# **第12章 零知识证明与隐私增强**

区块链自诞生之初，便以公开透明、去信任验证为核心设计原则。然而，随着区块链应用逐步渗透到金融、身份、社交等更广泛的场景，这种彻底的公开性也带来了严峻的隐私挑战。在多数公链中，链上交易信息、账户地址以及资金流动路径几乎完全公开，用户的链上活动可被持续追踪，甚至可能被用于画像与滥用。

隐私保护与可验证性的兼容，成为区块链技术亟需突破的核心矛盾。在此背景下，零知识证明（Zero-Knowledge Proof, ZKP）被视为是解决这一问题的基础密码学工具。ZKP 允许证明者（Prover）在不泄露任何敏感信息的前提下，向验证者（Verifier）证明某个声明的真实性[[1]](#footnote-0)。这一技术路径不仅重新定义了链上隐私保护的可能性，也为匿名交易、隐私币、可验证链下计算、跨链互操作以及合规证明等应用提供了全新的基础设施。

## 12.1 零知识证明基础与主流构造

零知识证明之所以成为区块链隐私增强的核心密码学支柱，在于其对“证明”概念的重新定义。与传统证明不同，ZKP 的关键在于验证者无需了解被证明内容的任何细节，也无需信任证明者，仅需通过协议执行即可确认声明的真实性。这一特性使其在身份认证、链上隐私交易、资产负债证明等场景中具备不可替代的价值。

### 12.1.1 零知识证明的基本原理

零知识证明最早由 Goldwasser、Micali 和 Rackoff 于 1985 年在论文《The Knowledge Complexity of Interactive Proof-Systems》中提出，其核心性质包括完备性、可靠性和零知识性。完备性（Completeness）是指若命题为真，诚实的证明者总能使验证者接受；可靠性（Soundness）是指若命题为假，作弊的证明者几乎不可能使验证者接受；零知识性（Zero-Knowledge）是指验证者无法从证明过程获得除命题真实性之外的任何额外信息。

在区块链语境中，这一革命性思想为隐私保护、身份认证和可信计算奠定了坚实基础。随着技术演进，zkSNARK（简洁非交互式零知识证明）、zkSTARK（透明且可扩展的零知识证明）、Bulletproofs（高效范围证明）等构造逐渐成为不同应用场景的核心方案。其中 zkSNARK 与 zkSTARK 更适合通用性证明系统，而 Bulletproofs 则在隐私交易与资产范围证明方面表现突出。

#### （1）零知识证明的定义与核心安全特性

在深入讨论零知识证明的技术细节之前，有必要首先明确其基本概念与核心安全属性。只有准确理解零知识证明的定义及其所必须满足的安全条件，才能为后续的算法设计与应用分析奠定坚实的理论基础。

零知识证明（Zero-Knowledge Proof, ZKP）最初被提出时，是一种基于交互式证明系统（Interactive Proof System）的密码学协议，如图 12-1 所示。它涉及两类参与方，证明者（Prover）拥有某个秘密信息，试图向验证者（Verifier）证明某个命题为真；验证者希望确认命题的真实性，但不希望或不能获取关于秘密的任何额外信息。零知识证明的目标是在证明交互结束后，验证者能够确信命题为真，但不会因此获得除“命题为真”之外的任何附加知识。换言之，证明既要“令人信服”，又要“保持无知”。

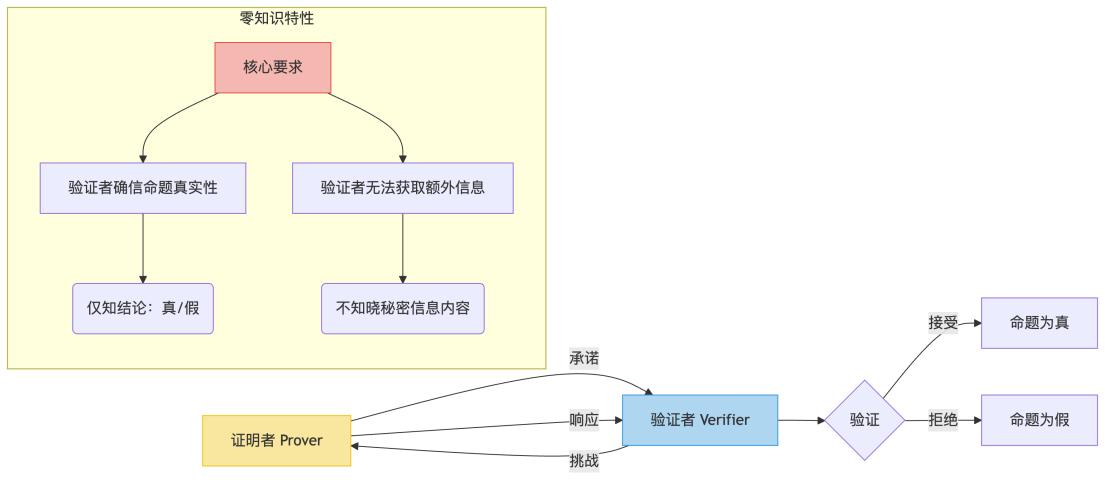


图 12-1 零知识的特性及验证过程

形式化地，一个零知识证明协议必须同时满足三个性质。一是完备性（Completeness），若命题为真，且证明者遵循协议，则诚实的验证者总会接受证明。即如果证明者确实掌握秘密，协议不会被错误拒绝。二是健全性（Soundness），若命题为假，即证明者实际上并不拥有秘密信息，则无论证明者如何尝试欺骗，诚实的验证者被误导接受的概率至多为可忽略量。健全性保证了验证者不会轻易被伪造证明欺骗。三是零知识性（Zero-Knowledge），如果证明者诚实执行协议，则验证者在整个交互过程中不会获得关于秘密本身的额外信息。形式化而言，存在一个模拟器（Simulator），它能够在不接触证明者秘密的情况下生成一段与真实证明过程在分布上不可区分的交互记录，从而保证验证者无法学到新的知识。

零知识性是区别于传统证明的根本特征，它确保证明过程在提供强有力说服力的同时，不会泄露任何潜在的敏感信息。

#### （2）Goldwasser-Micali-Rackoff 交互式证明模型

零知识证明最初是以交互式证明系统（Interactive Proof System）的形式提出的，其核心框架由 Goldwasser、Micali 与 Rackoff（GMR, 1985）奠定。一个典型的交互式零知识证明协议包含多个回合（rounds），如图 12-2 所示。在承诺阶段（Commitment），证明者根据其秘密与随机参数生成初始消息并提交给验证者；在质询阶段（Challenge），验证者随机选择挑战并发送给证明者；在响应阶段（Response），证明者基于质询与自身秘密计算并返回响应。经过多轮交互，验证者根据协议规则判断证明者是否确实掌握秘密。

GMR 证明了该系统满足以下性质。一是完备性，若命题为真，则诚实的证明者总能使验证者接受。二是健全性，若命题为假，即证明者并不掌握秘密，则即便证明者尝试欺骗，验证者被误导的概率也至多为可忽略量；通过增加交互轮数，该错误概率可被任意压缩。三是零知识性，存在一个多项式时间模拟器（PPT simulator），它能够在不接触证明者秘密的前提下，生成与真实交互记录在分布上不可区分的模拟记录。因此，验证者无法从交互中获得任何超出命题真实性之外的知识。

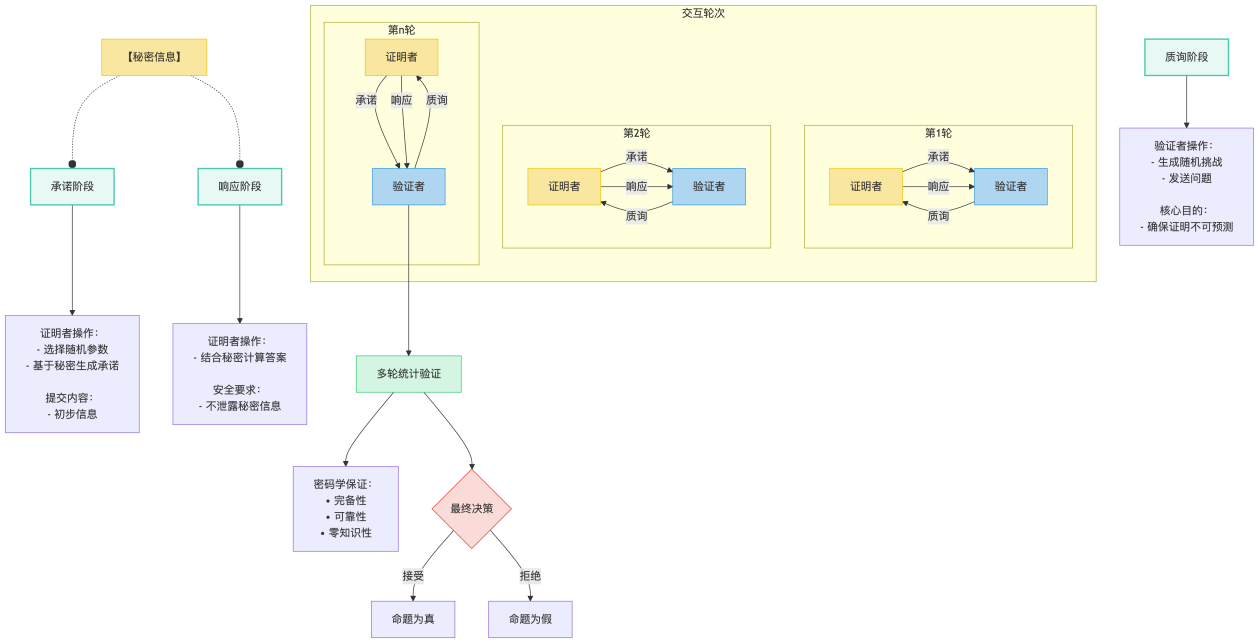


图12-2 交互式零知识证明过程

#### （3）交互式与非交互式 ZKP 的比较

尽管交互式零知识证明（IZK）在理论上完备，但在实际应用，尤其是在区块链环境中，其多轮交互存在明显局限。一是区块链交易通常要求一次性提交，无法支持高频交互；二是网络延迟与链上存储成本使交互式证明开销过大；三是验证者往往并不实时在线，难以进行实时质询。基于此，非交互式零知识证明（Non-Interactive Zero-Knowledge Proof, NIZK）逐渐成为区块链生态中的主流技术路径。

非交互式零知识证明最早由 Blum、Feldman 与 Micali 在 1988 年正式提出，其核心思想是在公共参考串（Common Reference String, CRS）模型下，证明者可以一次性生成一个证明，验证者仅需利用公开参数即可独立完成验证，无需额外交互。

将交互式证明转化为非交互式形式的一个重要方法是 Fiat–Shamir 转换（1986）。其基本思路是利用哈希函数的输出替代验证者的随机挑战，从而使证明者能够自发地产生挑战信息。具体过程如图12-3所示。在承诺阶段之后，证明者将先前提交的消息及公共参数输入哈希函数，得到挑战；随后进入响应阶段，证明者计算出证明并一并提交给验证者。这样，原本需要多轮的交互过程被压缩为单向提交，极大地提升了实际部署的可行性。

非交互式证明的优点包括单轮提交，适配区块链的一次性交易结构；高效的存证与验证，支持链下生成与链下验证；无需实时在线交互，便于离线计算与审计。其挑战则主要体现在多数构造依赖可信设置（Trusted Setup），证明规模和生成成本仍较高，Fiat–Shamir 转换的安全性通常建立在随机预言机模型（ROM）下，在标准模型中仍然存在开放问题。

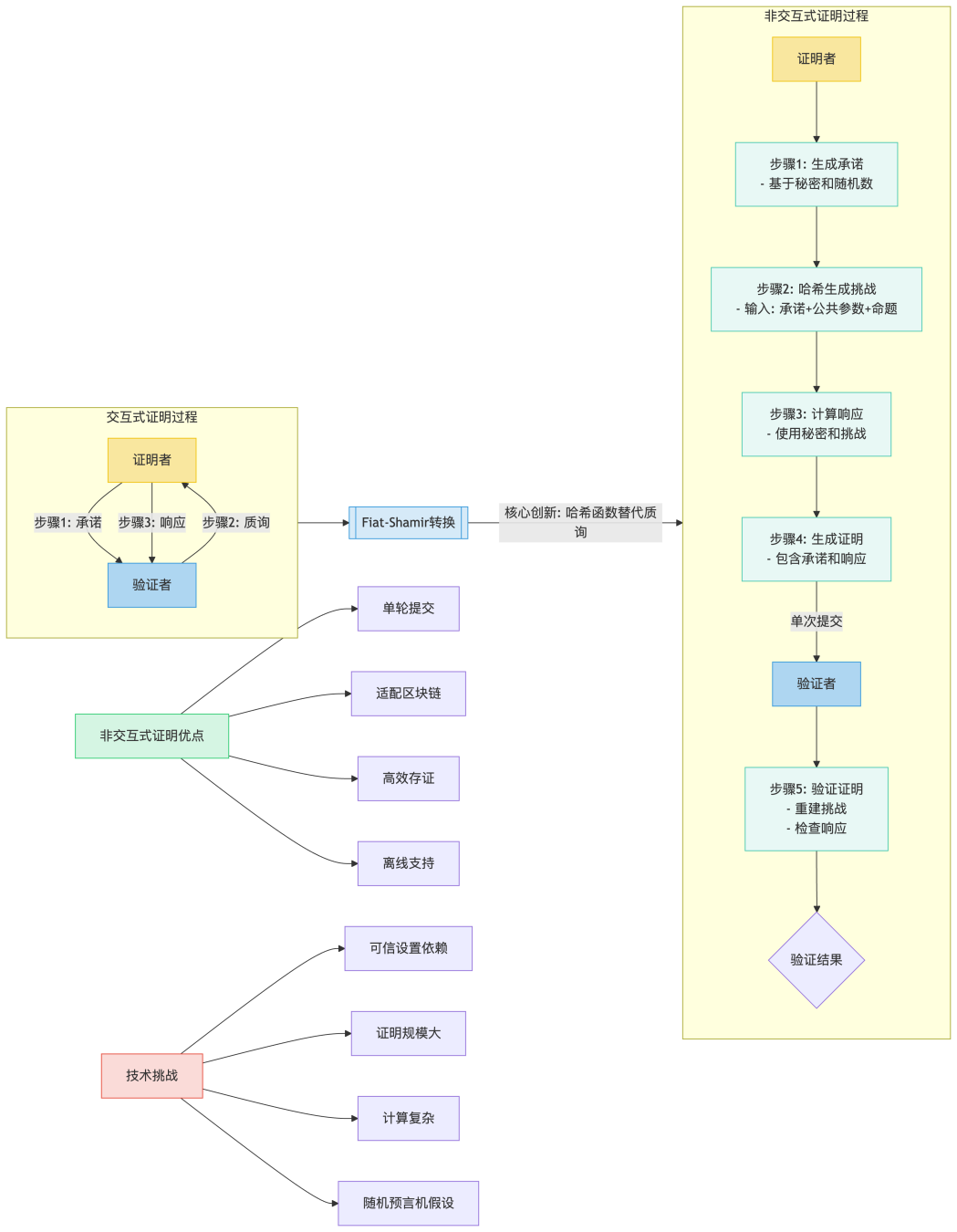


图 12-3 基于 Fiat-Shamir 转换的非交互式零知识证明过程

#### （4）区块链系统对 ZKP 的实际需求与适用场景

区块链的公开透明特性在提供系统安全性的同时，也带来了隐私保护、数据合规与链下验证的挑战。ZKP的应用主要体现在以下几个关键场景。一是隐私交易，代表性项目有 Zcash、Tornado Cash、Aztec。它们利用 ZKP 隐藏交易金额与对手方信息，从而提升链上隐私性并防止资金流被追踪。二是链上身份验证，如 zkID（泛指基于零知识证明的身份系统）、Polygon ID、Manta Network。用户可以证明自身满足某些条件（如年龄、国籍或通过 KYC），而无需公开具体身份，从而支持匿名投票、隐私 DAO 与可验证匿名身份。三是数据正确性证明，包括 ZK-Rollup、zkBridge、zkOracle 等。它们通过在链下执行复杂计算并在链上提交紧凑的零知识证明，保证计算结果的正确性，广泛应用于扩容、跨链通信和链外数据输入。四是隐私智能合约，如 Aztec Connect（已停止运营，但设计理念延续至 Noir 框架）、Railgun、Noir。这些项目使智能合约具备隐私执行能力，可隐藏调用参数、函数输入与交易金额。五是合规可验证路径，通过选择性披露（Selective Disclosure）机制，允许用户在保护隐私的同时，为监管机构提供可验证的合规审计能力，从而避免完全匿名系统带来的非法风险。

ZKP 通过完备性、可靠性与零知识性，提供了“证明而不泄露”的新范式。从交互式到非交互式的发展路径，特别是 Fiat–Shamir 转换的提出，使得非交互式零知识证明在区块链环境中成为主流技术选择，推动了匿名交易、隐私合约、身份认证及跨链验证的广泛落地。然而，ZKP 仍面临计算复杂度高、证明规模大以及可信设置依赖等工程性难题。更重要的是，ZKP 并不仅是技术工具，它在链上治理、金融合规与社会信任等层面可能引发制度性的重新设计。在隐私保护与监管合规之间寻求平衡，将是未来区块链隐私技术持续演进的核心议题。

