

密码学驱动的确定性逻辑时间轴

——面向机器协同与 AI 审计的时序基准

上海华湃特技术有限公司

邱勇 王哲浩

通讯作者：邱勇

邮箱：qiuyong@huapait.com.cn

2026 年 10 月

摘要

移动通信正从“人的通信”走向“机器的通信”。6G 核心场景对通信提出确定性时延、内生安全、零信令协同等全新要求，而现有系统“先对齐物理时钟，再通过信令交互分配资源”的底层逻辑遭遇根本性瓶颈。本文提出一种密码学驱动的确定性逻辑时间轴，由动态安全根基（DSF）驱动协议安全状态机（PSSM），在各节点独立演进逻辑状态 $S(n)$ ，并在逻辑决策时刻采样物理锚点 $T_{anchor}(n)$ ，形成逻辑-物理双锚定。行为记录通过密码学哈希链链接，验证端仅需持有相同 DSF 即可独立重算 $S(n)$ 、验证行为链完整性并追溯物理时间位置。本文系统阐述物理时间不可逆、物理时钟不可信、第三方存证不可独立验证三大根本困境，给出安全模型、与 Lamport 时钟/区块链存证/TEE 等方案的本质区别，并以 AI 调用审计为例验证其可行性。

关键词：密码学驱动的逻辑时间轴；动态安全根基；协议安全状态机；逻辑-物理双锚定；AI 行为审计；密码学时序基准

1 引言

1.1 一个未被认识到的范式转移

移动通信正经历一次根本性的范式转移。5G 及之前的系统，核心服务对象是“人”——手机上网、视频通话、移动支付。6G 的核心服务对象正在转向“机器”——工业控制器、自动驾

驶车辆、无人机蜂群、卫星星座、AI Agent。

这两类通信对象对网络的要求截然不同：

需求维度	人的通信	机器的通信
时延	尽量低，统计性满足即可	必须有数学可证明的上界
可靠性	99.9% 可接受	99.9999% 以上，长尾不可接受
同步	依赖 GPS/1588	不能依赖外部授时
信令	信令交互可接受	信令开销必须趋零
安全	外挂加密可接受	内生安全，物理层保护
组网	依赖基站基础设施	无基础设施自组织
移动性	切换中断数十毫秒可接受	零中断，UE 无感
功耗	可接受	极低功耗，长待机

这一转移的深层含义尚未被充分认识：**通信系统的底层逻辑本身需要被重新审视**。现有系统遵循一条不变的底层逻辑——先对齐物理时钟，再开展通信交互。物理时钟是系统的“心跳”，所有时序计算绑定于物理帧号。这一范式在人的通信场景下运行良好，但在机器通信场景下遭遇根本性瓶颈。

1.2 三个根本困境

困境一：物理时间不可逆。

事故发生后，时钟无法回拨，环境无法复现。传统方案只能依赖“存证”——把当时的数据记录下来，相信存证未被篡改。但“相信”不等于“证明”。日志记录只能证明“有人声称发生了某件事”，无法证明“某件事确实发生了”。区块链存证只能证明“数据在某个时刻被写入”，无法证明“数据在当时是正确的”。这一困境的根源在于：物理世界不可重放——无法回到“当时”去重新验证行为是否正确执行。

困境二：物理时钟不可信。

即使通过 GNSS 或网络参考时钟统一了物理时间，物理时钟仍具有可篡改、可欺骗、跨域不可独立验证、不可重放的固有局限。具体表现为：

- 模型签名方案：即使加上时间戳，该时间戳来自本地物理时钟，可被篡改，验证端无法独立重算；
- TEE 方案：认证报告是单节点产物，跨域时不同节点的报告时间戳无法直接对齐；
- ZKML 方案：证明计算正确性，但证明本身不含时间信息，无法回答何时、何模型、由谁执行；
- 运行时回执方案：回执中的时间戳来自本地物理时钟，可被篡改，且回执无法回溯模型身份；
- 统计检测方案：签名的对象包含时间戳的话仍是物理时间戳，不包含的话无法回答何时。

困境三：第三方存证不可独立验证。

现有方案——GPS 时间戳、NTP/PTP 同步、区块链存证、TEE 远程认证——均隐含假设验证者必须信任某个外部实体：GPS 卫星、时间服务器、区块链共识节点、硬件厂商。审计方必须信任存证平台的可信度，无法独立验证。在跨机构、跨信任域、跨物理时钟域场景中，这些假设全部失效。

1.3 本文的贡献

本文提出一种密码学驱动的确定性逻辑时间轴，作为机器协同与 AI 审计的统一时序基准。本文的核心贡献包括：

1. **问题定义**：系统阐述物理时间不可逆、物理时钟不可信、第三方存证不可独立验证三大根本困境，揭示现有方案在机器通信与 AI 审计场景下的共同失效根源。
2. **机制提出**：提出密码学驱动的动态安全根基（DSF）三元组、协议安全状态机（PSSM）、逻辑-物理双锚定、行为链密码学链接、候选集验证等核心机制，构建与物理时钟解耦、可独立验证、无限次重放的确定性逻辑时序基准。
3. **边界划定**：系统对比本方案与 Lamport 时钟、NTP/PTP、GPS、区块链存证、TEE、ZKML、Inference Receipts 等现有技术的本质区别，明确本方案的独特定位。
4. **安全模型**：给出不可伪造、可独立验证、不可篡改、前向/后向安全、密钥泄露分层控制等安全性质的形式化定义与分析。
5. **最小验证**：以 AI 调用审计为例，展示“逻辑状态重算 + 行为链完整性 + 物理锚点追溯”的完整闭环，并通过三节点工程演示验证严格同步与有界偏差两种模式。

2 相关工作与现有技术的根本局限

2.1 物理时间基准方案

GPS/GNSS/原子钟：蜂窝网络、电力系统、金融交易系统采用此方案。每个节点通过 GPS 接收机获取卫星原子钟提供的绝对物理时间，将行为记录打上物理时间戳后存储于日志系统。根本局限：依赖持续的外部授时信号，在室内、地下、水下、深空、强干扰环境下不可用；GPS 信号易受欺骗和干扰，存在单点失效风险；日志系统存在被具有管理员权限的用户篡改的风险。

IEEE 1588 / NTP：通过报文交换测量传播时延，补偿时钟偏差。该方案依赖“主从架构”且隐含假设所有节点的物理时钟遵循相同的计量规则，本质上是在“修正偏差”而非“建立新基准”；中心化存证平台存在单点篡改风险，审计方必须信任存证平台的可信度，无法独立验证。

