零知识证明（ZKP）

杨子凡

Jul 03, 2025

1　导言

在数字时代，我们面临一个根本性矛盾：如何既证明某个事实的真实性，又不泄露背后的敏感信息？想象向门卫证明俱乐部会员身份却不出示证件，或让银行验证资产达标却不透露具体金额。零知识证明（Zero-Knowledge Proof，ZKP）正是解决这一矛盾的密码学突破，其核心在于实现「数据可用不可见」。这项技术正在重塑区块链架构、身份认证系统和隐私保护方案，本文将系统拆解其数学原理、工程实现与前沿应用。

2　为什么需要零知识证明?

传统验证机制存在本质缺陷：密码验证需传输秘密，数字签名暴露公钥关联。当涉及医疗记录共享或金融反洗钱（KYC）时，这些方法迫使用户在隐私与合规间妥协。区块链领域更面临「不可能三角」困境 —— 可扩展性、去中心化与隐私性难以兼得。零知识证明通过数学约束替代数据披露，成为破局关键。例如匿名投票场景中，选民可证明自己属于合法选民集合却不泄露具体身份，实现隐私与可验证性的统一。

3　零知识证明的三大核心特性

完备性确保诚实证明者总能说服验证者：若命题为真且双方遵守协议，验证必然通过。可靠性防止作弊者伪造证明，其安全强度可表示为：当证明者作弊时，验证通过的概率不超过 2^{-k}（k 为安全参数）。最核心的零知识性通过模拟器概念严格定义 —— 验证者视角获取的信息与随机数据不可区分。形式化表述为：存在模拟算法 $\mathcal{S}$，对任意验证者 $\mathcal{V}^*$，满足以下分布等价：$[{\text{view}}_{\mathcal{V}^*}(x,w)}{(x,w) \in R} \approx {\mathcal{S}(x)}{(x,w) \in R}]$ 其中 R 为关系集合，view 包含验证过程所有交互数据。

4　从故事到数学：零知识证明的直观理解

阿里巴巴洞穴故事揭示交互证明的统计特性：证明者宣称知晓打开魔法门的咒语，验证者每次随机要求左/右通道。若证明者作弊，单次通过概率仅 50%，重复 20 次后作弊成功概率降至 9.5×10^{-7}。数学本质对应 NP 问题的知识证明：证明者拥有证据（witness）w，向验证者证明其满足关系 $R(x, w)=1$，其中 x 为公开陈述。例如证明佩尔方程 $x^2 - 2y^2 = 1$ 有整数解，却不泄露具体解向量 (x,y)。

5 零知识证明技术栈演进：从理论到实用

早期交互式证明依赖多轮挑战-响应，1986 年 Fiat-Shamir 启发式实现关键突破：将交互协议转为非交互式证明（NIZK）。核心思想是用哈希函数模拟验证者挑战，即 (\text{challenge} = \mathcal{H}(\text{transcript}))。现代 ZKP 体系呈现三足鼎立：zk-SNARKs 凭借恒定大小证明（约 288 字节）成为主流，但需可信设置；zk-STARKs 基于哈希函数抗量子攻击，代价是证明体积膨胀至 100KB；Bulletproofs 则专注高效范围证明，无需可信设置但验证成本较高。

6 深入 zk-SNARKs：最主流的实现原理

zk-SNARKs 技术栈分层构建：首先将计算问题算术电路化。例如验证 (a \times b = c) 可转化为乘法门约束。接着转化为 R1CS（Rank-1 Constraint System）约束系统，每个约束表示为向量内积：[(\vec{a}_i \cdot \vec{s}) \times (\vec{b}_i \cdot \vec{s}) = (\vec{c}i \cdot \vec{s})] 其中 (\vec{s}) 为包含变量值的状态向量。关键步骤是通过 QAP（Quadratic Arithmetic Program）将向量约束编码为多项式：在插值点 (x_k) 处，多项式需满足 (A(x_k) \cdot B(x_k) - C(x_k) = 0)。最终目标转化为证明存在多项式 (h(x)) 使得：[A(x) \cdot B(x) - C(x) = h(x) \cdot t(x)] 其中 (t(x) = \prod{k=1}^{n}(x - x_k)) 为目标多项式。
通过椭圆曲线配对（Pairing）实现同态隐藏：证明者计算 (g^{A(s)}, g^{B(s)}, g^{h(s)}) 等椭圆曲线点（(s) 为秘密点），验证者检查配对等式 (e(g^{A(s)}, g^{B(s)}) = e(g^{t(s)}, g^{h(s)}) \cdot e(g^{C(s)}, g)) 是否成立。
可信设置环节通过多方计算（MPC）降低风险，如 Zcash 的 Powers of Tau 仪式要求参与者协作生成 CRS 后销毁秘密碎片。新型可更新设置方案允许后续参与者覆盖前序密钥，实现向前安全。

