提升
RC2开放测试
8月
HC主链性能优化完善
9月
HSR/HC正式换币
2018
19 Q1
启动
HAILP协议开发
2019
19 Q2
完成
HAILP协议开发
19 Q3
开发完成
自动部署社区
（自治）决议功能
19 Q4
完成
基于分层、分片技术
实现高并发的DAG侧链开发
HyperExchange
2016
16 Q2
底层平台搭建
智能合约开发启动
2017
17 Q1
虚拟机架构完成
2018
18 Q1
跨链技术开发启动
跨链技术核心功能完成
（BTC/LTC）
跨链技术测试完成
18 Q3
跨链开发完成，主链测试
完成与HC主链的生态集成
完成资产通证化/HRC20标准开发
实现智能合约手续费的系统用户分配功能
完成智能合约执行环境优化
实现轻节点、轻储存
18 Q4
完成集成开发环境/IDE
完成SDK工具包
完成Dapp开发环境
2019
19 Q1
实现DAG跨链交易
19 Q2
完成高性能并行计算功能开发
19 Q3
RPPOM共识升级

五、风险提示

注意：正如这些条款中的其他地方所指出的，HCASH 币不是以货币，证券或任何其他形式的投资产品的形式设计或出售。因此，本节中提供的信息均不构成任何投资决策的依据，也不打算提供具体建议。基金会明确拒绝承担由于以下原因直接或间接导致的任何直接或间接损失或损害的任何和所有责任：（i）依赖本节中包含的任何信息，（ii）任何此类错误，遗漏或不准确之处信息或（iii）由此信息产生的任何行动。

您承认并同意购买、拥有和使用 HCASH 币存在风险。通过购买，持有和使用 HCASH 币，您明确承认并承担以下风险：

由于丢失私钥，保管错误或购买错误而失去 HCASH 币访问权限的风险

私钥或私钥组合对于控制和处理存储在您的购买者电子钱包或其他数字钱包或保管库中的 HCASH 币是必要的。因此，与您的购买者电子钱包或其他数字钱包或保管库存储 HCASH 币相关联的必要私钥丢失将导致这些币丢失。此外，任何获得此类私钥的第三方（包括获得您的购买者电子钱包或其他您使用的数字钱包或保管库服务的登录凭证）可能会盗取您的 HCASH 币。由于您选择接收和存储 HCASH 币的购买者电子钱包或其他数字钱包或保险库导致或与其相关的任何错误或故障，包括您自己未能正确维护或使用此类购买者电子钱包或其他数字钱包或保管库，也可能导致您的 HCASH 币丢失。此外，如果您未能严格按照规定的程序购买和接收 HCASH 币，也可能会导致您的 HCASH 币丢失。

与 HCASH 区块链相关的风险

由于 HCASH 币和平台基于 HCASH 区块链，因此 HCASH 区块链的任何故障，崩溃或放弃都可能对平台或 HCASH 币产生重大不利影响。此外，密码学或技术进步（如量子计算的发展）可能会给 HCASH 币和生态系统带来风险，包括 HCASH 币用于获得服务的效用，因为这会使得基于 HCASH 使用的密码学协议的共识机制无效。

挖矿攻击的风险

与基于以太坊区块链的其他分布式加密代币一样，HCASH 币在验证 HCASH 区块链上的 HCASH 币交易过程中也容易受到矿工的攻击，包括但不限于双花攻击，大多数挖矿权攻击，和自私挖矿攻击。任何成功的攻击都会给平台和 HCASH 币带来风险，包括但不限于准确执行和记录涉及 HCASH 币的交易。

黑客和安全漏洞的风险

黑客或其他恶意组织或组织可能会尝试以各种方式干扰平台或 HCASH 币，包括但不限于恶意软件攻击，拒绝服务攻击，基于共识的攻击，Sybil 攻击，smurfing 和欺骗等。此外，由于平台基于开源协议，因此存在第三方或基金会团队成员有意或无意地将弱点引入平台的核心基础架构的风险，这可能会对平台， HCASH 币，包括用于获取服务的 HCASH 币效用产生负面影响。

与 HCASH 币市场相关的风险

HCASH 币旨在仅作为代币持有者和有意成为 HCASH 币持有者的人员之间的交换方式。本基金会不会支持或以其他方式促进币的二级交易或外部估值。这限制了使用币来提供或接收服务的预期途径，因此可能会对您拥有的任何

