ACE Runtime——具备亚秒级密码学最终性的零知识证明原生区块链运行时

Jian Sheng
Wang Yeah LLC
jason@yeah.app

2026年3月10日

arXiv:2603.10242v1 [cs.CR] 10 Mar 2026

摘要

现有的高性能区块链（如 Solana）在关键执行路径上硬编码了每笔交易的签名验证，致使 $O(N)$ 签名检查成为主要的吞吐量瓶颈和硬件成本的重要来源。本文提出 ACE Runtime，一种构建于称为 ACE-GF 的身份–授权分离原语之上的区块链执行层。其核心创新是三阶段 证明–执行–证明 流水线架构：将授权验证从密码学签名转移到轻量级的基于HMAC的证明（在商用中央处理器上每笔交易 $\sim 1\ \mu$s），同时将零知识证明生成完全移出关键路径。由此形成的架构目标为 400 毫秒区块时间、 ~ 600 毫秒硬（密码学）最终性、 $O(1)$ 区块验证成本，并消除所有非构建者验证器的GPU要求。基于标准服务器硬件的分析建模预计，保守吞吐量为 $\sim 16{,}000$ 每秒交易数（TPS），经适度工程投入可扩展至 $\sim 32{,}000$ TPS。我们给出了详细算法、形式化安全分析以及模型驱动的定量比较，结果表明：在所述建模假设下，相对于规模化的 Solana，ACE Runtime 可将最终性延迟最多降低 $\sim 20\times$ ，并将每区块验证成本最多降低 $\sim 4{,}000\times$ 。

关键词： 区块链运行时、零知识证明、身份与授权分离、亚秒级最终性、Groth16、HMAC证明、后量子就绪

1 引言

高性能第一层区块链在交易吞吐量方面取得了显著进展。例如，Solana [5], 通过采用流水线式交易处理单元（TPU），将 Ed25519 签名验证卸载至图形处理器，实现了 400 毫秒的区块时间，并能每秒处理数千笔交易。然而，每笔交易都需要一次密码学签名这一基础设计带来了 $O(N)$ 验证成本，且该成本会随区块大小线性增长。这一瓶颈主要体现在三个方面：

1. **吞吐量上限**。 即使采用 GPU批量验证，每个签名耗时 $\sim 2\ \mu$s，包含 100,000 笔交易的区块也仅 SigVerify 就需要 200ms，占用一半的区块间隔。2. **硬件成本**。 每位 Solana 验证者——不仅是区块生产者——都必须配备能够进行批量签名验证的图形处理器，使最低硬件预算升至约 $7,500，每月运营成本则达到 $1,000–$3,000。3. **后量子漏洞**。 若转向 ML-DSA-44 [11] 等后量子签名方案，每笔交易授权数据将从 96 字节（Ed25519

签名加公钥）增至 $\sim$3，732 字节，使区块大小增加超过 $38\times$，令当前架构无法实际运行。

1.1 关键洞见：身份—授权分离

该ACE-GF 框架 [1] 将身份绑定与逐笔交易授权分离。用户无需为每笔交易附加完整的密码学签名，而是通过 HKDF，从根熵值（*REV*）派生出轻量级的基于 HMAC 的证明。 [7]证明该证明正确绑定至用户链上身份承诺（身份_承诺）的密码学证明，会在区块发布之后生成，而非在执行之前生成。

这种分离将验证模式从“每笔交易一个签名”转变为“每个区块一个简洁证明”，无论区块包含多少笔交易，都能实现 $O(1)$ 区块验证。

1.2 贡献

本文作出以下贡献：

1. **认证-执行-证明流水线**。我们设计了一种三阶段执行流水线，将轻量级证明检查和交易执行（位于关键路径上）与零知识证明生成（位于关键路径之外）解耦，目标是实现 400ms 的出块时间，并达到 $\sim$600 ms密码学硬最终性（第3.1节）。

2. **证明流水线**。 我们提出一种图形处理器并行证明生成方案，通过树形递归聚合16 [6]，在区块间隔内为每个区块生成一个 Groth16 [6] 证明（第 3.3 节）。

3. **双层最终性模型**。 我们形式化提出一种最终性模型，其中 软最终性（BFT 投票，$\sim$400 毫秒）和 硬最终性（零知识证明验证， $\sim$600毫秒），并证明在标准密码学假设下，硬最终性在计算上不可逆（第 3.4 节）。

4.**全面定量评估**。 我们提供原型微基准测试和基于模型的分析比较，表明 $O(1)$ 区块验证；区块传播带宽效率最高可达 $5\times$ （$\sim$1.9–$5\times$ 系统总带宽，取决于见证数据的 Gossip 调度），$\sim 20\times$ 正常路径上的硬最终性更快，并且非构建者验证者无需满足 GPU 要求（第 7 节）。

ACE Runtime 建立在 ACE-GF 身份框架 [1], 、ZK-ACE 证明系统 [2], 、AR-ACE 账户恢复协议 [3], 以及 CT-DAP 跨链转账机制 [4]之上，并引用了这些内容。

2 背景与动机

2.1 ACE-GF 密码原语

原子密码实体生成框架（ACE-GF） [1] 引入了一种确定性的多流密钥派生架构，该架构以一个称为根熵值（REV）的高熵秘密为根基。REV 使用 Argon2id [10] 进行封装，参数为（m=4096, t=3, p=1），且从不以明文形式存储。

定义 2.1 (多流密钥派生)。 给定根熵值REV和用途专用的信息字符串 信息$_i$，第 i 条密钥流定义为

$$k_i \leftarrow \mathsf{HKDF\text{-}SHA256}(\mathsf{REV},\ info_i,\ salt_i). \tag{1}$$

表 1：Solana 流水线上的 Ed25519 签名验证成本。

操作	CPU	图形处理器（批处理）
每笔交易验证	$\sim$76 μs	$\sim$2 μs
1,000-交易区块	76 毫秒	2 ms
10,000-交易区块	760 毫秒	20 毫秒
100,000-交易区块	7,600 毫秒	200 毫秒

ACE-GF 定义了七条规范密钥流：

流 1：HKDF(REV `"ACEGF-V1-ED25519-SOLANA"`)，用于 Solana 签名，

流 2：HKDF(REV `"ACEGF-V1-ED25519-POLKADOT"`)，用于 Polkadot，

流 3：HKDF(REV `"ACEGF-V1-SECP256K1-EVM"`)，用于 EVM 链，

流 4：HKDF(REV `"ACEGF-V1-SECP256K1-BTC"`)，用于比特币，

Stream 5：面向 Cosmos 的 HKDF(REV `"ACEGF-V1-SECP256K1-COSMOS"`)，

Stream 6：用于端到端加密的 HKDF(REV `"ACEGF-V1-X25519-IDENTITY"`) ，

Stream 7：用于后量子签名的 HKDF(REV `"ACEGF-V1-ML-DSA-44-PQC-IDENTITY"`)。

定义 2.2（上下文隔离）。 对于任意两个不同的上下文字符串 $c_1 \neq c_2$，

$$\mathsf{HKDF}(\mathsf{REV},\ \mathit{info},\ \mathit{context}{=}c_1) \neq \mathsf{HKDF}(\mathsf{REV},\ \mathit{info},\ \mathit{context}{=}c_2), \tag{2}$$

根据 HKDF 的伪随机函数假设，这两个派生密钥在计算上不可关联。

定义 2.3（零知识身份承诺）。 链上身份由 Poseidon 哈希承诺表示 [9]：

$$\mathsf{id_com} = \mathsf{Poseidon}(\mathsf{REV},\ \mathit{salt},\ \mathit{domain}). \tag{3}$$

这个 32 字节的值是链上可见的唯一 身份凭证。公钥从未公开，因此天然能够抵御“先收集，后解密”量子攻击。

2.2 Solana TPU 流水线分析

Solana 的交易处理单元通过四阶段流水线处理交易 ：

$$\mathsf{Fetch} \ \rightarrow\ \mathsf{SigVerify}\ (\mathrm{GPU}) \ \rightarrow\ \mathsf{Banking} \ \rightarrow\ \mathsf{Broadcast}.$$

签名验证阶段位于关键路径上，并带来 $O(N)$ 开销，且该开销随交易数量线性增长。表1 列出了 Ed25519 签名验证的单笔交易和单个区块开销。

两项关键观察促使我们采用这一设计：

1. **串行依赖**。 SigVerify 必须在开始执行之前完成，从而在关键路径上形成硬串行依赖。区块大小的任何增加都会直接增加端到端延迟。

2. **普遍 GPU 要求**。 由于每个验证者都必须独立验证每个签名，因此所有验证者都需要 GPU 硬件，而不仅仅是区块生产者。这提高了参与的经济门槛，并使验证者集合趋于中心化。