### 12.1.2 zkSNARK 的数学构造与工程实现

在零知识证明的发展路径中，zkSNARK（succinct non-interactive argument of knowledge，简明非交互式知识论证）是目前区块链领域应用最为广泛的证明系统之一，其核心优势在于证明长度极短、验证效率高，且工具链成熟，因此被广泛用于匿名交易、ZK-Rollup 与跨链验证等关键场景。

#### （1）zkSNARK 的原理与工作流程

zkSNARK 具有四个基本特征，一是简明性（Succinctness），证明通常仅为数百字节（如 Groth16 约 192 字节），验证时间在毫秒级，极适合链上执行；二是非交互性（Non-Interactive），通过 Fiat–Shamir 转换，证明可一次性生成，无需多轮交互；三是知识论证（Argument of Knowledge），证明不仅表明结论正确，还需要确保证明者确实掌握相应 witness（如私钥或交易细节）；四是零知识性（Zero-Knowledge），验证过程不会泄露任何关于秘密输入的信息。

zkSNARK 的工作流程通常分为四个阶段，如图12-4所示。一是电路设计（Circuit Construction），这个阶段要将需要证明的计算任务转化为算术电路或约束系统，最常用的形式为 R1CS（Rank-1 Constraint System），进一步转换为 QAP（Quadratic Arithmetic Program），为后续证明生成做准备。此阶段定义了需要证明的“陈述”的结构。二是可信设置（Trusted Setup），通过 MPC 仪式生成公共参考字符串（CRS），此阶段会产生一个必须安全销毁的有毒废物（toxic waste/tau）。如果秘密参数未安全销毁，可能导致伪造证明的风险。这是 zkSNARK 被批评“需要信任”的主要原因。三是证明生成（Proof Generation），证明者使用其私有输入（witness）和 CRS 中的公共参数，基于椭圆曲线密码学和双线性配对运算，生成一个非常简洁（succinct）的证明 π。这个过程计算量较大，但生成的结果很小。四是验证（Verification），验证者使用 CRS 进行少量双线性配对运算，即可快速验证证明 π 的有效性。输出结果为接受或拒绝，验证时间极短，且与原始计算复杂度无关。整个流程的核心密码学基础包括椭圆曲线、双线性配对、多项式约束系统及同态性质，实现了在不需要透露任何隐私信息的情况下，证明某个陈述的正确性。

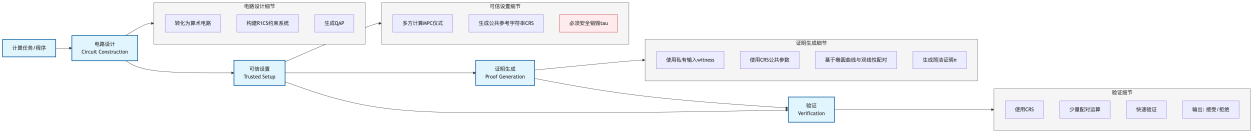


图12-4 zkSNARK 的工作流程

#### （2）椭圆曲线密码学、同态加密与配对运算

在深入理解 zkSNARK 的运行机制之前，有必要回顾其所依赖的若干基础代数与密码学工具。zkSNARK 的高效性与安全性并非源自单一密码原语，而是建立在电路代数化、多项式承诺技术与双线性配对等相互耦合的构件之上。

zkSNARK 的许多实现依赖于配对友好的椭圆曲线（pairing-friendly elliptic curves），其优点包括在给定安全等级下较短的公私钥和较高的运算效率。常用的配对曲线包括 BN254（在早期 Groth16 实现中常用）与 BLS12-381（广泛用于现代构造以获得更高的安全裕度与兼容性）。需要注意的是，配对运算相比普通椭圆曲线标量乘法代价更高，且对实现和参数选择的安全要求更严格。

将待验证的计算转换为代数形式是 zkSNARK 的第一步。常用的约束模型为 Rank-1 Constraint System (R1CS)。令为扩展的 witness 向量（包含公共输入与私有 witness），对每个约束  定义系数向量 ,,，则该约束要求。将所有约束集合起来即描述了整个电路的正确性。等价地，R1CS 可被转化为 Quadratic Arithmetic Program（QAP），进而表达为若干多项式整除关系，这为后续的多项式承诺与配对验证奠定了基础。

在现代 zkSNARK 中，常见的做法并非依赖传统意义上的同态加密，而是利用多项式承诺（polynomial commitment）（例如 KZG 承诺）将多项式值固定下来并以紧凑的方式公开承诺。随后，证明者通过构造多项式证明（evaluation proofs）并将这些承诺与验证方的随机挑战结合，达到在链上以常数复杂度验证多项式恒等式的目的。多项式承诺技术与 R1CS/QAP 的代数化步骤共同构成了 zkSNARK 的核心代数机制。

许多经典的 zkSNARK 构造（例如 Groth16）依赖双线性配对，其关键性质是对任意标量与曲线基点，有。这种双线性映射使得乘法关系或多项式关系能被映射到配对结果的幂次上，从而经由少量配对运算在验证时被高效检查。配对的存在正是使得某些 zkSNARK 既能达到 succinct 性又能实现高效验证的关键。

基于 QAP + KZG + 配对的经典 zkSNARK（如 Groth16）通常需要一次性的可信设置（trusted setup）以生成公共参考字符串（CRS）。若该过程产生的秘密参数（即所谓“有毒废料”）被保留或泄露，则攻击者就可以伪造任意证明。为降低这一风险，实践中常采用多方安全生成仪式（MPC ceremony），只要其中至少一方诚实并销毁其秘密份额，整体系统即保持安全。近年来，出现了若干致力于减少或消除可信设置需求的变体，例如 PLONK 等通用 SNARK 引入“通用/一次性可信设置”或更可复用的机制，而 STARK 类构造则以透明性（无需可信设置、基于哈希与多项式承诺的不同方案）为主要卖点，但通常会牺牲一定的证明规模或验证复杂度。

#### （3）Trusted Setup 的必要性与争议

虽然 zkSNARK 以其证明简洁与高效验证而被广泛采用，但其对 Trusted Setup（可信初始化） 的依赖长期以来成为安全性与信任模型讨论的焦点。在典型的 pairing-based zkSNARK（如 Groth16）中，系统部署前必须生成一组公共参考字符串（CRS），包括证明者使用的 proving key 与验证者使用的 verifying key。该参数生成过程通常通过一次性仪式（ceremony）完成；若过程中产生的秘密参数（俗称 toxic waste）被任何一方保留或泄露，则攻击者可伪造有效证明，直接破坏整个系统的安全性。

Trusted Setup 的主要风险体现在三个方面。一是信任单点，若仪式过度依赖少数参与者，系统可能遭遇单点攻击；二是一次性仪式的维护成本，电路或协议升级通常需要重新 setup，从而限制可扩展性；三是社区信任与合规挑战，若缺乏透明与公信力，社区可能质疑其去中心化属性，削弱合规接受度。

针对这些问题，学界与业界提出了多种缓解方案。一是采用 MPC 仪式并公开审计，确保“只要至少一方诚实即安全”；二是引入通用或可复用 CRS（如 PLONK、Marlin），避免每一电路均需独立的 setup；三是发展透明证明体系（如 STARKs），依赖哈希函数与代数编码理论而完全免除 trusted setup；四是利用递归证明、参数轮换与多协议协同等工程机制降低单次 setup 的长期风险。Trusted Setup 在部分 zkSNARK 构造中仍是必要前提，但随着 MPC 仪式、通用 CRS 与透明证明技术的不断进步，社区已逐步探索出多条技术路径，以减少甚至消除对一次性可信初始化的依赖。

#### （4）Groth16 协议与应用

在众多 zkSNARK 构造方案中，Groth16 协议被认为是迄今最成熟且应用最广泛的技术框架之一。Groth16 的显著优势包括证明规模极小（固定为 3 个椭圆曲线群元素，在 BN254 曲线上约 128 字节，在 BLS12-381 曲线上约 192 字节）、毫秒级的验证时间（通常只需 3 次双线性配对与少量标量乘法），以及对通用电路的支持。凭借这些特性，Groth16 已成为区块链隐私与扩容的核心密码学基元。

在构造上，Groth16 基于 pairing-friendly 椭圆曲线（如 BN254、BLS12-381），利用高效的双线性配对运算将电路约束系统（R1CS）映射为曲线上的代数方程。其证明流程包括三个环节，一是参数生成（Trusted Setup），为特定电路生成公共参数（proving key 与 verifying key）；二是证明生成，证明者结合私有输入与公共参数，通过随机性生成简洁证明；三是验证，验证者仅需常数次配对运算即可确认证明的正确性。

Groth16 的典型应用包括 Zcash（在 Sapling 协议中实现全链匿名交易，支持 z-address 与 t-address 混合使用）和 Tornado Cash（结合 Merkle 树与 zkSNARK，实现匿名存取款，保证资金一致性而不泄露关联路径）。凭借其高效性与成熟的工具链，Groth16 已成为链上隐私交易、ZK-Rollup、身份验证等场景的主流选择。

#### （5）zkSNARK 在区块链隐私与扩展性中的优势与局限

通过前文对 zkSNARK 数学基础、协议设计与应用实例的讨论，我们已经能够较为系统地理解其工作原理与构造逻辑。zkSNARK 的优势主要体现在几个方面，一是高效性，证明规模小、验证速度快，适合链上实时交互；二是非交互性，可通过一次性广播完成验证，降低网络与交互负担；三是通用性，能够支持任意复杂计算电路的约束表达，应用范围广泛。

然而，zkSNARK 也存在若干局限。首先是其核心设计依赖 Trusted Setup，难以彻底去信任化；其次是更新成本较高，每次电路修改均需重新生成参数，限制了协议的灵活性；再次是其基于双线性配对的安全性在量子计算威胁下将难以维持。

在现实中，zkSNARK 已在以太坊生态中形成了广泛应用路径，典型场景包括 zk-Rollup、隐私币（如 Zcash）、去中心化身份认证等。目前大多数系统仍以 Groth16 为主流方案，但新一代 Rollup 与隐私协议逐渐尝试采用 Plonk、zkSTARK 等不依赖电路特定 setup 的构造。

展望未来，zkSNARK 可能通过 MPC 仪式、通用/模块化设计、递归证明等路径逐步降低对 Trusted Setup 的依赖，并与 zkSTARK、Bulletproofs、Plonk 等新型证明系统形成互补。作为当前区块链生态中最成熟的零知识证明体系，zkSNARK 在效率、非交互性和简洁性上展现了密码学与系统设计的深度结合，但可信初始化、抗量子安全、证明电路灵活性等难题仍需继续攻克。随着透明证明与递归证明的进一步发展，zkSNARK 预计将在未来链上应用中继续发挥核心作用，并推动区块链的隐私保护与可扩展性创新。

### 12.1.3 zkSTARK 与新一代透明证明系统

zkSNARK 虽然在工程上取得了广泛应用，但其对“可信初始化”（Trusted Setup）的高度依赖始终被视为潜在的系统性风险。为了从根本上消除这一信任假设，zkSTARK（Zero-Knowledge Scalable Transparent ARguments of Knowledge）作为新一代透明零知识证明方案应运而生。zkSTARK 完全抛弃了 Trusted Setup，并通过哈希函数与信息论工具替代双线性配对运算，从而显著提升了系统透明性与抗量子攻击能力。

#### （1）zkSTARK 的哈希基础与抗量子安全性

zkSTARK 与 zkSNARK 在底层密码学假设上存在根本差异。zkSNARK 多数基于椭圆曲线密码学（ECC）与双线性配对，而 zkSTARK 主要依赖哈希函数的安全性（碰撞抗性、随机预言机模型），核心组件包括 Merkle 树承诺、FRI 协议以及 Fiat–Shamir 转换。这一结构使 zkSTARK 的安全性仅依赖成熟的哈希难题，无需引入复杂的数论假设。

与此相比，椭圆曲线离散对数问题在大规模量子计算下可被 Shor 算法高效破解，而哈希函数在量子攻击下最多遭遇平方根级别加速（Grover 算法），仍能通过增加安全参数维持安全性。因此，zkSTARK 相较 zkSNARK 拥有更强的抗量子安全能力。

尽管 zkSTARK 的证明规模与生成开销尚大于 zkSNARK，但其在透明性与长期抗量子安全方面的优势，使其被认为具备更广泛的未来应用前景。

#### （2）透明性、可扩展性与无需可信设置的优势

zkSTARK 相比 zkSNARK 最突出的特点在于其“透明性”（Transparency）设计，这意味着证明系统可以在完全无需信任第三方的前提下独立部署。透明性是指 zkSTARK 的公共参数可以完全由可验证的随机过程（如可追溯的随机种子）生成，不存在类似 zkSNARK 的 Trusted Setup，这一设计极大降低了系统后门风险和用户信任成本。zkSTARK 通过递归证明与高效的哈希压缩机制，支持超大规模数据证明，理论上可支撑数百万笔交易批量打包并生成单一有效性证明，具备更优的扩展性与未来链下批处理能力。

尽管 zkSTARK 的单次证明生成比 zkSNARK 更为复杂，证明文件规模也更大（通常在数百 KB 甚至数 MB 级别），但其批量处理与链下证明生成的优势正逐渐被新一代工程框架（如 recursive STARKs）所释放，成为适用于大规模区块链扩容的主流路径。

#### （3）STARK 与 SNARK 的对比分析

zkSTARK 与 zkSNARK 的技术差异见表12-1。深入理解 zkSTARK 与 zkSNARK 的技术差异，有助于开发者在不同应用场景中合理选择。

表12-1 zkSTARK 与 zkSNARK 的技术差异见

|  |  |  |
| --- | --- | --- |
| **维度** | **zkSNARK** | **zkSTARK** |
| 密码基础 | 椭圆曲线、双线性配对、多项式承诺 | 哈希函数、FRI、Merkle 树 |
| 可信设置 | 需要 Trusted Setup | 完全透明，无需可信设置 |
| 证明大小 | 极小（约 100–200 字节，典型为 128B） | 较大（数百 KB – MB） |
| 证明生成 | 计算代价高，复杂电路下耗时明显 | 计算更重，但支持批处理摊销 |
| 验证效率 | 极快（常数次 pairing 运算） | 较快（哈希为主），随规模增加成本可控 |
| 抗量子安全 | 不具备（受 Shor 算法威胁） | 具备（哈希基础，抵御量子攻击更强） |
| 批处理与递归 | 支持，但效率受限 | 支持递归与大规模批处理，效率逐步提升 |
| 应用现状 | 广泛应用于隐私币、Mixer、zk-Rollup 等 | 已落地于 StarkEx、StarkNet、DA 证明、递归验证等 |

从表12-1可见，zkSNARK 在证明规模与验证速度方面依然具有显著优势，特别适合单笔交易的链上快速验证与轻客户端应用；zkSTARK 则凭借透明性、抗量子安全性和批量扩展能力，在 rollup 与链下批处理场景展现出更强的系统级适用性。未来，两者可能在不同领域形成互补。

#### （4）zkSTARK 的工程化进展与代表性项目

zkSTARK 已进入工程化阶段，部分项目已投入生产环境。StarkWare 是该领域的核心推动者，其 StarkEx 平台支持链下撮合、链上批量结算，曾被 dYdX（V3）、Immutable X、Sorare 等协议采用；StarkNet 作为通用 Layer 2 网络，结合 Cairo 语言提供可编程性与递归证明支持，正在推动 zkSTARK 的生态化发展。

典型应用包括ZK-Rollup 扩容，StarkEx 批量交易处理显著降低以太坊主网压力；NFT 交易，Immutable X 基于 StarkEx 支持高频链上结算；数据可用性证明，zkSTARK 用于批量数据压缩提交，提高链上带宽利用率。

#### （5）zkSTARK 的发展方向与潜力挑战

尽管 zkSTARK 在抗量子安全性与透明性方面具备优势，但仍面临若干挑战。一是证明规模过大（数百 KB–MB），相较 zkSNARK（~128B）存储与带宽压力显著，未来可通过递归证明与聚合证明优化；二是证明生成速度较慢，虽可借助链下批处理与 GPU 并行优化，但对实时交易仍有限制；三是开发工具链不如 zkSNARK 成熟，尽管 Cairo 已成为 StarkNet 核心语言，但生态工具仍需完善。zkSNARK将与 zkSNARK 长期共存，前者适合大规模批处理与抗量子环境，后者仍适合轻量、低延迟场景。

总体而言，zkSTARK 作为 zkSNARK 的重要演化路径，凭借透明性、抗量子安全与扩展性，正成为区块链隐私保护与扩容的重要技术方向。随着递归证明、硬件加速与开发生态的完善，zkSTARK 有望成为下一代高性能区块链系统的重要基石。