2.2 逻辑时钟方案

Lamport 时钟：Lamport 于 1978 年提出，以及后续的向量时钟方案，在分布式系统中用于建立事件的偏序关系（“A 发生于 B 之前”）。核心特征：与物理时钟完全无关的单调递增软

件计数器，通过事件驱动和消息传递来更新，仅在节点间建立事件的因果顺序。根本局限：不记录行为的内容，不记录行为的输入输出，不产生任何形式的可验证证据链，更不具备审计或重放功能。现有技术中虽然有人将 Lamport 时钟用于日志合并或审计轨迹构建，但仅是利用其“排序”能力，既不绑定物理锚点，也不记录行为上下文。

混合逻辑时钟（HLC）：研究了有界偏差下的时间戳方案，解决时间戳的因果性与物理时间邻近性，但不涉及密码学承诺验证。

2.3 存证与审计方案

区块链存证：民航局已发布《基于区块链的民用无人驾驶航空器飞行数据存证技术要求》（MH/T 2015-2024），要求飞行数据通过区块链上链存证。根本局限：本质仍然是“外部存证”——数据在产生后被上传至区块链或中心化平台，其可信度依赖于联盟链节点的诚实性和共识机制；审计方必须信任存证服务商，无法独立验证数据的真实性；区块链存证必须事后上链，无法在行为发生时实时绑定，存在时间窗口内的篡改风险。

Inference Receipts / AI Integrity Receipts：在 AI 推理完成后生成轻量级密码学承诺记录，将模型身份、采样配置和输出 Token 绑定到一个防篡改的哈希链中。根本局限：事后验证而非事前控制——收据在推理完成之后生成，只能证明“某时某刻产生了某结果”，无法保证推理过程本身的确定性；不涉及熵源控制；明确承认不保证跨硬件或跨量化方式的通用确定性重放；安全模型建立在“诚实发射者”假设之上。

TEE（可信执行环境）：利用 Intel SGX、NVIDIA H100 机密计算等硬件能力，在隔离环境中运行模型并生成远程认证报告。根本局限：依赖特定硬件，存在侧信道攻击风险；每次验证只针对当前加载的模型，认证报告是孤立的，无法在跨节点、跨时钟域的分布式系统中提供一个统一的、不可篡改的时间基准来绑定模型身份和执行时刻。

ZKML（零知识机器学习）：将模型推理转化为算术电路并生成零知识证明。根本局限：计算开销极大，为大型模型生成零知识证明的计算成本是原始推理的 1000 倍到 10000 倍，目前仅能对极小的模型或推理的切片进行验证；只能证明计算正确性，无法提供行为模式的历史审计能力。

2.4 共同困境：物理时间不可逆

所有现有方案共享一个根本性的困境：**物理时间是不可逆的。**

事故发生后，时钟无法回拨，环境无法复现。传统方案只能依赖“存证”——把当时的数据记录下来，相信存证未被篡改。但“相信”不等于“证明”。日志记录只能证明“有人声称发生了某件事”，无法证明“某件事确实发生了”。区块链存证只能证明“数据在某个时刻被写入”，无法证明“数据在当时是正确的”。

这一困境的根源在于：**物理世界不可重放**——无法回到“当时”去重新验证行为是否正确执行。

3 密码学驱动的确定性逻辑时间轴：核心机制

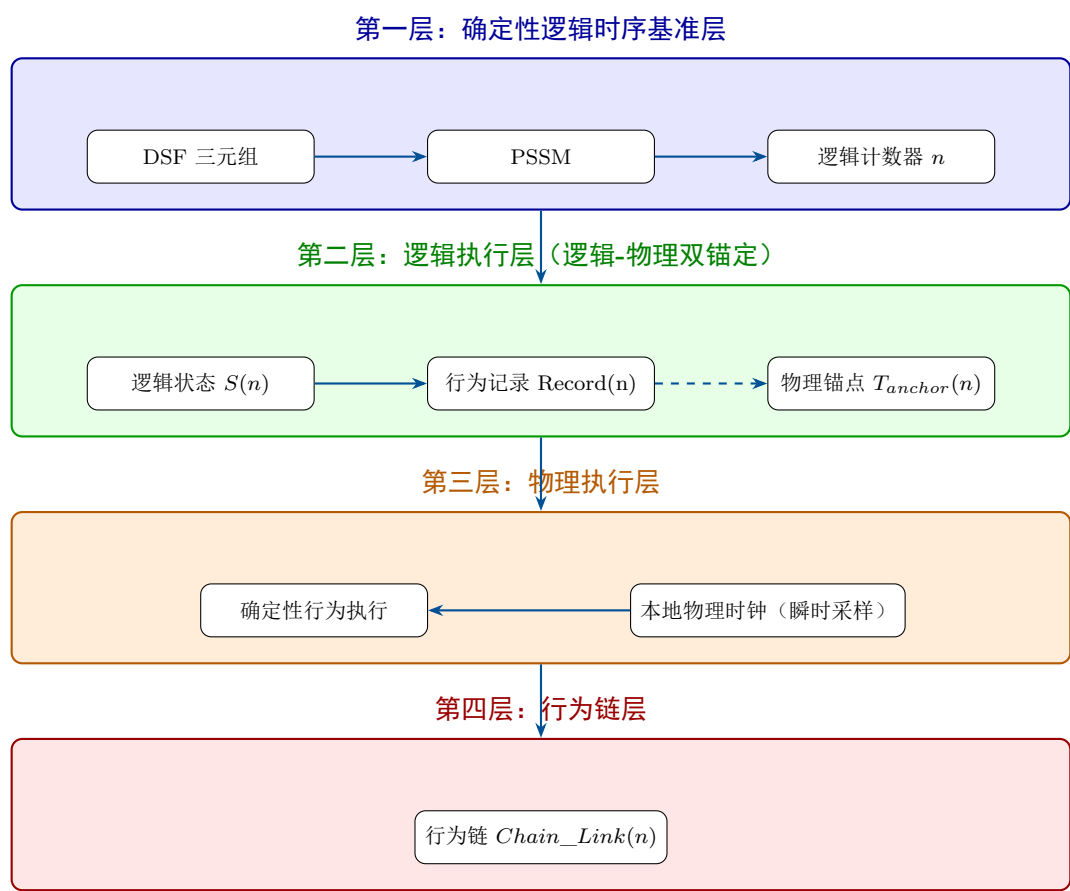


图 1: 确定性逻辑时间轴的四层架构与逻辑-物理双锚定设计

3.1 设计哲学

DSF 的设计哲学可以概括为三句话：

能计算的不传输，能确定的不随机，能逻辑的不物理。

原则	含义	对传统方案的颠覆
能计算的不传输	任何可通过共享信息计算得出的参数，绝不在空口传输	消除信令开销
能确定的不随机	将不可避免的碰撞转化为可预测的排队过程	消除统计性长尾
能逻辑的不物理	逻辑时间轴与物理时钟解耦，用密码学确定性对抗物理不完美	消除物理时钟依赖