表 2：认证-执行-证明流水线的关键路径分析。

阶段	延迟	关键?	瓶颈
AttestCheck	∼1–5 μ秒/交易	Yes	中央处理器（轻量级）
执行	∼10–50 μ秒/交易	Yes	状态输入/输出
区块发布	∼50 毫秒	Yes	网络
每笔交易证明	∼15 毫秒 × 128 并行	否（图形处理器异步）	GPU
2,000笔交易的总证明	∼240 毫秒	否（流水线处理）	GPU
树形聚合	∼30–60 毫秒	No	CPU/GPU
FC 发布	∼50 毫秒	否（下一个时隙）	网络

2.3 现有零知识增强型区块链的局限性

有些区块链系统采用零知识证明，但没有任何系统将其应用于身份与授权层，并以此实现 $O(1)$ 逐区块验证：

- **zkSync 和 StarkNet** [14, 15] 使用零知识证明进行状态压缩和二层汇总验证。单笔交易签名仍会在证明生成前，由 Rollup 排序器在 $O(N)$ 内进行验证。

- **Mina Protocol** [16] 采用递归 SNARK 组合来实现恒定大小的链状态，但每个区块的签名验证仍为 $O(N)$。

- **Aptos 和 Sui** [17, 18] 通过基于 DAG 的 BFT 共识实现快速最终性，但仍在关键路径上保留 $O(N)$ 逐笔交易签名验证。

据我们所知，尚无现有的一层区块链实现了 $O(1)$ 身份层验证与亚秒级硬最终性的结合。

3 系统设计

本节介绍 ACE Runtime 的核心贡献：认证-执行-证明流水线架构、认证验证机制以及证明流水线策略。

3.1 认证-执行-证明流水线

ACE Runtime 通过两个在连续时隙间重叠的阶段处理每个时隙。**阶段1**（证明 + 执行）位于关键路径上，并决定出块时间。**阶段 2**（证明）在 GPU 硬件上异步运行，并生成在后续时隙到达的密码学最终性证书。

算法 1 详细说明了完整的流水线逻辑。表格 2 分解了各阶段的耗时，并指出它们是否位于关键路径上。

关键路径仅包括 AttestCheck、执行和区块发布，总计约 300–400 毫秒。证明生成完全异步进行，并与后续时隙的执行重叠。

3.2 认证结构与验证

证明用基于 HMAC 的凭证取代传统的逐笔交易签名，该凭证通过 ACE-GF 密钥派生层次，由用户的 REV 派生而来。

算法 1 证明–执行–证明流水线调度器 Require: 交易池 T，当前时隙编号 SEnsure: 已发布区块 B_S，最终性证书 FC $_{S-1}$1: procedure 处理时隙(S) 2: txs $\leftarrow$ 选择交易(T，max$=2000$)▷ 阶段 1a：并行证明检查（$\sim$1–5 μs/交易，中央处理器）3: for all tx $\in$txs in parallel do 4: $k \leftarrow$HKDF(tx.rev_proxy "ACEGF-V1-MEMPOOL-ATTEST", domain) 5: expected $\leftarrow$ HMAC-SHA256(k, tx.obj_hash $\|$tx.domain) 6: if tx.attestation.credential $\neq$ expected then 7: Reject(tx) 8: end if 9: end for▷ 阶段 1b：并行执行（$\sim$10–50 μs/交易）10: for all tx $\in$通过 Sealevel 增强型调度器处理的有效交易 do_11: $\Delta_{tx} \leftarrow$执行(tx) 12: end for 13: $B_S \leftarrow$构建区块(valid_交易 Δ, 证明) 14: Broadcast(B_S) ▷ 获得 2/3 票数后实现软最终性 15: end procedure 16: procedure 异步证明(B_S) ▷阶段 2：GPU 并行，在关键路径之外 17: $\Pi \leftarrow \emptyset$18: for all tx $\in B_S$.在 GPU 上并行处理的交易 do 19: pub $\leftarrow$ [tx.id_com, tx.hash, tx.domain, tx.target, tx.rp_com]20: $\pi_{tx} \leftarrow$ Groth16.Prove(C$_{\text{ZK-ACE}}$, w_{tx}, pub) 21: $\Pi \leftarrow \Pi \cup \{\pi_{tx}\}$ 22: end for 23: while $|\Pi| > 1$ do ▷ 树形递归聚合 24: $\Pi \leftarrow$成对聚合(Π) 25: end while 26: FC $_S \leftarrow$ (Hash(B_S) S, $\Pi[0]$, Commit(B_S.id_coms))27: Broadcast(FC $_S$) ▷ 硬最终性 28: end procedure

定义 3.1 (证明)。 对于载荷为 P、在链 c 上时隙 s 中的交易，其证明是一个元组 $A =$(对象_哈希，id_com，域，凭证)，其中：

$$obj_hash = \text{SHA-256}(P), \tag{4}$$

$$domain = c \,\|\, s, \tag{5}$$

$$k_{attest} = \text{HKDF-SHA256}(\text{REV},\ \texttt{"ACEGF-V1-MEMPOOL-ATTEST"},\ domain), \tag{6}$$

$$credential = \text{HMAC-SHA256}(k_{attest},\ obj_hash \,\|\, domain). \tag{7}$$

证明的大小为 88–104 字节（32 字节 对象_哈希、16–32 字节 id_com、8 字节域、32 字节凭证）。

算法2 介绍了证明生成与验证流程。验证者在中央处理器上对每笔交易仅执行轻量级的格式和绑定检查，耗时 $\sim$1–5 μ秒。完整的密码学绑定——证明凭证源自一个REV ，且该 REV 的哈希值与已注册的ID_com相匹配——则由阶段 2 中的 Groth16 证明确立。

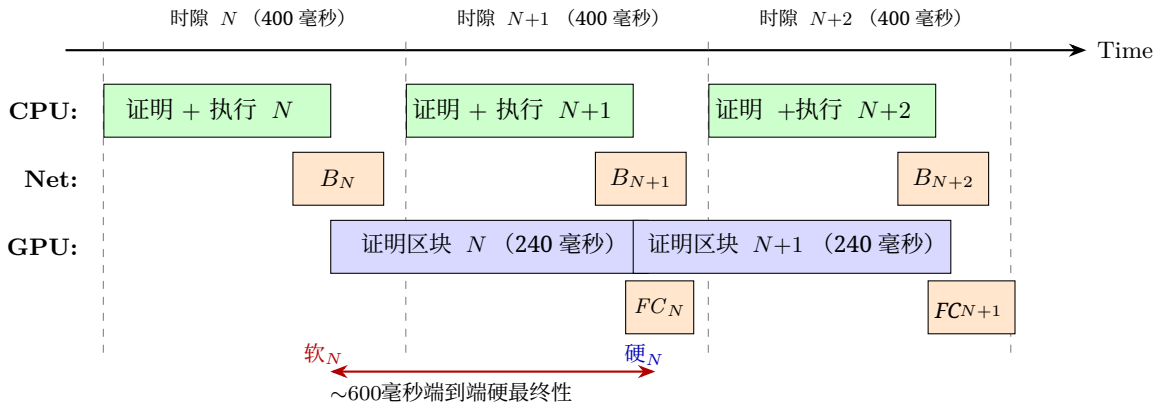

图 1：跨连续时隙的证明流水线。受中央处理器限制的证明检查和执行占据关键路径（绿色），而图形处理器证明生成（蓝色）和最终性证书发布（橙色）则与后续时隙并行进行。

表 3：每笔交易的数据格式对比。

组件	Solana	ACE Runtime
区块头	3 B	3 B
账户密钥	$N \times 32$B	$N \times 32$B
近期区块哈希	32 B	32 B
指令	变量	可变
授权	$N \times 64$B（签名）	90 B（证明）
典型总计	250–1,232 B	220–244 字节
公钥暴露	Yes	**None**

交易格式比较。表3 比较了 Solana 与ACE RUNTIME每笔交易的链上数据。

3.3 证明流水线

证明生成流水线在专用图形处理器硬件上持续运行（构建者角色），并与区块生产关键路径完全解耦。算法 3 详细说明了三阶段证明生成流程。

ZK-ACE 电路规范。 ZK-ACE 电路 [2]在 BN254 [8]上运行，并强制执行以下约束：

(i) 身份绑定：Poseidon(REV, salt, domain) = id_com。(ii) 密钥派生正确性：HKDF(REV, info, domain) = k_{attest}**。(iii) 证明正确性：HMAC(**k_{attest}**, 交易_哈希||domain) = 凭证。(iv) 策略约束（可选）：target ∈allowed_set。**