### 12.1.4 Bulletproofs、Plonk、Halo 2等新型证明系统

在 zkSNARK 与 zkSTARK 逐步成熟的过程中，零知识证明技术仍在不断演化，涌现出一系列在性能、安全性与开发体验上进一步优化的新型证明系统。它们不仅在特定应用场景中提供了更加灵活的设计选项，也推动了零知识证明向模块化、可编程、高效递归等方向持续迈进。

#### （1）Bulletproofs无需可信设置的小型证明方案

Bulletproofs 是由 Bünz 等学者于 2017 年提出的，旨在解决传统零知识证明（尤其是 zkSNARK）对于可信初始化的高度依赖问题。Bulletproofs 的最大特点是无需 Trusted Setup，并且具有较小的证明规模，适合链上空间有限但对隐私性有较高需求的应用场景。

Bulletproofs 基于内积证明（inner product argument），支持批量范围证明（aggregated range proofs），即证明某个值在特定范围内，而无需透露该值本身，但其扩展性仍有限，尤其对复杂电路不友好。这一特性使得 Bulletproofs 在机密交易（Confidential Transactions）中尤为重要。与 zkSNARK 不同，Bulletproofs 不需要复杂的同态加密或双线性配对运算，仅依赖于椭圆曲线加法和哈希函数，因此具备更强的工程可实现性。

Bulletproofs 的优势包括无需可信设置，避免了系统后门风险；证明规模小，通常在数 KB 级别，远小于 zkSTARK；适合范围证明、资产保密等链上简单隐私应用。其局限有验证时间线性增长，难以支撑复杂电路的高频应用；不支持批量证明，扩展性有限。

Bulletproofs 已被 Monero 等隐私币广泛采用，用于隐藏交易金额，同时也被部署在部分链下支付系统与去中心化交易协议中，用于提升资产保密性。虽然在复杂计算场景中逐渐被 zkSNARK 与 zkSTARK 替代，但 Bulletproofs 仍在链上轻量级隐私保护中占据重要位置。

#### （2）Plonk的通用电路支持与单次 Trusted Setup

Plonk（Permutation Argument for Knowledge）是近年提出的一种高性能 zkSNARK 证明系统，相较早期 zkSNARK，Plonk 的设计特点包括单次 Trusted Setup，只需进行一次系统参数生成，即可支持所有后续电路，无需为每个应用单独设置可信初始化，显著降低信任风险；同时能够灵活支持任意电路结构，开发者无需针对不同应用重复设计证明电路；验证效率高，其验证效率高于 Bulletproofs，但一般比 Groth16 慢一些；证明规模通常比较小，在不同实现中常在几 KB。Plonk 与 zkSNARK 的对比见表12-2。

表12-2 Plonk 与 zkSNARK 的对比

|  |  |  |
| --- | --- | --- |
| 维度 | zkSNARK（Groth16） | Plonk |
| Trusted Setup | 每个电路一次 | 单次系统级 |
| 电路通用性 | 低 | 高 |
| 验证效率 | 极高 | 较高 |
| 证明大小 | 极小 | 小（KB级） |
| 应用现状 | 广泛应用于隐私币、Mixer | 应用于 zk-Rollup、链上隐私等 |

Plonk 已被 Aztec Protocol、zkSync Era 等多个主流 zk-Rollup 项目采用，其通用性与单次 Trusted Setup 的优势正在成为新一代 zk 应用的基础架构选择。特别是在 EVM 兼容 zk-Rollup 的设计中，Plonk 提供了较好的电路开发灵活性，适合支持通用智能合约的可证明执行。

#### （3）Halo 2递归证明与无信任链路的可能性

Halo 2 是由 Electric Coin Company（Zcash 背后的开发团队）提出的下一代零知识证明系统，其核心突破在于递归证明（recursive proof）与完全无需 Trusted Setup 的链式验证能力。递归证明允许一个证明验证另一个证明，从而实现可验证计算的层层嵌套，这为链下计算、批量交易处理、可验证机器学习等复杂应用场景提供了理论基础。Halo 2 通过优化电路结构、多项式承诺（Plookup, polynomial IOPs）和 folding 技术优化递归性能，显著提升了递归证明的效率，使其具备链上递归验证的工程可行性。

与 zkSTARK 类似，Halo 2 完全无需可信设置，系统参数可通过透明过程公开生成，有效避免了 zkSNARK 可信初始化所带来的潜在安全问题。此外，Halo 2 支持灵活的电路更新与模块化设计，适合未来可升级证明系统的需求。Halo 2 已成为 Zcash 网络未来升级的核心密码技术，目标是实现私密交易、递归批处理、跨链验证等复杂功能的零知识化支持。递归证明的能力也为 zk-Rollup 的链下计算批量压缩与多层证明架构奠定了基础，可能在未来的 Layer 2 与跨链桥设计中发挥关键作用。

#### （4）模块化证明系统的发展趋势与未来路径

随着零知识证明技术的不断演化，证明系统正在从单一电路设计逐步走向模块化、可组合、可插拔的工程体系。包括 Plonk、Halo 2 在内的新一代证明系统，普遍支持通用电路与可编程证明路径，开发者可以通过模块化的电路模板快速组合隐私计算、资产验证、身份认证等功能，这极大降低了 zk 应用的开发门槛。未来零知识证明系统的主流趋势，将向支持递归证明与批量验证倾斜。递归证明不仅可显著降低链上负载，还可以支持可验证链下 AI、复杂供应链溯源等跨链跨应用场景。递归证明 + 模块化电路的组合，可能成为下一代可扩展 zk 基础设施的技术范式。

未来区块链生态中，可能同时存在 zkSNARK、zkSTARK、Plonk、Halo 2、Bulletproofs 等不同证明系统，这些不同的证明系统协同工作，例如轻量级交易使用 zkSNARK，批量链下计算使用 zkSTARK，链上递归验证使用 Halo 2，简单保密交易采用 Bulletproofs，不同证明系统通过桥接协议与互验证机制协同实现跨系统隐私保障。

模块化证明系统的发展离不开友好的开发工具链支持。当前包括 Circom、SnarkJS、Noir、Cairo 在内的 zk 编程语言与证明系统框架正在加速成熟，未来开发者将能够像编写传统智能合约一样快速设计 zk 电路，开发工具链的完善也将进一步推动零知识证明从实验室走向大规模生产应用。

从 Bulletproofs 的轻量级证明，到 Plonk 的通用电路与单次 Trusted Setup，再到 Halo 2 的递归证明与系统透明性，零知识证明系统正在走向更高的可扩展性、更强的开发友好性与更完善的工程实用性。新一代证明系统不仅提升了隐私保护与链上验证效率，更为未来的链下计算、跨链互操作、可验证 AI 与复杂金融系统的 zk 化奠定了基础。模块化、递归、多系统协同的证明架构正在成为行业共识，零知识证明正逐步从单一隐私工具演变为通用的链上安全计算基础设施。

## **12.2 匿名交易与隐私币机制解析**

零知识证明最初的设计动机之一，便是实现“可证明的隐私”，即在不泄露任何交易细节的前提下，向网络验证者证明交易的有效性。这一能力对区块链生态具有极高价值，特别是在资金流动、身份信息、账户关联等需要被保护的敏感场景下，零知识证明提供了链上隐私保护的重要技术基础。

然而，隐私交易在实际应用中并非单纯的技术优化，它也直接面对监管合规、链上追踪、反洗钱法规等现实挑战。如何在去中心化体系中实现强隐私保护，又如何在不破坏系统合法性的前提下，允许监管机构获得必要信息，成为链上隐私协议设计中的核心张力。

### 12.2.1 Tornado Cash链上匿名交易的典型路径

Tornado Cash 是以太坊生态中最具代表性的链上隐私协议之一，其通过 zkSNARK 实现的去中心化匿名交易模型，成为过去几年链上隐私探索的重要案例。Tornado Cash 不仅在技术上展示了零知识证明如何应用于交易隐匿，还因其遭遇美国 OFAC 制裁，成为链上隐私与监管博弈的标志性事件。

#### （1）Merkle Tree 的设计与存款承诺

在 Tornado Cash 的架构中，Merkle Tree 扮演了核心的数据组织角色。每笔存款都会成为 Merkle Tree 的一个叶子节点，树根（root）不断更新，记录当前资金池的全局状态。在 Tornado Cash 的隐私保护机制中，Merkle Tree 与 zkSNARK 的结合是系统匿名性实现的关键，前者提供了高效的数据结构以支持存在性证明，后者则以零知识证明确保验证过程中的隐私性与正确性。

Tornado Cash 的 Merkle Tree 本质上是一个“存款承诺集合”，其叶子节点存储的并非用户地址或金额，而是由用户存款时生成的 commitment（承诺值）。在存款阶段，用户本地生成两个随机数nullifier 和 secret，构成存款凭证 note。commitment 定义为：

commitment = H(nullifier, secret)

其中 H 为 zk 友好的哈希函数（如 MiMC 或 Poseidon）。该 commitment 被作为叶子节点写入 Merkle Tree，树的非叶子节点通过子节点哈希递归计算，最终得到唯一的根哈希 root，并记录在链上智能合约中。这种设计使得链上存储的只是哈希值集合，外界无法从 commitment 还原任何存款细节，保障了隐私性。

#### （2）提款阶段的 Merkle 路径与零知识证明

当用户提款时，需要证明“我的存款存在于资金池中”，但不能暴露是哪笔存款。此时，Merkle Tree 提供了路径证明机制，用户只需提供从自身 commitment 到树根的哈希链，即可证明包含性。

假设用户的 commitment 位于某叶子节点 L，则其 Merkle 路径包括兄弟节点哈希及各级父节点哈希。验证者（链上合约）通过路径重构得到 root，并与链上存储的 root 比对。若一致，则表明该 commitment 确实属于当前池。

然而，若直接暴露路径信息，会泄露 commitment 在树中的位置。因此 Tornado Cash 将该验证逻辑封装进 zkSNARK 电路，使证明过程零知识化，验证者只看到“该 commitment 存在于树中”的结论，而不能获知路径或 commitment 本身。

#### （3）防双花机制

为了避免同一笔存款被多次提款，Tornado Cash 引入了无效化因子 nullifier。在存款时，用户 note 中的随机数 nullifier 会通过哈希函数派生出：

nullifierHash = H(nullifier)

在提款时，用户在 zkSNARK 电路中证明自己掌握与该 nullifierHash 对应的 note，并提交该 nullifierHash 至链上。智能合约将 nullifierHash 与链上“黑名单”比对，若nullifierHash 已在“黑名单”上，则拒绝本次提款，从而防止双花。由于 nullifierHash 仅能由存款时的 note 唯一导出，外界无法从 nullifierHash 逆推出 note 或 commitment，因而不会破坏匿名性。

#### （4）zkSNARK 验证的整体逻辑

用户在提款时需生成并提交一份 zkSNARK 证明，该证明编码了三项核心逻辑。一是所有权证明，证明者掌握某个 note（即知道 nullifier 与 secret），且其哈希等于某一 commitment。二是存在性证明，该 commitment 属于当前 Merkle Tree（通过零知识化的路径验证）。三是防双花约束，对应的 nullifierHash 尚未出现在链上黑名单中。

验证者（智能合约）只需两步即可完成验证。一是检查 zkSNARK 证明的有效性；二是检查 nullifierHash 是否未被使用。若均通过，资金将从资金池发送至用户指定的新地址。整个过程中，智能合约从未接触用户的 note、commitment 或路径明文，外界也无法建立存款与提款地址的直接关联。

Tornado Cash 的核心设计可概括为，Merkle Tree通过哈希结构维护资金池状态，实现匿名集合与存在性证明；zkSNARK通过零知识证明将“所有权、存在性、防双花”三大条件统一编码，确保验证正确性而不暴露任何隐私；nullifier 确保单笔存款仅能提款一次，从而维持系统安全性。这种“Merkle Tree + zkSNARK” 的结合，既保证了去中心化环境下的匿名性，又在协议层面防止了作弊与双花，成为链上隐私技术的经典范式。具体过程见图12-5所示。

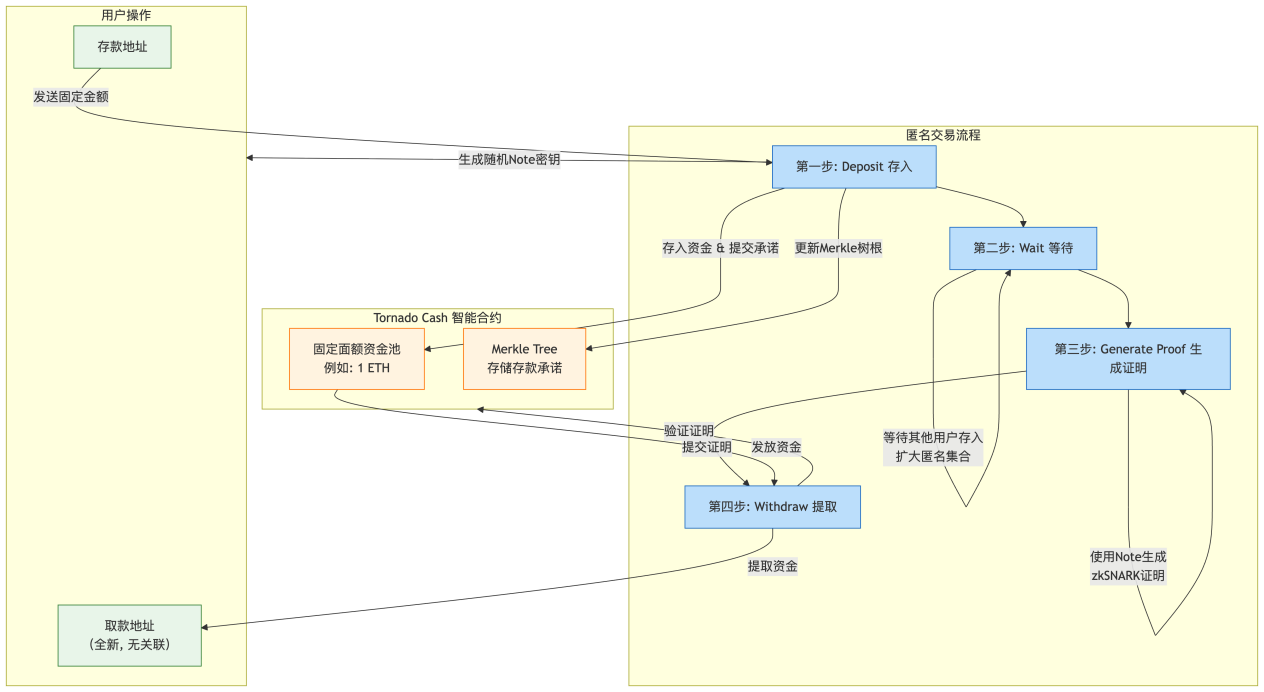


图12-5 Tornado Cash 实现匿名交易的核心四步工作流程

在图12-5中，Tornado Cash 实现匿名交易的核心工作流程共分四步。第一步是存入 (Deposit)，用户从原始地址向 Tornado Cash 合约的特定面值资金池存入固定金额资金，同时生成一个随机密钥(Note)作为取款凭证，合约将该笔存款的承诺记录到 Merkle 树中。第二步是等待 (Wait)，用户需要等待其他用户向同一个资金池存款，以此扩大匿名集合，增强隐私保护效果。第三步是生成证明 (Generate Proof)，用户使用保存的 Note 和 zk-SNARK 技术生成零知识证明，证明自己拥有资金池中某笔存款的有效凭证，而不透露具体是哪一笔。第四步是提取 (Withdraw)，用户将 zk-SNARK 证明提交给智能合约，验证通过后资金将被发送到用户指定的全新地址，彻底切断存款地址和取款地址之间的链上关联。整个过程通过固定面值资金池和 zk-SNARK 技术，确保交易的不可追踪性和隐私性。

### 12.2.2 Zcash的隐私币架构与隐私设计

Zcash 是全球最具代表性的隐私币之一，其创新在于将零知识证明（zkSNARK）嵌入区块链底层架构，实现了隐私保护与可选透明性的兼容。不同于 Tornado Cash 等基于混币池的外部隐私方案，Zcash 在协议层面构建了完整的隐私支付基础设施，其中双地址体系、Shielded Pool 与 Sapling 升级构成其核心机制。

#### （1）Shielded Pool 与承诺机制

Zcash 采用透明地址（t-address）与隐私地址（z-address）并存的设计。t-address 类似比特币地址，交易完全公开，适合审计与合规场景；z-address 依托 zkSNARK 实现“全额隐匿”，发送方、接收方及金额均不公开，仅证明交易在数学上有效。该体系支持 t→t、t→z、z→t、z→z 四类交易路径，使用户在隐私与透明之间具备灵活选择权。

Shielded Pool 是所有隐私交易的全局资产池，其特点在于，一是金额隐藏，通过 Pedersen 承诺，将金额与随机盲因子映射到椭圆曲线上，仅保证输入输出守恒，而不泄露金额本身。二是身份匿名，z-address 仅为加密标识，验证交易时用户需生成 zkSNARK 证明以证明拥有对应私钥，无需公开地址或身份。三是路径断链，交易在池内仅表现为总额变化，链上无法拼接形成可追溯路径。