3.2 从“物理时钟的奴隶”到“逻辑时间的主人”

传统通信系统遵循一条不变的底层逻辑：**先对齐物理时钟，再开展通信交互**。物理时钟是系统的“心跳”，所有时序计算绑定于物理帧号（SFN）。

DSF 跳出这个范式，将通信的时间维度分解为两个独立层次：

时间轴	定义	特点
逻辑时间轴	由逻辑计数器 n 和逻辑周期 T_{step} 定义，由 PSSM 驱动	与物理时钟解耦，按密码学规则独立演进
物理时间轴	由各节点本地物理时钟定义	仅在逻辑决策时刻被瞬时采样，用于物理执行

两个时间轴之间的关系是**锚定**，而非**驱动**。物理时钟提供执行时刻的参考，但不定义逻辑状态的内容。

关键认识：物理时钟是节拍源（Tempo Source），不是同步源（Sync Source）。它决定“多久走一步”，不决定“走到哪一步”。逻辑时间轴的内容由**密码学规则**独立决定。

3.3 DSF 三元组

DSF 由三元组定义：

$$DSF = (K_{sec}, Init_Anchor, Rule_ID)$$

要素	含义
K_{sec}	共享安全密钥，256 位，通过安全方式预共享，存储于硬件安全模块中
$Init_Anchor$	初始锚点，定义逻辑时间轴起点及其对应的物理启动锚点
$Rule_ID$	状态转移规则标识，定义逻辑节拍 T_{step} 、每个 epoch 包含的逻辑节拍数 N_{epoch} 、状态演进函数、以及安全窗口参数

3.4 协议安全状态机（PSSM）

通信双方基于共享 DSF 独立运行 PSSM：

$$S(0) = H(K_{sec} || Init_Anchor || "INIT")$$

$$K_e = H(K_{sec} || e || "SUBKEY"), \quad e = \lfloor n / N_{epoch} \rfloor$$

$$S(n+1) = H(K_e \| S(n) \| \text{"CHAIN_TAG"})$$

其中 H 为密码学哈希函数（如 SHA-256、SHA-3、BLAKE2、SM3）。

PSSM 的状态 $S(n)$ 具有以下性质：

性质	说明
不可伪造	不掌握 K_{sec} 的第三方无法预测或伪造 $S(n)$
可独立验证	任何持有相同 K_{sec} 的节点可独立重算 $S(n)$
不可篡改	一旦 $S(n)$ 被记录，任何修改都会导致后续哈希链断裂
前向安全	即使 $S(n)$ 被破解，也无法推知 $S(n+1)$
后向安全	即使设备被物理捕获，也无法恢复历史 $S(k)$ ($k < n$)

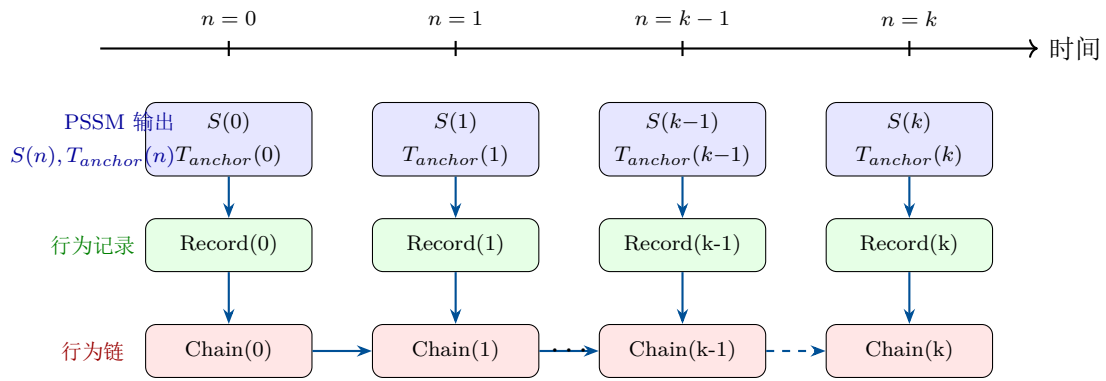


图 2: PSSM 状态演进与逻辑-物理双锚定时序示意图

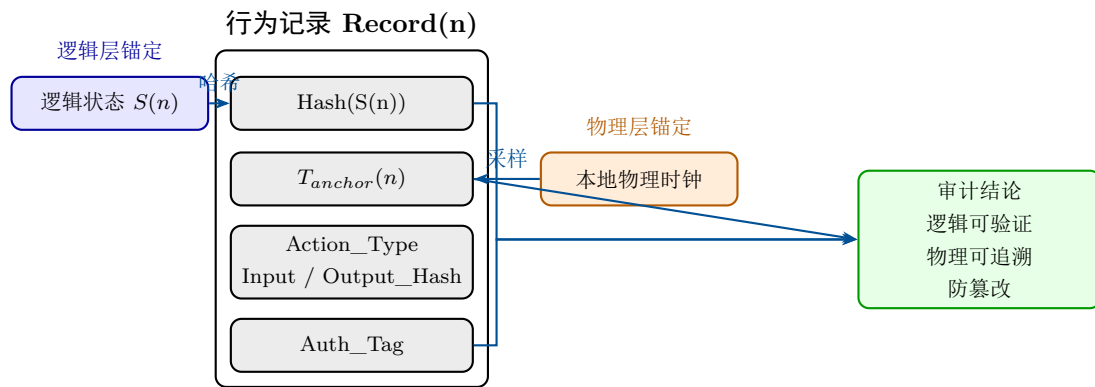
3.5 逻辑-物理双锚定

在逻辑时刻 n ，节点执行以下操作：

1. 输出当前逻辑状态 $S(n)$ ；
2. 将确定性行为与 $S(n)$ 绑定，生成行为记录 $Record(n)$ ；
3. 瞬时捕获本地物理时钟读数 $T_{anchor}(n)$ 作为物理锚点，与 $Record(n)$ 关联；
4. 将 $Record(n)$ 链接至行为链。