该电路包含约 1,400 个 R1CS 约束，可生成 256 字节的 Groth16 证明，并通过一次 Groth16 验证（在 BN254 上进行三次双线性配对运算）在 ~0.5 毫秒内完成验证。

算法 2 证明生成与验证

客户端：生成证明

函数 生成证明(REV，P, c, s)

$k \leftarrow$HKDF(REV "ACEGF-V1-MEMPOOL-ATTEST"，$c\|s$)

$h \leftarrow$ SHA-256(P)

凭证 $\leftarrow$HMAC-SHA256(k，$h\|c\|s$)

返回（h，标识_组件 $c\|s$，凭证） ▷ 88–总计 104 字节

结束函数

验证方：验证证明

函数 VerifyAttestation(A,交易)

如果 SHA-256($tx.payload$) $\neq A.obj_hash$ **则**

返回 拒绝 ▷ 载荷绑定检查

结束 if

如果 A.id_com $\notin$已注册身份 **则**

返回 拒绝 ▷ 身份存在性检查

结束条件

返回接受待处理证明_ _ ▷ 完整的密码学验证推迟至 Groth16 证明

结束函数

3.4 双层最终性模型

ACE Runtime 引入了一种双层最终性模型，既能快速保证活性，也能提供强大的密码学不可逆性。

定义 3.2（软最终性）。区块 B_S 在 $\geq \frac{2}{3}$ 的按质押权重加权的验证者投票确认该区块时，达到软最终性 。这发生在时隙开始后的 $t_{soft} \approx 400$毫秒。

定义 3.3（硬最终性）。区块 B_S 在 有效的 *Groth16* 最终性证书 FC_S 发布并由验证者验证后，达到硬最终性。在正常（构建者）路径上，这发生在时隙开始后的 $t_{hard} \approx 600$毫秒 。在故障恢复（备用）路径上，硬最终性可能最多延迟至 $t_{hard}^{\max} =$ （$K + K'$） $\times 400$ 毫秒 ≤ 2.4 秒 （定理 6.2）。

超时分为两个阶段（在定理 6.2中形式化）：一个构建器窗口 持续$K = 2$–3 个时隙（800 毫秒–1.2 秒），随后是一个备用窗口，持续 $K' = 2$–3 个额外时隙。构建者若未能在 K 个时隙内提交有效证明，将没收其全部质押；如果备用窗口期间也没有替代证明到达，该区块将回滚。表4 比较了各大区块链的最终性延迟。

定理 3.4 (硬最终性的不可逆性)。在 *BN254* 上的指数知识假设成立，且 *Poseidon* 和 *SHA-256* 具有抗碰撞性的前提下，已达到硬最终性的区块只能被能够伪造 *Groth16* 证明的对手逆转，而这需要攻破 q-*PKE* 假设。

证明。假设区块 B_S已通过证书 FC_S =(Hash(B_S) S，π，com) 达到硬最终性。逆转需要满足以下条件之一：

1. 为不同的区块 B'_S 生成有效证明 π'，且其时隙编号 S 和公共输入承诺相同，这与com中使用的 Poseidon 承诺的绑定性相矛盾；或 2. 为虚假陈述伪造 Groth16 证明，这与在 q-PKE 假设 [6]下 Groth16 的知识可靠性相矛盾。

算法 3 GPU 证明流水线

过程GPUProofPipeline
  $T \leftarrow 128$ ▷ GPU 线程并行（例如，RTX 4090）
  当运行时 **执行**
    $B \leftarrow$WaitForPublishedBlock
      ▷ **阶段 1：** 按交易并行生成证明（$\sim$1，每笔 400 个 R1CS 约束）
    $\Pi \leftarrow$ 并行映射(B.交易, GenZKACEProof T)
      ▷ $\lceil 2000/128 \rceil \times 15$ 毫秒 ≈ 240 毫秒
      ▷ **阶段 2：** 树形聚合（$\log_2 N$ 层）
    当 $|\Pi| > 1$ **时执行**
      $\Pi' \leftarrow \emptyset$
      对所有 $(i, i{+}1) \in \Pi$ **并行执行**
        $\Pi' \leftarrow \Pi' \cup \{$递归式Groth16($\Pi[i], \Pi[i{+}1]$)$\}$
      结束 for
      $\Pi \leftarrow \Pi'$
    结束 while
      ▷ 最后 2–3 层： $\sim$30–60 毫秒
    ▷ **阶段 3：** 发布最终性证书
    PublishFinalityCert(B.时隙 $\Pi[0]$)
      ▷ $\Pi[0]$：单个 Groth16 证明，256 B
  结束循环
结束过程

表 4：区块链平台的最终性延迟对比。

链	区块	Soft	硬†‡	回滚条件
Solana	400 毫秒	400 毫秒	$\sim$12 秒	Fork
以太坊	12 s	12 s	$\sim$15 分钟	Fork
ACE Runtime	**400 毫秒**	**400毫秒**	**$\sim$600毫秒**	无效/缺失的 FC 或分叉

†不同系统对硬最终性的定义各不相同。Solana：31 个已确认时隙（乐观回滚窗口） [5]。以太坊：两个 Casper FFG 纪元（BFT 最终性） [28]。ACE Runtime：已发布有效的最终性证书（基于密码学证明的最终性）。直接进行数字比较时，应考虑这些定义差异。

‡正常（构建者）路径。在故障恢复（备用）路径上，硬最终性可能需要最多 $\sim$2.4秒；参见定理 6.2。

无论哪种情况，对手都必须攻破一个计算困难假设，这表明硬最终性在计算上不可逆。 □

3.5 区块结构

定义 3.5 （ACE 区块）。ACE Runtime 区块 B 由以下部分组成：

- **区块头 (**$\sim$256 字节**)**：时隙编号、父哈希、状态根、交易 Merkle 根、证明 Merkle 根、PoH 哈希、领导者 ID_com、时间戳和交易数量。
- **区块体**：由交易向量（每笔 $\sim$154 字节）及其对应的证明（每个 $\sim$90字节）组成。值得注意的是，区块主体中不包含签名和证明；证明会在最终性证书中异步到达。

定义 3.6 （最终性证书）。 最终性证书 FC 是一种定长结构，由以下部分组成：区块哈希（32 字节）、时隙编号（8 字节）、聚合 Groth16 证明（256 字节），以及身份承诺列表的承诺（32 字节）。总大小为 300–400 字节，

算法 4 最终性状态机

状态： {待处理、 软、备份等待、 硬、 已回滚}

函数 更新最终性(S,事件)
  切换 事件
  情况 收到区块(B)
  验证 B.区块头 .证明_默克尔_根
  重新执行 B.交易
  若有效**则**
    投票(B.时隙)
  结束若
  若票数(B.时隙) $\geq \frac{2}{3} \cdot$ 总计_质押**则**
    B.状态 $\leftarrow$软 ▷ ∼400毫秒
  结束如果
  情况 收到最终性证书(FC)
  要求 *FC* .时隙 $= B$.时隙 **且** *FC* .区块_哈希 $=$ 哈希(B)
  如果 Groth16.验证(FC.证明 , *FC .pub,* 验证密钥) **然后**
    B.状态 $\leftarrow$硬 ▷ 正常或备用路径
  else if B.状态 $=$软 **then**
    B.状态 $\leftarrow$已回滚; 斜杠(B.构建者)
  结束如果 ▷ 在备份等待期间，忽略无效 FC；等待 $K+K'$
  情况超时(K 时隙) ▷ 构建者窗口到期
  如果 B.状态 $=$ 软 $\wedge \neg \exists$ *FC* **那么**
    削减(B.构建者) ▷ 立即罚没
    B.状态 $\leftarrow$备份等待 ▷ 进入备用窗口
  结束 if
  情况 超时($K+K'$时隙) ▷ 备用窗口到期
  如果 B.状态 $=$ 备份等待 $\wedge \neg \exists FC$ **则**
    B.状态 $\leftarrow$已回滚; 重新入队(B.交易)
  结束如果
结束函数

与区块大小无关。验证只需进行一次 Groth16 验证——在 BN254 上执行三次双线性配对运算——即可在 $\sim$0.5 毫秒内完成。

区块大小比较。 对于包含 2,000 笔交易的区块：

- **Solana：** $2,000 \times \sim 1,232$ B ≈ 2.4 MB（包括签名和公钥）。
- **ACE Runtime：区块头** ($\sim$256 B) + 笔交易（$2,000 \times \sim$154 B $= 308$ KB） + 份证明（$2,000 \times \sim$90 B $= 180$ KB） ≈ 488 KB，减少了 $\sim 5\times$ 。