HCASH 币造成流动性不足的风险。即使 HCASH 币的二级交易是由第三方交易所推动的，这种交易可能因为相对较新并且很少或者不受监管监督而更容易受到欺诈或操纵。此外，如果第三方确实将外部交换价值归于币（例如，以数字或法定货币计价），则此价值可能非常不稳定并减少至零。

未保险的损失风险

与某些其他金融机构的银行账户或账户不同，除非您专门为其购买了私人保险，否则 HCASH 币没有保险。因此，因此，如果发生损失或效用价值的损失，公共保险公司（例如联邦存款保险公司）或公司安排的私人保险不会向您提供追索权。

与不确定的法规和执法行动相关的风险

HCASH 币和分布式账本技术的监管状况在许多司法管辖区尚不清楚或未得到解决。很难预测监管机构如何或是否可以对这些技术及其应用（包括生态系统和 HCASH 币）应用现有法规。同样很难预测立法机构或监管机构如何或是否会对影响分布式账本技术及其应用（包括 HCASH 币）的法律法规进行更改。监管行为可能以各种方式对 HCASH 币产生负面影响，包括（仅出于说明目的）通过确定币的购买、销售和交付构成非法活动；或币作为一种受管制的工具，需要对这些文书或参与购买、销售和交付的部分或全部当事方进行注册或许可。如果监管行为或法律法规的变化导致在此类司法管辖区内运营非法，或在商业上不适宜获得必要的监管批准以在此类司法管辖区内运营，基金会可能会停止在辖区内的运营。

税收引起的风险

HCASH 币的税务特征不确定。您必须就购买币寻求自己的税务建议，这可能会对您造成不利的税务后果，包括预扣税、所得税和税务申报要求。

竞争协议的风险

可能会建立使用平台下相同的开源代码和协议的替代平台。该平台可能会与这些替代平台竞争，这可能会对采用平台和 HCASH 币产生负面影响，包括 HCASH 币用于币效用。

对平台或分布式应用程序缺乏兴趣的风险

平台可能不会被大量的个人，公司和其他实体所使用，或者更普遍的是，在分布式协议和分布式应用的创建和开发中公共利益受到限制。这种使用或兴趣的缺乏可能会对平台的发展以及 HCASH 币的潜在效用（包括其用于获得服务的效用）产生负面影响。

与平台开发相关的风险

尽管该平台将在币销售时进行部署和运营，但其仍在不断发展中，并可能随着时间的推移发生重大变化。其他参与者如何使用平台也不在基金会的控制范围内。这可能会产生风险，即 HCASH 币或平台，如进一步开发和使用，可能无法满足您在购买 HCASH 币时的预期。平台也可能会出现故障，或者长时间无法进行充分开发，这可能对平台和 HCASH 币的潜在效用（包括其用于获得服务的效用）产生负面影响。

基金会解散的风险

由于许多原因，包括但不限于以太（或其他密码和法定货币）价值的不利波动，HCASH 币的效用（包括其获得的服务效用）的降低，商业关系的失败或知识产权所有权的争议，基金会可能会解散。考虑到基金会在开发平台方面的作用以及基金会在推动平台发展方面的预期作用，基金会的解散可能对平台和 HCASH 币的效用产生不利影响。

缺失治理权引发的风险

因为 HCASH 币不赋予基金会任何形式的治理权利，所有涉及本基金会的决定均由基金会自行决定，包括但不限于决定终止对平台正在进行的开发的贡献或出售或清算基金会。如上所述，这些决定的后果可能会对平台和您持有的 HCASH 币的效用产生不利影响，包括 HCASH 币获得服务的效用。

新的和不断演变的法律影响分布式账本技术的风险

分布式账本和分布式应用生态系统以及扩展平台可能受制于各种联邦，州和国际法律和法规，包括金融服务，消费者隐私，数据保护，消费者保护，内容管理，网络中立性，网络安全，知识产权（包括版权，专利，商标和商业秘密法）等等。这些法律法规以及这些法律法规的解释或适用可能会发生变化。此外，可能颁布影响平台的新法律或法规，这可能对基金会，平台和 HCASH 币造成不利影响，包括 HCASH 币用于币效用。