行为记录 $Record(n)$ 至少包含以下信息：

字段	含义	用途
n	逻辑时刻标识	确定行为在逻辑时序中的位置
$S(n)$ $Hash(S(n))$	或 当前逻辑状态	逻辑重放验证
$T_{anchor}(n)$	物理锚点	物理追溯
$Action_Type$	行为类型标识	区分行为类型
$Input_Hash$	输入数据的哈希值	逻辑重放
$Output_Hash$	输出结果的哈希值	逻辑重放
$Auth_Tag$	认证标签	防篡改验证

图 3: 行为记录 $Record(n)$ 的数据结构与逻辑-物理双锚定绑定

3.6 行为链

各逻辑时刻的行为记录通过确定性演进链接，形成可验证的行为链：

$$Chain_Link(n, seq) = Hash(Chain_Link(n, seq - 1) || Record(n, seq))$$

$$Chain_Link(n, 0) = Hash(Chain_Link(n - 1, last) || Record(n, 0))$$

其中 seq 为同一逻辑时刻内多条记录的排序序号。行为链采用仅追加结构，编辑、插入、删除或重排任何记录都会破坏链式一致性。为防止截断攻击，验证端定期将行为链根哈希锚定至外部可信存储。

3.7 候选集验证：有界偏差下的密码学合规

严格同步在复杂网络环境中面临根本性挑战。IP 网络的时延抖动、晶振漂移的累积、跨域场景中统一时间基准的缺失，使得 $n_A = n_B$ 的要求在工程上难以持续满足。候选集验证通过构

造候选逻辑状态集，将同步要求从“严格相等”放宽为“有界偏差”，使验证端在不知道 Agent 真实逻辑计数器的条件下仍能完成密码学合规验证。

当 AI 系统与验证端经 IP 网络互联且存在有界偏差时，验证端以本地逻辑计数器值 n_{local} 为中心构造候选逻辑状态集：

$$\{S(n_{local} - K), \dots, S(n_{local}), \dots, S(n_{local} + K)\}$$

逐一计算候选状态的哈希值，与凭证中的 $S_commitment$ 比对。当存在 $S_{candidate}$ 使 $Hash(S_{candidate}) == S_commitment$ 时，确定 $n_{best} = n_{local} + \Delta n$ ， $\Delta n \in [-K, +K]$ 。

候选半径 K 的确定：

$$K_{min} = \max([\delta_{phy}/T_{step}] + K_{safety}, [\Delta_{sync}/T_{step}], [\theta + |R| \cdot T_{sleep}/T_{step}])$$

$$K_{cand_max} = \min(K_{e2e}, K_{hash})$$

其中 $K_{e2e} = [T_{e2e_max}/T_{step}]$ 为端到端时延约束对应的上界， K_{hash} 为哈希碰撞概率约束对应的上界。

候选集匹配基于哈希函数的单向性，验证端无法从 $S_commitment$ 反推 $S(n)$ ，只能通过逐一计算候选状态的哈希值进行精确匹配。这一放宽不是对安全性的妥协，而是将“有界同步”从物理层延伸至密码学验证层的技术贡献。

4 独立验证与无限次重放

4.1 验证流程

当一个行为需要被审计时，审计方执行以下操作：

步骤一：逻辑状态重算——验证逻辑绑定

审计方使用与行为执行节点相同的初始状态和演进规则，独立演进至目标逻辑时刻 n_{audit} ，计算得到 $S(n_{audit})$ 。验证行为记录中的逻辑状态标识是否与独立计算结果一致。若匹配，证明该行为记录确实是在逻辑时刻 n 生成的，没有被篡改。

步骤二：行为重放——验证逻辑正确性

审计方使用记录中的输入数据，在审计端独立重放行为的执行过程，计算输出结果，与记录中的输出结果标识比较。若匹配，证明该行为在逻辑时刻 n 确实产生了该输出。

步骤三：物理锚点追溯——验证物理真实性

审计方读取 $Record(n)$ 中的物理锚点 $T_{anchor}(n)$ ，将其映射到物理世界的时间轴，与外部日志、传感器记录、视频监控等其他物理证据进行交叉验证，确认该逻辑时刻对应的物理时间位置。

步骤四：链完整性验证

审计方验证从初始锚点到目标时刻的全部链接连续。

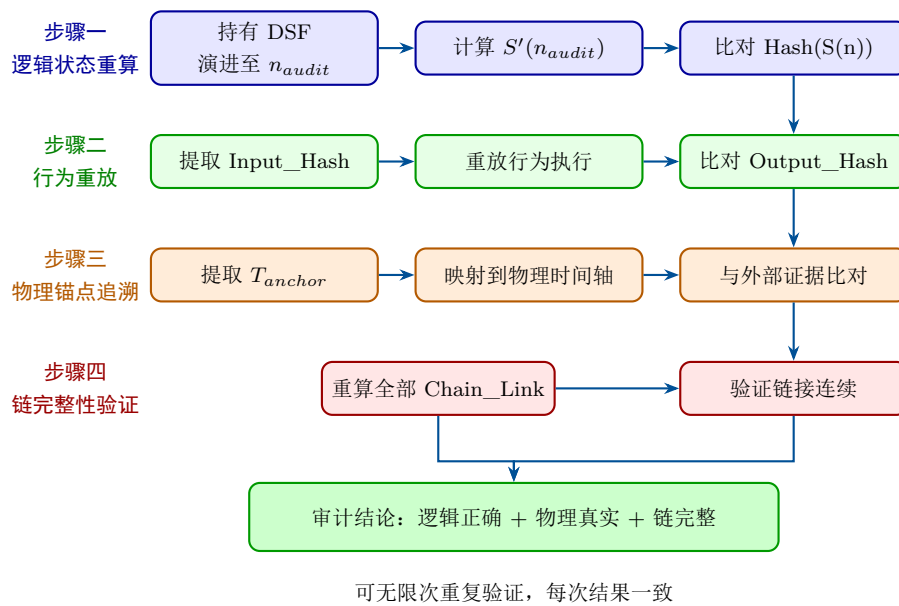


图 4: 独立验证与无限次重放的完整闭环流程

4.2 审计结论的全面性

通过上述四个步骤，审计方能够得出一个全面的审计结论：

“在逻辑时刻 n （对应物理时刻 $T_{anchor}(n)$ ），节点在逻辑状态 $S(n)$ 下，执行了类型为 $Action_Type$ 的行为，输入数据为 $Input_Hash$ ，输出结果为 $Output_Hash$ 。逻辑重放验证通过，物理追溯交叉验证通过，行为链完整未篡改。”