4 共识设计

ACE Runtime 采用了一种共识架构，将基于历史证明（PoH）的领导者调度与 BFT 投票相结合，并通过原生于 ACE-GF 的验证者身份管理加以增强。

4.1 通过历史证明实现可验证时钟

遵循 Solana 模型 [5], ACE Runtime 采用连续的 SHA-256 哈希链，作为可验证的时间流逝时钟：

$$H_0 \;\rightarrow\; H_1 = \mathsf{SHA\text{-}256}(H_0) \;\rightarrow\; H_2 = \mathsf{SHA\text{-}256}(H_1) \;\rightarrow\; \cdots \tag{8}$$

PoH 有两项用途：（1）无需就时间达成共识即可提供确定性的领导者调度；（2）通过建立可验证的事件排序，减少共识消息轮次。

PoH 机制与 ACE-GF 身份层正交，不会引入额外的密码学假设。

4.2 ACE-GF 原生验证者身份

与传统区块链中验证者密钥独立生成和管理的做法不同，ACE Runtime 通过 ACE-GF HKDF 层级结构，从验证者的 REV 派生所有验证者密钥材料：

$$k_{validator} = \mathsf{HKDF}(\mathsf{REV},\ \texttt{"ACEGF-V1-VALIDATOR-CONSENSUS"},\ validator_id), \tag{9}$$

$$k_{attest} = \mathsf{HKDF}(\mathsf{REV},\ \texttt{"ACEGF-V1-MEMPOOL-ATTEST"},\ domain), \tag{10}$$

$$k_{vote} = \mathsf{HKDF}(\mathsf{REV},\ \texttt{"ACEGF-V1-VALIDATOR-VOTE"},\ epoch). \tag{11}$$

这一设计具备以下几项特性：

1. **统一密钥管理**。 所有密钥均派生自单一 REV，从而消除管理多组独立密钥对的运营负担。

2. **上下文隔离**。 k_{vote} 的泄露不会暴露 k_{attest} 或 $k_{validator}$，因为每一项均使用不同的 HKDF 信息字符串派生。

3. **基于纪元的轮换**。 投票密钥可在每个纪元进行轮换，而无需更改验证者的链上身份（id_com 保持不变）。

4. **后量子就绪**。 ML-DSA-44 流（Stream 7）自创世起即可用，从而在需要时无缝过渡到后量子共识签名。

4.3 领导者选举

时隙 S 的领导者使用 PoH 输出和纪元级随机种子确定性地选出：

$$\mathsf{Leader}(S) = \mathcal{V}\left[\mathsf{hash}\big(\mathsf{PoH}(S) \parallel epoch_seed\big) \bmod |\mathcal{V}_w|\right], \tag{12}$$

其中，$\mathcal{V}_w$ 是按权益加权的位置索引的验证者集合。这确保了确定性（所有验证者计算出相同的时间表）、不可预测性（取决于 PoH 输出）以及按权益加权的公平性。

4.4 BFT 投票与区块确认

区块确认遵循标准的 BFT 投票流程：

1. 领导者发布区块 B_N，其中包含交易和证明。

2. 每个验证者都会验证证明 Merkle 根，重新执行交易，并投出使用 k_{vote} 签名的票。

3. 收到 $\geq \frac{2}{3}$ 权益加权投票后，区块 B_N 即达到软最终性。

4. 最终性证书 FC_N 到达后，验证者验证一个 Groth16 证明（$\sim$0.5 毫秒，仅使用中央处理器），区块 B_N 即达到硬最终性。

投票消息数量为每位验证者 $O(1)$ 条（标准 BFT 复杂度）。ACE-GF 身份层不会带来额外的单次投票开销。

5 执行层

5.1 Sealevel 增强型并行执行

ACE Runtime 采用 Sealevel 并行执行模型 [5], ，开发者可在该模型中声明每笔交易访问的账户集合。运行时会构建依赖图，并行执行彼此不冲突的交易，同时串行执行存在冲突的交易。

ACE Runtime 通过 ACE-GF 上下文隔离（定理 2.2）增强基础 Sealevel 模型。当同一身份在不同上下文中操作多个金库（例如，`treasury:0` 和 `payroll:0`）时，每个上下文都会派生出在密码学上相互独立的密钥材料。如果每个上下文所访问的链上状态仅由该上下文的密钥材料派生（即不访问共享的全局账户），运行时便可保证跨上下文操作在协议层面彼此独立，从而在这些情况下无需进行运行时冲突检测。

命题 5.1（上下文级并行性）。对于同一身份在不同上下文 $c_1 \neq c_2$ 下发出的任意两笔交易 tx_1 与 tx_2，其派生的状态空间在密码学上相互独立。若两笔交易均不访问其上下文派生地址空间之外的共享全局状态，则这些交易可以并行执行，而无需进行冲突检查。

证明。根据定理 2.2，$\mathsf{HKDF}(\mathsf{REV}, \ info\ c_1)$ 与 $\mathsf{HKDF}(\mathsf{REV}, \ info\ c_2)$ 生成在计算上不可关联的密钥。由这些密钥派生的链上账户占据互不相交的地址空间。因此，在交易$_1$ 与 交易$_2$，之间不会产生读写冲突或写写冲突，并行执行是安全的。 □

示例。考虑一名用户同时操作财务金库（上下文 `treasury:0`）和薪资金库（上下文 `payroll:0`）。在 Solana 上，开发者必须显式声明互不重叠的账户集合，并由运行时在执行时加以验证。在 ACE Runtime 上，基于 HKDF 的上下文隔离为独立性提供了协议级保证，使运行时能够在无需冲突检测的情况下并行执行跨上下文交易，前提是两笔交易均不触及各自上下文派生地址空间之外的共享全局状态（例如系统账户）。

5.2 EVM 兼容层

ACE Runtime 面向 EVM 兼容执行层，使现有 Solidity 和 Vyper 智能合约能够以极少修改或无需修改的方式部署。支持标准 EVM 操作码和 Gas 语义；当前原型尚未涵盖所有预编译合约或边缘情况下的 Gas 计量。此外，ACE Runtime 还提供四种 ACE-GF 专用预编译合约：

- `0x0100`: id_com_verify(id_com π) $\rightarrow$bool — 验证身份承诺 pro of.
- `0x0101`: context_derive(元组) $\rightarrow$address — 派生上下文专用地址。
- `0x0102`: 管理员_因子_检查（路径_标识） $\rightarrow$ 状态 — 检查管理因子授权。

- 0x0103：zkace_批量_验证（π，承诺）$\rightarrow$布尔值 — 批量验证 ZK-ACE证明。

这些预编译合约使智能合约开发者能够在EVM合约中直接利用ACE-GF身份原语（上下文隔离、ZK身份验证和批量证明检查）。

6 安全分析

我们分析三个主要威胁向量以及ACE运行时的量子安全态势 E.

6.1 证明伪造抗性

定理6.1（证明不可伪造性）。 令 $\mathsf{Adv}^{\mathrm{forge}}{}_{\mathcal{A}}$ 表示任意概率多项式时间（*PPT*）对手$\mathcal{A}$针对未经REV持有者授权的交易生成有效证明的优势。在*HMAC-SHA256*的伪随机函数（*PRF*）假设以及*HKDF-SHA256*的单向性下，

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{forge}} \;\leq\; \epsilon_{\mathrm{PRF}} + \epsilon_{\mathrm{OW}} \;=\; \mathsf{negl}(\lambda),$$

其中 ϵ_{PRF} 是针对 *HMAC-SHA256* 的 *PRF* 优势， ϵ_{OW} 是针对 *HKDF-SHA256* 的单向性优势， λ 是安全参数。具体而言，对于 $\lambda = 256$ ，已知最佳经典攻击的上界为 $O(2^{-256})$ ，而已知最佳量子攻击（通过格罗弗算法）的上界为 $O(2^{-128})$。

证明。 证明凭证的计算方式为 凭证 =HMAC-SHA256(k_{attest}, m)，其中k_{attest} = HKDF(REV,信息,域)。不知道 REV 的对手必须采取以下方式之一：

1. **恢复** REV。 这需要求逆 Argon2id（封装）或 HKDF-SHA256（派生）。对于 256 位密钥，已知最佳经典攻击成本的上界为 2^{256} 次求值。 2. **恢复** k_{attest}。 由于 k_{attest} 是 256 位的 HKDF 输出，暴力搜索成本在经典情况下的上界为 2^{256} 次求值。格罗弗算法将其降至至多 2^{128}, ，这在计算上仍然不可行。 3. **直接伪造 HMAC**。 在 HMAC-SHA256 的 PRF 假设下，其成功优势至多为 ϵ_{PRF}，根据假设该值可忽略不计。