#### （2）Sapling 升级与 Payment Address 设计

2018 年的 Sapling 升级显著优化了隐私交易性能与可用性。一是Diversified Payment Address，每个 z-address 可生成多个不可关联的接收地址，从根本上避免了长期使用同一隐私地址导致的追踪风险。二是Viewing Key，持有者可统一解密并查看所有关联交易，实现“地址不可关联 + 查看入口统一”。三是轻量化证明，Sapling 将 zkSNARK 证明生成的计算量降低到秒级，使隐私交易更易于在移动设备上执行。

相较 Tornado Cash 依赖用户手动管理一次性 Note，Zcash 通过自动派生地址 + note commitment + viewing key 形成了机制化的隐私保护体系，兼顾高强度隐私性与用户可用性。这使其成为区块链隐私保护领域的里程碑项目。

#### （3）zkSNARK 在匿名支付中的应用路径

Zcash 的隐私交易以 zkSNARK 为核心加密工具，实现了在不公开交易细节的前提下对交易有效性的链上验证。在具体机制上，zkSNARK 可用于以下四个方面的证明。一是输入与输出金额守恒，防止凭空增发；二是发送方确实拥有输入 notes 的所有权；三是输入 notes 未被重复花费；四是证明过程中不泄露金额、地址等敏感信息。其交易流程为，用户在本地钱包生成 zkSNARK 证明，该证明涵盖输入 notes 的 nullifiers（防止双花）、输出 notes 的 commitments（新资产承诺）及金额守恒关系。链上节点只需验证证明是否有效，而无需了解交易具体金额与地址。

在协议层面，Zcash 早期采用 Groth16 zkSNARK 协议，需要 Trusted Setup。为缓解单点信任，Zcash 社区先后组织了两轮多方参与的初始化仪式（2016 年 Sprout 六方仪式；2018 年 Sapling Powers of Tau 大规模仪式）。只要至少一名参与者安全销毁了自己的秘密贡献，系统就不会被破坏；但若所有秘密均泄露，则可能导致恶意方生成虚假证明，从而危及系统整体安全。尽管存在理论风险，zkSNARK 依然因其短证明和高效验证特性，成为 Zcash 早期最具可用性的匿名支付技术路径。

#### （4）透明地址与匿名地址切换的隐私权衡

Zcash 的 t-address / z-address 双地址设计，使用户可在隐私与透明之间自由选择，t-address 保持交易公开，便于合规与审计；z-address 提供完全匿名保护。然而，t ↔ z 的交互会在链上暴露交易模式。若用户将固定金额从 t-address 转入 z-address，再在短时间内将相同金额转出至 t-address，分析者可通过时间戳、金额模式与交互地址建立路径关联，从而显著削弱匿名性。这种现象被称为隐私侧写（privacy profiling）。

为降低风险，Zcash 建议用户，避免单次 t → z → t 路径，应多经由 z → z 转账打散资产；等待更多区块确认，以扩展匿名集；避免在资金流入与流出时使用相同网络环境（如 IP 地址）。

#### （5）Zcash 在隐私币生态中的制度价值

Zcash 不仅是技术创新，也是隐私制度探索的重要案例。其设计通过 Viewing Key 提供“选择性合规”功能，允许用户在需要时向监管或审计方披露部分交易数据，从而兼顾隐私保护与合规需求。相比于全匿名币（如 Monero）在合规市场面临的阻力，Zcash 因保留透明地址而更容易进入合规场景。

此外，Zcash 的多方 Trusted Setup 仪式动员了全球范围的参与者，被视为密码学社区的重要治理实验。其技术路线对后续隐私协议产生了深远影响。Tornado Cash 借鉴了其 zkSNARK 电路验证逻辑；Aztec Protocol 在 Sapling 基础上拓展了账户抽象与可编程隐私；Noir 等零知识 DSL 语言的电路设计也受其启发。

总体而言，Zcash 通过 zkSNARK 证明体系、t/z 双地址架构、Sapling 优化与 shielded pool 隐匿机制，奠定了区块链隐私支付的技术范式。其路径在“强隐私保护”与“可选合规”之间寻求制度平衡，成为隐私币生态中的标杆项目。未来，随着 zkSTARK 与递归证明的发展，Zcash 有望进一步消解 Trusted Setup 风险，并提升其在移动端与高频应用场景下的性能表现。

### 12.2.3 Aztec Protocol 与 Noir隐私合约的新路径

在 Tornado Cash 提供的匿名混币路径与 Zcash 构建的原生隐私币基础上，Aztec Protocol 进一步将隐私保护扩展到智能合约领域，开启了可编程隐私（Private Smart Contract）的新方向。Aztec 不仅在协议层探索了账户抽象与隐私保护的结合，还推出了高级零知识证明开发语言 Noir，从而降低了 ZK 电路编程的复杂度。

#### （1）Aztec Connect 的设计逻辑

Aztec Connect 是 Aztec Protocol 于 2021–2022 年运行的跨链隐私桥（已于 2022 年底停止运行），旨在为以太坊 DeFi 协议提供匿名访问路径。Aztec Connect 采用“双层架构”，以太坊主网作为结算与合规层，仅存储 Aztec 隐私层的状态根和桥接合约，负责资产锁定/解锁；Aztec 隐私层作为隐私操作层，运行在以太坊之上，处理匿名化的资产转移与 DeFi 调用。用户在隐私层不使用公开账户，而是持有加密 notes（由 Pedersen 承诺表示，例如），金额和身份均被隐藏。

用户交互的基本流程如图12-6所示。在入池环节，用户从主网透明账户将资产转入桥接合约，资产在主网锁定，隐私层生成对应的加密 note。主网仅能看到总流入，而无法识别用户与金额。在隐私操作环节，用户在隐私层发起操作（如借贷、兑换），隐私电路生成 zkSNARK 证明，证明操作合法（如满足 Aave 抵押率、Uniswap 曲线约束），但不泄露细节。隐私层将证明和结果提交给桥接合约，由其调用目标 DeFi 协议。在出池环节，用户在隐私层生成证明，表明其持有某一笔加密 note，对应主网桥合约解锁资产并转入指定透明地址。主网仅看到总流出，无法将其与入池地址对应。

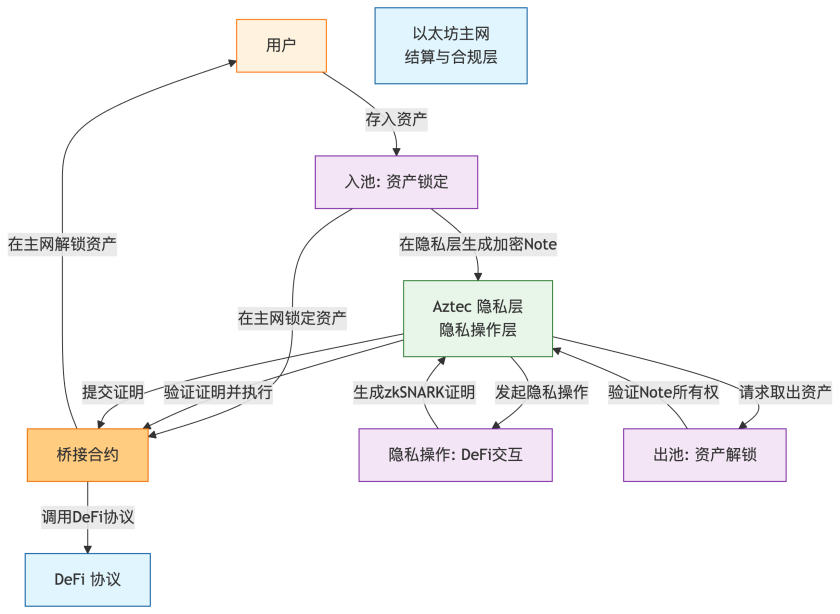


图12-6 Aztec Connect 项目的核心交互流程

为提升性能，Aztec Connect 使用 zkRollup 技术，将数百笔交易聚合为一个批次，链上仅需验证一份聚合证明，从而将验证开销从 O(n) 降至 O(1)，显著降低了 gas 成本。

#### （2）与 Tornado Cash 的对比

与 Tornado Cash 不同，Aztec Connect 并非一次性混币工具，而是支持持续性隐私操作和复杂合约交互。例如，用户可在隐私层进行循环借贷、组合资产管理等操作。Tornado Cash 面临严重监管打击，而 Aztec Connect 通过 zkRollup、可编程隐私和账户抽象，展现了更强的可扩展性和潜在合规性。其创新在于，不改变以太坊 DeFi 协议本身，而是在其上叠加一层隐私保护机制，使用户能够在保持 DeFi 便利性的同时，避免交易数据泄露和身份关联风险。

尽管 Aztec Connect 已停止运行，但其设计理念标志着隐私保护从“静态匿名存取”迈向“动态可编程合约执行”的新阶段，为未来隐私友好型区块链体系提供了重要的实验性范式。

#### （3）Account Abstraction + Privacy 的合约模型

Aztec Protocol 的另一核心突破是将 Account Abstraction（账户抽象）与隐私保护深度结合，构建了隐私合约的通用执行架构。这一模型主要在 Aztec 3 的设计愿景中提出。账户抽象（AA）允许用户自定义账户逻辑，打破了以太坊传统的 EOA（Externally Owned Account）与合约账户的硬性区隔。

在以太坊生态中，传统账户体系（EOA账户）的核心验证逻辑被硬编码在区块链底层，用户必须通过私钥签名证明“所有权”，且签名过程完全公开（链上可见签名信息与账户地址的关联），这既限制了账户功能的灵活性（如无法自定义多签、时间锁等规则），又暴露了用户的操作痕迹。

Aztec Protocol基于账户抽象（Account Abstraction，AA）理念，对这一体系进行了颠覆性重构。它将用户账户设计为可编程的zkSNARK电路，账户的所有验证逻辑（身份确认、交易权限、操作规则等）都被编码进电路中，通过零知识证明（zkSNARK）完成验证，既保留了账户控制的灵活性，又实现了全流程隐私保护。其核心设计可从“账户即电路”的本质、具体功能实现、自定义规则落地三个维度展开。

在Aztec的设计中，“用户账户”不再是简单的地址与私钥组合，而是一套自定义的zkSNARK电路。电路的核心作用是“定义验证规则”，即“满足哪些条件时，一笔交易或操作被视为合法”。传统EOA账户的验证逻辑是固定的（“私钥签名即合法”），且验证过程需要公开签名信息；而Aztec的账户电路则将验证逻辑转化为“数学约束”，验证时无需公开具体规则或操作细节，只需生成一个zkSNARK证明，也就是“我满足电路中定义的所有合法条件”。例如，一个简单的Aztec账户电路可能包含这样的约束，“交易发起者必须拥有某组私钥中的至少一个（多签逻辑），且当前时间已超过2025年1月1日（时锁逻辑）”。当用户发起交易时，电路会自动检查是否满足这些条件，若满足则生成证明，链上验证者只需确认证明有效，即可允许交易执行，全程看不到“哪组私钥签名”“具体时间限制”等细节。

Aztec通过“账户电路+zkSNARK”的组合，将传统账户的核心操作全部转化为隐私保护下的验证过程。一是身份验证证明“我有权操作账户”，但不暴露身份。传统EOA账户通过公开“私钥签名”证明身份（签名信息直接关联账户地址，链上可追踪），而Aztec的账户电路将身份验证逻辑编码为电路约束，通过zkSNARK证明“身份合法性”。账户电路中预定义“身份凭证”（如一组私钥、生物信息哈希、硬件设备签名等），用户操作时需向电路输入自己的凭证。通过电路验证“输入的凭证是否与预定义的合法凭证匹配”，若匹配则生成zkSNARK证明“我拥有该账户的合法身份凭证”。链上验证者仅需确认证明有效，即可认可用户的操作权限，但永远无法从证明中反推出“用户用了哪类凭证”“凭证的具体内容”（如私钥、生物信息等）。例如，一个绑定硬件钱包的Aztec账户，其电路会验证“输入的硬件签名是否与预存的硬件公钥匹配”，生成的证明仅体现“验证通过”，不会暴露硬件钱包的公钥或签名细节，彻底切断“操作-身份”的链上关联。

二是交易签名证明“交易符合我的意愿”，但不公开交易细节。传统交易中，签名会公开交易的金额、接收地址、Gas参数等细节，这些信息成为链上分析的关键线索，而Aztec的账户电路通过“加密参数+零知识证明”实现交易签名的隐私保护。用户发起交易时，先将交易参数（如转账金额、接收地址）加密后输入账户电路。电路验证“这些参数符合用户预设的规则”（如“单次转账不超过1000 USDC”“接收地址不在黑名单中”等），并生成zkSNARK证明“该交易参数经过我确认，且符合我的限制条件”。该证明中不包含任何明文参数，链上仅能看到“交易被验证通过”，接收方通过对应的解密密钥（由发送方通过隐私通道传递）才能知晓具体信息。例如，用户通过Aztec账户向某地址转账时，电路会验证“转账金额≤账户余额”“接收地址是用户预设的白名单地址”，生成的证明仅体现“参数合法”，链上看不到具体金额或接收地址，避免了交易特征被追踪。

三是交易授权证明“授权规则被满足”，但不公开授权过程。在复杂场景中（如企业账户、联名账户），交易可能需要多人授权、时间锁定或第三方审批，这些授权规则在传统账户中需公开记录，而Aztec通过账户电路实现“授权过程隐私化”。账户电路中预定义授权规则（如“需3/5签名者同意”“需管理员A或B审批”），规则以数学约束形式存在（如“输入的有效签名数量≥3”）。当用户发起需要授权的交易时，相关授权方的凭证（如签名）被输入电路，电路验证“是否满足预定义的授权条件”。验证通过后生成zkSNARK证明“授权规则已被满足”，但证明中不包含“谁参与了授权”“授权的具体顺序”等信息。例如，一个5人多签的Aztec账户，其电路约束为“输入的有效签名≥3”。当3位签名者完成授权后，电路生成证明“≥3个合法签名存在”，链上仅知“授权通过”，但无法得知是哪3人签了名，保护了授权者的隐私。

Aztec的设计打破了“隐私保护即功能简化”的误区。传统隐私协议往往为了匿名性，牺牲了账户的灵活性（如仅支持简单转账，不支持多签、恢复等复杂操作）；而Aztec通过“账户即电路”的架构，让用户既能享受零知识证明带来的全流程隐私（身份、交易、授权均不公开），又能像使用智能合约账户一样自定义验证规则，满足个人、企业、机构等不同场景的需求。这种融合的意义在于，它让“隐私”不再是DeFi的附加功能，而是账户体系的基础属性。用户可以在匿名的前提下，安全地进行多签管理、定时转账、社交恢复等操作，真正实现“我的账户我掌控，我的操作不泄露”。对于区块链向主流金融场景渗透而言，这种“灵活且匿名”的账户管理能力，无疑是消除用户隐私顾虑的关键一步。

Aztec 的隐私合约架构具有以下特点。一是支持状态隐私，合约账户余额、状态变量对外不可见；二是支持函数调用隐私，合约函数调用路径、输入参数对外隐藏；三是支持多用户共享隐私状态，多个用户可在同一隐私合约中进行隐私交互。这一架构区别于以太坊透明合约，实现了更高程度的链上匿名性与用户资产保护。Aztec 的隐私合约与传统智能合约的对比见表12-3。

表12-3 Aztec 的隐私合约与传统智能合约的对比

|  |  |  |
| --- | --- | --- |
| 特征 | 以太坊公开合约 | Aztec 隐私合约 |
| 账户结构 | EOA 与合约账户分离 | 账户抽象，合约即账户 |
| 交易透明度 | 全链可见 | 状态、金额、路径隐藏 |
| 交互路径 | 显式合约调用链 | 隐私合约链，路径对外不可见 |
| 交易验证 | 签名验证（ECDSA） | 零知识证明验证（证明交易参数和身份符合电路逻辑） |
| 资产显示 | 账户余额公开 | 账户余额隐私 |

Aztec 隐私合约的设计，使用户可在完全匿名的状态下参与链上复杂交互，显著扩展了隐私保护的适用范围。

#### （4）Noir高层 ZK DSL 语言的设计理念与优势

Noir 是由 Aztec Protocol 团队主导研发的一门面向 ZKP 的高级领域特定语言（Domain-Specific Language, DSL），其核心目标在于降低零知识电路的开发门槛，使开发者无需深入掌握椭圆曲线运算、约束系统建模等底层密码学细节，即可像编写常规程序一样构建安全且高效的证明逻辑。与传统 ZK 电路开发相比，Noir 的价值在于将零知识证明的构建过程“工程化”，推动其从实验性密码学走向规模化应用。

在零知识证明中，“电路”用于定义待证明的条件（如“某笔交易金额合法”“用户签名符合阈值规则”）。传统开发语言（如 Circom）通常要求开发者手动编写底层约束逻辑，代码冗长且难以维护。而 Noir 通过更高层的抽象语法，显著降低了复杂性。一是函数与模块化支持，Noir 允许将常用电路逻辑封装为函数（如“验证数字签名”“计算哈希”），并进一步组织为模块（如“转账模块”“权限验证模块”）。这使得电路逻辑可以像积木一样复用与组合，避免冗余编码。二是逻辑复用与组合能力，在多步骤验证场景中（如先验证身份，再检查余额，最后确认时锁），开发者可直接通过函数调用链与模块嵌套来构建逻辑流程，而无需重新设计底层约束。这种“乐高式”抽象降低了零知识应用开发的复杂度，使其从“密码学专家的专属工作”转变为“普通开发者可参与的工程开发”。