这一结论同时包含了行为在逻辑世界中的正确性验证和物理世界中的真实性验证，形成了完整的审计闭环。

4.3 无限次重放

这一验证过程与物理时间的可重放性无关，因此可以无限次重复，每次均得到相同结果。因为所有验证对象都是确定性逻辑时序基准的产物，不依赖物理世界的可重放性。

物理世界不可逆，逻辑世界可逆——将行为的发生时刻锚定在可重放的确定性逻辑时序基准上，使“当时发生了什么”可在任意时刻被独立验证和无限次重放。

5 安全模型

5.1 威胁模型

本文考虑以下攻击者能力：

- 攻击者可截获、篡改、重放、延迟或丢弃任意网络消息；

- 攻击者可物理捕获任意节点，读取其存储的所有数据；
- 攻击者可控制部分节点，使其发送任意消息；
- 攻击者不掌握 K_{sec} 和当前 epoch 子密钥 K_e 。

5.2 安全性质

不可伪造性：不掌握 K_{sec} 的攻击者无法伪造有效的 $S(n)$ 或 $Auth_Tag$ 。即使攻击者截获了历史 $S(n)$ ，由于哈希链的单向性，无法推知 $S(n+1)$ 。

可独立验证性：任何持有相同 DSF 的验证端可独立重算 $S(n)$ ，无需与行为执行节点交互，无需外部时间源，无需第三方存证。

不可篡改性：行为链采用哈希链链接，任一记录被篡改，从该时刻起后续所有链接全部断裂。

前向安全：即使 $S(n)$ 被泄露，攻击者无法推知 $S(n+1)$ 。

后向安全：即使设备被物理捕获，攻击者无法恢复历史 $S(k)$ ($k < n$)。

密钥泄露分层控制：系统将逻辑时间轴划分为连续的时间段（epoch），每个 epoch 自动切换子密钥 K_e 。即使某时间段的子密钥泄露，也仅影响该时间段。

5.3 密钥管理

K_{sec} 的安全性依赖于分发、存储与轮换三个环节的工程实现。

分发： K_{sec} 通过安全通道预共享，包括 TLS/DTLS 加密通道、IPsec 隧道、预共享密钥协商、可信配置中心分发、硬件安全模块注入等方式。在审计场景中，DSF 参数可独立于业务系统单独配置，也可从业务连接的 DSF 派生：

$$K_{sec_audit} = H(K_{sec_business} \parallel \text{"AUDIT_DERIVE"})$$

存储： K_{sec} 存储于硬件安全模块（HSM）或可信执行环境（TEE）中，永不离开安全边界。 K_{sec} 仅用于派生时效性子密钥 K_e ，不直接参与 $S(n)$ 的计算，也不以明文形式暴露于常规内存。

轮换：系统将逻辑时间轴划分为连续的 epoch，每个 epoch 包含 N_{epoch} 个逻辑节拍（典型值 10^6 ）。每个 epoch 自动切换子密钥：

$$K_e = H(K_{sec} \parallel e \parallel \text{"SUBKEY"}), \quad e = \lfloor n / N_{epoch} \rfloor$$

在 $T_{step} = 1 \text{ ms}$ 的配置下，每个 epoch 约 16.7 分钟。

密钥泄露的影响分析：

泄露对象	影响范围
单个 K_e	仅影响该 epoch 内生成的 $S(n)$ ；历史 epoch 的 $S(k)$ (k 属于更早的 epoch) 由历史 K_e 派生，无法从当前 K_e 反推；未来 epoch 的 K_e 亦无法预测
K_{sec}	需通过安全通道下发新的 DSF 三元组，重新建立逻辑时间轴；已生成的历史行为链仍可通过已有的审计流程独立验证，但其密码学可信度取决于 K_{sec} 泄露前是否已被外部锚定

侧信道攻击的缓解： K_{sec} 与 K_e 的派生在安全边界内完成， $S(n)$ 的演进不依赖任何可观测的物理量（物理锚点 T_{anchor} 仅被采样、不参与 $S(n)$ 计算）。因此，攻击者即使观测到节点的功耗、电磁辐射或执行时间，也无法从物理侧信道推断出 K_{sec} 。

5.4 与现有安全模型的对比

安全性质	区块链存证	TEE	Inference ceipts	Re-	本发明
不可伪造	依赖共识	依赖硬件	依赖诚实发射者		密码学强制
可独立验证	需接入网络	需信任硬件 厂商	需信任发射者		持有 DSF 即可
不可篡改	共识保证	硬件保证	哈希链		哈希链 + 密码学
前向安全	不涉及	不涉及	不涉及		有
后向安全	不涉及	不涉及	不涉及		有
密钥泄露控制	不涉及	不涉及	不涉及		分层控制

6 与现有技术的本质区别

6.1 与 Lamport 时钟的本质区别

对比维度	Lamport 时钟	本发明
记录内容	事件序号（仅顺序）	逻辑时刻 + 逻辑状态 + 输入输出哈希 + 物理锚点
物理可追溯	无	物理锚点
逻辑与物理解耦	是	是
逻辑与物理锚定	否	是
行为内容记录	否	是
可验证证据链	否	是
审计与重放	否	是
独立验证	否	是

Lamport 时钟需要节点间通信来维护一致性，且仅建立事件的偏序关系；本发明中的确定性逻辑时序基准在单节点上完全自包含，无需通信即可独立演进，且为每个逻辑时刻提供完整的逻辑状态。

定量对比：Lamport 时钟的单次事件计数操作是一次整数递增，耗时纳秒级；本发明的单次 PSSM 状态更新是一次 SHA-256 哈希，软件实现约 $0.5\ \mu\text{s}$ ，硬件实现约 $0.01\ \mu\text{s}$ ，是 Lamport 时钟的 10^3 至 10^5 倍。但 Lamport 时钟仅提供事件排序，不产生可验证证据；本发明在 $0.5\ \mu\text{s}$ 的开销内完成了逻辑状态演进、逻辑锚定和密码学绑定。在 $T_{\text{step}} = 1\ \text{ms}$ 的配置下， $0.5\ \mu\text{s}$ 占逻辑节拍的 0.05%，性能代价可忽略。

6.2 与区块链存证的本质区别

对比维度	区块链存证	本发明
存证时机	事后上链	行为发生时实时绑定
信任锚点	共识机制	密码学规则
验证方式	需接入网络	持有 DSF 即可离线验证
时间基准	物理时间戳	逻辑时间轴 + 物理锚点
重放能力	无	无限次重放
单点篡改风险	共识节点合谋	密码学保证