因此，该证明方案的安全性可归约为 HMAC-SHA256 的 PRF 安全性和 HKDF-SHA256 的单向性。请注意，这是一个对称密钥安全性归约，与 Ed25519 等签名方案基于离散对数的安全性不同；不过，两者在 128 位后量子安全级别上均可达到相当的具体安全强度。 □

6.2 缺失证明攻击

恶意构建者可能发布包含有效证明的区块，却未能在必需时间范围内提交相应的 Groth16 证明。

定义 6.2（两阶段证明超时协议）。 区块发布后，协议分两个阶段进行：

1. **构建器窗口**（K 个时隙， $K = 2$–3，即 800 ms–1.2 s）：当选的构建者必须发布有效的最终性证书。若其发布，则该区块正常最终确定。2. **备用窗口**（K' 个额外时隙， $K' = 2$–3）：若构建者失败，任意由$\lceil 2n/3 \rceil$ 名验证者组成的法定人数均可生成并发布替代终局性证书（见下文）。无论如何，构建者的全部质押都将被罚没。

若在 $K + K'$ 个时隙内（即最多 2.4 s）未发布有效的最终性证书，则该区块将被回滚，所有受影响的交易将重新排队。

定义 6.3 (见证可用性**)**。 如果在备用窗口开始时，至少有 B_S 名验证者通过见证可用性 gossip 协议收到了 中每笔交易的加密见证包，则区块 $\lceil 2n/3 \rceil$ 满足见证可用性。可形式化验证的实现方式（例如，Danksharding B_S 中的纠删码分发与数据可用性采样） [13]能够为这一条件提供密码学保证；具体协议留待未来研究。

命题 6.4 (缺失证明攻击的经济非理性**)**。在标准理性行为者模型下——也就是说，构建者是一个策略性单一主体，其收益仅由质押奖励、区块奖励和罚没惩罚决定，且没有与网络中断相关的外部经济头寸——故意扣留证明对任何持有正质押的构建者而言，都是严格劣势策略。

证明。 设 s 表示构建者的质押量， r 表示区块奖励。攻击成本为 s （全额罚没），而攻击收益至多是造成暂时的服务中断 $(K + K')$ ×400 毫秒 $\leq 2.4\text{s}$ （同时计入定理 6.2中的构建者窗口和备用窗口）。由于$s \gg r$ 这是质押协议设计所决定的，且中断时间有上限，因此在上述收益模型下，任何持有正质押的理性构建者都不会选择这一策略。

此外，备用证明机制还提供了次级活性保证。提交到内存池的每笔交易都附带一个加密见证包——即私有输入 w_{tx} ，由$\mathcal{C}_{\text{ZK-ACE}}$所需，并使用活跃验证者集合持有的门限公钥加密。这些数据包通过专用的见证可用性 *gossip* 协议进行传播。该协议与区块传播通道并行运行，但彼此独立；关键在于，加密见证数据不会包含在区块主体中，因此不会影响链上每笔交易的占用空间（244 B，见第7.4节）。

如果原始构建者未能在构建器窗口（K 个时隙）内生成证明，则备用窗口将开启（定理 6.2）。在此期间，任何由 $\lceil 2n/3 \rceil$ 名验证者组成的法定人数，只要通过 gossip 收到加密数据包，就可以协作对见证材料进行门限解密，并独立生成聚合 Groth16 证明，从而获得部分区块奖励。由于解密需要超多数法定人数，任何单个验证者或少数派联盟都无法访问原始见证；不过，执行解密的法定人数确实会看到明文见证，因此隐私保证属于委员会机密，而非对所有参与方都实现完全零知识。

因此，备用机制可确保活性，前提是定理 6.3 中的见证可用性条件 $\lceil 2n/3 \rceil$ 成立：即，验证者在备用窗口开始时已收到加密数据包。在此条件下，备用证明可在 K'个时隙的窗口内完成，且区块达到硬最终性而无需回滚。若不满足见证可用性，则系统将回退至定理 6.2中的回滚路径。 □

备注 6.5。上述分析未考虑具有外部动机的对手（例如，MEV 提取、跨市场空头头寸或竞争性破坏）或非理性行为者。要将经济安全论证扩展到此类情形，需要额外的机制设计假设（例如，MEV 销毁或提议者—构建者分离），这将留待未来研究。

6.3 量子安全分层

ACE Runtime 通过四个安全层，为抵御量子对手提供纵深防御：

1. **身份层**。 链上身份 id_com = Poseidon(REV,盐,域)是一个哈希原像问题。链上不暴露任何公钥，从而使“先收集，后解密”，

表 5：ACE Runtime 各层的量子安全态势。

层	原语	经典	量子
身份	Poseidon哈希	256位	128位（格罗弗）
证明	HMAC-SHA256	256位	128位（格罗弗）
证明	Groth16/BN254	128位	失效（Shor）
后量子密码模式	ML-DSA-44 + 零知识	128位	128位

表 6：每笔交易授权验证成本（代表性微基准测试和估算值）。

操作	CPU	图形处理器（批处理）	来源
Ed25519 验证（Solana）	∼76 μ秒	∼2 μ秒	[24]
ECDSA secp256k1 验证	∼50 μ秒	∼3 μ秒	[25]
ML-DSA-44 验证（后量子密码）	∼200 μ秒	N/A	[26]（约）
AttestCheck（ACE）	∼1–5 μ秒	不适用（中央处理器已足够）	章节 7.6
Groth16 验证（每个区块）	∼500 μ秒	N/A	[27]

稍后”攻击不再适用。量子计算机无法从 id_com 中提取 REV（哈希原像抗性在量子计算下仍然保持，格罗弗对256位原像至多提供平方根加速）。

2. **认证层。** HMAC-SHA256 在面对量子对手时仍保有 128位安全性（受格罗弗算法的平方根加速界限约束），这被认为在可预见的未来足够安全。

3. **证明层。** 基于 BN254 的 Groth16 并非 抗量子安全，因为基于配对的假设会被肖尔算法破解。不过，证明仅用作临时授权确认——不会泄露任何密钥材料，并且每个区块都会生成新的证明。迁移到抗量子SNARK（例如基于 STARK 的证明系统）的路径已明确，不需要更改身份层或认证层。

4. **后量子密码（PQC）模式。** ACE-GF 密钥派生层次的 Stream 7 从创世起便提供 ML-DSA-44 密钥 [11]。ZK-ACE 电路可以将 ML-DSA-44 验证（原生支持 ∼200 μs）压缩为相同的固定大小 Groth16 证明，而不会增加任何逐区块验证成本。待此类证明系统成熟后，即可实现完整的后量子密码操作（用基于格的 SNARK 替代 Groth16）。

7 定量评估

我们结合具有代表性的库级微基准测试与简单的分析模型，进行定量分析。除非明确标注为原型实测数据，本节中的数值均应视为基于模型的估算。第 7.6 节给出了 Rust 参考实现的实测数据，用于验证分析估算。

7.1 每笔交易验证成本

表 6 比较了不同密码学方案的每笔交易授权验证成本。

在上述模型中，AttestCheck 操作在中央处理器上的速度约为 Ed25519 验证的 15×–76× 倍。即使与 GPU 批处理的 Ed25519（∼2 μs）相比，基于 HMAC 的

表 7：基于模型的区块验证时间对比。无论区块大小如何，ACE Runtime 均提供恒定的$O(1)$验证。

区块大小	Solana（图形处理器）	Solana（中央处理器）	ACE（1 个 Groth16）加速比	
1,000 笔交易	2 ms	76 毫秒	**0.5 毫秒**	4×
10,000 笔交易	20 毫秒	760 毫秒	**0.5 毫秒**	40×
100,000 笔交易	200 毫秒	7,600 毫秒	**0.5 毫秒**	400×
1,000,000 交易	2,000 毫秒	76,000 毫秒	**0.5 毫秒**	4,000×

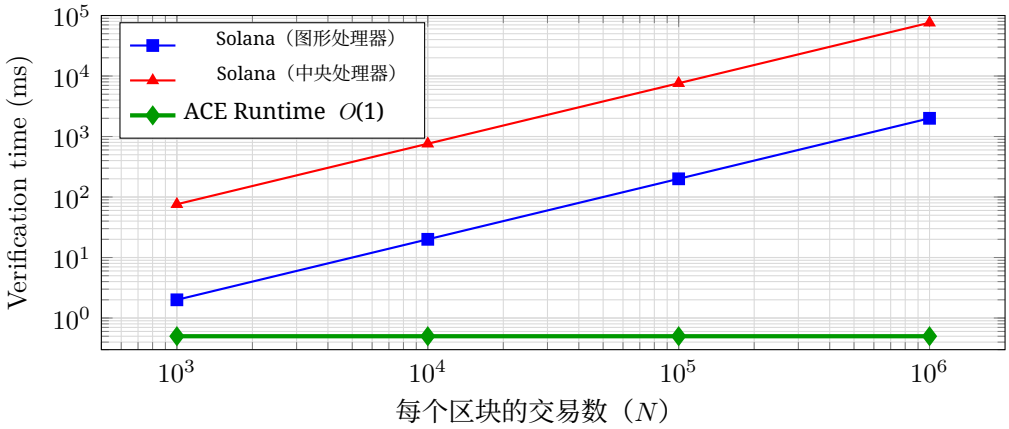

图 2：区块验证时间随区块大小的变化。Solana 呈线性扩展（$O(N)$），而 ACE Runtime 无论交易数多少，验证时间均保持恒定（$O(1)$），为 0.5 毫秒。