Aztec 的核心创新之一是“账户即电路”的理念，即用户账户的验证逻辑由 zk 电路直接定义。Noir 作为 Aztec 的官方语言，从语法层面原生适配该架构，开发者能够用简洁代码定义自定义账户的验证规则。例如，在设计多签账户时，开发者无需手写“签名数量大于等于阈值”的底层约束，而只需在 Noir 中编写类似如下的伪代码：

fn verify\_multisig(signatures: [Signature; 5]) -> bool {

let valid\_count = signatures.iter().filter(|s| s.is\_valid()).len();

valid\_count >= 3

}

上述代码可视为抽象示例，编译器将其转换为符合 PLONKish 电路规范的约束逻辑，并与 Aztec 账户架构集成。对于时间锁、社交恢复等功能，Noir 提供了预置接口（如 check\_timestamp(lock\_time)），开发者可直接调用，而无需关心底层时间戳到约束系统的映射细节。这一机制为企业多签、定时转账、社交恢复等隐私账户的多样化设计提供了便利。

需要指出的是，Aztec 2 已停止运行，Aztec 3 仍在研发阶段，上述功能多处于设计与测试阶段，尚未在生产环境全面落地。

零知识证明的效率（证明生成速度与验证耗时）高度依赖电路的约束数量。Noir 的编译器具备自动优化能力，能够识别冗余约束并进行合并。例如，当开发者代码中出现数学等式间的重复关系时，编译器会自动约简，确保生成的电路既正确又高效。Noir 采用证明系统无关的设计（proof-system agnostic），但目前主要与 PLONKish 系统集成。实践表明，相较于传统手工编写电路，Noir 在典型场景下能显著减少约束数量，并缩短验证时间。这种“一键优化”能力使开发者无需具备电路优化的专门知识，也能开发出性能优良的零知识应用，尤其适用于高频交易与大规模验证等性能敏感的场景。

零知识电路一旦存在逻辑漏洞，可能导致严重后果（如无效交易通过验证）。为此，Noir 内置了测试框架，支持以 #[test] 注解编写单元测试，开发者可像普通程序测试那样验证电路逻辑。例如：

#[test]

fn test\_transfer\_amount() {

let invalid\_amount = -100;

assert(!is\_valid\_amount(invalid\_amount));

let valid\_amount = 500;

assert(is\_valid\_amount(valid\_amount));

}

测试框架可便捷地覆盖边界值与异常输入场景（如金额为 0、签名不足、签名无效），从而降低逻辑漏洞的概率。与传统电路开发需要额外编写测试工具不同，Noir 的开发—测试—编译一体化流程显著提升了电路的安全性与工程效率。

Noir 的技术特点可以概括为三点。一是工程化抽象，使电路编写接近普通编程，降低开发门槛；二是与 Aztec 的原生适配，为账户抽象与隐私账户设计提供语言层支持；三是优化与测试工具链，保证了电路的性能与安全性。因此，Noir 不仅是 ZK 技术在语言层的创新，也推动了零知识证明从“实验室密码学”向“规模化应用工具”转变。当开发者能够在不深入底层密码学的情况下构建隐私应用时，零知识证明才可能真正大规模融入 DeFi、身份认证、跨境支付等场景，成为区块链隐私保护的基础设施。表 12-4 展示了 Noir 与 Circom、ZoKrates 的典型对比。

表 12-4 Noir 与 Circom、ZoKrates 的比较

|  |  |  |  |
| --- | --- | --- | --- |
| 特征 | Circom | ZoKrates | Noir |
| 语法复杂度 | 电路级语法 | 类 Solidity 语法 | 类 Rust 语法，抽象更高级 |
| 编译优化 | 基础优化 | 基础优化 | 电路级深度优化 |
| 账户抽象支持 | 不支持 | 不支持 | 原生支持（适配 Aztec 架构） |
| 开发工具链 | 基础工具 | 基础工具 | 集成编译、调试、测试全套工具 |

Noir 的出现标志着零知识开发语言正从底层电路建模走向更高层的可读性与工程化实践，其对隐私保护协议的未来演进具有深远意义。

#### （5）Aztec 与 Tornado、Zcash 的技术与治理对比

Aztec 在技术路径与治理结构上展现出与 Tornado Cash、Zcash 明显不同的战略选择，如表 12-5所示。

Tornado Cash 基于单次匿名提款机制，强调强对抗监管的立场；Zcash 作为原生隐私币，通过 Shielded Pool 提供匿名支付，但合约支持有限；而 Aztec 则依托 zkRollup 实现批量隐私交易和可编程隐私逻辑，更适合复杂的链上应用。

在治理模式上，Aztec 由 Aztec Labs 主导，正探索逐步去中心化路径，并研究账户抽象与可选合规机制，以在隐私保护与监管要求之间取得平衡，相比 Tornado Cash 的对抗式立场和 Zcash 的中间路径，体现出更灵活的适配性。

表12-5 Aztec 与 Tornado Cash、Zcash 在技术路径和治理结构上的区别

|  |  |  |  |
| --- | --- | --- | --- |
| **特征** | Tornado Cash | Zcash | Aztec Protocol |
| 技术基础 | zkSNARK + Merkle Tree | zkSNARK + Shielded Pool | zkSNARK + zkRollup |
| 交互方式 | 单次匿名存取 | 原生隐私币 | 隐私交易聚合 + 可编程隐私逻辑 |
| 合约支持 | 不支持 | 不支持 | 支持账户抽象与隐私合约（Noir 语言） |
| 扩展性 | 依赖单笔交易 | 原生匿名支付 | 批量隐私交易 + zkRollup 优化 |
| 治理模式 | 无链上治理 | 基金会主导 + 社区参与 | Aztec Labs 主导 + 逐步去中心化 |
| 合规态度 | 强烈对抗监管 | 积极与监管对话 | 探索账户抽象与可选合规机制 |
| 技术路线 | 静态混币池 | 原生隐私币 | 动态隐私合约 + 高层 DSL 支持（Noir） |

Aztec Protocol 代表了零知识证明在链上隐私应用的新一代范式，从单一交易隐私扩展至全流程合约隐私；从静态混币池演进至可编程隐私逻辑；从底层约束式开发迈向高层语言（Noir）驱动的工程化开发。与 Tornado Cash 和 Zcash 相比，Aztec 在技术路径上更注重扩展性、灵活性与可编程性，为未来隐私友好型应用奠定了基础。

然而，Aztec 仍面临多重挑战。其 zkRollup 架构计算复杂度较高，对移动端支持不足；Noir 语言与工具链生态尚处早期；账户抽象与跨协议隐私交互的合规风险仍有待验证。随着隐私协议逐渐成为区块链基础设施的重要组成部分，Aztec 提供的“账户抽象 + 隐私合约”组合，可能成为未来“选择性隐私”与“可合规匿名”并存的重要技术路径。

## **12.3 匿名性的链上设计挑战与监管张力**

尽管ZKP技术为链上隐私保护提供了强有力的工具，但隐私的全面实现不仅是技术工程问题，更是治理结构与监管合规的复杂平衡。匿名交易、隐私币、隐私合约的兴起，正在挑战区块链世界一贯推崇的“透明性”原则，重新定义交易数据的可见性边界。

现实中的链上金融系统往往需要面对两个互相冲突的目标，一方面，用户期望实现资产、身份与交互路径的充分隐私；另一方面，监管机构对金融透明、AML、制裁合规（OFAC）的要求日益严苛。隐私保护的技术演进、系统设计选择与监管适配路径，已经成为区块链行业未来发展的核心分歧点之一。在这一背景下，链上匿名性的设计正面临四个技术维度与系统挑战，这就是如何隐藏交易金额？如何隐藏交易对手？如何隐藏账户身份？如何隐藏交互路径？

### 12.3.1 匿名性技术的设计维度与系统难题

链上匿名性从来不是一个单一维度的技术目标，而是一个由多个层次、多个技术路径构成的复杂系统。简单的地址混淆或交易金额隐藏，远不足以保障用户的链上隐私。真正完善的链上匿名设计，往往涉及金额、交易对手、账户身份以及交互路径的全面隐匿。

#### （1）金额隐藏

在比特币等传统区块链系统中，交易金额是明文存储的。这意味着攻击者可以通过分析交易金额，追踪用户的资金流向，甚至推断出用户的财务状况和交易意图。为了保护用户的隐私，需要一种机制能够在不泄露交易金额的情况下，确保交易的有效性和安全性。

CT（Confidential Transactions，机密交易）的核心思想是用承诺（Commitment）替代明文金额。承诺是一种加密技术，允许用户将某个值（如交易金额）隐藏在一个加密的“承诺”中。验证者可以在不知道具体数值的情况下，验证这个承诺是否有效。在 CT 中，交易金额被隐藏在一个承诺中，验证者可以通过数学方法验证交易输入与输出之和是否相等，而无需知道具体的金额数值。这样既保护了隐私，又确保了交易的合法性。

CT 主要采用的加密技术是 Pedersen 承诺，这是一种基于椭圆曲线加法的同态承诺。Pedersen 承诺具有加法同态性，用来在“看不到明文金额”的前提下做余额守恒校验，但它本身不是加密算法本身。 例如，如果C1是对金额 a 的承诺，C2 是对金额 b 的承诺，那么 C1 + C2 将是对金额 a + b 的承诺。在 CT 中，交易的输入和输出金额都被隐藏在 Pedersen 承诺中。验证者可以通过验证输入承诺的总和是否等于输出承诺的总和，来确保交易的合法性。即使验证者不知道具体的金额数值，也可以通过数学方法验证金额的平衡。

范围证明是 CT 的另一个关键应用，用于确保加密金额处于合法区间（如非负）。在金额被加密的情况下，验证者无法直接查看金额的具体数值，范围证明的作用是确保加密金额是非负的，并且没有超过某个合理的范围。这样可以防止用户创建负金额或进行溢出攻击。

CT 的设计理念已经被多个项目采用，以实现金额隐藏和隐私保护。Monero 是一个以隐私保护为核心的加密货币项目，它采用了类似 CT 的技术，通过隐藏交易金额和地址，实现了高度的匿名性。Monero 同时使用了 Bulletproofs 技术来实现高效的范围证明。Elements 是 Blockstream 的一个侧链项目，它支持包括 CT在内的多种隐私保护技术。通过 CT，Elements 项目可以在不泄露交易金额的情况下，确保交易的合法性。Liquid 是一个支持资产发行和隐私保护的区块链网络，它采用了 CT 技术来隐藏交易金额，同时通过范围证明确保金额的合法性。Liquid 网络允许用户在隐私保护的情况下进行资产交易。但Liquid/Elements 的 CT 默认仅隐藏金额（在 Liquid 上还隐藏资产类型），地址仍可见；若要隐藏收款人身份，需要配合一次性地址/隐身地址等机制，比如Monero 就通过 stealth address + ring signatures + RingCT 组合实现了以上功能。

CT的技术难点与系统挑战包括证明规模庞大、无法隐藏交易对手与身份，以及与 ZK 技术集成有限。CT 中的范围证明规模通常非常大，占用了大量链上空间。尽管 Bulletproofs 已显著缩小了证明规模，但仍无法满足 Rollup 等高频交易场景的极致压缩需求。CT 仅隐藏金额，地址信息仍然公开，容易被交易图谱分析攻击。此外，CT 基于加法同态加密，与 zkSNARK 等电路证明系统天然存在架构上的差异，难以高效兼容。

CT 所用的曲线与承诺体系可被 zk 电路表示，但把“CT 风格的本地验证”直接搬到链上并获得“SNARK 级别的极致压缩与聚合验证”的工程效率并不理想。与 rollup 的批量证明/聚合方案相比，Bulletproofs 的证明规模与验证成本在大规模吞吐下更为吃紧。因此“融合 zk-SNARK/zk-STARK 以统一在一个小规模证明里”是一个现实的优化方向。

尽管 CT 提供了金额隐藏的有效路径，但单独使用时隐私效果有限，必须与 Mixers、Stealth Address 等机制组合应用，才能提供完整匿名性。未来可能的发展在于将 CT 技术与 zk-SNARK、zk-STARK 证明体系深度融合，实现金额、地址、路径的全面隐匿。

#### （2）交易对手隐藏

zk-Mixer 是一种基于零知识证明（zk-SNARKs）的去中心化隐私协议，旨在彻底打断交易输入与输出的路径联系，其核心机制是利用 Merkle Tree 保存用户的存款承诺（commitments）。当用户将资金存入 zk-Mixer 智能合约时，系统会生成一个加密承诺并写入 Merkle 树，用户自己则保留与该承诺相关的秘密凭证（note）和用于防双花的标识符（nullifier）。在提取资金时，用户提交一份零知识证明，证明其 note 对应的承诺存在于 Merkle 树中且尚未花费，但并不暴露具体是哪一笔存款。合约在验证证明有效后，会将资金转移到用户指定的新地址。这样，外部观察者就无法将提款与具体的存款对应起来，从而打断了交易路径关联。

zk-Mixer 的隐私保护效果取决于匿名集（anonymity set）的大小。匿名集越大，攻击者越难以通过统计或链上分析去匿名化。为此，许多 zk-Mixer（如 Tornado Cash）采用固定面额池（如 1 ETH、10 ETH、100 ETH 等）来汇聚更多用户的存款，从而提升匿名性。与传统混币器相比，zk-Mixer 的优势在于不依赖参与者的实时协作，不存在中心化中介，且借助零知识证明确保了交易的正确性与资金归属。在最新的研究与迭代中，zk-Mixer 进一步引入多种优化路径，包括支持非固定面额存款、批量提款机制、递归证明以压缩验证开销，以及更高效的 Merkle Tree 构造以降低电路复杂度。这些改进旨在扩大匿名集规模、提升系统吞吐量和可扩展性。

然而，zk-Mixer 仍面临多重挑战。一是链上追踪残留，虽然 Mixer 内部交易关系被打断，但跨合约流转和与中心化交易所交互时，资金流向仍可能被链上分析部分关联。二是合规与监管压力，2022 年 Tornado Cash 被美国 OFAC 列入制裁名单，显示出 zk-Mixer 技术面临极高的监管风险。三是流动性瓶颈，如果匿名集过小，用户交易可能暴露在“去匿名攻击”风险之下。未来 zk-Mixer 设计，可能需要在链下身份验证、链上隐私增强与可选合规模块之间找到平衡，才能在监管夹缝中维持其生存与发展。

#### （3）账户身份隐藏

Stealth Address（隐身地址）用于增强接收方身份的隐私保护，其核心思想是通过一次性地址生成机制，使得每笔交易的接收地址都是唯一的，从而避免了接收方身份的直接暴露。在 Monero等典型实现中，接收方拥有两对密钥，View Key（查看密钥）用于扫描区块链，识别是否有交易属于自己；Spend Key（支出密钥）用于花费属于自己的资金。其典型交易流程如图12-7所示。

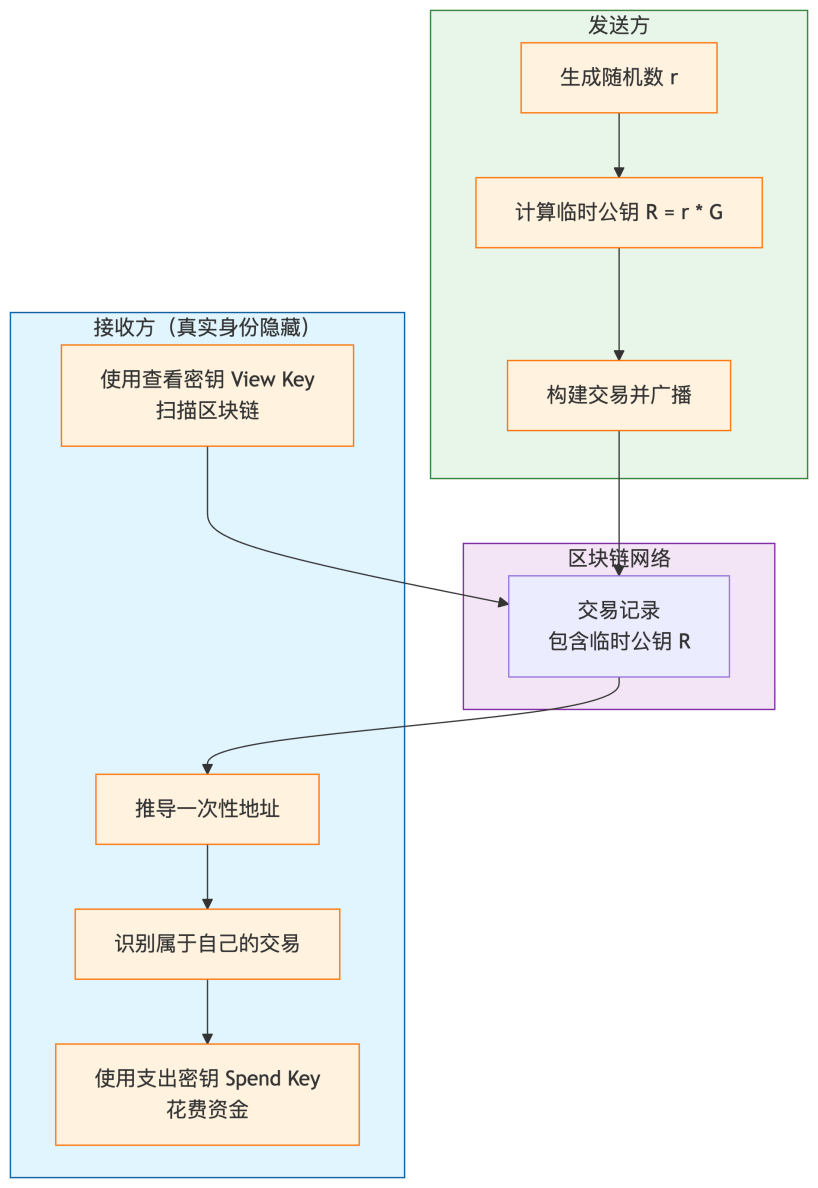


图12-7 Stealth Address（隐身地址）的典型交易流程示意图

图12-7右侧是发送方流程。发送方首先生成一个随机数 r 作为临时私钥，然后使用椭圆曲线乘法计算临时公钥 R = r \* G（G为椭圆曲线基点），将临时公钥 R 包含在交易中并广播到区块链网络。中部是区块链记录，交易被记录在公开账本上，其中包含临时公钥 R，所有交易信息公开可见，但接收方身份得到保护。