7 零知识证明实现实战：开发者视角

主流开发库如 circom 提供领域特定语言（DSL）定义电路。以下电路证明用户知晓满足 (a \times b = c) 的秘密整数：

```
pragma circom 2.0.0;
template Multiplier() {
   signal input a; // 私有输入
   signal input b; // 私有输入
   signal output c; // 公开输出
   c <== a * b; // 约束声明
}
component main = Multiplier();
```

代码解析：`signal` 声明电路信号，`input` 标注私有输入，`output` 为公开输出。`<==` 操作符同时进行赋值与约束绑定。编译流程为：1）电路编译为 R1CS 约束系统；2）基于 CRS 生成证明密钥（pk）与验证密钥（vk）；3）证明者用 pk 和私有输入生成证明 (\pi)；4）验证者用 vk 和公开输入验证 (\pi)。
性能优化是落地关键。Prover 计算瓶颈在于多标量乘法（MSM）和快速傅里叶变换（FFT），GPU 加速可提升

30 倍性能。递归证明技术将证明作为另一电路输入，实现证明聚合。以下伪代码展示递归验证逻辑：

```
// Nova 方案中的步进电路
fn step_circuit(
    z_i: [F; 2], // 当前状态
    U_i: RelaxedR1CS, // 当前证明
    params: &Params // 参数
) -> ([F; 2], NIFSVerifierState) {
    let (z_{i+1}, U_{i+1}) = fold(U_i, z_i); // 证明折叠
    (z_{i+1}, U_{i+1})
}
```

通过连续折叠（folding）多个证明，最终只需验证单个聚合证明，链上验证成本从 (O(n)) 降为 (O(1))。

8 零知识证明的杀手级应用场景

区块链扩容领域，zkRollup 将千笔交易压缩为单个证明提交至 Layer1。以 zkSync 为例，其电路处理签名验证、余额检查等逻辑，使 TPS 从以太坊的 15 提升至 3,000+。隐私保护场景中，Tornado Cash 混币器使用 Merkle 树证明成员资格：[\exists \ \text{path}: \ \text{root} = \text{Hash}(\text{leaf}, \text{path})] 用户证明自己属于存款集合却不暴露具体叶子节点。身份合规领域，zkKYC 方案允许用户证明年龄满足 (\text{age} \geq 18) 而不泄露生日日期。去中心化存储协议 Filecoin 的 PoRep 电路则验证存储提供方正确编码数据，电路规模达 1.25 亿个约束。

9 挑战与未来方向

当前瓶颈集中在证明生成效率，例如证明 Zcash 交易需 7 秒（8 核 CPU）。硬件加速方案如 FPGA 实现 MSM 模块可提升 100 倍吞吐。开发体验方面，高阶电路语言如 Halo2 的 PLONKish 算术化方案支持自定义门：

```
// Halo2 自定义乘法门
meta.create_gate("mul", |meta| {
    let a = meta.query_advice(col_a, Rotation::cur());
    let b = meta.query_advice(col_b, Rotation::cur());
    let c = meta.query_advice(col_c, Rotation::cur());
    vec![a.clone() * b.clone() - c.clone()]
});
```

未来方向包括透明设置（zk-STARKs）、并行化证明（Nova）及 ZK 协处理器。跨领域融合如 ZKML 实现模型推理可验证：用户提交预测请求，服务端返回结果与 ZKP，证明推理过程符合预定模型架构。

零知识证明本质是密码学的优雅舞蹈 —— 用数学约束替代数据暴露。开发者无需理解全部数学细节，可从 circom 玩具电路入门实践。随着硬件加速突破和开发者工具成熟，互联网基础设施正经历从「可选隐私」到「默认隐私」的范式迁移。零知识证明作为隐私计算的基石，将持续重塑我们对数据价值的认知边界。