证明检查在该模型中也能在商用中央处理器上达到相当或更优的性能，且无需专用硬件。

7.2 区块验证时间：$O(N)$ 与 $O(1)$

ACE Runtime 最显著的性能优势 在于将区块验证从 $O(N)$ 转变为$O(1)$ 。表 7 展示了区块验证时间随区块大小变化的情况。

验证复杂度可概括如下：

$$T_{\text{Solana}}(N) = O(N) \cdot t_{\text{sig}}, \quad \text{linear growth with block size,} \tag{13}$$

$$T_{\text{ACE}}(N) = O(1) \cdot t_{\text{Groth16}} = 0.5 \text{ ms}, \quad \text{constant.} \tag{14}$$

7.3 链上授权数据

表 8 比较了每区块授权数据占用，凸显了 ACE Runtime 在后量子场景中的显著优势。

恒定大小的证明对后量子迁移尤为重要。从 Ed25519 切换到 ML-DSA-44 时，Solana 的授权数据将增加 $\sim$38× （每笔交易从 96 字节增至 3,732 字节）；而 ACE Runtime 无论零知识电路中使用何种底层签名方案，每个区块都能维持固定的 256 字节证明。

7.4 带宽受限的 TPS

我们分析网络带宽所设定的理论吞吐量上限：

表 8：每区块授权数据比较。ACE 将所有逐笔交易授权聚合为一个 256 字节证明。

区块大小	Solana（Ed25519）	Solana（ML-DSA-44）	ACE（ZK-ACE）	ACE 对比 Solana 后量子密码
1K 笔交易	96 KB	3,732 KB	**0.256 KB**	14,578× 更小
1 万笔交易	960 KB	37,320 KB	**0.256 KB**	145,781× 更小

$$\begin{aligned}\text{Solana per-tx:}\quad & \sim 154\text{ B (payload)} + 64\text{ B (sig)} \\ & + 32\text{ B (pubkey)} + \sim 982\text{ B (overhead)} \approx 1{,}232\text{ B}, \\ \text{ACE per-tx:}\quad & \sim 154\text{ B (payload)} + 90\text{ B (attestation)} = 244\text{ B}.\end{aligned}$$

在网络带宽为 1 Gbps 时：

$$\text{TPS}_{\text{Solana}} = \frac{125\text{ MB/s}}{1{,}232\text{ B}} \approx 101{,}000\text{ tx/s}, \tag{15}$$

$$\text{TPS}_{\text{ACE}} = \frac{125\text{ MB/s}}{244\text{ B}} \approx 512{,}000\text{ tx/s}. \tag{16}$$

ACE Runtime 通过消除每笔交易的签名和公钥，使带宽受限吞吐量上限 $\sim 5\times$ 更高，从而加快区块传播。

见证 Gossip 带宽开销。备份证明机制（第 6.2 节）引入了一个独立的见证可用性 Gossip 通道，用于传输加密见证包。每个见证包包含一笔交易的私有电路输入（对于 $\sim$1,400 约束的 C $_{\text{ZK-ACE}}$ 电路，估计每笔交易为 $\sim$200–400 B）。对于正常（构建者）证明路径，这些流量不在关键路径上：不会影响区块传播延迟，也不会影响 $\sim$600ms 的正常路径最终性时间。在故障恢复（备用）路径上，见证 Gossip 延迟会计入硬最终性上界 $\sim$2.4 s（定理 6.2）。见证包可通过后台 Gossip 在多个时隙内传播。即使在最坏情况下，所有见证包都与区块一同在单个时隙内传输，每笔交易的网络占用合计也仅为 $244 + 400 = 644$ B，仍比 Solana 的 1,232 B 少 $\sim 1.9\times$ 。因此，上述 $5\times$ 优势指的是仅区块传播界限；考虑 Gossip 调度后，系统总带宽优势保守估计为 $\sim 1.9\times$–$5\times$ 。

在标准硬件上采用保守工程设计时，可实现的吞吐量为 $\sim$16,000 TPS，通过适度优化（8 路并行执行分片）可扩展至 $\sim$32,000 TPS。

7.5 硬件成本对比

表 9 比较了不同节点角色的硬件要求与成本。

取消非构建者验证器的GPU要求可将硬件成本降低约 $\sim$30%–50%，并显著降低验证者参与的经济门槛，从而促进更加去中心化的网络。

7.6 参考实现测量

为验证上述分析模型，我们实现了 ACE Runtime 流水线的 Rust 参考原型，并使用 Criterion 统计基准测试框架 [23] 在 Apple M3 Pro 中央处理器（12 个核心、36GB 内存）上测量其性能。除非另有说明，所有报告的延迟均为 Criterion 默认配置下的样本中位数（p50）（每个基准测试 100

表9：按角色划分的硬件要求和估算成本。

组件 Solana	验证-tor	ACE验证器（非构建者）	ACE构建者
CPU	24+ 个核心（~$2,000）	16+ 个核心（~$1,500）	24+ 个核心（~$2,000）
RAM	512 GB+ (~$3,000)	256–512 GB (~$1,500–3,000)	512 GB+ (~$3,000)
GPU	**必需费用** (~$2,000)	**Not 需要 ($0)**	必需费用 (~$2,000)
总计 每月运营费用	~$7,500+ $1,000–3,000	~$3,500–5,000 $600–2,000	~$7,500 $1,000–3,000

表 10：实测密码学原语延迟（Apple M3 Pro，单线程；除非另有说明，否则为 100 个样本的样本中位数）。

操作	延迟
证明生成（HMAC-SHA256）	2.27 μ秒
证明验证（完整检查）	830 纳秒
载荷绑定检查（SHA-256）	334 纳秒
HKDF 单密钥派生	1.18 μ秒
HKDF 证明密钥派生	1.22 μ秒
HKDF 全部 7 条规范流	7.87 μ秒
身份承诺（SHA-256）	272 纳秒
SHA-256 哈希（输入 244 B）	544 纳秒
REV 封装（Argon2id）*	3.57 毫秒

* 10 个样本（Argon2id 故意设计得较慢）。

个样本，每个基准测试的测量窗口为 5 秒）。阶段 1a 和阶段 1b 使用 Rayon 工作窃取并行；当前的模拟阶段 2 实现采用顺序执行。单操作基准测试均为单线程运行。基准测试框架（`benches/crypto_bench.rs`、 `benches/pipeline_bench.rs`）包含在参考实现中，可通过`cargo bench`复现。

代码可用性。参考实现可公开获取，地址为 https://github.com/ya-xyz/ace-chain/tree/master/ace-runtime。为确保结果可复现，本节的测量数据对应以下固定版本： https://github.com/ya-xyz/ace-chain/tree/b805ea90a8c3ab1ce5fa62cf842dccfb9452fc9f/ace-runtime。除 `cargo bench` 外，还可通过 `cargo run -release-bin prototype_bench - 50`运行端到端原型运行器。

密码学微基准测试。表10 汇总了核心密码原语各项操作的测量延迟。

每笔交易测得的证明生成时延为 ~2.3 μ秒，证实了表 6 中 5 μ秒 的估算。完整的证明验证（载荷绑定SHA-256哈希校验 +HMAC-SHA256凭证重计算）在 ~830纳秒内完成。流水线的阶段 1a AttestCheck 执行较轻量的结构检查（载荷绑定、域匹配、身份注册表查找），无需重新计算HMAC凭证；在批大小为2,000时，其摊销结果为 ~129纳秒/交易（表 11）。这两个数值均表明，AttestCheck 60×–90× 比Ed25519验证更快

表 11：测得的流水线阶段耗时（Rayon 并行，Apple M3 Pro）。阶段 2 使用模拟证明器；实际的 Groth16 证明受 GPU 限制，因此未包含在这些 CPU 测量中。