左侧是接收方流程。接收方使用自己的查看密钥 (View Key) 持续扫描区块链，根据临时公钥 R 和自身公钥推导出唯一的一次性地址，通过查看密钥验证哪些交易是发送给自己的。当需要时，使用支出密钥 (Spend Key) 对属于自己的一次性地址中的资金进行花费。通过这种机制，每笔交易都使用不同的一次性地址，使得外部观察者无法通过区块链上的地址关联到接收方的真实身份，有效增强了交易隐私性。只有拥有相应查看密钥的接收方才能识别属于自己的交易，而只有拥有支出密钥的接收方才能花费这些资金。

Monero采用 Stealth Address 作为隐私三大支柱之一，结合环签名（Ring Signatures）与 RingCT（机密交易）提供强匿名性。Elements / Liquid均支持 Stealth Address，用于增强资产交易的隐私性。但Stealth Address 的扫描过程计算量较大，对低算力设备并不友好。同时跨链 Stealth Address 尚无通用标准。

与去中心化身份（DID）和零知识证明结合的研究正在兴起，目标是实现“选择性披露”与“合规 KYC”，但目前也仍处于工程探索阶段。未来账户身份隐藏可能与账户抽象、隐私合约、DID 深度融合，形成更复杂的隐私身份架构。

#### （4）交互路径隐藏

链上匿名设计的最后一块拼图，往往被忽视。即便隐藏了金额、地址与账户身份，若用户的链上交互路径（包括函数调用顺序、合约访问历史）依然全程公开，仍可能通过行为模式识别泄露身份。因此，交互路径的隐私保护正成为隐私增强型区块链的重要发展方向。

zk-Rollup 在批量交易聚合的同时，提供了构建隐私执行环境的可能性。其核心思路是，交易在链下处理与排序，外部观察者无法直接看到单笔交易顺序；仅提交零知识证明来保证所有交易的正确性，而不公开每一笔交易细节；通过专门设计的证明电路，可以进一步隐藏交易路径与函数调用参数，从而避免行为模式泄露。需要强调的是，主流 zk-Rollup（如 zkSync、StarkNet）主要聚焦扩容与数据压缩，并不原生支持隐私。但通过扩展电路，可以演化出 Privacy Rollup 或私密合约系统，例如 Aztec 所探索的路径。

zkEVM 则试图将零知识证明与以太坊虚拟机指令集结合，使得更复杂的智能合约也能在隐私模式下执行。其目标是在保持与以太坊生态兼容的前提下，支持私密交易与合约调用，通过优化证明电路，实现高效的 zk 证明生成与验证。

在传统智能合约中，函数调用路径、参数和返回值完全公开，易于被链上分析利用。为此，研究者提出了 Shielded Function（私密函数调用）的概念，其核心机制包括加密合约状态，防止外部观察者推断内部逻辑与数据；隐藏函数参数与调用路径，避免从调用依赖图中泄露用户意图；合约执行的正确性通过零知识证明链上验证，而不暴露任何敏感细节。Aztec Connect 曾是这一方向的代表，通过加密合约状态与零知识证明实现私密 DeFi 交互。当前 Aztec 团队正在研发新一代 Aztec 3，探索“公私合约混合 Rollup”架构。zkSync Era 虽然是 zkRollup 的代表，但目前主要聚焦扩容，其隐私功能尚未落地。

但zk-Rollup 的批量证明计算成本高，影响了隐私 Rollup 的扩展性；隐私路径与合规审计的“可解释性”要求存在冲突；私密路径设计需要与账户抽象、电路设计深度耦合，工程门槛极高。未来可能通过“选择性路径披露”和“审计密钥”机制，在隐私与监管之间形成妥协。

综上，链上匿名性的设计并非单一维度的技术堆叠，而是涉及金额、交易对手、账户身份、交互路径四个核心隐私维度的系统性工程。每一层隐私防护都有明确的密码学技术路线，同时伴随着计算复杂度、工程瓶颈与监管张力。完整的链上匿名解决方案，需要统筹 zk-SNARK、zk-Rollup、账户抽象、DID、Shielded Function 等多项技术，并在合规框架内逐步推进。

### 12.3.2 匿名交易与全球监管冲突

随着DeFi与隐私保护技术的不断发展，匿名交易成为区块链生态的重要组成部分。然而，链上匿名性与全球监管体系的紧张关系亦愈加凸显。AML、KYC与OFAC等监管框架与链上匿名机制存在根本性的设计冲突。这场冲突不仅是技术问题，更是法律、治理与价值观的全面碰撞。

#### （1）AML、KYC、OFAC成为全球监管的链上适用难题

为了防范跨国洗钱、恐怖融资与非法交易，全球金融监管体系逐步建立了AML、KYC以及制裁名单（如 OFAC）等核心机制。然而，当这些监管工具试图迁移至去中心化、匿名化的区块链生态时，便遭遇了结构性挑战。

AML 法规旨在阻止非法资金进入金融系统，传统路径依赖银行、支付机构等中心化节点执行身份审查与交易监控。公链交易无需许可、地址生成无需身份验证，使得传统 AML 监控难以落地。匿名交易协议的设计目标（隐藏资金流向与账户关联性）与 AML 的核心诉求（追踪资金流动、防范非法融资）存在显著张力。但值得注意的是，一些项目（如 Zcash）已尝试引入“选择性披露”机制，以便在特定场景下满足合规需求。

KYC 是 AML 的技术前提，确保每笔交易可追溯至现实世界主体。然而，Tornado Cash、Zcash、Aztec 等链上匿名协议通常无需许可与身份验证，开放运行。一旦智能合约部署上链，难以下架或修改，且 DeFi 协议往往缺乏“运营主体”可供追责。这导致链上 KYC 机制的落地缺乏执行抓手。

美国财政部海外资产控制办公室（OFAC）于 2022 年首次将 Tornado Cash 智能合约地址列入制裁名单，开创了链上地址制裁的先河。其逻辑是 Tornado Cash 被大规模用于黑客洗钱与规避国际制裁。此举标志着监管机构开始将智能合约本身视为可制裁对象。但这一做法引发广泛争议，智能合约是否可以被认定为“主体”？开发者是否需要为协议的滥用负责？匿名协议是否存在“合法用途”，如隐私保护薪酬支付、公益捐赠？目前，围绕 Tornado Cash 的法律挑战在美国与欧洲持续进行，尚未形成最终定论。这表明，在匿名性与合规性问题上，监管与技术社区仍处于博弈与探索阶段。

#### （2）Tornado Cash 制裁事件回顾与监管逻辑解析

在全球 AML 与 KYC 合规路径日益关注链上交易的背景下，Tornado Cash 案件成为监管对去中心化匿名协议采取强力措施的标志性事件。2022 年 8 月 8 日，OFAC 将 Tornado Cash 纳入制裁清单，理由包括其被 DPRK 关联的 Lazarus 组织广泛用于处理黑客所得资金与规避制裁。该举动被广泛视为首次将智能合约本身纳入 SDN（Specially Designated Nationals ，特别指定国民）清单的先例，迅速引发了关于链上隐私、技术中立与监管边界的全球性讨论。

OFAC 的执行逻辑有两条主线。一是功能性制裁，即便协议去中心化，但若其主要功能与实际用途被大量用于洗钱与规避制裁，仍可被指定为制裁对象；二是地址/实体并行路径，在传统“冻结资产”难以适配智能合约的情况下，OFAC 通过将合约地址与 Tornado Cash 实体一并纳入清单、发布与更新 FAQ 的方式，从交互入口层面实施限制。

制裁发布后，链上出现针对名人地址的小额“dusting”转账[[2]](#footnote-1)，以凸显制裁在技术执行层面的局限。对此，OFAC 随后明确不会因非自愿接收受制裁来源的微额资金而追责，并为制裁前存款提供特许许可申请通道，以合规方式取回资金。

法律、技术与社会层面的争议持续发酵，争议集中在以下几个主题上。其一，智能合约是否构成可制裁的“财产/主体”，财政部是否越权？其二，区块链能否对地址进行有效隔离与访问禁止？其三，隐私技术的合理用途是否应被保留？2022 年 11 月，OFAC 对 Tornado Cash“撤销并重新指定”以补强法源与事实，其后在美国的多起诉讼与上诉中，法院对财政部权限边界提出了不同程度的限制与审查。 在个人责任层面，Tornado Cash 开发者 Alexey Pertsev 于 2022年8月10日在荷兰被捕，2024年5月14日被荷兰法院以洗钱罪判处 64 个月有期徒刑，案件对“开源开发自由 vs. 合规责任”的讨论产生重大影响。

### 12.3.3 可选择合规路径与未来设计方向

在 Tornado Cash 事件之后，链上隐私协议的发展陷入了新的十字路口。一方面，链上用户的隐私需求依然强烈，ZKP、账户混淆与链上匿名交易的优化技术也在持续推进；另一方面，全球监管趋势表明，对于完全匿名且不可追踪的交易结构将采取更为强硬的打击手段。面对这种结构性冲突，越来越多开发者开始探索“可选择合规”（compliance-optional）与“选择性披露”（selective disclosure）的解决方案，希望在保护用户隐私的同时，为监管留下合理的审查与取证路径。这一设计方向并非简单地在隐私与合规之间折中，而是试图通过密码学创新、协议设计与治理架构的协同，打造出灵活、安全、可持续的隐私金融基础设施。

#### （1）View Key / Audit Key 机制详解

在可选择合规（compliance-optional）与选择性披露（selective disclosure）的隐私金融架构中，View Key（Viewing Key）是最具代表性的技术工具之一。

View Key 机制最早由 Zcash 提出，用于其 shielded 地址（z-address）交易的有条件审计。Zcash 使用 zk-SNARK 对交易金额、发送方和接收方加密，从而实现链上高度匿名性。在功能原理上，用户可以为 z-address 生成一个 View Key（官方称为 viewing key），允许第三方仅以只读方式访问该地址的交易信息，而无法动用资金或操作合约，其核心特征包括只读性、可授权性和非全局可见性。只读性（Read-only）是指持有 View Key 的第三方只能读取交易明细，不能执行转账或改变合约状态；可授权性（User-controlled）是指地址所有者可自主决定向哪些对象披露 View Key，实现有条件的交易透明；非全局可见性（Non-global visibility）是指仅持有 View Key 的实体能够解密交易数据，保护隐私不被全网公开。这一机制实现了“有条件可审计”的隐私保护路径，既满足用户对匿名性的需求，也为监管或审计提供技术接口。

Audit Key（审计密钥）是一种概念性扩展，用于在协议层提供统一、可控的审计接口，目前尚未成为标准化功能。其功能原理是在特定隐私协议或实验性设计中，Audit Key 可由 DAO 或多签/MPC 分布式治理机构持有，用于在满足特定条件（如涉嫌非法交易或大额异常交易）时触发链上交易溯源或披露操作。其设计上的挑战包括，谁有权持有和启用 Audit Key？在何种情况下可启动审计？如何防止审计权限被滥用或引发对用户隐私的过度侵犯？Audit Key目前主要存在于实验性隐私协议或研究性设计中（如部分 ZKP 或 Aztec 隐私合约探索），尚未在主流隐私币或标准协议中广泛部署。

View Key 是链上匿名交易可选择合规路径的核心工具，已在 Zcash 等项目实际应用。Audit Key 则属于协议层概念性设计，旨在提供条件性链上审计，但仍处于实验阶段。二者结合，可以在保持用户隐私的同时，为监管提供有限的可审计接口，是链上隐私与合规技术融合的重要方向。

#### （2）可选择披露 vs 全链隐私的平衡设计

要理解可选择披露（selective disclosure）设计的重要性，必须首先认识全链隐私架构（full-chain privacy）的理论优势和现实局限。完全匿名的区块链协议，如 Monero 和早期 Tornado Cash，能够提供高度隐私保护。Monero 通过环签名（Ring Signatures）、隐匿地址（Stealth Address）和机密交易（RingCT）隐藏交易双方账户和金额；Tornado Cash 则利用 zk-SNARK 技术在智能合约池中混淆交易，实现交易金额和地址的匿名化。全链隐私的核心优势在于，交易不可被链上追踪，账户身份高度匿名，交易金额和交互路径全方位隐藏。然而，这类架构同样存在明显的监管风险，匿名交易可能被用于洗钱、诈骗、规避国际制裁等非法活动。

针对上述问题，部分新兴隐私协议提出了“可选择披露”的机制，即在默认情况下交易信息保持全链加密，而用户可自主授权对特定交易记录的查看。技术实现方式包括 View Key（可视密钥）、审计合约（Audit Contract）或监管桥接（regulatory bridge）等，使用户在享有隐私的同时，为必要的合规审查和法律取证预留路径。例如，Aztec Protocol 的账户抽象（account abstraction）与隐私合约（private contract）模型支持通过 Noir 编程语言定义条件性披露逻辑，如当交易金额超过预设阈值时自动触发披露授权请求。这一设计在理论上提供了一条在保护用户隐私与满足监管要求之间的技术折衷路径，为隐私金融的可持续发展提供了可能性。

#### （3）zk-SNARK 合规电路模板的可行性

从用户层面的可选择披露延伸至协议层，还可以在 zk-SNARK 电路设计中嵌入合规验证逻辑。此类电路可用于生成“交易符合规则”的零知识证明，例如交易金额超过设定阈值时生成证明以表明已获得 View Key 授权；与黑名单地址交互时生成证明表明交易符合合规要求；特定交易类型（如跨境大额转账）需包含合规证明输入。通过 zk-SNARK 验证，外部验证者无需知晓交易具体细节，即可确认交易满足预设合规约束。

合规电路设计面临若干挑战。一是合规标准具有地域性，难以实现全球统一；二是电路更新可能涉及系统性升级，这也增加了协议的复杂度；三是用户对合规路径的信任依赖于治理机制的去中心化和透明性。未来的解决方案可能包括模块化电路设计、链上治理的透明审计，以及智能合约化的合规规则嵌入。

#### （4）可验证匿名身份（ZK-ID）机制设计

ZK-ID（Zero-Knowledge Identity）是一类基于零知识证明的链上身份验证技术，其目标是在不暴露真实身份的前提下，实现身份合规验证。用户通过传统 KYC 完成身份认证后，可获得匿名凭证，在链上交易或交互时使用零知识证明生成证明，表明“用户属于已认证群体”，而无需公开任何个人信息。

该机制的优势包括用户链上身份与现实身份解耦，提高隐私保护；满足监管身份验证要求；支持去中心化协议开放合规门槛。当前典型项目包括 Polygon ID、ZKPass 和 Semaphore，它们在匿名 KYC、可验证凭证和零知识群体认证方面提供了工程实现。

面临的主要挑战，一是KYC 发行机构可能受到监管要求强制披露数据；二是凭证伪造与重复使用问题，这需要结合零知识证明及链上防重放机制缓解；三是跨链和多协议 ZK-ID 的互操作性，需要标准化证明格式与验证逻辑以支持多链环境。

#### （5）模块化隐私层的工程路径

随着隐私协议设计的持续深入，业界逐渐意识到模块化、可插拔的隐私架构可能成为未来主流。未来隐私协议可能逐步走向模块化架构，即核心交易协议与隐私功能解耦，不同隐私层（如金额隐藏、地址隐藏、路径隐藏）可按需接入，支持不同链、不同协议间的隐私标准互操作。

模块化隐私层的技术优势，一是降低隐私功能的开发与部署门槛，二是支持用户自主选择隐私强度，三是便于快速适配不同合规要求。已有项目如 zkRouter、Aztec、Polygon Miden 正在探索这种可插拔式隐私层设计。