区块链存证只能证明“数据在某个时刻被写入”，无法证明“数据在当时是正确的”；本发明通过逻辑状态重算和行为重放，能够证明“行为在逻辑时刻 n 确实产生了该输出”。

6.3 与 TEE/ZKML 的本质区别

对比维度	TEE	ZKML	本发明
信任基础	硬件厂商	数学证明	密码学规则
跨域验证	困难	困难	天然支持
时间绑定	物理时间戳	无时间信息	逻辑时间轴 + 物理锚点
计算开销	低	极高	低（哈希 +HMAC）
行为审计	单次认证	无	完整行为链
无限重放	否	否	是

6.4 与 Inference Receipts 的本质区别

对比维度	Inference Receipts	本发明
作用时机	推理完成后	推理开始前锚定
锚定对象	记录写入时刻（因果链终点）	推理发起时刻（因果链起点）
能否证明推理发起时刻	否	是
熵源控制	不涉及	$S(n)$ 作为 PRNG 种子
跨环境一致性	明确不保证	密码学保证
安全模型	诚实发射者假设	密码学强制
独立验证	需外部账本	持有 DSF 即可

定量对比：Inference Receipts 在推理完成后生成收据，仅需一次哈希计算，开销极低（公开资料称低于推理时间的 0.006%），但其锚定对象是因果链终点，且不涉及熵源控制。本发明在推理开始前锚定 $S(n)$ 并以其作为 PRNG 种子，额外开销为一次 PSSM 状态更新（约 $0.5\ \mu s$ ）和一次 HMAC 派生（约 $1\ \mu s$ ）；这一增量开销换取了两个关键能力——推理起点的可证明性和跨环境的密码学一致性。两者的开销在同一数量级，但可信度模型本质不同。

7 最小验证：AI 调用审计

7.1 场景描述

某金融机构部署了一套 AI 交易决策系统。该系统使用第三方模型发布方训练的深度学习模型，部署在自有的推理服务器上，通过 API 向交易员客户端提供服务。监管机构需要对该 AI 系统的决策过程进行审计。

7.2 系统配置

参数	值	说明
K_{sec}	256 位预共享密钥	收发双方一致
$Init_Anchor$	2026-10-01 00:00:00 UTC	逻辑时间轴起点
$Rule_ID$	哈希链, $T_{step} = 1ms$	逻辑节拍
N_{epoch}	10^6	每个 epoch 包含的逻辑节拍数
Δ_{sync}	$10\mu s$	物理锚点偏差容限
K	3	候选半径

7.3 验证流程

步骤 1：逻辑状态重算

监管机构持有相同 DSF，独立运行 PSSM 至逻辑时刻 n ，计算 $S(n)$ ，验证与 $Record(n)$ 中的 $Hash(S(n))$ 一致。

步骤 2：安全窗口验证

检查 n 是否落在预定义的交易时段逻辑计数器区间内，并验证操作属性（如金额）是否满足阈值条件。

步骤 3：行为链完整性验证

验证从初始锚点到目标时刻的全部哈希链接连续。

步骤 4：物理锚点追溯

使用 $Record(n)$ 中的物理锚点 $T_{anchor}(n)$ 将逻辑时刻映射到物理时间轴，与银行核心系统日志、数据库事务时间戳进行交叉验证。

7.4 审计结论示例

“在逻辑时刻 n （对应物理时刻 $T_{anchor}(n)$ ），AI 审批系统在逻辑状态 $S(n)$ 下处理了贷款申请 $ID = 12345$ ，审批结果为‘批准’，额度为 50 万元。逻辑重放验证通过，物理追溯交叉验证通过（与银行核心系统日志时间戳偏差小于 1ms），行为链完整未篡改。”

7.5 验证结果

验证项	结果
逻辑状态重算	$S_{recalc}(n) = S_{record}(n)$, 错误 0 次
安全窗口验证	正确识别合规/违规操作
行为链完整性	篡改任一记录导致链断裂
物理锚点追溯	与外部证据偏差 $< \Delta_{ext}$
无限次重放	每次验证结果一致

7.6 工程演示：三节点局域网实现

为验证本文核心机制的工程可行性，我们实现了一个三节点局域网演示系统。三个节点角色如下：

- **Node-A (Agent)**: 主动发起 AI 调用请求，接收执行回执，将凭证和回执提交审计中心；
- **Node-B (大模型推理服务)**: 接收调用凭证，执行推理（确定性函数模拟），生成执行回执并维护行为链；
- **Node-C (合规/审计中心)**: 独立重算 $S(n)$ ，验证逻辑绑定、Auth_Tag、安全窗口、金额阈值、行为链完整性。

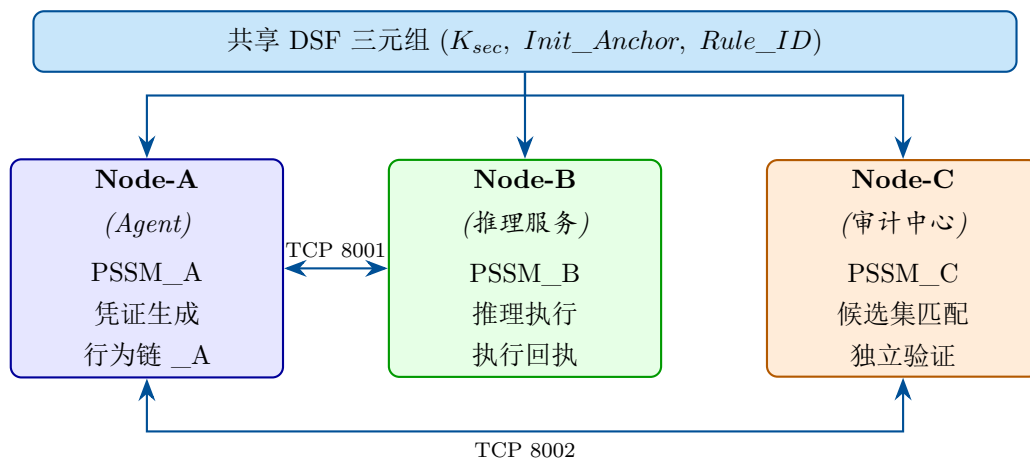


图 5: 三节点演示系统架构。三个节点共享同一套 DSF 三元组，各自独立运行 PSSM。

三节点通过 TCP socket 通信，仅依赖 Python 标准库，代码约 600 行。演示覆盖两种模式：
严格同步模式 (K=0): Node-C 与 Agent 逐拍对齐，用凭证中的明文逻辑计数器推进 PSSM 并验证。三次测试的 n_{exec} 为 1、2、3，Node-C 本地 n_{local} 逐次推进为 0、1、2，逻辑绑定验证全部通过。