阶段	100 笔交易	500 笔交易	2,000 笔交易	2K 时每笔交易
1a：AttestCheck†	60.6 μ秒	117.8 μ秒	258.3 μ秒	129 纳秒
1b：执行†	55.1 μ秒	81.6 μ秒	141.9 μ秒	71 纳秒
1b：状态根（默克尔）	28.2 μ秒	135.1 μ秒	538.4 μ秒	269 纳秒
2：证明（模拟）	214.3 μ秒	1.09 毫秒	4.19 毫秒	2.10 μ秒
区块构建	101.6 μ秒	515.9 μ秒	2.02 毫秒	1.01 μs
端到端	**521** μs	**1.95** 毫秒	**7.69** 毫秒	**3.85** μs

†Rayon 工作窃取并行。

表12：定位矩阵：ACE Runtime 与现有区块链平台的比较。

平台	签名验证	区块验证	硬最终性	后量子密码路径	GPU要求	层
Solana [5]	$O(N)$ 图形处理器	$O(N)$	~12 秒	None	所有节点	L1
Aptos/Sui [17, 18]	$O(N)$ 中央处理器	$O(N)$	~2 s	None	None	L1
以太坊 [12]	$O(N)$ 中央处理器	$O(N)$	~15 分钟	以太坊改进提案已规划	None	L1
zkSync [14]	$O(N) \to O(1)$	$O(1)$	继承第一层	继承第一层	证明者	L2
StarkNet [15]	$O(N) \to O(1)$	$O(1)$	继承第一层	原生 STARK	证明者	L2
Mina [16]	$O(N)$	$O(1)$*	~15 分钟	None	递归式	L1
ACE Runtime	**O(1)**/区块	**O(1)**	~600 毫秒	**第 1 天**	**仅限构建者**	**L1**

*Mina 的 $O(1)$ 区块级授权 指的是链级状态证明，而非逐区块授权验证。ACE Runtime 提供 s 验证，并在第一层实现亚秒级硬最终性。

在中央处理器上。

流水线阶段耗时。 表 11 报告了批大小为100、500和2,000笔交易时端到端流水线各阶段的耗时。阶段1a和阶段1b通过Rayon并行执行；阶段 2使用模拟证明器，通过基于哈希的证明执行树形聚合（在GPU 硬件上进行实际的Groth16证明时，证明过程将占据阶段 2的主要成本）。

在 2,000 笔交易的情况下，阶段 1a（AttestCheck）的每笔交易摊销成本为~129纳秒，这表明该原型具有良好的并行扩展性（单线程基线对比留待未来工作）。完整的中央处理器端流水线（不包括受 GPU 限制的 Groth16 证明）在 8 毫秒内完成，证实中央处理器阶段并非第 4 节所假设的 400 毫秒时隙预算中的瓶颈。

对生产吞吐量的影响。按测得的端到端中央处理器成本3.85 μ秒 每笔交易（2,000 笔交易的批次）计算，单个流水线实例仅凭中央处理器即可处理 ~260,000 笔交易/秒。由于这些测量中阶段 2 使用的是模拟证明器，我们 推断 预期的生产瓶颈是图形处理器上的 Groth16 证明，而非中央处理器流水线——这支持了将证明卸载给专用构建者角色的设计决策（第 3.3 节）。要验证这一推断，需要与生产级 Groth16 后端（如 arkworks 或 Rapidsnark）集成，这留待未来工作。

8 相关工作

表12 对ACE Runtime与现有区块链平台在关键维度上进行了全面比较。

高性能第一层区块链。 Solana [5] 开创了基于 PoH 的高吞吐量架构，区块时间为 400 毫秒。然而，其硬最终性需要 31 次确认（~12秒），且每位验证者都必须运行 GPU 硬件来执行 SigVerify。Aptos [17] 和 Sui [18]通过基于 DAG 的 BFT 共识实现更快的最终性，但仍需对每笔交易进行 $O(N)$ 验证。ACE Runtime 实现了与 Solana 相同的区块时间，同时将硬最终性时间从 12 秒缩短至 600 毫秒，并消除了普遍的 GPU 要求。

ZK Rollup 和 L2 解决方案。 zkSync [14]和 StarkNet [15]采用零知识证明进行状态压缩和 Rollup 验证，实现了 $O(1)$ 对第二层交易的状态验证。然而，在 Rollup 排序器内部，仍需在 $O(N)$ 中验证各笔交易的签名，且最终性取决于底层 Layer-1 结算。ACE Runtime 在主权 Layer-1 链的身份与授权层应用零知识证明，实现 $O(1)$ 验证，且无需承受 Layer-1 最终性延迟。

递归证明链。 Mina Protocol [16] 采用递归 SNARK 组合，以维持固定大小的链状态证明。这解决了链同步问题，但并未降低逐区块授权验证的成本，该成本仍为 $O(N)$。ACE Runtime 通过区块级 $O(1)$ 授权验证，补充了链级压缩。

身份层创新。 去中心化身份（DID）框架和自我主权身份（SSI）协议 [20] 探索了替代性的身份模型，但尚无任何一种被整合进区块链执行层，以消除逐笔交易签名验证。ACE-GF 框架 [1] 提供了使 ACE Runtime 成为可能的密码学基础。

9 局限性与未来工作

我们承认当前 ACE Runtime 设计存在若干局限性，并概述未来研究的方向。

执行受限吞吐量。 在没有状态分片的情况下，ACE Runtime 的执行受限 TPS 与 Solana 相当（标准硬件上为 ~16,000–32,000 TPS）。授权层优化消除了一个瓶颈，但并不能从本质上扩展状态 I/O 层。利用 HKDF 上下文隔离属性的基于上下文的状态分片，是一篇关于原生 ACE-GF 状态分区的配套论文的主题 [21]。

Groth16 量子脆弱性。 基于 BN254 的 Groth16 证明系统并不具备量子安全性。虽然证明的短暂性（每个区块重新生成）限制了该脆弱性的影响，但全面过渡到抗量子证明系统（例如基于 STARK 的系统或基于格的 SNARK）仍然是可取的。零成本 PQC 迁移路径在一篇配套论文中有详细阐述 [22]。

生态系统启动。 作为新的 Layer-1 平台，ACE Runtime 需要从零开始构建开发者工具、钱包基础设施和应用生态系统。EVM 兼容层（第 5.2节）通过支持部署现有 Solidity 合约来缓解这一问题，但 ACE-GF原生应用需要新的开发模式。

证明者市场设计。 当前设计假定区块构建者同时担任证明生成器。在去中心化证明者市场中，多个相互竞争的证明者竞价生成最终性证书，这将改善活性保证并降低中心化风险。我们将此类市场的机制设计留待未来工作。

可信设置。 Groth16 需要电路专用可信设置仪式。虽然存在通用设置替代方案（例如 PLONK [19]），但它们会带来更大的证明大小或更高的验证成本。我们计划在未来的工程报告中评估设置可信假设与验证效率之间的权衡。

10 结论

本文介绍了 ACE Runtime，这是一种利用身份–授权分离来重构交易处理流水线的区块链运行时。ACE Runtime 以轻量级 HMAC 证明取代逐笔交易密码学签名，并将授权证明聚合为单次逐区块 Groth16 验证。分析建模和形式化分析表明，与当前最先进的方案相比，ACE Runtime 有望带来多项改进：

1. 亚秒级硬最终性。设计上可在 $\sim$600ms 内实现密码学最终性；相比之下，Solana 为 $\sim$12s（$\sim$20×），以太坊为 $\sim$15 分钟（$\sim$1,500×）。需要注意的是，这些系统采用了不同的最终性定义（参见表 4）。2. $O(1)$ 区块验证。无论区块大小如何，区块授权验证的成本都保持恒定；基于模型的预测显示，相较于 Solana 的 $O(N)$ SigVerify，规模化后加速比最高可达 4,000× 。3. 无 GPU 验证者。非构建者验证者无需 GPU 硬件，可降低 30%–50% 的硬件成本，并降低参与验证的门槛。4. 后量子就绪。身份层和认证层从一开始就具备抗量子性，而零知识压缩架构确保采用后量子签名方案不会产生逐区块验证开销。5. 带宽效率。移除逐笔交易签名和公钥后，链上逐笔交易数据减少 $\sim$5× （仅计算区块传播）。若将备用证明所需的链下见证 Gossip 开销计入，总系统带宽优势为$\sim$1.9 ×–5× ，具体取决于 Gossip 调度（第 7.4 节）。

身份—授权分离是区块链设计中一个常被忽视的架构维度。ACE Runtime 表明，将这种分离与现代零知识证明系统相结合，可在最终确定性、验证效率、硬件可及性和抗量子韧性方面带来数量级的提升。这些优势源自协议层设计，而非实现层优化，表明身份—授权分离是下一代区块链系统的一项基本架构原则。