多链时代，隐私层的互通成为一大工程难题。关键问题包括不同链的地址格式、哈希算法与加密曲线如何兼容；跨链桥中的隐私数据如何安全转移；多链合规路径的统一标准如何制定。未来可能出现的趋势是跨链隐私网关（zkBridge）与链下数据索引节点（zkIndexer）协同，实现链上链下、单链跨链的统一隐私层。

未来，链上隐私保护将不再是单一维度的技术对抗，而是成为一个涵盖密码学设计、合约工程、链上治理、全球监管协同的综合性课题。可选择合规路径的广泛应用，有望推动隐私金融从灰色地带走向合规主流，成为支持链上经济持续发展的重要基石。

## 12.4 零知识证明在链下计算与 Rollup 中的应用

在前文的讨论中，我们主要聚焦于零知识证明技术在链上隐私保护与合规机制中的应用。然而，零知识证明的潜力远不止隐私领域，它也在区块链扩容路径中扮演着极为关键的角色。随着以太坊等公链的使用规模持续扩大，链上计算与存储负载面临严峻挑战，如何在不牺牲去中心化和安全性的前提下，显著提升交易吞吐量，成为整个行业的核心问题。ZK-Rollup 作为最具前景的扩容技术之一，正是在这一背景下崭露头角。

### 12.4.1 ZK-Rollup链上扩容的结构与工作原理

ZK-Rollup（Zero-Knowledge Rollup）是一种将零知识证明技术应用于区块链扩容的重要路径。通过将大量交易在链下批量处理，并生成简洁的有效性证明提交至链上，ZK-Rollup在显著提升系统吞吐量的同时，依然保持极高的安全性和数据完整性。

#### （1）ZK-Rollup的整体架构与状态更新流程

ZK-Rollup的核心设计思路，是将链下批量处理（off-chain batch processing）与链上有效性证明（on-chain validity proof）有机结合，形成高效且安全的数据处理闭环。其基本架构和关键组成部分包括，一是链下交易批处理器（Sequencer），负责收集、排序用户提交的交易，并在链下执行状态更新；二是状态根哈希（Merkle Root），所有账户状态通过Merkle树结构组织，每次交易批次会生成新的状态根哈希，用于链上验证；三是零知识有效性证明（Validity Proof），对每个交易批次生成简洁的zk-SNARK或zk-STARK证明，用以证明状态转换过程的正确性；四是链上验证合约（Verifier Contract），智能合约负责校验提交的零知识证明，并在证明有效时，更新链上记录的状态根哈希。ZK-Rollup的交易处理与状态更新流程通常如图12-8所示。

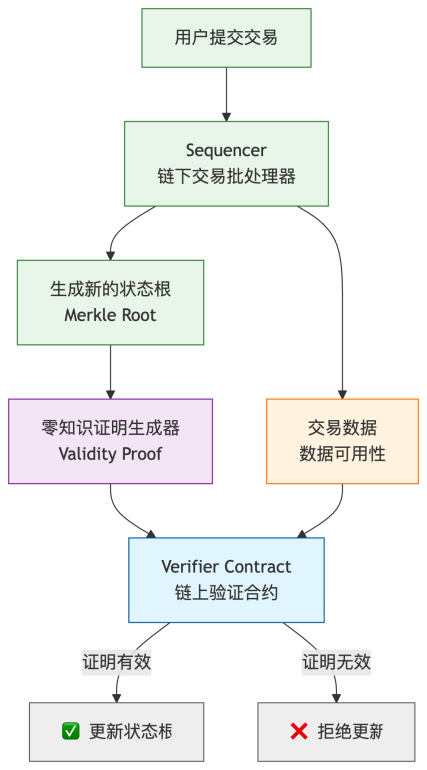


图12-8 ZK-Rollup 交易处理与状态更新流程

#### （2）有效性证明的链上校验机制

ZK-Rollup 的安全核心在于有效性证明的链上验证，确保链下执行的全部交易遵循协议规则，且状态更新不会被篡改。链下批处理器在每轮交易处理结束后，生成一份 zk-SNARK 或 zk-STARK 证明，证明输入状态根哈希正确，提交的交易批次有效且顺序正确，输出状态根哈希与链下计算结果一致等断言。该证明与状态摘要一起被提交至主链。

主链上的验证合约执行以下步骤。首先接收并解析证明与状态更新数据，然后使用零知识证明验证器进行快速检验。如果证明通过，更新链上状态根哈希；如果证明失败，拒绝交易批次。由于 zk-SNARK 证明规模通常非常小（不到几百字节），链上验证所需 Gas 远低于处理全部交易，能够显著降低主链负载。ZK-Rollup 与 Optimistic Rollup 的核心对比见表12-6。

表12-6 ZK-Rollup 与 Optimistic Rollup 的核心对比

|  |  |  |
| --- | --- | --- |
| 维度 | ZK-Rollup | Optimistic Rollup |
| 验证机制 | 有效性证明（Validity Proof） | 挑战机制（Fraud Proof） |
| 提现等待时间 | 极短（几分钟） | 通常 7 天争议期 |
| 链上数据提交 | 状态根 + 证明 | 完整交易数据 |
| 证明体积 | 极小（几十到几百字节） | 不适用 |
| 链上验证复杂度 | 极低 | 不适用 |
| 用户体验 | 高 | 中等 |
| 安全假设 | 无需信任第三方 | 假设有诚实挑战者在线 |

ZK-Rollup 依赖数学证明确保状态转换正确，因此不需要争议期，用户可以快速提取资产。而 Optimistic Rollup 采用乐观假设，链上默认交易有效，若有错误则依赖挑战者提出欺诈证明，需等待争议期结束才能确认。虽然 ZK-Rollup 在用户体验和安全性上占优，但其证明生成计算量大，适合小型、规则清晰的交易批次（如支付、转账等）。Optimistic Rollup 则在支持复杂智能合约方面更成熟，且开发难度相对较低。

#### （3）ZK-Rollup 的优势、局限与发展前景

ZK-Rollup 的优势在于，一是高安全性，无需信任挑战者或第三方，依赖零知识数学证明保证交易正确性；二是快速最终性，无挑战期，提现等待时间极短；三是扩容效果显著，链上仅验证简洁证明，显著降低存储与计算负担；四是高度兼容性，支持不同类型的主链与侧链结构。

其局限则有证明生成计算复杂，目前生成 zk-SNARK 仍需大量计算资源，可能导致批处理延迟；开发难度高，涉及复杂的电路设计、密码学实现与证明优化；存在兼容性问题，通用 zk-EVM 的开发仍处于早期阶段，完全兼容现有 EVM 合约存在技术挑战。

从发展前景来看，随着硬件加速（如 GPU、FPGA）、证明电路优化（如 Plonk、Halo 2）以及 zk-EVM 的持续发展，ZK-Rollup 有望成为下一代区块链扩容的主流路线。未来的发展可能包括通用 zk-EVM 的完善，实现与现有以太坊生态的无缝兼容；更高效率的证明生成算法，支持高频、高复杂度交易；与链下执行环境（如 zk-Coprocessor）的深度结合，扩展链下计算可信边界。

ZK-Rollup 通过零知识有效性证明实现了链上扩容、安全性和隐私保护的高度结合，成为目前最被看好的 Layer 2 技术之一。相比 Optimistic Rollup，ZK-Rollup 提供了更优的用户体验与安全保证，但也面临更高的技术复杂度与计算门槛。随着证明系统、zk-EVM 和链下计算的持续演进，ZK-Rollup 有望在未来区块链基础设施中占据核心地位。

### 12.4.2 ZK-EVM的兼容性与工程挑战

零知识以太坊虚拟机（zk-EVM）是将零知识证明技术与以太坊虚拟机（EVM）深度融合的扩容方案，其目标是在无需修改现有智能合约的前提下，实现对以太坊智能合约的高效零知识证明生成与验证，从而原生支持 ZK-Rollup 的扩展路径。

#### （1）主流 zk-EVM 项目的技术路线

当前主流的 zk-EVM 项目可大致分为兼容优先和性能优先两类不同的发展路径，不同的设计取向决定了它们在实际应用中的适配难度与扩展性空间。

zkSync Era是由 Matter Labs 开发的 zk-Rollup 扩展方案，其核心设计目标是在提高吞吐量的同时兼容 EVM 智能合约。zkSync Era 在早期版本中采用了 zkPorter 模块，通过链下数据可用性（Data Availability）实现极高 TPS，后续版本逐步支持更广泛的智能合约兼容。zkSync Era 的关键设计包括通过基于 Solidity 的 zkEVM 电路实现对EVM 的兼容，允许现有合约迁移，提供更简洁的证明生成路径以降低验证成本，在账户抽象、原生跨链桥等功能上有所创新，提升整体开发体验。zkSync Era 于 2023 年 3 月上线主网，现已支持 Solidity 0.8.x 版本，兼容 EVM 操作码，并计划扩展对 Rust 和 C++ 等高阶语言的支持。其 LLVM 编译器将 Solidity、Vyper 和 Yul 等代码编译为可执行代码，提升了开发者体验。

Polygon zkEVM专注于完全兼容 EVM 字节码，试图做到开发者无需修改任何智能合约即可直接迁移到 Layer 2。Polygon zkEVM 的技术特点有支持现有以太坊开发工具链（如 Remix、Hardhat），优先实现 EVM 字节码电路的 zk 证明，兼容 Solidity、Vyper 编译后的合约。其性能瓶颈主要集中在证明生成时间，但大幅降低了迁移门槛。Polygon zkEVM 于 2023 年 3 月上线主网，支持现有以太坊开发工具链，兼容 Solidity、Vyper 编译后的合约。

Scroll强调与以太坊节点行为完全一致，其架构设计旨在与以太坊节点执行环境保持高保真同步。Scroll 的设计要点包括完整仿真以太坊客户端，确保链上状态与以太坊主网保持完全一致；兼顾开发者体验与安全性，适合生态原生无缝衔接；与以太坊基金会紧密合作，提升兼容性标准。

#### （2）ZK-EVM 兼容性等级

以太坊基金会对 ZK-EVM 的兼容性划分提出了 Type 1–Type 4 四个等级标准，不同类型反映了对 EVM 兼容的深度与实现路径的复杂性，具体如表12-7所示。需要说明的是，Type 1–4 的划分并非互斥，部分项目在不同阶段可能跨越类型边界，如 zkSync 就经历了由 Lite 到 Era 的演进。

表12-7 ZK-EVM兼容性等级

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 维度 | Type 1 | Type 2 | Type 3 | Type 4 |
| 兼容性 | 字节码级完全兼容 | 字节码部分兼容+优化 | Solidity语言兼容 | 全新语言不兼容EVM |
| 优势 | 工具链完全支持 | 兼顾性能与兼容 | 无需修改源码 | 极致性能 |
| 工具链 | 零适配 | 需部分适配 | 需重新编译部署 | 全新工具链 |
| 挑战 | 证明生成复杂 | 平衡难度大 | 迁移成本较高 | 生态隔离 |
| 代表 | Scroll, Polygon zkEVM | zkSync Era | 早期zkEVM | StarkNet, zkSync Lite |

Type 1是与以太坊主网完全字节码兼容，执行结果完全一致，其特点是支持所有现有开发工具，最小迁移成本，代表项目包括Scroll和Polygon zkEVM，其挑战在于证明生成极为复杂，当前性能受限。

Type 2兼容字节码执行，但允许对某些复杂指令的执行逻辑进行优化。其特点是兼顾兼容性与性能，可能需要部分工具链适配，典型项目是zkSync Era（部分阶段），其挑战是维护兼容性同时提升效率，平衡难度大。

Type 3支持 Solidity 等高级语言，但不兼容底层字节码，其特点是开发者无需修改源码，但需要重新编译部署。早期 zkEVM 的实现路径是这个类型的典型项目，其挑战是现有工具链无法完全兼容，用户迁移成本较高。

Type 4不兼容 EVM，需要设计全新的 zk 扩展语言与环境，其特点是追求极致性能，典型项目如 StarkNet、zkSync Lite，其挑战是生态隔离严重，难以快速吸引现有开发者。

这种兼容性等级划分，为开发者和项目方提供了明确的技术路线选择依据。

#### （3）CIRCUIT 级与 Bytecode 级兼容的工程难题

ZK-EVM 的最大挑战之一，是如何在零知识证明电路（CIRCUIT）中有效支持 EVM 的复杂执行逻辑。电路级兼容需要对 EVM 指令集的所有操作编写证明电路，包含存储操作、内存操作、跳转逻辑等复杂结构。而且在处理动态数据结构（如 CALLDATA、动态数组）时，电路设计体积极其庞大，证明生成时间大幅增加，需要大量硬件并行优化。

Bytecode 直接兼容 EVM 字节码，但要求证明电路对每一条 EVM 操作码（Opcode）进行映射，包括 gas 消耗、异常处理等所有细节均需与以太坊主网节点保持一致，任何微小偏差都会导致链上状态不一致，影响验证正确性。

一些项目通过构建专门的 zk-EVM 编译器，将高层语言优化为更适合电路生成的中间表示（IR），通过并行证明、批量验证等手段降低证明开销。

#### （4）zkEVM 的性能现状、开发体验与未来趋势

在全面了解了各类 zkEVM 项目的技术路径、兼容性等级以及工程实现中的核心难点之后，有必要进一步考察 zkEVM 当前在实际应用中的表现情况。性能瓶颈、开发者生态的成熟度以及未来潜在的优化方向，构成了 zkEVM 能否被广泛采用的重要基础。

从性能现状来讲，zkEVM 证明生成时间仍在分钟级别，短期内延迟较高。但由于无需挑战期，其资金提现速度通常快于 Optimistic Rollup。批量交易可以大幅降低单位交易的证明成本，适合高频批处理场景。验证开销相对较低，适合链上快速验证。

在开发者体验方面，Type 1与 Type 2 zkEVM 项目已经基本兼容现有 Solidity、Vyper 开发环境，部分项目支持 Remix、Hardhat、Truffle 等工具链原生接入。调试与错误追踪尚不成熟，zkEVM 原生的开发者支持生态仍在建设中。zkEVM 调试工具和开发者文档仍处于早期建设阶段，尚不如 Optimistic Rollup 成熟。

从未来趋势来看，递归证明（如 Halo 2）的应用将大幅降低单次证明成本，推动实时性优化。zkEVM 专用编译器与电路设计将趋于模块化，支持更复杂的合约调用与状态管理。预计未来 2–3 年，Type 1 兼容的 zkEVM 项目将逐步进入主流应用，成为 Layer 2 扩展的重要组成部分。

ZK-EVM 作为零知识证明与以太坊生态融合的关键技术路径，正在成为下一代区块链扩展方案的技术核心。虽然当前 zkEVM 在性能、证明生成时间、开发者工具支持等方面尚存在显著挑战，但其在兼容性、去信任扩展与隐私保护方面展现出的巨大潜力，预示着 zkEVM 极有可能成为公链未来竞争格局中的重要基石。未来，随着电路优化、递归证明、批量验证等技术的持续成熟，zkEVM 有望实现高性能、高兼容性与良好开发体验的平衡，推动去中心化应用迈入新一轮扩展与隐私保护并进的发展阶段。

### 12.4.3 ZK 在跨链与预言机中的应用拓展

随着区块链生态的多链化与模块化发展，链与链之间的状态互操作与跨链通信成为制约系统扩展性与安全性的核心问题。传统跨链桥大多依赖多签节点或中介验证者进行跨链消息确认，其信任模型存在显著脆弱性。一旦这些验证节点遭到攻击或协同作恶，用户资产可能面临不可逆转的损失。历史上，Ronin Bridge（2022）、Wormhole Bridge（2022）及 Harmony Horizon（2022）等跨链桥均因安全漏洞而导致数亿美元级别的资产损失，充分揭示了此类架构的系统性风险。

#### （1）zkBridge无需信任的跨链验证新架构

为克服上述局限，zkBridge（零知识跨链桥）引入ZKP技术，通过在源链上生成目标链状态的有效性证明（validity proof），并在目标链链上直接验证该证明，从而在无需信任外部中继者的情况下实现跨链数据传递。zkBridge 的典型设计如 Herodotus，提出了基于 zk-SNARK 的存储证明（zkStorage Proof），能够在目标链上验证源链账户、合约存储等状态信息，而无需依赖中心化跨链服务。

zkBridge 的主要优势体现在以下几个方面。一是无需信任第三方，跨链验证完全依赖链上验证 zk-proof，实现全流程去中心化；二是安全性得到了提升，避免了传统桥接架构中的多签节点被劫持或篡改的风险；三是通用数据跨链能力增强，不仅支持资产转移，还能实现账户状态、合约存储、治理结果等复杂数据的跨链同步。

然而，zkBridge 仍然面临工程与性能挑战。一是证明生成开销大，源链完整状态或历史数据的 zk 证明生成通常需要大量计算资源，且延迟在秒级至分钟级，难以支撑高频实时跨链场景；二是源链依赖性强，EVM 系列链由于具备 Merkle Patricia Trie 状态树结构，较易生成 zkStorage Proof；而 Cosmos、Solana 等非 EVM 链缺乏统一的状态导出机制，增加了 zkBridge 的适用难度；三是开发生态尚不成熟，尽管已有初步原型实现，但通用化、标准化工具链仍处于早期阶段。