有界偏差模式 (K=3): Node-C 本地 $n_{local} = 2$, 与 Agent 的 n_{tx} 存在 ± 1 拍偏差。Node-C 构造候选集 $\{S(-1), \dots, S(5)\}$ (有效范围 $[S(0), S(5)]$), 逐一计算哈希, 成功识别出三次调用的真实 n_{best} (1、2、3), 并基于 n_{best} 完成 Auth_Tag、安全窗口、金额阈值和行为链完整性验证。两次演示的审计结果如下:

测试	Action	金额	n_{best}	逻辑绑定	Auth_Tag	金额阈值	结论
1	TRADE	500,000	1				PASS
2	TRADE	800,000	2				PASS
3	TRADE	5,000,000	3				REJECT

两次演示共享同一套 DSF 三元组、PSSM 演进规则、行为记录格式和行为链链接机制, 区别仅在验证端的候选集匹配部分。这印证了候选集验证将“有界同步”从物理层延伸至密码学验证层的技术路径: PSSM、双锚定、行为链等基础机制完全复用, 仅验证端的逻辑计数器对齐要求从“严格相等”放宽为“有界偏差”。

演示还验证了行为链的篡改检测能力: 修改任一条记录的字段会导致从该时刻起后续所有链链接断裂, 验证端通过独立重算即可发现。同时, 演示验证了候选集验证的计算开销: K=3 时单次匹配耗时约 0.1 ms, 满足跨域合规验证的实时性要求。

7.7 性能开销分析

本节从哈希计算、导频匹配、开窗接收三个方面给出 PSSM 与行为链的定量性能开销, 为工程部署提供性能预算参考。

操作	算法	软件耗时	硬件耗时
PSSM 状态更新	SHA-256	$\sim 0.5 \mu s$	$\sim 0.01 \mu s$
导频序列生成	HMAC-SHA256	$\sim 1 \mu s$	$\sim 0.02 \mu s$
行为链链接	SHA-256	$\sim 0.5 \mu s$	$\sim 0.01 \mu s$
加密密钥派生	HKDF	$\sim 2 \mu s$	$\sim 0.05 \mu s$
单拍总耗时	—	$< 3 \mu s$	$< 0.1 \mu s$

在 $T_{step} = 1 \text{ ms}$ 的配置下, 单拍计算开销约占逻辑节拍的 0.3%, 余量 300 倍以上; 即使采用纯软件实现, 也完全满足实时要求。

指标	数值
单次候选集匹配（K=3）	~ 0.1 ms
单次候选集匹配（K=500）	~ 12 ms
500 次 API 调用平均逻辑跨越	726 拍
500 次 API 调用平均物理耗时	726 ms
500 次 API 调用耗时标准差	189 ms
审计通过率	100%（500/500）

候选集匹配耗时随候选半径 K 线性增长，但与逻辑计数器 n 无关（即不随运行时长增长）。在跨域合规验证场景（ $K \leq 3$ ）中，单次匹配耗时约 0.1 ms，满足实时性要求；在卫星通信场景（ $K = 5 \sim 10$ ）中，单次匹配耗时约 0.2~0.4 ms，仍在可接受范围内。

与 ZKML 的对比：ZKML 的计算开销是原始推理的 1000 至 10000 倍，目前仅能对极小的模型或推理的切片进行验证。本发明的核心操作仅为 SHA-256 哈希和 HMAC 派生，计算开销为微秒级，可在生产级 AI 系统中直接部署。

8 应用前景

确定性逻辑时间轴可作为以下场景的统一时序底座：

应用领域	核心价值
AI Agent 合规控制	跨信任域、跨物理时钟域的事前密码学强制约束
Token 生成确定性	跨环境、跨硬件、跨框架的 Token 采样路径一致
AI 输出确定性	推理起点锚定、版本管理、检索复用
无人机集群审计	飞行行为可追溯、可审计、可重放
金融交易审计	交易决策的密码学证据链
工业控制审计	控制指令的确定性时序绑定
卫星星座管理	跨星、跨地面站的确定性协同
分布式数据库审计	事务执行的一致性与完整性验证
智能合约合规	合约执行的密码学时序绑定

8.1 案例一：金融交易审计

场景：某跨国投资银行部署了一套分布式高频交易系统，由全球 5 个数据中心的数百台交易服务器组成。监管机构要求每笔交易决策可追溯、可验证，且监管方需独立验证而不依赖银行的内部日志。

核心需求：

- 交易顺序可验证：跨数据中心的交易必须有一致的全局顺序；
- 监管可追溯：每笔交易必须能映射到具体的模型版本、输入数据和执行时刻；
- 无第三方存证：监管方不能依赖银行的日志系统或外部存证平台。

DSF 方案：以全局事务序号作为确定性逻辑时间轴。在每一逻辑时刻 n ，交易服务器执行以下操作：

1. PSSM 输出逻辑状态 $S(n)$ ；
2. 订单接收、订单匹配、成交确认等行为与 $S(n)$ 绑定，生成行为记录 $\text{Record}(n)$ ；
3. 瞬时采样本地物理时钟，记录 $T_{\text{anchor}}(n)$ ；
4. $\text{Record}(n)$ 链接至行为链。

预期效果：监管机构持有相同的 DSF 参数，独立运行 PSSM 至目标逻辑时刻，重算 $S(n)$ ，验证行为记录与逻辑时刻的绑定关系，并通过 $T_{\text{anchor}}(n)$ 将逻辑时刻映射到物理时间轴，与银行核心系统日志、数据库事务时间戳进行交叉验证。整个验证过程不依赖银行的内部日志系统。

8.2 案例二：工业控制审计

场景：某智能制造车间部署了多台工业机器人，负责协同装配。每台机器人上运行 AI 决策模块，中央控制器下发任务指令。行业规范要求控制指令的时序确定性可审计、事故可追溯。

核心需求：

- 控制指令的确定性时序绑定：每条指令必须绑定到确定的逻辑时刻；
- 事故可追溯：发生事故后，审计方需独立重放控制指令序列，定位责任；
- 跨节点协同：多台机器人需在同一逻辑时间轴上协同。

DSF 方案：中央控制器与各机器人在任务开始前通过安全通道共享 DSF。此后双方独立运行 PSSM，逻辑计数器严格相等。每条控制指令与 $S(n)$ 绑定，并通过行为链记录。各机器人的行为记录可独立链接至各自的行为链，中央控制器维护全局行为链。

预期效果：事故发生后，审计方（如安全生产监管部门）持有 DSF 参数，独立重算 $S(n)$ ，验证控制指令与逻辑时刻的绑定关系，重放控制指令序列，并与现场传感器记录进行交叉验证，定位责任节点和逻辑时刻。