致谢

作者感谢 ACE-GF 研究社区就身份—授权分离和零知识证明系统设计所进行的讨论。

参考文献

[1] J. Wang，“ACE-GF：一种用于原子密码实体的生成式框架”，*arXiv* 预印本 *arXiv:2511.20505*，2025。 https://arxiv.org/abs/2511.20505

[2] J . Wang, “ZK-ACE: Identity-Centric Zero-Knowledge Authorization for Post-Q 量子区块链系统”，arXiv预印本 arXiv:2603.07974，2026年。 https://arxiv.org/abs/2603.07974[3] J. Wang，“基于 ACE-GF 的后量子密码证明中继：无需路径上证明的轻量级内存池传播”，arXiv预印本 arXiv:2603.07982，2026年。 https://arxiv.org/abs/2603.07982[4] J. Wang，“通过休眠授权路径实现条件触发的加密资产控制”，arXiv预印本 arXiv:2603.07933，2026年。 https://arxiv.org/abs/2603.07933[5] A. Yakovenko，“Solana：一种高性能区块链的新架构”，Solana Labs白皮书，2018年。 https://solana.com/solana-whitepaper.pdf[6] J. Groth，“关于基于配对的非交互式论证的大小”，载于《密码学进展——EUROCRYPT 2016》，施普林格，第305–326页，2016年。[7] H. Krawczyk和P. Eronen，“基于HMAC的提取扩展密钥派生函数（HKDF）”，RFC 5869，互联网工程任务组，2010年5月。[8] P. S. L. M. Barreto和M. Naehrig，“素数阶配对友好椭圆曲线”，载于《密码学精选领域——SAC 2005》，施普林格，第319–331页，2006年。[9] L. Grassi、D. Khovratovich、C. Rechberger、A. Roy和M. Schofnegger，“Poseidon：一种面向零知识证明系统的新型哈希函数”，载于USENIX安全研讨会，2021年。[10] A. Biryukov、D. Dinu和D. Khovratovich，“Argon2：面向口令哈希及其他应用的新一代内存困难函数”，载于IEEE欧洲安全与隐私研讨会（EuroS&P），2016年。[11] 美国国家标准与技术研究院，“后量子密码学：数字签名方案”，NIST FIPS 204（ML-DSA），2024年。[12] V. Buterin等，“以太坊：新一代智能合约和去中心化应用平台”，以太坊白皮书，2014年。[13] V. Buterin、D. Feist、D. Loerakker、G. Kadianakis、M. Garnett、M. Reitwiessner和A. Beregszaszi，“EIP-4844：分片数据块交易”，以太坊改进提案，2022年。 https://eips.ethereum.org/EIPS/eip-4844[14] Matter Labs，“zkSync：以太坊扩容与隐私引擎”，技术报告，2020年。 https://zksync.io[15] StarkWare，“StarkNet：无需许可的去中心化 ZK-Rollup”，技术报告，2021年。 https://starknet.io[16] J. Bonneau、I. Meckler、V. Shapiro和E. Srinivasan，“Mina：大规模去中心化加密货币”，Mina Protocol白皮书，2020年。[17] Aptos Labs，“Aptos区块链：安全、可扩展、可升级的Web3基础设施”，技术报告，2022年。[18] Mysten Labs，“Sui：具备高吞吐量、低延迟和资产导向编程模型的新一代智能合约平台”，技术报告，2022年。

[19] A. Gabizon、Z. J. Williamson和O. Ciobotaru，“PLONK：用于通用非交互式知识论证的拉格朗日基置换”，*IACR ePrint*档案，报告编号2019/953，2019年。[20] M. Sporny、D. Longley、M. Sabadello、D. Reed、O. Steele和C. Allen，“去中心化标识符（DID）v1.0”，W3C建议标准，2022年7月。[21] J. Wang，“上下文分片状态：利用身份—授权分离实现原生区块链状态划分”，编写中，2026年。[22] J. Wang，“知识系统化：区块链系统中的后量子密码学迁移——挑战、方法与零成本范式”，编写中，2026年。[23] B. Heisler，“Criterion.rs——Rust中以统计数据为驱动的微基准测试”，https://bheisler.github.io/criterion.rs/book/，2024年。[24] I. A. Lovecruft和H. de Valence，“ed25519-dalek：Rust中快速且正确的Ed25519签名与验证”，https://github.com/dalek-cryptography/ed25519-dalek，2023年。[25] P. Wuille等，“libsecp256k1：用于secp256k1曲线上的ECDSA签名及密钥操作的优化C语言库”，https://github.com/bitcoin-core/secp256k1，2024年。[26] D. Stebila和M. Mosca，“Open Quantum Safe”，https://openquantumsafe.org/，2024年。[27] arkworks贡献者，“arkworks：用于开发和编写zkSNARK程序的生态系统”，https://arkworks.rs/，2024年。[28] V. Buterin和V. Griffith，“Casper友好最终性小工具”，*arXiv*预印本 *arXiv:1710.09437*，2017年。

最终性时间线比较

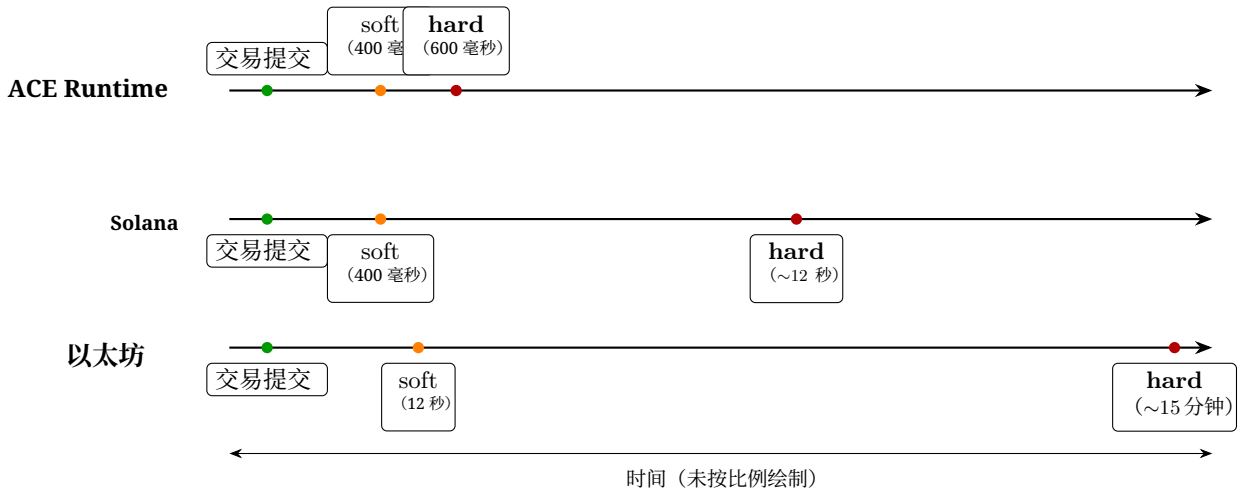

图 3：最终性时间线对比。ACE Runtime 的目标是实现 ~600 毫秒硬最终性，而 Solana 需要 ~12 秒，以太坊则需要 ~15 分钟。

B ZK-ACE 电路约束细分

表 13：ZK-ACE 电路约束细分。

约束组	描述	约计数量
身份绑定	Poseidon(REV，盐值，域) $=$ id_com	$\sim$400
密钥派生	HKDF(REV，信息，域) $= k_{attest}$	$\sim$500
证明检查	HMAC(k_{attest}，m) $=$ 凭证	$\sim$400
策略（可选）	目标 $\in$允许_集合	$\sim$100
总计		**$\sim$1,400**

证明大小：256 字节（基于 BN254 的 Groth16）。验证时间： $\sim$0.5 毫秒（单次 Groth16 验证，3 次双线性配对运算）。

C ACEBlock 和最终性证书格式

表 14：ACE 区块头字段。

字段	Type	大小（字节）
`slot_number`	`u64`	8
`parent_hash`	`[u8; 32]`	32
`state_root`	`[u8; 32]`	32
`tx_merkle_root`	`[u8; 32]`	32
`attest_merkle_root`	`[u8; 32]`	32
`poh_hash`	`[u8; 32]`	32
`leader_id_com`	`[u8; 32]`	32
`timestamp`	`u64`	8
`tx_count`	`u32`	4
区块头总计		**$\sim$256**

表 15：最终性证书格式（固定大小，与区块大小无关）。

字段	Type	大小（字节）
`block_hash`	`[u8; 32]`	32
`slot_number`	`u64`	8
`groth16_proof`	`[u8; 256]`	256
`public_inputs`	`Commitment([u8; 32])`	32
总计		**328**