根据现有文献与产业实践，zkBridge 技术目前处于 TRL 4–5（实验室验证到原型验证阶段）。虽然 Herodotus、Succinct Labs 等团队在以太坊与 Layer 2 之间完成了跨链状态验证实验，但距离大规模商用尚需克服证明生成效率、跨链延迟与开发工具链标准化等瓶颈。

综上，zkBridge 为解决跨链通信的安全与信任难题提供了一条零信任化、加密安全的全新路径。随着递归证明、批量验证与跨链电路优化技术的成熟，zkBridge 有望在未来取代依赖中介验证者的传统跨链桥，成为多链互操作的重要基础设施。

#### （2）zk-Oracles成为链外数据的零知识验证路径

预言机是区块链系统获取链外数据的核心基础设施，其去中心化程度与数据可靠性直接决定了DeFi、RWA以及链上衍生品等应用的金融安全。然而，当前主流预言机（如 Chainlink、Pyth、Band Protocol）大多依赖数据源的可信性与节点报告机制，这意味着链上系统仍需对外部节点维持一定的信任假设，难以彻底消除潜在的作恶与操纵风险。

为降低信任开销，zk-Oracles（零知识预言机）提出利用 ZKP，将链下数据采集与验证过程转换为可在链上高效验证的 zk-proof。其典型工作流程为，数据提供方在链下本地生成 zk-proof，用以证明“数据 X 来源于 Y，且满足 Z 条件”，而无需暴露底层采集细节；链上预言机合约仅验证该 zk-proof 的有效性，而不依赖数据上传者的信誉；数据消费者（如 DeFi 协议）根据链上验证结果执行合约逻辑，从而在无需信任外部节点的前提下完成数据接入。

在具体实现方面，Supra Oracles 已提出结合可验证随机函数（VRF）与 ZKP 的设计，使链上能够验证随机性与数据正确性；Herodotus 则探索了基于 zk-SNARK 的链上存储证明，用于验证链外数据与链上状态间的可信映射。这些方案展现了 zk-Oracles 在去信任化数据传输中的潜在价值。

与传统预言机相比，zk-Oracles 具备以下显著优势。首先是去信任，预言机节点无需完全可信，链上可直接验证 zk-proof；其次是隐私保护，zk 证明可隐藏数据采集路径或部分细节，减少数据源暴露风险；最后还有抗操纵性，zk-Oracles 可以显著降低节点串谋、恶意延迟、价格操纵等风险。

然而，zk-Oracles 仍存在一些技术挑战。首先是证明生成与验证的性能开销大，当前 zk-proof 生成需要秒级甚至分钟级延迟，仅适合中低频数据同步；其次是动态数据难以适配，例如高频金融市场价格的零知识证明生成难度极大，限制了场景应用；最后还有标准化生态不足，目前 zk-Oracles 工具链尚处于实验性阶段，难以广泛部署。

根据产业落地情况，zk-Oracles 技术目前处于 TRL 4–5（实验室到原型验证阶段）。已有 Supra、Herodotus、Succinct Labs 等团队展示了 zk 证明驱动的预言机原型，但距离支持大规模、高频链外数据输入仍存在显著的性能与标准化瓶颈。

综上，zk-Oracles 为链上引入高可信数据提供了一种“无需信任第三方”的创新路径，尤其在金融安全与跨链通信场景中展现出广阔应用潜力。随着 zk-SNARK、zk-STARK 以及递归证明等技术的成熟，zk-Oracles 有望成为 DeFi 与 Web 3 基础设施的重要组成部分。

#### （3）zk-Oracles 与传统预言机的对比分析

zk-Oracles 与主流预言机机制对比见表12-8。从表12-8可以看出，zk-Oracles 在信任模型与隐私保护方面具有显著优势，适合对安全性要求极高的链上应用，如高价值 RWA 资产映射、链上法律合约验证、跨链治理同步等场景；但在性能方面，现阶段 zk-Oracles 的计算复杂度与时延仍限制了其在部分高速数据场景（如 AMM 价格更新）中的应用。

表12-8 zk-Oracles 与主流预言机机制对比

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目 | 数据来源验证 | 信任模型 | 性能 | 隐私保护 |
| Chainlink | 多节点价格聚合 | 需部分信任节点 | 中等 | 无 |
| Pyth | 授权节点直连数据源 | 需信任部分授权节点 | 高速 | 无 |
| Band Protocol | 多节点投票确认 | 需信任节点网络 | 中等 | 无 |
| zk-Oracles | 零知识证明链上验证 | 极低信任需求 | 低—中 | 支持 |

值得注意的是，传统预言机与 zk-Oracles 并非完全对立，未来预言机网络很可能形成混合型架构，高频数据仍依赖传统节点快速传输，核心链外状态或重要价格点则通过 zk-Oracles 提供可验证的安全保障。

#### （4）ZK 驱动的数据基础设施未来展望

zkBridge 与 zk-Oracles 的发展，预示着未来区块链跨链通信与数据同步机制将从当前的“部分去中心化”逐步走向“极低信任化”发展，这是 Web 3 数据基础设施的一次范式跃迁。zkBridge 有望成为链与链之间的原生安全层，跨链通信无需依赖第三方中继；zk-Oracles 将为链上合约提供更安全、可验证的数据输入接口，降低外部依赖。

未来值得重点关注的技术演进路径包括zk-proof 生成速度的优化，如更高效的证明电路设计、并行计算加速，zk-Rollup 与 zkBridge 的融合可以实现批量跨链状态同步，zk-Oracles 与 RWA 资产链上映射的深度集成支持链上债券、实物资产等关键数据的可信接入，以及zk-Coprocessor 与链下 AI 计算的可验证连接，跨链、链下推理、链上验证一体化。可以预见，随着零知识证明技术的持续进步，基于 zk 的跨链桥与预言机体系将成为新一代区块链基础设施的核心组成部分，为去中心化生态构建更坚固的数据流通与资产互操作网络。

### 12.4.4 ZK Coprocessor 与未来链下计算路径

随着区块链应用的复杂性不断提升，单纯依赖链上执行环境（如 EVM）的计算能力已难以满足高性能场景与多样化应用需求。从高频金融交易到机器学习模型推理，链上执行的成本高昂、吞吐受限，因此链下计算就成为必然选择。然而，链下计算如何被链上安全、可信地验证，成为下一代区块链架构的核心挑战。ZKP 为这一问题提供了关键突破口。通过设计 ZK Coprocessor（零知识协处理器）与 zkML（零知识机器学习验证）等新型架构，链上系统可以将复杂计算任务外包至链下环境，同时以 zk-proof 的形式安全、高效地验证链下计算结果，兼顾数据隐私与计算正确性。

#### （1）ZK Coprocessor成为链下计算的可验证外包路径

ZK Coprocessor意指基于零知识证明的链下协处理器，旨在解决链上虚拟机无法高效处理复杂计算的问题。其基本设计逻辑是将大规模矩阵运算、机器学习推理、复杂数学公式等复杂计算任务发送给链下 ZK Coprocessor，Coprocessor 在链下完成计算，同时生成证明该计算正确性的零知识证明，然后将计算结果与 zk-proof 一并返回链上，由链上智能合约进行证明验证。智能合约基于验证结果执行后续逻辑，无需信任链下计算节点。

这一机制的核心价值在于大幅降低了链上的 Gas 消耗，链上只需验证 succinct proof，计算主负担全部链下转移；AI 推理、图像处理、统计分析等高负载复杂计算任务均可链下安全处理。同时保障了计算正确性，链上通过 zk-proof 验证计算结果，防止链下节点作弊。代表性项目如 RISC Zero 正在开发通用型 zkVM（零知识虚拟机），支持链下 Rust、C 等高阶语言程序的可信执行与零知识证明生成。此类 zk Coprocessor 正逐步成为 zk-Rollup、zkBridge、去中心化 AI 计算等系统的底层通用模块。目前 ZK Coprocessor 在大规模计算的证明生成仍然昂贵，尤其是 AI 推理类计算的证明时间和电路复杂度仍然是瓶颈，成熟度仍在“早期实验/PoC 阶段”，尚未形成大规模商用。

#### （2）zkML成为链下 AI 推理的零知识验证路径

在 Web 3 与 AI 融合的趋势下，机器学习（ML）模型的推理结果正逐渐成为链上决策的重要输入，例如去中心化推荐系统、基于 AI 的链上身份评分、自动化链上合约管理等。然而，AI 模型的执行通常依赖链下环境，且难以保证模型计算未被篡改。如何在保障数据隐私的同时，向链上证明 AI 推理的正确性，是 zkML（零知识机器学习）试图解决的关键问题。

zkML 的基本思路是在链下运行 AI 模型推理，输入可加密或隐藏；使用专门设计的 zk-proof 生成电路证明模型推理步骤的正确性；将模型结果与 zk-proof 提交至链上，供智能合约验证。zkML.org、Giza、Modulus 等项目正在推进 zkML 的工程实现，涵盖了线性模型（如逻辑回归）的零知识证明高效电路设计、多层感知机（MLP）和卷积神经网络（CNN）等基础模型的 zk-proof 优化，以及对 Transformer 等深度模型支持的逐步尝试。

当前 zkML 的技术挑战主要有因为证明电路复杂度高而导致的证明生成时间长和适用场景有限，因为模型参数规模大而带来的电路优化与高效哈希支持问题；因为隐私需求复杂而带来的如何在不泄露训练数据与模型细节的前提下完成验证的问题。尽管如此，zkML 作为链上 AI 合规、去中心化 AI 推理验证的未来方向，仍具有极高的研究价值与应用潜力。

#### （3）数据隐私与计算正确性的统合范式

传统链下计算方案面临两个核心问题，一是结果是否正确，链上难以验证，容易被作恶节点操纵；二是输入数据如果涉及隐私问题，链上传输原始数据，容易被链上第三方窥探。ZK Coprocessor 与 zkML 提供了一种全新的统合范式，链上只验证 zk-proof，无需信任链下节点；输入数据加密，链上合约验证结果正确性但无法获知原始数据。这种范式特别适合去中心化 AI 服务，链上验证 AI 推理结果而无需暴露训练数据。对于链下 RWA 数据处理，金融机构在链下计算资产收益、风险参数，链上只需验证正确性。此外，ZK Coprocessor 也支持与 Rollup、跨链桥、预言机系统无缝集成，为多链、链下数据协同提供了统一的可验证计算基础。目前这类系统大多还处于实验性 PoC 阶段，尚未进入大规模商用。

#### （4）零知识证明与链下执行环境的深度融合

未来区块链系统的扩展路径，必然依赖链下计算能力的充分释放与链上验证机制的不断完善。ZK Coprocessor 将成为链下执行环境（Off-chain Execution Environment）与链上智能合约的核心桥梁。

当前的技术路径包括以下几个方面。一是zkVM（零知识虚拟机）开发，支持通用编程语言（Rust、C、LLVM IR 等）程序的 zk-proof 自动生成，以显著降低链下开发门槛；二是递归证明优化，支持将多个链下计算证明嵌套为单一证明，以提升验证效率；三是模块化 zk-Coprocessor 标准化，如 zkWASM、zkLLVM 等不同底层架构，形成可插拔链下计算协议；四是高效证明系统的开发，Halo 2、Plonky2、Nova 等支持递归、快速生成的新型证明系统将进一步缩短证明生成时延，扩大 zk-Coprocessor 的适用范围。未来，链下 AI 推理、链下复杂金融计算、链下用户身份验证、链下治理流程，都可以在 zk-Coprocessor 框架下实现统一的链上验证。

#### （5）ZK 驱动的去中心化可信计算网络未来展望

ZK Coprocessor、zkML、链下计算验证架构的持续演进，将推动区块链从“账本层去中心化”迈向“计算层去中心化”的新阶段。我们有望看到在去中心化 AI 服务平台方面，用户无需信任中心化 AI 服务商，链上即可验证模型推理结果；在多链协同计算网络方面，链下协处理节点就可以服务多个区块链，通过 zk-proof 实现全链互操作；在数据所有权保护机制方面，用户数据在链下私密处理，链上只公开验证结果，赋予用户更强的数据主权。但这些进展目前仍处于研究/实验性阶段，成熟度尚低。

未来“链上状态 + 链下计算 + 零知识验证”的混合架构可能成为DApp的主流。如果这一判断成立的话，那么ZK Coprocessor 将成为这一体系的技术支柱，推动区块链迈向更高性能、更强隐私、更可信的全新生态。

## **本章小结**

本章系统梳理了零知识证明（Zero-Knowledge Proof, ZKP）在区块链系统中的原理、密码构造、应用路径和面临的工程挑战，旨在帮助读者建立从理论基础到协议设计再到隐私合规的完整知识体系。

首先，我们从零知识证明的基本定义与安全特性入手，回顾了 Goldwasser-Micali-Rackoff 所提出的交互式证明模型，并深入分析了 ZKP 完备性、可靠性与零知识性等三大核心属性。通过对交互式与非交互式 ZKP 的比较，我们理解了 Fiat-Shamir 转换的重要意义，以及区块链系统为什么偏好使用非交互式零知识证明。

随后，我们详细剖析了当前主流的零知识证明系统。zkSNARK 作为链上应用最广泛的证明系统，凭借 succinct（简洁）与 non-interactive（非交互）的优势，在隐私币与 ZK-Rollup 等场景得到广泛应用，但也因 Trusted Setup 的信任假设受到一定争议。zkSTARK 作为新一代证明系统，突破了 Trusted Setup 的限制，基于哈希函数的透明性设计提升了抗量子安全性，同时在可扩展性与证明大小之间做出了新的权衡。

我们还介绍了 Bulletproofs、Plonk、Halo 2 等新型证明系统，它们为递归证明、模块化电路设计和通用计算支持开辟了更多路径，显示出零知识证明技术正在快速向更高效、更透明、更通用的方向演进。

在应用层，本章重点讨论了链上匿名交易与隐私币机制。我们从 Tornado Cash、Zcash 到 Aztec Protocol 的不同设计切入，深入理解了 zkSNARK 在隐私交易中的作用，比较了不同项目在技术架构、隐私保护深度与合规风险之间的取舍。通过 Tornado Cash 制裁事件的分析，我们认识到链上隐私协议与全球监管体系存在的结构性张力。

针对隐私保护与合规监管之间的张力，本章进一步讨论了链上若干可行的合规路径，包括 View Key 与 Audit Key 机制、基于 zk-SNARK 的合规电路模板、以及匿名 KYC（ZK-ID）等设计。这些方案为在隐私与合规之间寻找平衡提供了潜在的技术支撑。

在扩展性部分，我们深入讲解了ZK-Rollup 的结构与运作原理，明确了链下批量计算与链上 succinct 验证的效率协同，比较了 ZK-Rollup 与 Optimistic Rollup 的安全模型差异。通过对 zkEVM 的技术路径（如 zkSync Era、Polygon zkEVM、Scroll）的剖析，我们理解了 CIRCUIT 级与 Bytecode 级兼容性的工程难题，以及 zkEVM 在性能、开发体验与未来演进路径中的现状与潜力。

更进一步，本章前瞻性地探讨了零知识证明在跨链与预言机中的应用拓展，如无需信任的 zkBridge 设计与 zk-Oracles 的链外数据验证方案。相比传统预言机，基于 ZKP 的去信任设计在安全性与通用性方面展现出更强的适配能力。

最后，我们聚焦于未来的链下计算架构，重点解析了 ZK Coprocessor 与 zkML（零知识机器学习验证）的技术路径。ZK Coprocessor 将成为高复杂度链下计算的可验证外包模块，而 zkML 则为 AI 推理结果的链上可信验证提供可能，推动链上系统从“账本层去中心化”迈向“计算层去中心化”。这一趋势将促进区块链与 AI、数据隐私保护、复杂金融建模等领域的深度融合。

通过本章学习，读者应能够掌握以下核心能力。一是系统理解零知识证明的原理与主流系统差异；二是能够评估 ZKP 在链上隐私保护、扩展性、合规设计中的应用价值；三是对 ZK-Rollup、zkEVM、zkBridge、zk-Oracles、zkML 等前沿技术具备工程级理解；四是能够思辨隐私与监管的冲突，分析未来可选择合规路径的技术可行性。

零知识证明已成为区块链技术的重要基石，它不仅为隐私保护提供工具，更正在重塑链上计算、跨链验证与数据处理的安全框架。下一步，随着证明系统工程优化、zkEVM 标准成熟、zkML 逐步落地，ZKP 有望成为 Web 3 世界中通用的隐私与可信计算基础设施。

1. 本书作者2024年下半年在电子工业出版社出版了《Web 3中的零知识证明》（ISBN：9787121488115），如有更深入全面的需求可以参考。 [↑](#footnote-ref-0)
2. 攻击者向大量公开地址发送极小金额（“灰尘”），使这些地址被动收到受制裁资金，从而触发黑名单标记，以此暴露制裁在技术执行层面的局限。 [↑](#footnote-ref-1)