9 结论

本文提出了一种密码学驱动的确定性逻辑时间轴，作为机器协同与 AI 审计的统一时序基准。本文的核心贡献在于：

- 1. **问题定义：**系统阐述物理时间不可逆、物理时钟不可信、第三方存证不可独立验证三大根本困境，揭示现有方案在机器通信与 AI 审计场景下的共同失效根源。
- 2. **机制提出：**提出 DSF 三元组、PSSM、逻辑-物理双锚定、行为链、候选集验证等核心机制，构建与物理时钟解耦、可独立验证、无限次重放的确定性逻辑时序基准。
- 3. **边界划定：**系统对比本方案与 Lamport 时钟、NTP/PTP、GPS、区块链存证、TEE、ZKML、Inference Receipts 等现有技术的本质区别。
- 4. **安全模型：**给出不可伪造、可独立验证、不可篡改、前向/后向安全、密钥泄露分层控制等安全性质的形式化定义与分析。
- 5. **最小验证：**以 AI 调用审计为例，展示“逻辑状态重算 + 行为链完整性 + 物理锚点追溯”的完整闭环，并通过三节点工程演示验证严格同步（K=0）与有界偏差（K=3）两种模式。

物理世界不可逆，逻辑世界可逆。将行为的发生时刻锚定在可重放的确定性逻辑时序基准上，使“当时发生了什么”可在任意时刻被独立验证和无限次重放。

这一能力不依赖物理时钟的精度、不依赖网络条件、不依赖任何第三方存证服务——因为所有验证对象都在确定性逻辑时序基准上被永久定格。传统方案依赖“存证”（相信数据未被篡改），本发明提供“证明”（可通过独立重放验证数据正确性），两者的可信度存在本质区别。

9.1 已知限制与未来工作

本节的目的是明确当前工作边界，为后续研究提供方向。

已知限制：

限制	说明
链尾截断检测	当前版本的行为链验证仅检测相邻记录的链式连续性，不检测链尾截断。截断攻击需通过外部锚定机制（定期将链根哈希锚定至第三方时间戳服务或监管机构的独立存储）来防护，该机制已在专利中提出，但在本文的工程演示中尚未实现
T_{step} 不参与密码学计算	T_{step} 仅用于将逻辑时刻映射到物理时刻，不参与 $S(n)$ 的计算。若 T_{step} 配置错误，密码学验证仍会通过，但物理时刻映射将出错。实际部署中需通过外部证据核对 T_{step} 的正确性
单节点事件驱动	本文的工程演示为单节点事件驱动 PSSM。周期性 PSSM、多节点协同在后续工作中实现
候选集匹配的 K 上界	K 的增大虽然提升了有界偏差的容忍度，但也增加了计算开销。K 的上界由端到端时延约束和哈希碰撞概率约束共同确定，在大规模部署中需要进一步优化

未来工作：

1. **外部锚定机制**：实现将行为链根哈希定期锚定至外部可信存储的接口，闭合链尾截断检测的缺口；
2. **多节点协同**：将单节点事件驱动 PSSM 扩展至多节点周期性 PSSM，支持分布式协同审计；
3. **不同网络条件下的系统仿真**：在局域网、城域网、广域网、卫星链路等不同网络条件下，系统评估候选集验证的成功率、K 的取值、单次匹配耗时等指标；
4. **第三方独立复现**：邀请独立机构在干净环境中复现本文的工程演示，出具独立验证报告；
5. **形式化安全证明**：对不可伪造性、前向安全、后向安全等安全性质给出形式化证明。

AI 使用声明

本文的 LaTeX 排版、格式整理、文字润色和结构建议由 AI 辅助完成。研究设计、技术方案、核心机制、实验验证和结论均由作者完成。AI 未参与研究问题的提出、技术路线的选择、实验数据的生成或结果的解释。

参考文献

- [1] L. Lamport, “Time, clocks, and the ordering of events in a distributed system,” *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.
- [2] D. L. Mills, “Internet time synchronization: the network time protocol,” *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.
- [3] IEEE, “IEEE 1588-2019 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” 2019.
- [4] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
- [5] V. Buterin, “Ethereum white paper,” 2014.
- [6] V. Costan and S. Devadas, “Intel SGX explained,” *IACR Cryptology ePrint Archive*, vol. 2016, p. 086, 2016.
- [7] T. Liu, Y. Xie, and J. Zhang, “ZKML: A survey on zero-knowledge machine learning,” *arXiv preprint arXiv:2403.10326*, 2024.
- [8] Thinking Machines Lab, “Batch-invariant kernels for deterministic LLM inference,” 2026.
- [9] Model Witness: Cryptographic attestation for AI inference, 2026.
- [10] AIIR: AI Integrity Receipts, 2026.
- [11] S. S. Kulkarni, M. Demirbas, B. Madappa, et al., “Logical physical clocks,” in *International Conference on Principles of Distributed Systems*, Springer, 2014, pp. 17–32.

- [12] 中国民用航空局, “MH/T 2015-2024 基于区块链的民用无人驾驶航空器飞行数据存证技术要求,” 2024.
- [13] 3GPP, “TR 38.821: Solutions for NR to support non-terrestrial networks (NTN),” 2023.
- [14] ITU-R, “IMT-2030 Framework and overall objectives of the future development of IMT for 2030 and beyond,” 2023.

A 符号表

符号	含义
DSF	动态安全根基
K_{sec}	共享安全密钥
$Init_Anchor$	初始锚点
$Rule_ID$	状态转移规则标识
$PSSM$	协议安全状态机
$S(n)$	逻辑状态
$T_{anchor}(n)$	物理锚点
n	逻辑计数器
T_{step}	逻辑节拍周期
N_{epoch}	每个 epoch 包含的逻辑节拍数
K_e	时效性子密钥
$Chain_Link(n)$	行为链链接值
K	候选半径
Δ_{sync}	物理锚点偏差容限
Δ_{ext}	物理锚点与外部证据的容限

B 缩略语

缩略语	全称
DSF	Dynamic Security Foundation
PSSM	Protocol Security State Machine
TEE	Trusted Execution Environment
ZKML	Zero-Knowledge Machine Learning
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
NTP	Network Time Protocol
PTP	Precision Time Protocol
HLC	Hybrid Logical Clock
DCI	Downlink Control Information
DMRS	Demodulation Reference Signal
PRF	Pseudo-Random Function
HMAC	Hash-based Message Authentication Code
HKDF	HMAC-based Key Derivation Function