此外，平台的用户和开发人员可能会受到特定行业的法律法规或许可要求的限制。如果其中任何一方未能遵守任何这些许可要求或其他适用的法律或法

规，或者此类法律法规或许可要求变得更为严格，则可能会对平台和 HCASH 币产生不利影响，包括 HCASH 币获得服务的效用。

HCASH 币价值与功能相关的具体风险

利用 HCASH 币在平台上推出的新功能可能会因基金会无法控制的原因而延迟，并可能最终证明不成功。HCASH 币的值取决于币的效用。价值可能受到市场状况和其他因素的影响。

将 HCASH 币转换为其他加密货币或法定货币的能力将取决于币的交易市场的发展。基金会没有义务促进或支持 HCASH 币的交易。

HCASH 基金会不会承诺币未来的表现或价值，包括固有价值的承诺以及继续付款的承诺。HCASH 基金会也不能保证 HCASH 币将保留任何特定价值。

未预料到的风险

加密货币（如 HCASH 币）是一种新的未经测试的技术。 除本节所包含的风险外，还有与您购买，持有和使用 HCASH 币有关的其他风险，包括基金会无法预期的风险。 此类风险可能会进一步成为本节所讨论的意外变化或风险组合。

六、参考文献

[1] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/ bitcoin.pdf. Oct 2008

[2] Vitalik Buterin. Ethereum White Paper : A Next-Generation Smart Contract and Decentralized Application Platform. https://github.com/ethereum/wiki/ wiki/White-Paper.

[ABL+17] Divesh Aggarwal, Gavin K Brennen, Troy Lee, Miklos Santha, and Marco Tomamichel. Quantum attacks on bitcoin, and how to protect against them. *arXiv preprint arXiv:1710.10377*, 2017.

[BBD09] Daniel J Bernstein, Johannes Buchmann, and Erik Dahmen. *Postquantum cryptography*. Springer Science & Business Media, 2009.

[BDH11] Johannes Buchmann, Erik Dahmen, and Andreas Hu¨lsing. Xmss: a practical forward secure signature scheme based on minimal security assumptions. *Post-Quantum Cryptography*, pages 117–129, 2011.

[BHH+15] Daniel J Bernstein, Daira Hopwood, Andreas Hu¨lsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. Sphincs: practical stateless hash-based signatures. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 368–397. Springer, 2015.

[CFS01] Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In Asiacrypt, volume 2248, pages 157–174. Springer, 2001.

[DDLL13] L´eo Ducas, Alain Durmus, Tancr`ede Lepoint, and Vadim Lyubashevsky. Lattice signatures and bimodal gaussians. In Advances in Cryptology– CRYPTO 2013, pages 40–56. Springer, 2013.

[DLL+17] L´eo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehl´e. Crystals-dilithium: Digital signatures from module lattices. Technical report, Cryptology ePrint Archive, Report 2017/633, 2017.

[dOL15] Ana Karina DS de Oliveira and Julio L´opez. An efficient software implementation of the hash-based signature scheme mss and its variants. In International Conference on Cryptology and Information Security in Latin America, pages 366–383. Springer, 2015.

[DOTV08] Erik Dahmen, Katsuyuki Okeya, Tsuyoshi Takagi, and Camille Vuillaume. Digital signatures out of second-preimage resistant hash functions. PQCrypto, 5299:109–123, 2008.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In ACNS, volume 5, pages 164–175. Springer, 2005.

[Duc14] L´eo Ducas. Accelerating bliss: the geometry of ternary polynomials. IACR Cryptology ePrint Archive, 2014:874, 2014.

[EFGT17] Thomas Espitau, Pierre-Alain Fouque, Benoit Gerard, and Mehdi Tibouchi. Side-channel attacks on bliss lattice-based signatures – exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. Cryptology ePrint Archive, Report 2017/505, 2017. https://eprint.iacr.org/2017/505.

[Flu17] Scott Fluhrer. Further analysis of a proposed hash-based signature standard. Technical report, Cryptology ePrint Archive, Report 2017/553, 2017.

[GLP12] Tim Gu¨neysu, Vadim Lyubashevsky, and Thomas P¨oppelmann. Practical lattice-based cryptography: A signature scheme for embedded systems. In *CHES*, volume 7428, pages 530–547. Springer, 2012.

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 197–206. ACM, 2008.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

[Kat] Jonathan Katz. Analysis of a proposed hash-based signature standard, rev. 4.

[Kat16] Jonathan Katz. Analysis of a proposed hash-based signature standard. In Security Standardisation Research, pages 261–273. Springer, 2016.

[LM95] Frank T Leighton and Silvio Micali. Large provably fast and secure digital signature schemes based on secure hash functions, July 11 1995. US Patent 5,432,852.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In An- nual International Conference on the Theory and Applications of Cryptographic Techniques, pages 738–755. Springer, 2012.

[MBDG14] Carlos Aguilar Melchor, Xavier Boyen, Jean-Christophe Deneuville, and Philippe Gaborit. Sealing the leak on classical ntru signatures. In International Workshop on Post-Quantum Cryptography, pages 1–21. Springer, 2014.

[MCF17] Dr. David A. McGrew, Michael Curcio, and Scott Fluhrer. Hash-Based Signatures. Internet-Draft draft-mcgrew-hash-sigs-08, Internet Engineering Task Force, October 2017. Work in Progress.

[Mer89] Ralph C Merkle. A certified digital signature. In *Conference on the Theory and Application of Cryptology*, pages 218–238. Springer, 1989.

[MW17] Daniele Micciancio and Michael Walter. Gaussian sampling over the integers: Efficient, generic, constant-time. In *Advances in Cryptology - CRYPTO 2017 - 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 20-24, 2017, Proceedings, Part II*, pages 455–485, 2017.

[Nak08] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008.

[NSW05] Dalit Naor, Amir Shenhav, and Avishai Wool. One-time signatures revisited: Have *they become practical? IACR Cryptology ePrint Archive, 2005:442, 2005.*

[PBY17] Peter Pessl, Leon Groot Bruinderink, and Yuval Yarom. To bliss-b or not to be - attacking strongswan's implementation of post-quantum signatures. Cryptology ePrint Archive, Report 2017/490, 2017. https://eprint.iacr.org/2017/490.

[PCG01] Jacques Patarin, Nicolas Courtois, and Louis Goubin. Quartz, 128-bit long digital signatures. In Cryptographers' Track at the RSA Conference, pages 282–297. Springer, 2001.

[Pei10] Chris Peikert. An efficient and parallel gaussian sampler for lattices. In Annual Cryptology Conference, pages 80–97. Springer, 2010.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303– 332, 1999.

[Szy04] Michael Szydlo. Merkle tree traversal in log space and time. In *Eurocrypt*, volume 3027, pages 541–554. Springer, 2004.

[3] Bonneau, J., Clark, J., Felten, E., Kroll, J., Miller, A. and Narayanan, A. 2014. Mixcoin:

Anonymity for Bitcoin with Accountable Mixes. In Financial Cryptography.

[4] Maxwell, G. 2013. Coinjoin: Bitcoin privacy for the real world. Bitcoin forum thread. https://bitcointalk.org/index.php?topic=279249.0.

[5] Chiesa, A., Garman, C., Miers, I., Virza, M., Ben-Sasson, E., Green, M., and Tromer, E.

2014. Zerocash: Practical decentralized anonymous e-cash from bitcoin. In IEEE Symposium on Security and Privacy (S&P).

[6] Miers I., Garman C., Green M., and Rubin A. 2013. Zerocoin: Anonymous distributed e-cash from bitcoin. In IEEE Security and Privacy (S&P), pages 397{411, 2013.

[7] Torres, W. A. A., et al. Post-Quantum One-Time Linkable Ring Signature and Application to Ring Confidential Transactions in Blockchain (Lattice RingCT v1. 0). *Cryptography ePrint Archive Version*.

[8] Noether, S. (2015). Ring SIgnature Confidential Transactions for Monero. *IACR Cryptology ePrint Archive*, *2015*, 1098.

[9] Bünz, B., Bootle, J., Boneh, D., Poelstra, A., Wuille, P., & Maxwell, G. (2017) Bulletproofs: Short Proofs for Confidential Transactions and More. *IACR Cryptology ePrint Archive*, *2017*, 1066.

[10] Kluczniak, K. & Au, MH. (2018) Fine-Tuning Decentralized Anonymous Payment Systems based on Arguments for Arithmetic Circuit Satisfiability. *IACR Cryptology ePrint Archive*, *2018*, 176.

[11]HOFFSTEIN J，PIPHERⅠJ，SILVERⅠMAN J H．NTRⅠU：A ring — based public key cryptosystem

[12]LYUBASHEVSHY V，PEIKERⅠT C，RⅠEGEV O．On ideal lattice and learning with errors over rings